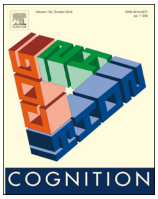
Cognition 201 (2020) 104295

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/cognit



The effects of declaratively maintaining and proactively proceduralizing novel stimulus-response mappings

Silvia Formica*, Carlos González-García, Marcel Brass

Ghent University, Faculty of Psychology and Educational Sciences, Department of Experimental Psychology, Henri Dunantlaan 2, B-9000 Ghent, Belgium

ARTICLE INFO

Keywords:
Working memory
Instructions
Retro-cues
Attentional prioritization

ABSTRACT

Working memory (WM) allows for the maintenance and manipulation of information when carrying out ongoing tasks. Recent models propose that representations in WM can be either in a declarative format (as content of thought) or in a procedural format (in an action-oriented state that drives the cognitive operation to be performed). Current views on the implementation of novel instructions also acknowledge this distinction, assuming these are first encoded as declarative content, and then reformatted into an action-oriented procedural representation upon task demands. Although it is widely accepted that WM has a limited capacity, little is known about the reciprocal costs of maintaining instructions in a declarative format and transforming them in an action code. In a series of three experiments, we asked participants to memorize two or four S-R mappings (i.e., declarative load), and then selected a subset of them by means of a retro-cue to trigger their reformatting into an action-oriented format (i.e., procedural load). We measured the performance in the implementation of the proceduralized mapping and in the declarative recall of the entire set of memorized mappings, to test how the increased load on one component affected the functioning of the other. Our results showed a strong influence of declarative load on the processing of the procedural component, but no effects in the opposite direction. This pattern of results suggests an asymmetry in the costs of maintenance and manipulation in WM, at least when procedural representations cannot be retrieved from long term memory and need to be reformatted online. The available resources seem to be first deployed for the maintenance of all the task-relevant declarative content, and proceduralization takes place to the extent the system can direct attention to the relevant instruction.

1. Introduction

In everyday activities, the information acquired from the environment needs to be manipulated and reformatted in a way that makes it suitable to perform the necessary cognitive or physical actions to achieve specific goals. The cognitive system that is assumed responsible for the storage and processing of such information is working memory (WM; Baddeley & Hitch, 1974).

Classical working memory models assume that different subsystems operate on different contents such as visual and verbal information (Baddeley, 2012; Baddeley & Hitch, 1974). Interestingly, the distinction between declarative and procedural representations that is fundamental in the domain of long term memory (Anderson, 1982) has not until recently found its way into WM models (Oberauer, 2009, 2010). In his theoretical framework, Oberauer (2009) conceptualizes WM as an attentional system divided in two analogous components, each of which deals with different types of representations. Declarative WM is responsible for holding the necessary information accessible, whereas procedural WM contains the cognitive operations that have to be carried out (Oberauer, 2009, 2010). In its original formulation, the model by Oberauer assumes the two components to operate independently. Nevertheless, studies using the Psychological Refractory Period setup provide initial evidence in favor of an interdependence in the functioning of declarative and procedural WM (Janczyk, 2017).

The distinction between these two types of WM representations is also relevant in the cognitive control domain that deals with the implementation of instructions (Brass, Liefooghe, Braem, & De Houwer, 2017; Meiran, Pereg, Kessler, Cole, & Braver, 2015). Crucially, this field is focused mainly on novel instructions, therefore excluding a role of prior practice or training (i.e., procedural long term memory) in the implementation of the new instructed behavior (Meiran, Liefooghe & De Houwer, 2017). Here, a novel stimulus-response (S-R) mapping is thought to be first encoded in a declarative format and subsequently reformatted into a procedural representation capable of driving the instructed behavior (Brass et al., 2017; Ruge & Wolfensteller, 2010; Wenke, Gaschler, Nattkemper, & Frensch, 2009). Evidence in support of

---

* Corresponding author.
E-mail address: silvia.formica@ugent.be (S. Formica).

https://doi.org/10.1016/j.cognition.2020.104295
Received 2 October 2019; Received in revised form 1 April 2020; Accepted 6 April 2020

this view comes from the phenomenon of *goal neglect*, which refers to the inability to carry out such reformatting, leading to the failure in implementing the required behavior, while being able to recall the task instructions (Bhandari & Duncan, 2014; Duncan, Emslie, Williams, Johnson, & Freer, 1996; Milner, 1963). Recently, an analogous attentional manipulations highlighted the role of attention in the reformatting from declarative to procedural code, supporting the existence of these two different formats (González-García, Formica, Liefooghe, & Brass, 2020). Furthermore, brain imaging research suggests a similar dissociation between declarative and procedural representations at the brain level (Bourguignon, Braem, Hartstra, De Houwer, & Brass, 2018; Demanet et al., 2016; González-García, Formica, Wisniewski, & Brass, 2019; Hartstra, Kühn, Verguts, & Brass, 2011; Muhle-Karbe, Duncan, De Baene, Mitchell, & Brass, 2017).

Notably, it has been demonstrated that the transformation from a declarative into a procedural format only takes place when the task requires the implementation of the instructions, but not if the participant is simply asked to maintain the instruction for subsequent recognition (Liefooghe, Wenke, & De Houwer, 2012; but see also Liefooghe & De Houwer, 2018). Therefore, both at the behavioral and at the neural level, there is evidence for the dissociation between 'knowing' the content of the instruction (i.e. a declarative representation of the S-R contents in WM) and 'doing' (i.e. representing in a procedural format) the instructed behavior (Brass et al., 2017; Demanet et al., 2016; Muhle-Karbe et al., 2017). Once this reformatting takes place, the resulting action-bound representation can act like a prepared reflex upon stimulus presentation (Hommel, 2000; Meiran et al., 2017). This intention-based reflexivity (IBR) has been demonstrated by showing that merely instructed (thus, not yet implemented) novel S-R mappings can elicit congruency effects on a secondary diagnostic task, sharing the same stimuli and response sets as the instructed mappings, but with a different task rule (Liefooghe et al., 2012; Meiran et al., 2017, 2015).

Crucially for the goal of our experiments, previous studies have shown that the reformatting of declarative instructions into action-oriented representations, a process that is sometimes referred to as *proceduralization* (Brass et al., 2017), can be disrupted when adding concurrent WM load (Cohen-Kdoshay & Meiran, 2006; Meiran & Cohen-Kdoshay, 2012). These studies showed an absence of compatibility effects in the presence of a concomitant secondary task whose rules changed in every block, suggesting that proceduralization does not occur reflexively in high WM load settings. Furthermore, it has been demonstrated that proceduralization seems to be subject to a capacity limit. Liefooghe and colleagues reported the results of an unpublished experiment in which they did not observe IBR effects with four instructed S-R mappings in the inducer task and, analogously, Cohen and collaborators only observed reflexive responses with restricted sets of instructed mappings (Cohen, Jaudas, & Gollwitzer, 2008; Liefooghe et al., 2012). Therefore, it seems that only a limited amount of information can be rapidly reformatted into the procedural state, but it remains unclear to what extent the proceduralization process and the quality of the resulting representations are affected by declarative and procedural load, separately. This raises the issue of interdependence between procedural and declarative WM representations, as creating novel procedural representations might benefit or affect the underlying declarative content. A large amount of research has been devoted to investigating the capacity limits of WM and the mechanisms that determine them (Cowan, Morey, Chen, Gilchrist, & Saults, 2008; Oberauer, Farrell, Jarrold, & Lewandowsky, 2016). Nevertheless, most of the time the proposed distinction between declarative and procedural formats is not addressed in the assessment of WM capacity. Traditionally, tasks such as visual recognition, change detection, memory span and recall have been used to investigate the number of items that can be maintained in declarative WM and how increasing their number and the duration of the interval between encoding and test affects the quality of their representations (Barrouillet, Bernardin, & Camos, 2004; Barrouillet, Bernardin, Portrat, Vergauwe, & Camos, 2007; Conway et al., 2005; Fukuda, Awh, & Vogel, 2010; Luck & Vogel, 1997; Marois & Ivanoff, 2005; Oberauer et al., 2016; Rouder, Morey, Morey, & Cowan, 2011). On the other hand, manipulating the load on procedural WM is less straightforward and usually involves varying the number and the characteristics (e.g., arbitrary mappings vs fixed sequences) of stimulus-response mappings to be implemented (Gade, Druey, Souza, & Oberauer, 2014; Shahar, Teodorescu, Anholt, Karmon-Presser, & Meiran, 2017; Shahar, Teodorescu, Usher, Pereg, & Meiran, 2014).

To gain insights into the reciprocal influence of procedural and declarative representations, one strategy is to add load on one of the two components to measure whether this is detrimental for the functioning of the other component and vice-versa. Few studies tried to investigate the cross-component effects of load by means of dual task paradigms, with one task thought to rely mainly on declarative WM (i.e., a task in which maintaining the information is sufficient, such as a recognition task) and one task taxing primarily the procedural component (i.e., a task involving the implementation of condition-action rules, as in a choice reaction task, CRT) (Barrouillet, Corbin, Dagry, & Camos, 2015; Gade et al., 2014; Vergauwe, Camos, & Barrouillet, 2014). These studies have mixed results: some advocate in favor of an interdependence between maintenance and processing of information in WM (Barrouillet et al., 2015; Vergauwe et al., 2014), whereas other results seem to support a clear distinction and independence between the two representational states (Gade et al., 2014).

In the present study, we adapted the dual task paradigm to the field on novel instruction implementation, with the goal of investigating how the number of instructions that needs to be maintained declaratively affects the effectiveness of the proceduralization process, and vice versa. In order to do so, we used a retro-cuing paradigm (Myers, Chekroud, Stokes, & Nobre, 2018; Myers, Stokes, & Nobre, 2017; Rerko & Oberauer, 2013; Souza & Oberauer, 2016) that has been adapted to investigate the transformation of declarative into procedural representations (González-García et al., 2020). In a series of three experiments, we presented novel S-R mappings in every trial and asked participants to memorize these mappings for later recognition. Subsequently, a subset of these mappings was selected by means of a retrocue to be prepared for implementation. Thus, this paradigm allowed us to manipulate the declarative load (two or four) by varying the total number of mappings presented at encoding, whereas the procedural load was modulated by the number of mappings selected by the retrocue (one or two). The choice of these loads was motivated by previous research suggesting that instruction-based congruency effects are not elicited when more than two mappings have to be prepared for implementation (González-García et al., 2020; Liefooghe et al., 2012). Therefore, we varied the procedural load within this maximum limit and, consequently, chose two and four as the possible declarative loads to keep the overall task doable for the participant. We reasoned that simply maintaining the content of a variable number of instructions would primarily tap into the declarative component of WM, whereas proactively preparing to implement a subset of the stored instructions, reformatting them into action-oriented representations, would increase the load mainly on the procedural component of WM.

This experimental setup allowed us to control for two important factors. First, increasing the load on the procedural component is not adding additional load on the declarative one. The proceduralized mappings are selected by the retro-cue within the whole pool of mappings that need to be maintained in their declarative format for recognition. In this way, varying the load on the procedural component (i.e., proceduralizing one or two mappings) is not introducing new declarative content, in contrast to what might have been the case in previous studies (Gade et al., 2014). Moreover, new S-R mappings are presented in every trial. This is crucial to ensure the procedural representations of the selected mappings are created following their selection and not retrieved from long-term memory, as could be the case with non-arbitrary or over-trained mappings (Shahar et al., 2017, 2014).

In Experiment 1, we first tested the load effects in our dual task setting (CRT followed by recognition task). Concerning within-component load effects, we predicted each load manipulation to have a clear effect on the corresponding WM component (i.e., high declarative load is detrimental for the recognition task and high procedural load affects the CRT). Crucial to our main question, our paradigm allowed us to investigate whether declarative load impairs the proceduralization of S-R mappings and, conversely, if proceduralization affects the underlying declarative representations. We aimed at adjudicating between three possible scenarios: 1) the effects of the two load manipulations could be completely independent, resulting in only within-component load effects (i.e., declarative load affecting only the declarative task and procedural load affecting only the procedural task); 2) the effects could be symmetrical, with proceduralization being affected by the number of items maintained and the recognition task being influenced by the amount of proceduralized mappings (i.e., both loads affecting both tasks); 3) the load effects could be asymmetrical, if one load manipulation affected both tasks and the other did not. Next, we conducted two additional experiments to replicate our findings and control for potential confounds. Specifically, in Experiment 2 the order of the two tasks (CRT and recognition) was randomized, and in Experiment 3 the proportion of the two tasks was manipulated.

2. Experiment 1

2.1. Methods

2.1.1. Participants

Thirty-six undergraduate students from Ghent University took part in the experiment (mean age = 19.16, *SD* = 3.39, all females) in exchange for course credits and 5 euros. Sample size was calculated before data collection, assuming a medium effect size ($d = 0.3$) and aiming at a power of 0.85 to detect a significant interaction. All participants had normal or corrected-to-normal vision and twenty-nine reported to be right-handed. Data from six participants were discarded due to low task performance (individual mean accuracy exceeded by two standard deviations the group mean accuracy in at least one of the two tasks and/or accuracy in any of the two tasks or in response to catch trials was below 60%) resulting in a final sample size of thirty participants and a power of 0.78. All participants for this and the following experiments gave their informed consent prior to the beginning of the experiment, in accordance with the Declaration of Helsinki, and the protocols were consistent with the general ethical protocol of the Faculty.

2.1.2. Materials

The same set of stimuli as in a previous study was used (González-García et al., 2020). It consisted of 1550 pictures, grouped in two macro categories: animate non-human animals and inanimate objects (vehicles and musical instruments) (Brady, Konkle, Alvarez, & Oliva, 2013; Brodeur, Guérard, & Bouras, 2014; Griffin, Holub, & Perona, 2007; Konkle, Brady, Alvarez, & Oliva, 2010). All images had their background removed, were centered in a 200 × 200 pixels square and were converted to black and white.

2.1.3. Procedure

Stimuli presentation and response collection were performed with Psychopy (Peirce, 2007). Each trial started with a white fixation cross presented in the middle of the screen. Participants were instructed to press the spacebar to display the encoding screen and to hold it for as long as needed to memorize all the presented S-R mappings, with a maximum encoding time of 10 s. The encoding screen contained 2 or 4 mappings. Each mapping consisted of a new image associated with a bimanual response: “index” referred to both index fingers (keys “r” and “i”) and “middle” referred to both middle fingers (keys “e” and “o”). These responses were used instead of the more traditional left and right options to avoid automatic motor activations due to lateralized responses (Bundt, Bardi, Abrahamse, Brass, & Notebaert, 2015). Encoding screens with four mappings included two images of animals and two images of inanimate objects, whereas encoding screens with two mappings contained images from only one category. This was done to minimize the interference caused by having to memorize more than two semantically related items. It is worth pointing out that when the retro-cue selected two mappings, these always belonged to the same category, in order to make the conditions with procedural load of 2 similar across declarative loads. Participants were not informed in advance of the nature of the stimuli in the encoding screens. Once the participant released the spacebar (or after the 10 s deadline expired) the encoding screen disappeared and was replaced by a white fixation cross. Participants had to press again the spacebar and hold it for as long as desired (or for a maximum of 5 s) to display the retro-cue. It consisted of a red frame, selecting one or two of the spatial locations previously occupied by the mappings. Importantly, when the retro-cue selected two mappings, these were always both belonging to the upper or the lower row in the encoding screen. Participants were instructed that the retro-cue signaled which mapping(s) could be probed for the CRT and should therefore be prepared for implementation. Crucially, the retro-cue was not informative for the recognition task, as all the encoded mappings would be tested at the end of the trial. After the retro-cue disappeared, a white fixation cross appeared again on the screen for 2 s (cue-target interval, CTI). Right after the CTI, participants were presented with a CRT, which is considered a working memory task relying heavily on procedural WM (for simplicity, we will refer to this as “procedural task”). One single probe image was presented in the middle of the screen. The probe was always one of the images selected by the retro-cue (thus, the retro-cue was always valid, except for catch trials, see below) and the participant had to respond pressing the keys corresponding to the associated finger mapping (middle or index, bimanually). One crucial difference between high and low Procedural Load is that, when only one mapping was selected, the procedural task could be approached as a simple detection task (i.e., press the keys as soon as the probe appears on the screen) whereas when two proceduralized mappings were prepared, participants had to process the probe to perform the correct response. To address this issue and try to equate both conditions, 24 catch trials were added to the task (6 for each of the four load combinations). In these trials, instead of one of the selected images, a new image was presented as probe, and in this case participants were asked to press the spacebar. By adding these catch trials, even in the one-item condition, participants had to fully process the probe image before giving a response. As soon as the response was given or after a maximum response time of 2 s, the image was replaced by a fixation cross, staying on screen for 500 ms. Next, participants were presented with the recognition task (i.e., “declarative task”). The declarative probe contained the same number of S-R mappings as the encoding screen, therefore the retro-cue was completely irrelevant with respect to this second task. Participants had to say whether the mappings in the declarative probe screen were the same of those encoded at the beginning of the trial, by pressing with both fingers of one hand for “yes” and both fingers of the other hand for “no” (the sides for “yes” and “no” were the same for the whole experiment and counterbalanced between subjects; labels with “yes” and “no” were displayed at the bottom of the probe screen to reduce complexity of the task). In 50% of the trials, the memory probe showed the same mappings as in the encoding (“yes” response). In the other 50% of trials, two or more mappings were “non-matching”, that is, images were presented associated with a different response with respect to encoding (“no” response). Orthogonally, also the position of the mappings on the screen in the memory probe was manipulated: they could be presented in the same spatial location as in the encoding or in a different spatial location. Note that the task required to confirm the identity of the mappings, therefore the associations of images and responses, and not their spatial positions.

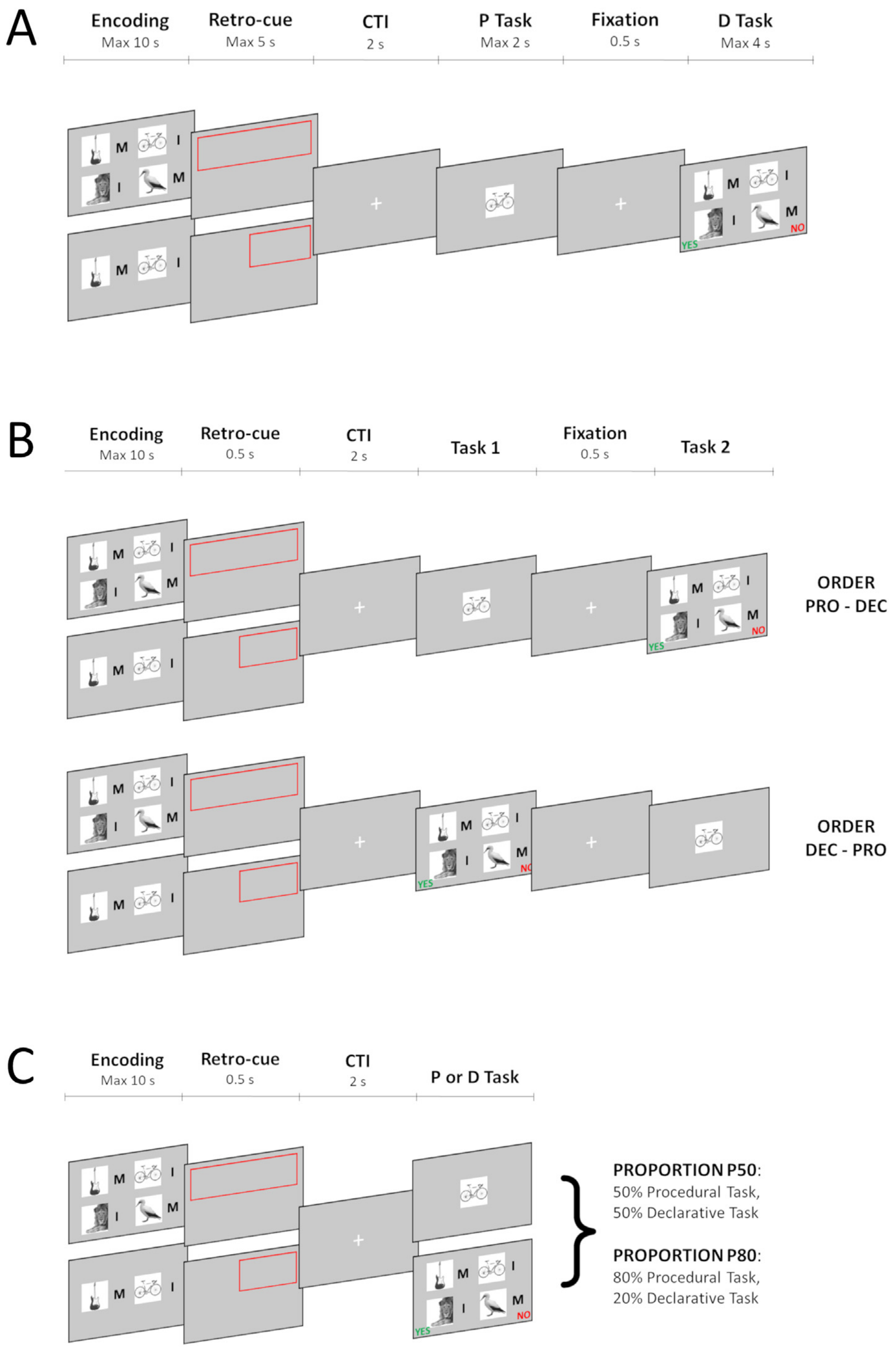


Fig. 1. Trial structure for each experiment. In all three experiments, four or two mappings are presented at the beginning of each trial (Declarative Load manipulation), followed by a retro-cue selecting two or one mapping (Procedural Load manipulation). A) Experiment 1: After the 2 s CTI, the procedural task is administered, followed by a short fixation (500 ms) and the declarative task. B) Experiment 2: Order manipulation. Order Pro-Dec is an exact replication of Experiment 1. In Order Dec-Pro the order of the two tasks is reversed. The two orders are randomized within blocks. C) Experiment 3: Proportion manipulation. In each trial, only one of the two tasks is presented. In blocks with ProportionP50, each task is presented in 50% of trials. In blocks with ProportionP80, 80% of trials contained a procedural task and 20% involved a declarative task (González-García et al., 2020).

This manipulation was introduced to discourage participants to rely on the perceptual similarity between the encoding and the probe arrays. In half of the non-match trials, a new image replaced one of the images of the encoding array. Again, this was done to make sure participants encoded all mappings. Maximum response time for the declarative task was 4 s. Finally, a red fixation cross stayed on screen for 1 s, signaling the inter-trial interval (Fig. 1A).

As mentioned before, we considered "Declarative Load" the number of S-R mappings presented in the encoding array (2 or 4) and "Procedural Load" the number of mappings selected by the retro-cue for the procedural task. Participants were instructed to use the information provided by the retro-cue to prepare to respond to the procedural probe (therefore, preparing to implement the mapping) while still maintaining in WM all the mappings for the second, declarative task. These load manipulations lead to four orthogonal conditions: high Declarative and high Procedural Load (Declarative, 4 mappings – Procedural, 2 mappings; D4P2), high Declarative and low Procedural Load (D4P1), low Declarative and high Procedural Load (D2P2) and low Declarative and low Procedural Load (D2P1). Participants completed 4 experimental blocks, each containing 54 trials, for a total of 216 trials. Specifically, for each load combination participants performed 48 regular trials and 6 catch trials.

Prior to the task, participants performed a practice session. Each mini-block of practice consisted of 14 trials, including all possible combinations of Declarative and Procedural Load and at least one catch trials. The only difference with the main task was the presence of feedback at the end of each trial, signaling the accuracy of the response or encouraging participants to respond faster in case no response was registered within the maximum response time. Performance was assessed at the end of each mini-block: if accuracy was above 80% in both tasks, practice was concluded, otherwise a new mini-block started, up to a maximum of 4 blocks. S-R mappings used during the practice were never presented again during the main task. The total duration of the experiment, including practice, main task and breaks was approximately 75 min.

2.1.4. Data analysis

A factorial design with one factor for Declarative Load (2 levels, high and low) and another for Procedural Load (2 levels, high and low) was used. Reaction times (RTs) and error rates (ER) were separately entered into 2 × 2 repeated measure ANOVAs. For completion, both frequentist and Bayesian statistics are computed. Bayes Factors (BF) quantify the amount of evidence supporting the hypothesis being tested: $BF_{10} > 3$ is considered evidence in favor of the alternative hypothesis, whereas $BF_{10} < 0.3$ (i.e., 1/3) represents evidence in favor of the null hypothesis. BFs falling between these two arbitrary boundaries are interpreted as inconclusive (Jeffreys, 1998; Rouder, Morey, Speckman, & Province, 2012; Schonbrodt & Wagenmakers, 2017). All statistical analyses for this and the next experiments were performed in JASP (Jasp Team, 2019).

In both tasks, only RTs of correct trials were used for the analyses. RTs were trimmed at the individual level by discarding trials in which response latency exceeded by 2 standard deviations the mean RT, separately for each of the four experimental condition. To confirm that the approach we adopted did not bias our results, we additionally performed all analyses for all experiments removing trials according to other trimming regimes, as suggested by (Jones, 2019). Namely, we also used 3 standard deviations from the mean, an “unbiased” criterium (i.e., removing RTs faster than 200 ms and exceeding 1500 ms for the procedural task and 3500 ms for the declarative task), and the Measure of Spread S_n as proposed by Rousseeuw and Croux (Rousseeuw & Croux, 1993). Notably, all these trimming regimes gave qualitatively similar results, suggesting that our results and their interpretations do not depend on the specific trimming regime we adopted.

Additionally, for Experiment 1 and 2, only trials in which the subject was accurate in both the declarative and procedural tasks were used.

For Experiment 1, our trimming procedure resulted in an average of 1.00 excluded trial ($SD = 0.82$) for the D4P2 condition, 2.03 ($SD = 0.80$) for D4P1, 2.17 ($SD = 0.90$) for D2P2, 2.10 ($SD = 0.91$) in the procedural task. Concerning the declarative task, we removed an average of 0.97 trials ($SD = 0.60$) for D4P2, 1.63 ($SD = 1.05$) for D4P1, 1.80 ($SD = 0.75$) for D2P2 and 2.20 ($SD = 0.83$) for D2P1.

2.2. Results

2.2.1. Procedural task

Repeated measures ANOVA on RTs revealed a significant main effect of Procedural Load ($F_{1,29} = 186.46$, $p < 0.001$, $\eta_p^2 = 0.86$, $BF_{10} > 100{,}000$) and Declarative Load ($F_{1,29} = 110.29$, $p < 0.001$, $\eta_p^2 = 0.79$, $BF_{10} > 100{,}000$). RTs were slower for both high Procedural Load ($Mean = 0.94$ s, $SD = 0.20$) compared to low Procedural Load ($Mean = 0.57$ s, $SD = 0.12$), and high Declarative Load ($Mean = 0.82$ s, $SD = 0.27$) compared to low Declarative Load ($Mean = 0.69$ s, $SD = 0.20$). Also the interaction between Declarative and Procedural Loads resulted to be significant ($F_{1,29} = 27.31$, $p < 0.001$, $\eta_p^2 = 0.48$, $BF_{10} = 28.67$). More specifically, the effect of Procedural Load was larger ($F_{1,29} = 194.8$, $p < 0.001$, $BF_{10} > 100{,}000$) in trials with high Declarative Load, compared with trials with low Declarative Load ($F_{1,29} = 134.1$, $p < 0.001$, $BF_{10} > 100{,}000$). The combination of high load on both WM components lead to the slowest RTs (D4P2, $Mean = 1.03$ s, $SD = 0.20$), followed by D2P2 ($Mean = 0.85$ s, $SD = 0.16$), D4P1 ($Mean = 0.62$ s, $SD = 0.13$) and, lastly, D2P1 ($Mean = 0.54$ s, $SD = 0.11$) (Fig. 2A, left panel).

Similarly, an ANOVA on ER showed a significant main effect of Procedural Load ($F_{1,29} = 17.00$, $p < 0.001$, $\eta_p^2 = 0.37$, $BF_{10} = 239.73$), with high load causing more errors ($Mean = 0.07$, $SD = 0.07$) than low load ($Mean = 0.03$, $SD = 0.05$), and a significant main effect of Declarative Load ($F_{1,29} = 28.31$, $p < 0.001$, $\eta_p^2 = 0.49$, $BF_{10} > 100{,}000$), driven by more errors in trials with four encoded mappings ($Mean = 0.08$, $SD = 0.08$), compared with trials with two encoded mappings ($Mean = 0.02$, $SD = 0.03$). The interaction of the two loads resulted to be not significant ($F_{1,29} = 1.22$, $p = 0.287$, $\eta_p^2 = 0.04$, $BF_{10} = 1.41$) (Fig. 2A, right panel).

At the same time, error rates in response to procedural catch trials did not differ ($t_{29} = 1.21$, $p = 0.236$, $d = 0.22$, $BF_{10} = 0.38$) across high ($Mean = 0.10$, $SD = 0.09$) and low ($Mean = 0.13$, $SD = 0.11$) Procedural Load.

2.2.2. Declarative task

RTs showed only a significant main effect of Declarative Load ($F_{1,29} = 868.03$, $p < 0.001$, $\eta_p^2 = 0.97$, $BF_{10} > 100{,}000$). Participants responded faster in trials with two mappings ($Mean = 1.42$ s, $SD = 0.24$) compared to trials with four mappings ($Mean = 2.20$ s, $SD = 0.32$). Procedural Load and the interaction of the two loads showed no significant effect ($F_{1,29} = 1.25$, $p = 0.27$, $BF_{10} = 0.26$ and $F_{1,29} = 1.76$, $p = 0.195$, $BF_{10} = 0.36$, respectively) (Fig. 2B, left panel).

The ANOVA on error rates yielded a significant main effect of Declarative Load ($F_{1,29} = 84.33$, $p < 0.001$, $\eta_p^2 = 0.74$, $BF_{10} > 100{,}000$). Participants were more accurate in responding to low Declarative Load trials ($Mean = 0.07$, $SD = 0.07$) compared to high Declarative Load trials ($Mean = 0.18$, $SD = 0.08$). Procedural Load and the interaction of the two loads showed moderate evidence for a null effect ($F_{1,29} = 0.98$, $p = 0.330$, $BF_{10} = 0.24$ and $F_{1,29} < 0.01$, $p = 0.981$, $BF_{10} = 0.23$, respectively) (Fig. 2B, right panel).

2.3. Discussion

The results of this first experiment confirmed the expected within-component load effects: the higher the number of mappings to encode, the worse the performance in the recognition task and, analogously, the higher the number of mappings to proceduralize, the worse the performance in the CRT, reflecting the effectiveness of our load manipulations. Interestingly, the results of this experiment indicate an asymmetrical cross-task load effect. Increasing the encoding and declarative maintenance demands lead to a significantly lower performance in the procedural task (occurring in between encoding and recognition) compared to a low declarative load condition with only two mappings. Conversely, the procedural load showed no effect on the recognition task, which is heavily taxed only by the declarative load in both RTs and error rates.

Nevertheless, the order in which the two tasks were presented might have played a role in determining this asymmetrical effect. Since the declarative task occurred always after the procedural one, it is possible that the procedural load became irrelevant and/or was dropped from WM immediately after responding to the procedural task, therefore causing no effect on the subsequent recognition task.

3. Experiment 2

To investigate whether this asymmetry was due to the order in which the two tasks were presented, in a second experiment the order of the two tasks was randomized. In this way, the recognition task was equally likely to occur immediately after the retro-cue, when the action-oriented representations for the procedural task are highly activated and still relevant, or after implementing the mapping (as in Experiment 1).

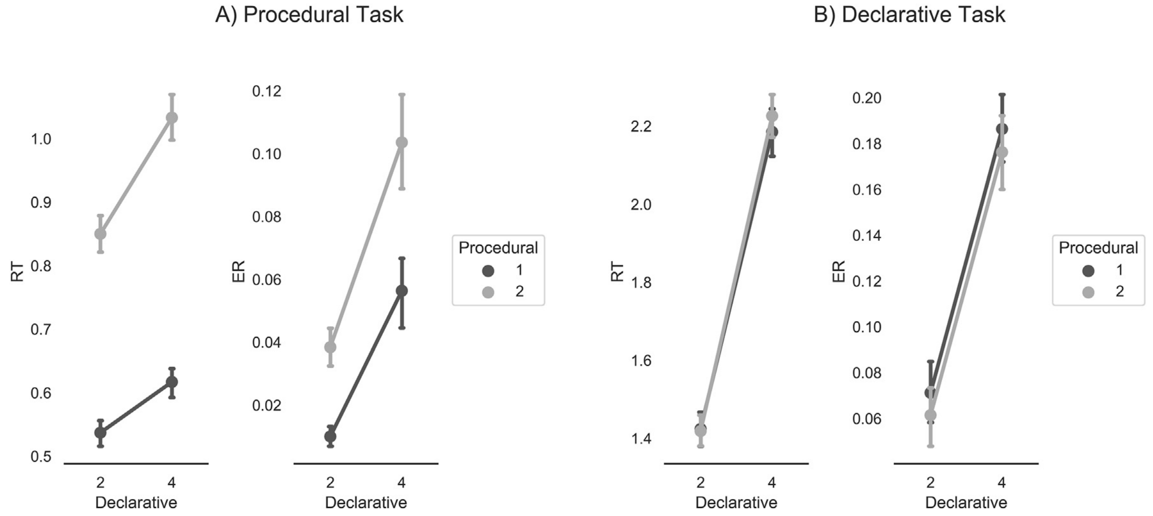


Fig. 2. A) Reaction times (left panel) and Error rates (right panel) for the procedural task in Experiment 1. B) Reaction times (left panel) and Error rates (right panel) for the declarative task in Experiment 1. Error bars represent standard error mean.

3.1. Methods

3.1.1. Participants

Thirty-five undergraduate students from Ghent University took part in the experiment (mean age = 21.20, *SD* = 5.30, 22 females, 31 right-handed) in exchange for course credits and 5 euros. Three participants did not complete the entire experiment and were therefore excluded from further analyses. The same exclusion criteria were applied as in Experiment 1, and this resulted in two participants being discarded, leading to a final sample size of thirty and power of 0.80 to detect a significant two-way interaction.

3.1.2. Procedure

The encoding phase was the same as in Experiment 1. In contrast to Experiment 1, in Experiment 2 the retro-cue was not self-paced, but appeared 250 ms after the offset of the encoding screen and stayed on screen for 0.5 s. This was done because in Experiment 1 there was no significant difference in the duration of the self-paced retro-cue between high and low Procedural Load (t_{29} = 0.493, p = 0.626, BF_{10} = 0.22) and its mean duration was 0.55 s (*SD* = 0.33). The order of the two tasks was pseudo-randomized. In 50% of trials, the procedural task appeared first, and the declarative task was second (exactly as in Experiment 1, from now on we will refer to this as “Order Pro-Dec”). In the remaining 50% of trials, the declarative task was presented first and the procedural task second (“Order Dec-Pro”) (Fig. 1B). The main aim of this manipulation was to investigate the performance in the declarative task in Order Dec-Pro, as the task needed to be performed while still maintaining the prepared mappings ready for implementation. It is worth noting that the procedural task in Order Dec-Pro is performed after the presentation of the recognition screen. Since this could be displaying correct mappings, thus being another chance of encoding, or incorrect mappings, potentially leading to interference, the performance in the procedural task was analyzed only for trials following Order Pro-Dec. In this experiment, participants completed a total of 288 trials, divided in 6 blocks. Each block contained combinations of all load conditions and task orders, and 8 catch trials. Specifically, for each load and order combination, participants performed 30 regular trials (for a total of 240) and 8 catch trials (for a total of 48). Importantly, in each block the number of switches between task orders was the same and the total number of trials with Order Pro-Dec and trials with Order Dec-Pro was equated. This was done to ensure that participants did not favor one specific strategy and considered each trial equally likely to belong to both orders up to the presentation of the first probe. Participants performed a practice session before the main task, with the same structure as in Experiment 1. The total duration of the experiment, including the practice and the breaks, was approximately 75 min.

3.1.3. Data analysis

Given the characteristics of the experimental design, the performance in the procedural task was analyzed only for the trials of Order Pro-Dec. RTs and ER were analyzed by means of 2 × 2 repeated measures ANOVAs with Procedural Load (high vs low) and Declarative Load (high vs load) as independent variables. On the contrary, RTs and ER of the declarative task were entered in 2 (Declarative Load: high vs low) x 2 (Procedural Load: high vs load) x 2 (Order: Pro-Dec vs Dec-Pro) repeated measure ANOVAs. Data trimming was performed as in Experiment 1. For the procedural task (Order Pro-Dec), we excluded an average of 0.80 trials (*SD* = 0.65) for D4P2, 1.10 (*SD* = 0.60) for D4P1, 1.27 (*SD* = 0.70) for D2P2 and 1.10 (*SD* = 0.65) for D2P1. For the declarative task, the following number of trials were removed from our analyses: 0.80 (*SD* = 0.70) for D4P2 – Order Pro-Dec, 0.63 (*SD* = 0.66) for D4P1 – Order Pro-Dec, 1.13 (*SD* = 0.56) for D2P2 – Order Pro-Dec, 1.37 (*SD* = 0.55) for D2P1 – Order Pro-Dec, 0.87 (*SD* = 0.62) for D4P2 – Order Dec-Pro, 0.67 (*SD* = 0.54) for D4P1 – Order Dec-Pro, 1.13 (*SD* = 0.56) for D2P2 - Order Dec-Pro and 0.97 (*SD* = 0.66) for D2P1 – Order Dec-Pro.

3.2. Results

3.2.1. Procedural task[1]

In the RTs, we found a significant main effect of Declarative Load ($F_{1,29}$ = 85.72, p < 0.001, η_p^2 = 0.75, BF_{10} > 100,000) and Procedural Load ($F_{1,29}$ = 38.75, p < 0.001, η_p^2 = 0.57, BF_{10} > 100,000). Specifically, participants were faster when the retro-cue selected only one mapping (*Mean* = 0.80 s, *SD* = 0.14) compared

---

[1] The results presented in this section refer only to Order Pro-Dec. Concerning the effects of the declarative and procedural loads, the ANOVAs including also the variable Order gave qualitatively similar results. The effect of Order was found to be significant for both RTs ($F_{1,31}$ = 59.23, p < 0.001, η_p^2 = 0.67, BF_{10} > 100,000) and ER ($F_{1,29}$ = 41.56, p < 0.001, η_p^2 = 0.59, BF_{10} > 100,000). Responses were slower and less accurate in Order Dec-Pro. This is in line with our hypothesis that performing the CRT after being presented with the (matching or non-matching) recognition screen, as is the case in Order Dec-Pro, might introduce unwanted confounds. Additionally, for RTs also the interaction Order x Declarative load was significant ($F_{1,29}$ = 20.70, p < 0.001, η_p^2 = 0.42, BF_{10} = 4.11).

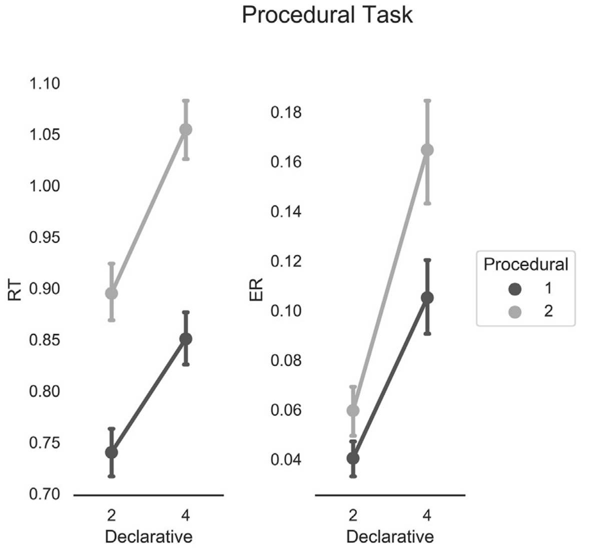


Fig. 3. RTs (left panel) and ER (right panel) of the procedural task in Experiment 2. Note that only results for Order Pro-Dec are reported). Error bars represent standard error mean.

with two mappings (*Mean* = 0.97 s, *SD* = 0.16). Similarly, in trials with high Declarative Load RTs were slower (*Mean* = 0.95 s, *SD* = 0.16) than in trials with low Declarative Load (*Mean* = 0.82 s, *SD* = 0.14). Additionally, the interaction of the two loads reached the threshold for significance ($F_{1,29}$ = 4.01, p = 0.055, η_p^2 = 0.12, BF_{10} = 1.90). In trials with four mappings, high Procedural Load caused slower ($F_{1,29}$ = 47.41, p < 0.001, BF_{10} > 100,000) RTs (*Mean* = 1.05 s, *SD* = 0.16) than low Procedural Load (*Mean* = 0.85 s, *SD* = 0.15). Analogously, RTs were faster ($F_{1,29}$ = 22.13, p < 0.001, BF_{10} = 427.6) when both loads were low (*Mean* = 0.74 s, *SD* = 0.13) compared with trials in which the Declarative Load was low, but the retro-cue selected two mappings (*Mean* = 0.89 s, *SD* = 0.15) (Fig. 3).

The ANOVA on error rates revealed a significant main effect of Declarative Load ($F_{1,29}$ = 40.12, p < 0.001, η_p^2 = 0.58, BF_{10} > 100,000); participants made less errors when they only had to maintain two mappings (*Mean* = 0.05, *SD* = 0.05), compared with four mappings (*Mean* = 0.13, *SD* = 0.10). The main effect of Procedural Load was also significant ($F_{1,29}$ = 11.25, p = 0.002, η_p^2 = 0.28, BF_{10} = 21.27). High Procedural Load led to more errors (*Mean* = 0.11, *SD* = 0.08) than low Procedural Load (*Mean* = 0.07, *SD* = 0.06). The interaction between loads resulted to be not significant, although the Bayes Factor suggests anecdotal evidence in favor of the alternative hypothesis ($F_{1,29}$ = 3.21, p = 0.084, η_p^2 = 0.10, BF_{10} = 3.17) (Fig. 3).

Participants responded equally accurately (t_{29} < 0.001, p = 1, d < 0.001, BF_{10} = 0.19) to procedural catch trials in the high (*Mean* = 0.13, *SD* = 0.14) and low (*Mean* = 0.13, *SD* = 0.12) Procedural Load conditions.

3.2.2. *Declarative task*

The ANOVA on RTs showed a significant main effect of Declarative Load ($F_{1,29}$ = 426.70, p < 0.001, η_p^2 = 0.94, BF_{10} > 100,000) and of Order ($F_{1,29}$ = 44.48, p < 0.001, η_p^2 = 0.60, BF_{10} > 100,000). Participants were slower with high Declarative Load (*Mean* = 2.23 s, *SD* = 0.32) compared with low Declarative Load (*Mean* = 1.50 s, *SD* = 0.20), and in trials of Order Dec-Pro (*Mean* = 1.94 s, *SD* = 0.27) than trials of Order Pro-Dec (*Mean* = 1.79 s, *SD* = 0.25). No effect of Procedural Load on RTs was found ($F_{1,29}$ = 0.07, p = 0.79, BF_{10} = 0.07) nor any interaction (all *Fs* < 3.07, *ps* > 0.09, all BF_{10} < 0.78). The Bayesian counterpart of this ANOVA revealed that the BF_{10} for the effect of the Procedural Load was 0.07, and the BF_{10} for the interactions of Declarative x Procedural Load and Procedural Load x Order were 0.07 and 0.10, respectively, therefore providing strong evidence towards the null hypothesis that the Procedural Load does not have an effect on the RTs of the declarative task (Fig. 4A).

For error rates, we found a significant main effect of Declarative Load ($F_{1,29}$ = 125.67, p < 0.001, η_p^2 = 0.81, BF_{10} > 100,000): participants made more errors in trials with four mappings (*Mean* = 0.19, *SD* = 0.11) compared with trials with two mappings (*Mean* = 0.06, *SD* = 0.06). Interestingly, also the effect of Procedural Load ($F_{1,29}$ = 6.58, p = 0.016, η^2p = 0.18, BF_{10} = 1.21) and the interaction of Procedural Load and Order ($F_{1,29}$ = 5.62, p = 0.025, η_p^2 = 0.16, BF_{10} = 1.29) were found to be significant, although the associated Bayes factors indicate inconclusive evidence in favor of the alternative hypothesis. Surprisingly, high Procedural Load led to less errors (*Mean* = 0.11, *SD* = 0.08) than low Procedural Load (*Mean* = 0.14, *SD* = 0.09). Planned comparisons revealed significant differences between high (*Mean* = 0.10, *SD* = 0.07) and low Procedural Load (*Mean* = 0.15, *SD* = 0.11) only in Order Dec-Pro ($F_{1,29}$ = 11.75, p = 0.002, BF_{10} = 19.35), whereas no differences were found between high (*Mean* = 0.12, *SD* = 0.09) and low Procedural Load (*Mean* = 0.12, *SD* = 0.07) in Order Pro-Dec ($F_{1,29}$ = 0.02, p = 0.89, BF_{10} = 0.20). Specifically, error rates differed between the two Orders only for trials with low Procedural Load ($F_{1,29}$ = 6.67, p = 0.015, BF_{10} = 3.18), with significantly more errors in Order Dec-

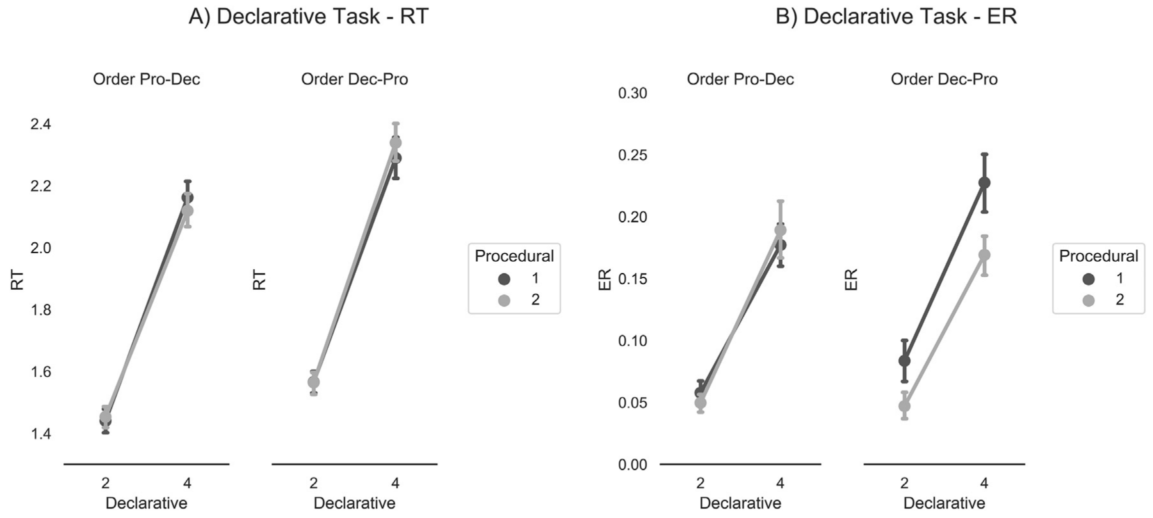


Fig. 4. Results of the declarative task in Experiment 2. A) RTs in Order Pro-Dec (left panel) and Order Dec-Pro (right panel). B) Error rates in Order Pro-Dec (left panel) and Order Dec-Pro (right panel). Error bars represent standard error mean.

Pro (*Mean* = 0.15, *SD* = 0.11) compared with Order Pro-Dec (*Mean* = 0.12, *SD* = 0.07). No differences were found ($F_{1,29}$ = 1.03, p = 0.319, BF_{10} = 0.31) in trials with high Procedural Load between Order Pro-Dec (*Mean* = 0.12, *SD* = 0.09) and Order Dec-Pro (*Mean* = 0.11, *SD* = 0.07). (Fig. 4B).

3.3. Discussion

The pattern of results in Order Pro-Dec replicates the results of Experiment 1, showing strong within- and cross-component load effects on the procedural task but only an effect of declarative load on the declarative task.[2] Surprisingly, in Order Dec-Pro, the procedural load significantly affected the declarative task, in which participants made more errors when the retro-cue selected one mapping (low procedural load) compared to the condition in which the retro-cue selected two mappings (high procedural load).

This effect seems to be driven by a higher cost in the low procedural load condition, suggesting that maintaining one single mapping ready for implementation can be detrimental for the recall of the whole set of declarative representations. We assume the retro-cue to have a twofold effect: the selected mappings are first brought into the focus of attention and then reformatted into a highly prioritized state optimized for action (Myers et al., 2017). Whether the observed effect is due to the attentional prioritization of the selected mappings (Myers et al., 2018; Rerko & Oberauer, 2013) or, in contrast, to the decrease in the quality of unattended declarative representations (Bays, Catalao, & Husain, 2009; Luck & Vogel, 1997), is a question that is difficult to directly address with the current dataset.

Moreover, the present experimental design cannot provide definitive evidence to confirm that the selected mappings are immediately reformatted into a procedural representation. For instance, these results could also be explained by assuming the retro-cue to be only selecting the subset of relevant mappings, but not triggering their reformatting. Since it was equally likely to be presented with the procedural or the declarative task immediately after the retro-cue, it might have been strategically advantageous for participants to use the information provided by the retro-cue only to select the mapping(s), and to postpone the reformatting to the moment of the onset of the procedural probe.

In an attempt to provide evidence supporting the role of the retro-cue in triggering the immediate proceduralization of the selected mappings and to further investigate the quality of uncued declarative representations, we ran a third experiment in which only one of the two tasks was presented in each trial. The proportion of the two tasks varied between blocks, to create a condition in which it was strategically optimal to reformat the selected mapping(s) into a prioritized procedural representation (González-García et al., 2020). Therefore, the third experiment was meant to investigate the cross-component load effects across blocks with different proportions of procedural and declarative trials in a single-task approach.

4. Experiment 3

While proceduralization is traditionally assessed through compatibility effects in inducer-diagnostic experimental settings (González-García et al., 2020; Liefooghe et al., 2012; Meiran et al., 2015), nesting diagnostic trials within our experimental design would have increased dramatically the difficulty of the task, making it effectively not doable. Therefore, in Experiment 3 we manipulated the proportion of procedural and declarative probes to manipulate the strategical advantage of proceduralizing. Thus, for each trial, only one of the tasks would appear, either the procedural or the declarative one, but the proportion of trials with the procedural and the declarative tasks changed between blocks. The aim of this manipulation was to induce an expectation regarding which task would be more likely to occur, therefore creating a condition in which it was optimal to proceduralize the selected mapping. Crucially, previous research using a similar design has shown that a high proportion of trials with a procedural task, in detriment of declarative probes, leads to instruction-based congruency effects only for selected mappings (Whitehead & Egner, 2018b), suggesting that in such a setting, retro-cues trigger the transformation of mappings into an action-oriented (procedural) representation (González-García et al., 2020).

Therefore, this third experiment has two main goals. First, comparing the cross-component load effects across different proportion conditions. If the results are similar across conditions, it is reasonable to assume that the ongoing cognitive processes are analogous, implying that proceduralization is taking place even when the number of procedural trials is not predominant over declarative trials (as in Experiment 2). In contrast, differences in the effects of declarative and procedural load manipulations across proportions would suggest that the strategical approach to our task depends on the proportion of trial type, possibly implying that when the participant does not have a clear expectation of which task is going to be presented first, the selected mappings are kept in a declarative format to be proceduralized at probe presentation. Second, assessing the effect of procedural load on the declarative task in single-task trials. If the effect we found in Experiment 2 is due to a decrease in the quality of unselected declarative representation, this should be present also in this third experiment. On the other hand, the absence of such an effect would suggest that procedural load exerts an influence on the declarative task only when the mappings are still maintained in a highly prioritized procedural format for an upcoming CRT.

4.1. Methods

4.1.1. Participants

Forty-four participants took part in this third experiment (mean age = 18.88, *SD* = 1.51, 32 females, 38 right-handed) in exchange for course credits and 5 euros. Recruitment of participants was analogous to Experiment 1 and 2 and they were discarded based on performance following the same criteria. This resulted in eight participants being discarded, providing a final sample of thirty-six participants and a power of 0.86.

4.1.2. Procedure

The experimental session consisted of 288 trials, whose structure was the same as in Experiment 2 up to the cue-target interval. In the present experiment, the probe consisted of either the procedural or the declarative task, hence each trial contained only one of the two tasks (Fig. 1C). Crucially, the trials were divided in six blocks of equal length. Three of them had an equal number of declarative and procedural trials (i.e., 50% of procedural probes (72), we will refer to this condition as

---

[2] For completeness, we additionally checked whether performance in our task was influenced by switch costs between Orders. Therefore, we also run all the ANOVAs including the additional factor Sequence with two levels (Switch and Repeat), referring to whether the Order of the two tasks (Pro-Dec or Dec-Pro) was the same (Repeat) or different (Switch) in trial t-1 with respect to trial t.

Procedural Task. For RTs, neither the factor Sequence ($F_{1,29}$ = 0.08, p = 0.776, η_p^2 = 0.003, BF_{10} = 0.02) nor any interaction with other factors (all ps > 0.142 and all BF_{10} < 0.03) resulted to be significant. Analogously, the factor Sequence ($F_{1,29}$ = 0.32, p = 0.576, η_p^2 = 0.01, BF_{10} = 0.02) and all its interactions (all ps > 0.29 and all BF_{10} < 0.02) were not significant for Error rates.

Declarative Task. The ANOVA on the RTs for the Declarative task yielded similar results: the factor Sequence had no significant effect ($F_{1,29}$ = 0.15, p = 0.698, η_p^2 = 0.005, BF_{10} = 0.02), as well as all its interactions (all ps > 0.309 and all BF_{10} < 0.02). Finally, the same pattern was observed for Error rates, with no effects of Sequence ($F_{1,29}$ = 0.13, p = 0.717, η_p^2 = 0.01, BF_{10} = 0.02) and associated interactions (all ps > 0.11 and all BF_{10} < 0.02).

Therefore, we concluded that our observed asymmetric effects of Declarative and Procedural Loads are not modulated by the switch cost.

ProportionP50, analogous to the order manipulation of Experiment 2); in the other three blocks, 80% of trials (114) involved the procedural task, whereas the remaining trials (30) ended with a declarative recognition task (ProportionP80), as in González-García et al. (2020). ProportionP50 and ProportionP80 contained 12 and 18 catch trials, respectively, and each cell of the design had approximately the same number of trials for each load combination. The block order was randomized, but participants were informed at the beginning of each block on the current proportion of procedural and declarative tasks. This was done to emphasize the advantage of proceduralizing in the condition with many procedural trials. Prior to the main task, participants performed a practice session, with a structure similar to the previous experiments. In the practice session the number of procedural and declarative trials was the same, to make participants familiarize equally with both tasks. The total duration of this experiment was approximately 75 min.

4.1.3. Design

RTs and error rates were entered in 2 (Declarative Load: high vs low) x 2 (Procedural Load: high vs low) x 2 (Proportion: ProportionP50 vs ProportionP80) repeated measure ANOVAs. Data trimming was performed as in Experiment 1 and 2. For the procedural task, we excluded the following average number of trials per condition: 0.47 (*SD* = 0.60) for D4P2 – P50, 0.75 (*SD* = 0.49) for D4P1 – P50, 0.78 (*SD* = 0.41) for D2P2 – P50, 0.78 (*SD* = 0.41) for D2P1 – P50, 0.72 (*SD* = 0.69) for D4P2 – P80, 1.94 (*SD* = 0.74) for D4P1 – P80, 1.08 (*SD* = 0.64) for D2P2 – P80 and 1.67 (*SD* = 0.64) for D2P1 – P80. For the declarative task, we trimmed the following number of trials, averaged for condition: 0.44 (*SD* = 0.50) for D4P2 – P50, 0.44 (*SD* = 0.50) for D4P1 – P50, 0.53 (*SD* = 0.50) for D2P2 – P50, 0.61 (*SD* = 0.54) for D2P1 – P50, 0.05 (*SD* = 0.23) for D4P2 – P80, 0 (*SD* = 0) for D4P1 – P80, 0.17 (*SD* = 0.37) for D2P2 – P80 and 0.08 (*SD* = 0.28) for D2P1 – P80.

4.2. Results

4.2.1. Procedural task

For RTs, we obtained a significant main effect of Declarative Load ($F_{1,35} = 84.36, p < 0.001, \eta_p^2 = 0.71$, $BF_{10} > 100{,}000$), of Procedural Load ($F_{1,35} = 37.60$, $p < 0.001$, $\eta_p^2 = 0.52$, $BF_{10} > 100{,}000$) and of Proportion ($F_{1,35} = 6.59$, $p = 0.015$, $\eta_p^2 = 0.16$, $BF_{10} = 2.25$). Higher Declarative Load led to slower RTs (*Mean* = 0.96 s, *SD* = 0.29) than low Declarative Load (*Mean* = 0.77 s, *SD* = 0.22); analogously, with high Procedural Load, RTs were slower (*Mean* = 0.98 s, *SD* = 0.25) compared with low Procedural Load (*Mean* = 0.76 s, *SD* = 0.25). Additionally, participants were faster in ProportionP80 (80% of procedural trials; *Mean* = 0.84 s, *SD* = 0.25) compared to ProportionP50 (50% of procedural trials; *Mean* = 0.89 s, *SD* = 0.29), confirming that the proportion manipulation was effective in favoring the procedural task in ProportionP80. The interaction of Declarative Load and Procedural Load also resulted to be significant ($F_{1,35} = 7.05, p = 0.012$, $\eta_p^2 = 0.17$, $BF_{10} = 1.06$). Planned comparisons revealed that the effect of Declarative Load was larger in trials in which the retro-cue selected two mappings ($F_{1,35} = 109.87, p < 0.001$, $BF_{10} > 100{,}000$) then one mapping ($F_{1,35} = 40.08$, $p < 0.001$, $BF_{10} > 100{,}000$). Specifically, with high Procedural Load, response latencies were longer when the encoded mappings were four (*Mean* = 1.09 s, *SD* = 0.24) compared with two (*Mean* = 0.87 s, *SD* = 0.21). Analogously, with low Procedural Load participants were slower with four (*Mean* = 0.84 s, *SD* = 0.28) compared with two mappings (*Mean* = 0.67 s, *SD* = 0.20) (Fig. 5A). This pattern of results is analogous to that of Experiment 1. The factor Proportion did not interact significantly neither with Declarative Load ($F_{1,35}$ 0.11, $p = 0.75$, $BF_{10} = 0.36$), nor with Procedural Load ($F_{1,35} = 0.16, p = 0.69$, $BF_{10} = 0.33$). The three-way interaction between all factors was also not significant ($F_{1,35} = 0.56$, $p = 0.46$, $BF_{10} = 0.03$).

The ANOVA on error rates yielded a significant main effect of both the Declarative ($F_{1,35} = 34.84$, $p < 0.001$, $\eta_p^2 = 0.50$, $BF_{10} > 100{,}000$) and the Procedural Load ($F_{1,35} = 17.16, p < 0.001$, $\eta_p^2 = 0.33$, $BF_{10} = 6957.48$). As in the previous experiments, higher loads implied worse performance, both for the declarative component (for D4: *Mean* = 0.19, *SD* = 0.16; for D2: *Mean* = 0.08, *SD* = 0.09) and the procedural component (for P2: *Mean* = 0.17, *SD* = 0.15; for P1: *Mean* = 0.10, *SD* = 0.11). The interaction of Declarative Load and Proportion was also significant, although the Bayes Factor associated with this effect does not support the frequentist statistics ($F_{1,35} = 4.77$, $p = 0.036$, $\eta_p^2 = 0.12$, $BF_{10} = 0.15$). Planned comparisons revealed a larger effect of the Declarative Load in ProportionP80 ($F_{1,35} = 43.10$, $p < 0.001$, $BF_{10} > 10{,}000$), with more errors with four mappings (*Mean* = 0.20, *SD* = 0.15) than with two (*Mean* = 0.07, *SD* = 0.07); compared to ProportionP50 ($F_{1,35} = 17.55, p < 0.001, BF_{10} = 146.5$), in which high Declarative Load led to an average error rate of 0.17 (*SD* = 0.17) while low Declarative Load trials had an error rate of 0.09 (*SD* = 0.10) (Fig. 5B). The factor Proportion resulted to be non-significant ($F_{1,35} = 0.76, p = 0.78, BF_{10} = 0.08$), as well as its interaction with Procedural Load ($F_{1,35} = 0.34$, $p = 0.56$, $BF_{10} = 0.06$) and with Procedural and Declarative Loads ($F_{1,35} = 0.88$, $p = 0.35$, $BF_{10} = 0.01$).

Error rates in response to procedural catch trials were entered in a repeated measures ANOVA in 2 (Procedural Load: 1 vs 2) x 2 (Proportion: ProportionP50 vs ProportionP80). The main effects of Procedural Load ($F_{1,35} = 0.60, p = 0.47, BF_{10} = 0.15$) and Proportion ($F_{1,35} = 0.17, p = 0.73, BF_{10} = 0.14$) resulted to be not significant. The interaction was marginally significant, but the Bayes Factor provided evidence in favor of the null hypothesis ($F_{1,35} = 4.33$, $p = 0.045$, $\eta_p^2 = 0.11$, $BF_{10} = 0.08$).

4.2.2. Declarative task

The ANOVA on RTs showed only a significant main effect of Declarative Load ($F_{1,35} = 289.70$, $p < 0.001$, $\eta_p^2 = 0.89$, $BF_{10} > 100{,}000$), with slower reaction times for the higher load (*Mean* = 2.35 s, *SD* = 0.44) compared to low load (*Mean* = 1.68 s, *SD* = 0.28), and a main effect of Proportion ($F_{1,35} = 6.55, p = 0.015$, $\eta_p^2 = 0.16$, $BF_{10} = 8.49$). Participants were faster in ProportionP50 (*Mean* = 1.96 s, *SD* = 0.46), where the declarative task occurred more often, compared to ProportionP80 (*Mean* = 2.06 s, *SD* = 0.53) (Fig. 6A). Notably, neither the effect of Procedural Load ($F_{1,35} = 1.53$, $p = 0.224$, $\eta_p^2 = 0.04$, $BF_{10} = 0.18$) nor its interaction with Declarative Load ($F_{1,35} = 0.58, p = 0.45, \eta_p^2 = 0.01, BF_{10} = 0.14$) were significant. Again, the Bayes Factor provided moderate evidence in favor of the null hypothesis. The interactions of the factor Proportion with Declarative Load ($F_{1,35} = 0.72$, $p = 0.401$, $\eta_p^2 = 0.02$, $BF_{10} = 0.44$) and with Procedural Load ($F_{1,35} = 2.63, p = 0.114, \eta_p^2 = 0.07, BF_{10} = 0.20$), as well as the three-way interaction ($F_{1,35} = 0.00, p = 0.972, \eta_p^2 = 0.00$, $BF_{10} = 0.01$) were not significant.

For error rates, we found only a significant main effect of Declarative Load ($F_{1,35} = 53.72$, $p < 0.001$, $\eta_p^2 = 0.61$, $BF_{10} > 100{,}000$). As expected, participants were less accurate with four mappings (*Mean* = 0.23, *SD* = 0.19) than with two mappings (*Mean* = 0.10, *SD* = 0.12). Crucially, neither the effect of Procedural Load was significant ($F_{1,35} = 0.34, p = 0.57, \eta_p^2 = 0.01, BF_{10} = 0.06$), nor any interaction (all *F*s < 1.67, *p*s > 0.2 and all $BF_{10} < 0.15$). These values are usually considered moderate to strong evidence in favor of the null hypothesis that the Procedural Load has no effect on the error rates of the declarative task (Jeffreys, 1998; Rouder et al., 2012). Here, the factor Proportion was not significant ($F_{1,35} = 1.85$, $p = 0.183$, $\eta_p^2 = 0.05$, $BF_{10} = 0.15$). Moreover, it did not interact significantly with Declarative Load ($F_{1,35} = 1.66$, $p = 0.206$, $\eta_p^2 = 0.04$, $BF_{10} = 0.16$), with Procedural Load ($F_{1,35} = 0.33$, $p = 0.571$, $\eta_p^2 = 0.01$, $BF_{10} = 0.02$), nor with both of them ($F_{1,35} = 1.02, p = 0.320$, $\eta_p^2 = 0.03$, $BF_{10} = 0.00$) (Fig. 6B).

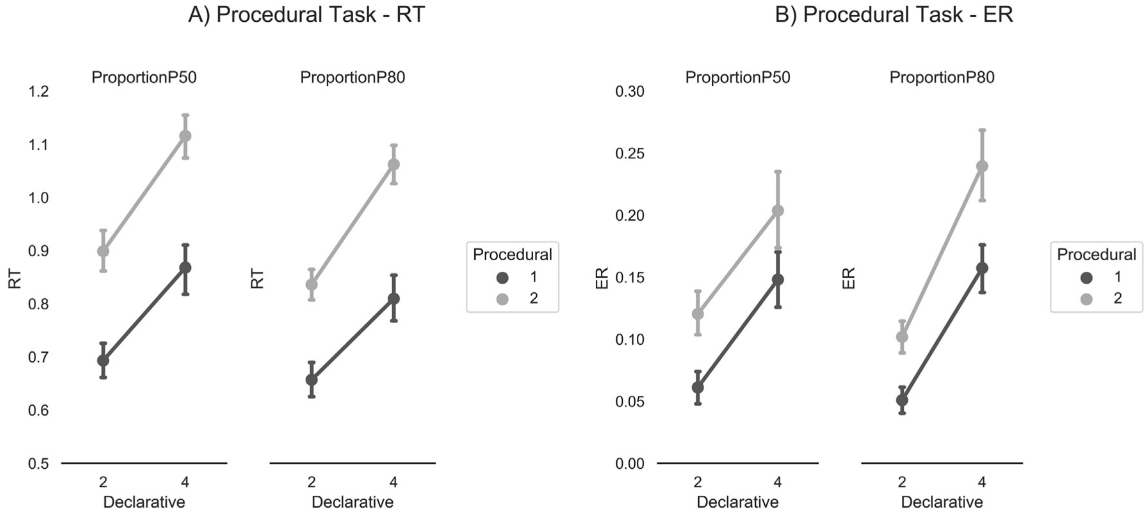


Fig. 5. Results of the procedural task in Experiment 3. A) RTs in ProportionP50 (left panel) and ProportionP80 (right panel). B) Error rates in ProportionP50 (left panel) and Proportion80 (right panel). Error bars represent standard error mean.

4.3. Discussion

In this third experiment, the crucial manipulation was the proportion of procedural and declarative probes between blocks. The effect of proportion, significant in the RTs of both tasks, followed a clear pattern: in ProportionP80, participants expected the procedural task more often than the declarative task, leading to faster reaction times in the former and slower reaction times in the latter, compared to ProportionP50. This suggests the effectiveness of the proportion manipulation in inducing an expectation for the upcoming task (Whitehead & Egner, 2018b).

In line with the previous two experiments, the procedural task was highly affected by both declarative and procedural load. On the contrary, the declarative task was uniquely influenced by the declarative load in both RTs and ER, replicating the one-way cross-component load effect.

These results allow us to draw two conclusions. First, the effects of declarative and procedural loads are consistent across proportions. Assuming in ProportionP80 proceduralization is occurring (González-García et al., 2020), we can infer it is occurring to a similar extent in ProportionP50 (and thus Experiment 2), supporting the assumption that our procedural load manipulation effectively targets the procedural component of WM and cannot be reduced to a simple orienting of attention towards the declarative representations of the selected items.

Furthermore, the effect of procedural load on error rates of the declarative task that was found in Order Dec-Pro of Experiment 2, is not replicated here. The lack of such effect suggests that in a single-task context, as soon as the declarative task appears, the procedural load becomes irrelevant, as the participant is aware that the procedural task will not follow. Nevertheless, the selection operated by the retro-cue does not come at a cost for the uncued representations, because selecting one mapping (and thus leaving a higher number of unattended items) does not impair the performance in the recall of the whole set of encoded mappings (Myers et al., 2018).

Additionally, we also found a significant interaction of declarative load and proportion in error rates of the procedural task. Although we did not have a clear prediction, we believe this interaction is the result of a more detrimental effect of high declarative load in highly procedural experimental setting, suggesting that declarative load has a larger influence when the system is tuned to implement rather than memorize.

5. General discussion

In the present study, we aimed at investigating whether declarative

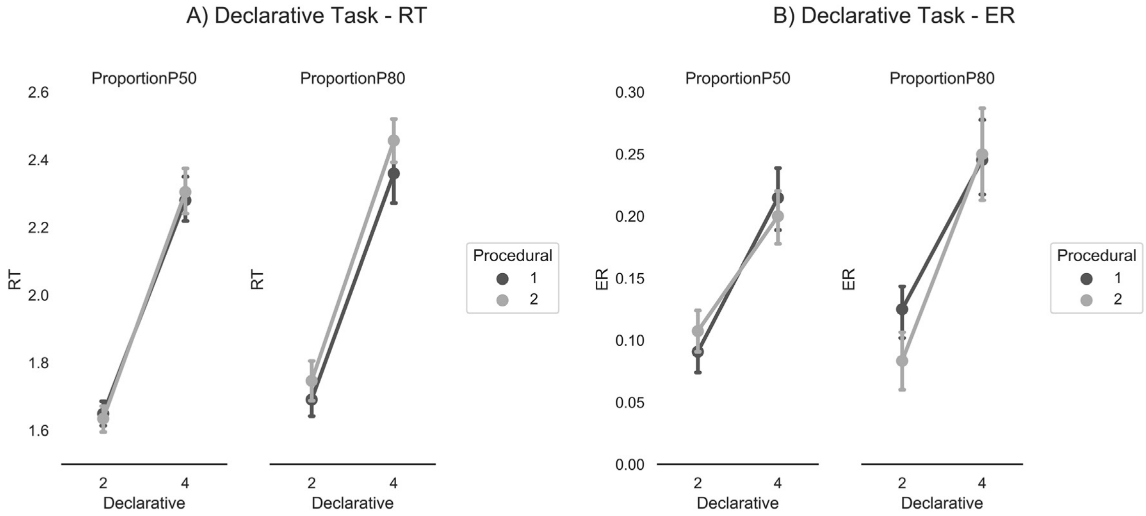


Fig. 6. Results of the declarative task in Experiment 2. A) RTs in ProportionP50 (left panel) and ProportionP80 (right panel). B) Error rates in ProportionP50 (left panel) and ProportionP80 (right panel). Error bars represent standard error mean.

Table 1
Overview of the effects of load manipulations across experiments and tasks. Abbreviations: D = Declarative load; P = Procedural load; D*P = Interaction of Declarative and Procedural loads.

		Procedural task			Declarative task		
		D	P	D*P	D	P	D*P
Exp 1	RTs	✓	✓	✓	✓	✗	✗
	ER	✓	✓	✗	✓	✗	✗
Exp 2	RTs	✓	✓	✓	✓	✗	✗
	ER	✓	✓	✗	✓	✓	✗
Exp 3	RTs	✓	✓	✓	✓	✗	✗
	ER	✓	✓	✗	✓	✗	✗

load has an influence on the proceduralization of novel S-R mappings. At the same time, we wanted to test if the process of proactively transforming mappings in an action-oriented format affects their underlying declarative representations. To achieve this goal, we independently manipulated the number of S-R mappings to be simply maintained declaratively and the number of mappings to be recoded in a procedural format, and we measured the accuracy of their recognition (i.e. declarative task) and implementation (i.e. procedural task). Virtually every WM task requires the contribution of both components, in that they all involve some declarative content to work on and the procedural execution of a response. However, we assumed our two tasks to rely prominently on one or the other, probing their functioning separately. Throughout three experiments, we found a consistent effect of the declarative load on the procedural task, both on RTs and error rates (see Table 1). Specifically, maintaining a higher number of mappings in a declarative format led to slower RTs and more errors in the procedural task. On the contrary, the declarative task remained largely unaffected by the number of mappings proactively transformed and prepared for implementation. Although it is reasonable to conceive proceduralizing and holding action-oriented representations as costly in terms of resources, here we show that this does not come at the expense of the maintenance of the whole set of mappings in a declarative format. Conversely, holding a larger number of declarative mappings has a detrimental effect on the strength of newly created procedural representations, resulting in less efficient performance in their implementation.

Our results suggest that, in the unique case of creating action-oriented representations from novel instructions, declarative and procedural components might be organized in a hierarchical fashion, with procedural WM grounded in declarative WM. From this perspective, the two components seem to share the same pool of resources, hierarchically distributed to first guarantee the maintenance of the declarative representations before being deployed for their procedural processing.

This interpretation fits nicely with current views on the implementation of novel instructions: these are first received and encoded in a declarative format (usually as verbal content). Only when they enter a state of attentional prioritization and if the task demands require it, instructions are then reformatted in an action-oriented code (Brass et al., 2017; González-García et al., 2020; Wenke et al., 2009; Whitehead & Egner, 2018b). The asymmetry we found in cross-component load effects supports the idea that proceduralization cannot be simply reduced to a deeper processing of the declarative instruction, as this would be reflected in an effect of the retro-cue (i.e., of procedural load) on the declarative task. On the contrary, our results are coherent with the existence of two fundamentally distinct (but partially interdependent) representational codes, one that deals with "knowing" the content of the instruction (i.e., maintaining it declaratively) and the other with its actual implementation through a condition-action rule ("doing") (Brass et al., 2017). Crucially, the present study is showing for the first time that the effectiveness of the reformatting process, and thus the strength of the newly created procedural representation, depends not only on the expectations towards its prospective use, but also on the total amount of declarative information that needs to be maintained. It is worth noting that the effect of declarative load cannot be attributed to a failure in the encoding of the mappings or to a decrease in the quality of their representations, as participants were successful in recognizing them in the declarative task. Moreover, additional exploratory analyses showed no significant effect of the attentional selection exerted by the retro-cue on the recognition accuracy of the images in the mappings. This result suggests that proceduralization does not affect the quality of the underlying declarative representations, and that the effect of the retro-cue cannot be reduced to mere attentional prioritization of the selected items. In these regards, our results are in line with the phenomenon of *goal neglect*: the content of the instruction is preserved, but its implementation is inefficient (Bhandari & Duncan, 2014).

The number of S-R mappings that can be proceduralized at once, and exert reflexive-like congruency effects on a secondary task, has been usually considered to be limited to two, even in the absence of declarative load manipulations (Liefooghe et al., 2012). The mechanisms limiting the emergence of IBR and the efficiency in implementing new tasks are not fully understood, although a broad consensus exists around the idea that WM capacity plays a crucial role (Cohen-Kdoshay & Meiran, 2006; Meiran & Cohen-Kdoshay, 2012; Meiran, Pereg, Givon, Danieli, & Shahar, 2016; Pereg & Meiran, 2019). More specifically, both are thought to be reduced in the presence of concurrent WM load, usually in the form of a concomitant secondary task (but see also Pereg & Meiran, 2019, in which effects of WM load were observed on the implementation of the mappings, but not on IBR measures). Our results support the assumption that the efficiency of new instructions implementation depends on the available resources, and further extends on it, showing that even "purely" declarative concurrent load is detrimental for implementing new instructions.

Crucially, in our experiments the mappings to be implemented are selected from a larger set by means of a retro-cue. A recent study (González-García et al., 2020) highlighted the role of attentional selection in the transformation of declarative mappings in a procedural format. Coherent with the idea that retro-cues select relevant declarative representations in WM by orienting attention towards them and then trigger their reformatting (Myers et al., 2017), this study argued that being readily accessible is a necessary condition for S-R mappings to be proceduralized (González-García et al., 2020). The actual reformatting in a procedural code can only take place following the attentional prioritization of the relevant information, bringing the declarative representation in the focus of attention. The proactive preparation of new instructions for future implementation is therefore conceived as a capacity-limited mechanism, with attention playing a key role in determining these limits. Interestingly, one question that derives from our findings concerns the stage at which attention affects the proceduralization process.

One possibility is that the detrimental effect of high declarative load on the procedural task is caused by a less efficient attentional prioritization of the selected mappings. We reasoned that selecting and prioritizing mappings could be more difficult from a larger set of declarative representations compared to a small set, leading to a weaker input for the reformatting process. In this view, the high declarative load would place a constraint not at the stage of recoding the mapping from a declarative to an action-oriented format itself, but rather at the earlier selection of the relevant mappings.

An alternative is to interpret our results within the framework of the Time-Based-Resource-Sharing model (Barrouillet et al., 2004). According to this perspective, both maintenance and processing in WM tasks require attention, which constitutes a limited pool of resources shared between the two. Successful maintenance relies on the periodical refreshing of memory traces achieved by attentional focusing: switching attention away from an item leads to its time-dependent decay. With respect to our task, declaratively maintaining for recognition a higher number of mappings (i.e. high declarative load) requires frequent switches of attention to preserve all the items from the decay, therefore reducing the amount of time and resources to devolve to proceduralization, and consequently the quality of the resulting procedural representations (Barrouillet et al., 2004, 2007; Lépine, Bernardin, & Barrouillet, 2005; Vergauwe et al., 2014).

Finally, our results and interpretations resonate with the concept of *updating* of information in WM, which refers to the retrieval, transformation and substitution of existing representations (Kessler & Meiran, 2006, 2008; Kessler & Oberauer, 2014). This framework also supports a bistate WM system, enabling maintenance of relevant representations and their update, when the task requires it, by means of a gate regulating the interplay of posterior and prefrontal regions through the basal ganglia (Kessler & Oberauer, 2014). Crucially, the authors manipulated both the memory set size (i.e., number of items in the lists) and the number of items to be updated in each list, in an orthogonal design in line with our load manipulations. Their findings showed that the time needed for updating depends on both the number of items to be updated (i.e., procedural processing) and on the size of the set to be maintained (i.e., declarative load). Their interpretation of this cost refers to the opening and closing of the gate that allows for such updating (Kessler & Meiran, 2006, 2008; Kessler & Oberauer, 2014).

One related and still open debate concerns the fate of the uncued declarative representations. WM models positing a continuous pool of resources shared among all the maintained representations assume that the benefit for the cued items comes at the cost of reducing the resources allocated to the uncued representations, thus leading to a decrease in their quality and in the accuracy of their recall (Bays et al., 2009; Luck & Vogel, 1997; Ma, Husain, & Bays, 2014; Pertzov, Bays, Joseph, & Husain, 2013). However, studies involving multiple retrocues showed that a second cue redirecting attention towards a previously unselected item exerts a benefit analogous to that of the first cue (Rerko & Oberauer, 2013). These findings suggest that the fate of uncued items depends on the expectations towards their subsequent use: if it is sure that the uncued items will never be probed (for example, in a single cueing paradigm with valid cues), it is beneficial to forget the unselected items (Williams, Hong, Kang, Carlisle, & Woodman, 2013). On the other hand, if the uncued representations are still relevant for a subsequent task or a second cue, they are maintained, while the selected items are prioritized in the focus of attention (Myers et al., 2018; Rerko & Oberauer, 2013; Souza & Oberauer, 2016). Our experimental setting is in line with this second scenario: the retro-cue selects the mappings relevant for the CRT, but all the mappings should be retained for the recognition task. If it was true that resources are subtracted from unselected declarative representations, leading to a decrease in their quality, an effect of procedural load on recognition should be evident in all three experiments. More specifically, this account would predict worse performance with one proceduralized mapping, as this condition implies a larger number of unselected (and thus degraded) declarative mappings. On the contrary, this pattern of results is only found in Order Dec-Pro of Experiment 2. This is the only case in which the declarative task was presented before the CRT, while the selected mappings were still in the prioritized state, ready for implementation. We therefore interpreted the cost arising in the declarative task for the low procedural load condition as due to the retrieval of the unselected items in the focus of attention, which is a necessary step to compare the recognition screen with the encoded representations of the mappings (Ecker, Lewandowsky, Oberauer, & Chee, 2010; Shepherdson, Oberauer, & Souza, 2018). The low procedural load condition implies a larger set of uncued items that need to be retrieved, while still maintaining the selected mapping in its prioritized state. On the contrary, in the high procedural load condition, two mappings are already prioritized and therefore easily accessible for the recognition task. Attentional prioritization makes the mappings more readily accessible for recognition, thus having a larger number of mappings prioritized leads to better performance. However, when the declarative task is presented after the CRT (as it is the case in all other conditions), the prioritized status of the selected items is no longer relevant, as all the mappings will be probed in the declarative task. This would explain why we did not find an effect of procedural load on the declarative task in all the conditions in which the CRT was administered before the recognition task. According to this view, there is no qualitative difference between selected and unselected declarative representations as long as they are all needed to accomplish the task, but they only differ in the degree of their prioritization and accessibility (Myers et al., 2018).

Therefore, the present study is coherent with previous evidence in highlighting the important role of attention for the proceduralization of novel instructions into an action-oriented code (González-García et al., 2020). Furthermore, our current results suggest that attentional limitations might be one of the underlying causes of the capacity limits of proceduralization. Future research should aim at clarifying at what stage the process is disrupted with increased declarative and procedural load. First, if reformatting takes place to the extent resources are left available from maintaining and prioritizing relevant declarative representations, the emergence of procedural representations should be impaired when declarative WM is overloaded and attention is fully devoted to the constant recall and refreshing of the stored items. Moreover, a retro-cue that is relevant also for the declarative task, allowing participants to completely drop the unselected declarative representations, should free attentional resources and therefore reduce the effect of declarative load on the procedural task and, conversely, introduce an effect of procedural load (i.e., number of selected items) on the declarative task. Finally, the effect of maintaining and proceduralizing an higher number of mappings should be investigated in future studies.

In conclusion, the current series of experiments integrates previous evidence in suggesting that the flexible transformation of declarative instructions into action-oriented representations depends on 1) the amount of resources left available from maintaining all the necessary declarative information, in a hierarchical fashion, 2) the extent to which attention can prioritize specific information (González-García et al., 2020), 3) whether task demands are optimally met by means of a strong procedural representation (Brass et al., 2017; Liefooghe et al., 2012; Wenke et al., 2009; Whitehead & Egner, 2018a, 2018b). This view can be accommodated within Oberauer's WM model (Oberauer, 2009), by considering the implementation of novel instructions as a special case in which the procedural representation cannot be retrieved from long-term memory but requires the active prioritization and reformatting of the corresponding declarative representation (Brass et al., 2017).

Open practices statement

Raw and preprocessed data for all the experiments can be found at https://osf.io/u68jg/ and none of the experiments was preregistered.

Funding

C.G.G. and S.F. were supported by the Special Research Fund of Ghent University (Belgium) BOF.GOA.2017.0002.03.

CRediT authorship contribution statement

Silvia Formica: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing - original draft, Writing - review & editing, Visualization.**Carlos González-García:**Conceptualization, Methodology, Writing - review & editing.**Marcel Brass:**Conceptualization, Methodology, Writing - review & editing, Supervision, Funding acquisition.

Declaration of competing interest

None.

References

Anderson, J. R. (1982). Acquisition of cognitive skill. *Psychological Review, 89*(4), 369–406. https://doi.org/10.1037/0033-295X.89.4.369.

Baddeley, A. (2012). Working memory: Theories, models, and controversies. *Annual Review of Psychology, 63*(1), 1–29. https://doi.org/10.1146/annurev-psych-120710-100422.

Baddeley, A., & Hitch, G. (1974). Working memory. *Psychology of Learning and Motivation, 8*, 47–89. https://doi.org/10.1016/S0079-7421(08)60452-1.

Barrouillet, P., Bernardin, S., & Camos, V. (2004). Time constraints and resource sharing in adults' working memory spans. *Journal of Experimental Psychology: General, 133*(1), 83–100. https://doi.org/10.1037/0096-3445.133.1.83.

Barrouillet, P., Bernardin, S., Portrat, S., Vergauwe, E., & Camos, V. (2007). Time and cognitive load in working memory. *Journal of Experimental Psychology: Learning Memory and Cognition, 33*(3), 570–585. https://doi.org/10.1037/0278-7393.33.3.570.

Barrouillet, P., Corbin, L., Dagry, I., & Camos, V. (2015). An empirical test of the independence between declarative and procedural working memory in Oberauer's (2009) theory. *Psychonomic Bulletin and Review, 22*(4), 1035–1040. https://doi.org/10.3758/s13423-014-0787-y.

Bays, P. M., Catalao, R. F. G., & Husain, M. (2009). The precision of visual working memory is set by allocation of a shared resource. *Journal of Vision, 9*(10), https://doi.org/10.1167/9.10.7 7–7.

Bhandari, A., & Duncan, J. (2014). Goal neglect and knowledge chunking in the construction of novel behaviour. *Cognition, 130*(1), 11–30. https://doi.org/10.1016/j.cognition.2013.08.013.

Bourguignon, N. J., Braem, S., Hartstra, E., De Houwer, J., & Brass, M. (2018). Encoding of novel verbal instructions for prospective action in the lateral prefrontal cortex: Evidence from univariate and multivariate functional magnetic resonance imaging analysis. *Journal of Cognitive Neuroscience, 26*(3), 1–15. https://doi.org/10.1162/jocn_a_01270.

Brady, T. F., Konkle, T., Alvarez, G. A., & Oliva, A. (2013). Real-world objects are not represented as bound units: Independent forgetting of different object details from visual memory. *Journal of Experimental Psychology: General, 142*(3), 791–808. https://doi.org/10.1037/a0029649.

Brass, M., Liefooghe, B., Braem, S., & De Houwer, J. (2017). Following new task instructions: Evidence for a dissociation between knowing and doing. *Neuroscience and Biobehavioral Reviews, 81*(June), 16–28. https://doi.org/10.1016/j.neubiorev.2017.02.012.

Brodeur, M. B., Guérard, K., & Bouras, M. (2014). Bank of Standardized Stimuli (BOSS) phase II: 930 new normative photos. *PLoS One, 9*(9), e106953. https://doi.org/10.1371/journal.pone.0106953.

Bundt, C., Bardi, L., Abrahamse, E. L., Brass, M., & Notebaert, W. (2015). It wasn't me! Motor activation from irrelevant spatial information in the absence of a response. *Frontiers in Human Neuroscience, 9*. https://doi.org/10.3389/fnhum.2015.00539.

Cohen, A. L., Jaudas, A., & Gollwitzer, P. M. (2008). Number of cues influences the cost of remembering to remember. *Memory and Cognition, 36*(1), 149–156. https://doi.org/10.3758/MC.36.1.149.

Cohen-Kdoshay, O., & Meiran, N. (2006). The representation of instructions in working memory leads to autonomous response activation: Evidence from the first trials in the flanker paradigm. *The Quarterly Journal of Experimental Psychology, 1*. https://doi.org/10.1080/17470210600896674 1–1.

Conway, A. R. A., Kane, M. J., Bunting, M. F., Hambrick, D. Z., Wilhelm, O., & Engle, R. W. (2005). Working memory span tasks: A methodological review and user's guide. *Psychonomic Bulletin & Review, 12*(5), 769–786. https://doi.org/10.3758/BF03196772.

Cowan, N., Morey, C. C., Chen, Z., Gilchrist, A. L., & Saults, J. S. (2008). Theory and measurement of working memory capacity limits. *Psychology of Learning and motivation - Advances in research and theory. 49. Psychology of Learning and motivation - Advances in research and theory* (pp. 49–104). Elsevier Masson SAS. https://doi.org/10.1016/S0079-7421(08)00002-9.

Demanet, J., Liefooghe, B., Hartstra, E., Wenke, D., De Houwer, J., & Brass, M. (2016). There is more into 'doing' than 'knowing': The function of the right inferior frontal sulcus is specific for implementing versus memorising verbal instructions. *NeuroImage, 141*, 350–356. https://doi.org/10.1016/j.neuroimage.2016.07.059.

Duncan, J., Emslie, H., Williams, P., Johnson, R., & Freer, C. (1996). Intelligence and the frontal lobe: The organization of goal-directed behavior. *Cognitive Psychology, 30*(3), 257–303. https://doi.org/10.1006/COGP.1996.0008.

Ecker, U. K. H., Lewandowsky, S., Oberauer, K., & Chee, A. E. H. (2010). The components of working memory updating: An experimental decomposition and individual differences. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 36*(1), 170–189. https://doi.org/10.1037/a0017891.

Fukuda, K., Awh, E., & Vogel, E. K. (2010). Discrete capacity limits in visual working memory. *Current Opinion in Neurobiology, 20*(2), 177–182. https://doi.org/10.1016/j.conb.2010.03.005.

Gade, M., Druey, M. D., Souza, A. S., & Oberauer, K. (2014). Interference within and between declarative and procedural representations in working memory. *Journal of Memory and Language, 76*, 174–194. https://doi.org/10.1016/j.jml.2014.07.002.

González-García, C., Formica, S., Liefooghe, B., & Brass, M. (2020). Attentional prioritization reconfigures novel instructions into action-oriented task sets. *Cognition, 194*, 104059. https://doi.org/10.1016/j.cognition.2019.104059.

González-García, C., Formica, S., Wisniewski, D., & Brass, M. (2019). *Frontoparietal action-oriented codes support novel task set implementation. BioRxiv,* 830067. https://doi.org/10.1101/830067.

Griffin, G., Holub, A., & Perona, P. (2007). Caltech-256 Object Category Dataset. Retrieved from https://authors.library.caltech.edu/7694/.

Hartstra, E., Kühn, S., Verguts, T., & Brass, M. (2011). The implementation of verbal instructions: An fMRI study. *Human Brain Mapping, 32*(11), 1811–1824. https://doi.org/10.1002/hbm.21152.

Hommel, B. (2000). Prepared reflex: Automaticity and control in stimulus-response translation. In Control of cognitive processes: Attention and performance XVIII (pp. 247–273).

Janczyk, M. (2017). A common capacity limitation for response and item selection in working memory. *Journal of Experimental Psychology: Learning Memory and Cognition, 43*(11), 1690–1698. https://doi.org/10.1037/xlm0000408.

Jasp Team (2019). *JASP (Version 0.11.1)[Computer software].* Retrieved from https://jasp-stats.org/.

Jeffreys, H. (1998). *The theory of probability*. Oxford University Press.

Jones, P. R. (2019, July 15). A note on detecting statistical outliers in psychophysical data. *Attention, perception, and psychophysics*Springer New York: LLC. https://doi.org/10.3758/s13414-019-01726-3.

Kessler, Y., & Meiran, N. (2006). All updateable objects in working memory are updated whenever any of them are modified: Evidence from the memory updating paradigm. *Journal of Experimental Psychology: Learning Memory and Cognition, 32*(3), 570–585. https://doi.org/10.1037/0278-7393.32.3.570.

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

Kessler, Y., & Oberauer, K. (2014). Working memory updating latency reflects the cost of switching between maintenance and updating modes of operation. *Journal of Experimental Psychology: Learning Memory and Cognition, 40*(3), 738–753. https://doi.org/10.1037/a0035545.

Konkle, T., Brady, T. F., Alvarez, G. A., & Oliva, A. (2010). Conceptual distinctiveness supports detailed visual long-term memory for real-world objects. *Journal of Experimental Psychology: General, 139*(3), 558–578. https://doi.org/10.1037/a0019165.

Lépine, R., Bernardin, S., & Barrouillet, P. (2005). Attention switching and working memory spans. *European Journal of Cognitive Psychology, 17*(3), 329–345. https://doi.org/10.1080/09541440440000014.

Liefooghe, B., & De Houwer, J. (2018). Automatic effects of instructions do not require the intention to execute these instructions. *Journal of Cognitive Psychology, 30*(1), 108–121. https://doi.org/10.1080/20445911.2017.1365871.

Liefooghe, B., Wenke, D., & De Houwer, J. (2012). Instruction-based task-rule congruency effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 38*(5), 1325–1335. https://doi.org/10.1037/a0028148.

Luck, S. J., & Vogel, E. K. (1997). The capacity of visual working memory for featuresand conjunctions. *Nature, 390*(1996), 279–281.

Ma, W. J., Husain, M., & Bays, P. M. (2014). Changing concepts of working memory. *Nature Neuroscience, 17*(3), 347–356. https://doi.org/10.1038/nn.3655.

Marois, R., & Ivanoff, J. (2005). Capacity limits of information processing in the brain. *Trends in Cognitive Sciences, 9*(6), 296–305. https://doi.org/10.1016/j.tics.2005.04.010.

Meiran, N., & Cohen-Kdoshay, O. (2012). Working memory load but not multitasking eliminates the prepared reflex: Further evidence from the adapted flanker paradigm. *Acta Psychologica, 139*(2), 309–313. https://doi.org/10.1016/j.actpsy.2011.12.008.

Meiran, N., Liefooghe, B., & De Houwer, J. (2017). *Powerful.* instructions: Automaticity without practice. Current Directions in Psychological Sciencehttps://doi.org/10.1177/0963721417711638.

Meiran, N., Pereg, M., Givon, E., Danieli, G., & Shahar, N. (2016). The role of working memory in rapid instructed task learning and intention-based reflexivity: An individual differences examination. *Neuropsychologia, 90*, 180–189. https://doi.org/10.1016/j.neuropsychologia.2016.06.037.

Meiran, N., Pereg, M., Kessler, Y., Cole, M. W., & Braver, T. S. (2015). The power of instructions: Proactive configuration of stimulus–response translation. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 41*(3), 768–786. https://doi.org/10.1037/xlm0000063.

Milner, B. (1963). Effects of different brain lesions on card sorting: The role of the frontal lobes. *Archives of Neurology, 9*(1), 90–100. https://doi.org/10.1001/archneur.1963.00460070100010.

Myers, N. E., Chekroud, S. R., Stokes, M. G., & Nobre, A. C. (2018). Benefits of flexible prioritization in working memory can arise without costs. *Journal of Experimental Psychology: Human Perception and Performance, 44*(3), 398–411. https://doi.org/10.1037/xhp0000449.

Myers, N. E., Stokes, M. G., & Nobre, A. C. (2017). Prioritizing information during working memory: Beyond sustained internal attention. *Trends in Cognitive Sciences, 21*(6), 449–461. https://doi.org/10.1016/j.tics.2017.03.010.

Oberauer, K. (2009). *Design for a Working Memory, 51*(09), 45–100. https://doi.org/10.1016/S0079-7421(09)51002-X.

Oberauer, K. (2010). Declarative and procedural working memory: Common principles, common capacity limits? *Psychologica Belgica, 50*(3&4), 277–308. https://doi.org/10.5334/pb-50-3-4-277.

Oberauer, K., Farrell, S., Jarrold, C., & Lewandowsky, S. (2016). What limits working memory capacity? *Psychological Bulletin, 142*(7), 758–799. https://doi.org/10.1037/bul0000046.

Peirce, J. W. (2007). PsychoPy-Psychophysics software in Python. *Journal of Neuroscience Methods, 162*(1–2), 8–13. https://doi.org/10.1016/j.jneumeth.2006.11.017.

Pereg, M., & Meiran, N. (2019). Rapid instructed task learning (but not automatic effects of instructions) is influenced by working memory load. *PLoS One, 14*(6), e0217681. https://doi.org/10.1371/journal.pone.0217681.

Pertzov, Y., Bays, P. M., Joseph, S., & Husain, M. (2013). Rapid forgetting prevented by retrospective attention cues. *Journal of Experimental Psychology: Human Perception and Performance, 39*(5), 1224–1231. https://doi.org/10.1037/a0030947.

Rerko, L., & Oberauer, K. (2013). Focused, unfocused, and defocused information in working memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 39*(4), 1075–1096. https://doi.org/10.1037/a0031172.

Rouder, J. N., Morey, R. D., Morey, C. C., & Cowan, N. (2011). How to measure working memory capacity in the change detection paradigm. *Psychonomic Bulletin and Review, 18*(2), 324–330. https://doi.org/10.3758/s13423-011-0055-3.

Rouder, J. N., Morey, R. D., Speckman, P. L., & Province, J. M. (2012). Default Bayes factors for ANOVA designs. *Journal of Mathematical Psychology, 56*(5), 356–374. https://doi.org/10.1016/J.JMP.2012.08.001.

Rousseeuw, P. J., & Croux, C. (1993). Alternatives to the median absolute deviation. *Journal of the American Statistical Association, 88*(424), 1273–1283. https://doi.org/10.1080/01621459.1993.10476408.

Ruge, H., & Wolfensteller, U. (2010). Rapid formation of pragmatic rule representations in the human brain during instruction-based learning. *Cerebral Cortex, 20*(7), 1656–1667. https://doi.org/10.1093/cercor/bhp228.

Schonbrodt, F. D., & Wagenmakers, E. J. (2017). Bayes factor design analysis: Planning for compelling evidence. *Psychonomic Bulletin and Review, 2014*, 1–15. https://doi.org/10.3758/s13423-017-1230-y.

Shahar, N., Teodorescu, A. R., Anholt, G. E., Karmon-Presser, A., & Meiran, N. (2017). Examining procedural working memory processing in obsessive-compulsive disorder. *Psychiatry Research, 253*(January), 197–204. https://doi.org/10.1016/j.psychres.2017.03.048.

Shahar, N., Teodorescu, A. R., Usher, M., Pereg, M., & Meiran, N. (2014). Selective influence of working memory load on exceptionally slow reaction times. *Journal of Experimental Psychology: General, 143*(5), 1837–1860. https://doi.org/10.1037/a0037190.

Shepherdson, P., Oberauer, K., & Souza, A. S. (2018). *Working memory load and the retro-cue effect: A diffusion model account, 44*(2), 286–310.

Souza, A. S., & Oberauer, K. (2016). In search of the focus of attention in working memory: 13 years of the retro-cue effect. *Attention, Perception, & Psychophysics, 78*(7), 1839–1860. https://doi.org/10.3758/s13414-016-1108-5.

Vergauwe, E., Camos, V., & Barrouillet, P. (2014). The impact of storage on processing: How is information maintained in working memory? *Journal of Experimental Psychology: Learning Memory and Cognition, 40*(4), 1072–1095. https://doi.org/10.1037/a0035779.

Wenke, D., Gaschler, R., Nattkemper, D., & Frensch, P. A. (2009). Strategic influences on implementing instructions for future actions. *Psychological Research Psychologische Forschung, 73*(4), 587–601. https://doi.org/10.1007/s00426-009-0239-x.

Whitehead, P. S., & Egner, T. (2018a). Cognitive control over prospective task-set interference. Journal of Experimental Psychology. *Human Perception and Performance, 44*(5), 741–755. https://doi.org/10.1037/xhp0000493.

Whitehead, P. S., & Egner, T. (2018b). Frequency of prospective use modulates instructed task-set interference. *Journal of Experimental Psychology: Human Perception and Performance, 44*(12), 1970–1980. https://doi.org/10.1037/xhp0000586.

Williams, M., Hong, S. W., Kang, M.-S., Carlisle, N. B., & Woodman, G. F. (2013). The benefit of forgetting. *Psychonomic Bulletin & Review, 20*(2), 348–355. https://doi.org/10.3758/s13423-012-0354-3.

Muhle-Karbe, P. S., Duncan, J., De Baene, W., Mitchell, D. J., & Brass, M. (2017). Neural coding for instruction-based task sets in human Frontoparietal and visual cortex. *Cerebral Cortex, 27*(3), 1891–1905. https://doi.org/10.1093/cercor/bhw032 (New York, N.Y.: 1991).