Running head: WORKING MEMORY UPDATING AND REMOVAL

Removal of information from working memory: A specific updating process

Ullrich K. H. Ecker, Stephan Lewandowsky

School of Psychology, University of Western Australia

Klaus Oberauer

Department of Psychology, University of Zurich, and

School of Psychology, University of Western Australia

Word count (brief report): 4422 excluding references; 5200 including references

(approximate count due to use of LaTeX)

Ullrich Ecker

School of Psychology

University of Western Australia

Crawley, W.A. 6009, AUSTRALIA

Telephone: +618 6488 3257

Fax: +618 6488 1006

E-mail: ullrich.ecker@uwa.edu.au

Abstract

Previous research has claimed that working memory (WM) updating is one of three primary central executive processes, and the only one to reliably predict fluid intelligence. However, standard WM updating tasks confound updating requirements with generic WM functions. This article introduces a method for isolating a process unique to WM updating, namely the removal of no-longer relevant information. In a modified version of an established updating paradigm, to-be-updated items were cued before the new memoranda were presented. Longer cue-stimulus intervals—that is, longer removal-cue time—led to faster updating, showing that people can use the removal-cue time to remove old information before receiving new information. This benefit of removal-cue time was found only for partial updating (i.e., 1 or 2 items out of 3), not for complete updating of the entire memory set. The reduction of updating time during the cue-stimulus interval was used to measure individual's removal speed. Removal speed was measured reliably but was uncorrelated to WM capacity. We conclude that (1) removal of outdated information can be experimentally isolated and measured reliably, (2) removal speed is a unique WM updating ability, and (3) the view of WM updating as a core executive process that uniquely predicts fluid abilities is overstated.

Keywords: working memory; updating; removal; individual differences

Removal of information from working memory: A specific updating process

Imagine you ask a colleague for his phone extension and he replies: "It's 3266. No, hang on, in my new office it's actually 3257". Ideally, one should easily discard the last two digits of the outdated information given (i.e., "66") and replace them in working memory with the correct digits (i.e., "57"). However, this updating of working memory content is no trivial task, and outdated information often continues to affect memory (?, ?, ?).

Working memory updating has been identified as one of three primary central executive processes (?, ?). Updating has been claimed to be the only executive process to predict fluid intelligence (?, ?, ?). However, most updating tasks used in previous research (e.g., ?, ?) not only require memory updating but arguably also measure general working memory (WM) abilities. This has led some researchers to conclude that updating tasks constitute reliable assays of general WM capacity (?, ?).

This creates an unsatisfactory situation. If WM updating tasks measure just the same as other WM tasks such as complex span tasks, then why call them updating tasks? Both conceptually and theoretically, updating can be distinguished from maintenance and processing in WM. If updating is to be established as a non-redundant construct, it must be isolated and measured separately from other WM processes.

In a recent individual-differences study, we identified a processing component that was independent of general WM capacity and unique to situations that demanded memory updating (?, ?). In that study we analyzed the processing components involved in widely used WM updating tasks, and we identified three separable components: retrieval, transformation, and substitution. The only component process that was unique to WM updating tasks was the substitution of information in memory. To illustrate those components, consider the scenario of a restaurant manager advising a chef early in the evening that they were expecting 20 patrons. If the manager later advised the chef that

twice as many guests were expected as before, the chef will first need to *retrieve* the initial expectation (i.e., 20), then *transform* it (i.e., $2 \times 20 = 40$), and then *substitute* the outdated information with the updated information (i.e., 40). ? found that retrieval and transformation operations covaried with general WM capacity, but that the substitution component did not. This finding was interpreted as showing that substitution is the only process that uniquely represents WM updating, without being "contaminated" by any association with working memory. One implication of this analysis is that previous studies measuring WM updating did not separate variance unique to updating from the variance of generic WM processes. As a consequence, the conclusions concerning the predictive relation between WM updating and fluid intelligence (?, ?, ?) arguably were not based on a proper measure of WM updating, but may instead reflect the well-known association between higher cognitive functions and general WM capacity (?, ?, ?).

In this article, we further decompose the components of WM updating. In ? (?), we suggested that information substitution can be further subdivided into the *removal* of outdated information and the *encoding* of new information. For example, the chef would need to remove the number 20 from memory before encoding the updated number 40 into the vacant memory slot. As encoding is a simple and generic operation involved in many cognitive tasks, we argue that it is the removal process that likely lies at the heart of memory updating. Accordingly, we focus on the removal of information from WM. Here we show that (1) removal of outdated information can be separated experimentally from encoding of new information, (2) that removal can be measured reliably, and (3) that removal speed is independent of WM capacity.

Our removal measure is based on the work of ? (?), who used a variant of the classic updating paradigm developed by ? (?). In this paradigm, items (e.g., letters or digits) are presented in a set of individual frames. Items are then repeatedly updated by presenting new items in some frames. On each updating step, between one and n items are updated,

where n is the set size (i.e., the number of items to be held in memory at any point in time, which is equal to the number of frames). Participants had to press a key at the end of each step to indicate that they finished updating; the next set of new stimuli was presented immediately after the key press.

? (?) proposed a distinction between local and global updating. Local updating refers to changes made to individual items, whereas global updating refers to the integration of all items in the current memory set after individual items were changed. A key piece of evidence for this distinction comes from the observation (Experiment 3 in ?, ?) that updating RTs increased with the number of to-be-updated items up to $n - 1$ items, but updating was much faster again when all n items were to be replaced on a given step. Thus, updating latencies depended in a non-monotonic fashion on the number of to-be-updated items. ? (?) explained this non-monotonicity by assuming that partial updates require a complex sequence of (1) unbinding of the integrated representation of the previous memory set, (2) substitution of some but not all items (i.e., the actual local updating), followed by (3) re-binding the new set as part of the global updating process. In contrast, when the entire set is updated, steps (1) and (2) can be omitted, the old set is simply discarded and a new memory set is encoded and globally updated.

Our interpretation of the non-monotonicity of updating latencies is a specific instantiation of the ideas of ? (?), motivated by a computational model of working memory, SOB (?, ?, ?, ?). SOB is a two-layer neural network in which items (represented in one layer) are associated to position markers (represented in the other layer) through Hebbian learning, which rapidly modifies the matrix of connection weights between the two layers. In the present updating paradigm, the position marker would represent the location of the item's frame on the screen. Forgetting in SOB is entirely based on interference; there is no time-based decay. This assumption is corroborated by a growing body of evidence (?, ?, ?).

To avoid overloading of the system in the absence of decay, an interference model requires a mechanism to remove outdated information; such a mechanism is implemented in the most recent version of SOB (see ?, ?). Removal of a specific item involves retrieving that item by cueing with its position marker, and "unlearning" the association between that item and its position. Unlearning is computationally implemented as Hebbian anti-learning. Thus, in SOB the idea of "unbinding" in the theory of ? (?) refers specifically to the unbinding of selected items from their position markers. Both encoding and removal of individual items take time (cf. ?, ?, ?, ?). By contrast, wholesale removal of an entire memory set can be achieved by simply resetting the entire weight matrix. We assume that wholesale clearance of the weight matrix is a very rapid process, compared to the removal of individual items. This explains why updating the entire memory set is faster than partial updating.

In SOB, removal of old information and encoding of new information are described as two separate processing steps. It follows that updating should be facilitated if a cue about what information needs to be removed is given ahead of the to-be-encoded new information. In standard WM updating tasks—including the one used by ? (?)—removal can only begin when the new item(s) are presented. For example, when updating the telephone extension from from 3266 to 3257, one can only begin removing the "66" when given the "57". Hence updating times in such a task will include both time for removal and time for encoding. In our new updating task we present cues indicating which items are to be updated before presenting the new to-be-encoded stimuli. People can use the cue only to selectively remove old items from the memory set, not to encode new information. By varying the cue-stimulus interval, we vary the available time for removal. If longer available removal time leads to faster updating RTs, people must have used the cue-stimulus interval for removal or unbinding. Based on the distinction between slow selective removal and fast resetting of the weight matrix, we predict that longer

removal-cue times should lead to a substantial time gain in partial updating, but little gain on updating steps replacing the entire memory set. Moreover, the speed-up in updating with vs. without pre-encoding removal provides a pure measure of an individual's removal speed, which we can then use to investigate how removal of outdated information from WM is correlated to other variables.

Experiment 1

Experiment 1 used a letter updating task in which each trial consisted of an encoding stage, an updating stage with multiple updating steps, and a final recall stage. Each updating step could update the memory set either partially or entirely. Before each updating step, participants were warned which item(s) were about to be updated.

Methods

Participants

Seventy-four University of Western Australia undergraduates participated for partial course credit. Five participants were removed because of missing data ($n = 1$), misunderstanding of task instructions ($n = 2$), or outlying recall performance (more than 3 standard deviations from the mean; $n = 2$). The final sample thus comprised 69 participants (51 females, 18 males; mean age 20.8 years; age range 18-34 years).

Apparatus

The experiment was run with the aid of MatLab and the Psychophysics toolbox (?, ?, ?). Participants were tested individually in booths, sitting about 70 cm from a 17-in. thin-film transistor monitor.

Stimuli, Design, and Procedure

The letter updating task involved a single row of three black, rectangular frames. Following a fixation cross presented for one second, each trial began with the simultaneous 2-second presentation of 3 black consonants (ranging from B to Z) in the frames. The minimum alphabetic distance between the to-be-encoded letters was 2, to avoid sequences such as X, Y, Z.

A series of updating steps followed. On each step, new letter(s) were presented in 1, 2, or 3 frames. The to-be-updated frames at each step turned bold and red before the new content was displayed, potentially allowing participants to remove outdated memory content before encoding the new memoranda. Cueing was either short (i.e., 200 ms before presentation of the new items) or long (i.e., 1500 ms). The longer cue should be sufficient time for removal, whereas the shorter cue should be just enough time to focus attention on the to-be-updated frames without permitting removal.

Participants pressed the space bar once they had encoded the new content. The new memoranda remained on the screen until a response was made or the maximum updating time of 5 seconds was reached. After each updating step, the frames were blanked for 500 ms or 1800 ms in the long and short removal-cue condition, respectively, ensuring equal retention intervals in both conditions (i.e., 500 ms + 1500 ms in the long removal-cue condition and 1800 ms + 200 ms in the short removal-cue condition).

The number of updating steps varied from 1 to 21; the sequence finished with a constant probability of 10% after each updating step. This resulted in an unpredictable number of updating steps, and because each step was equally likely to be the last, participants had an equal incentive to carry out each updating step independent of the duration of the sequence. Sequences had a mean number of about 9 updating steps. The number of to-be-updated frames and removal-cue duration were chosen randomly at each step.

After the updating phase of each trial, participants recalled the current contents of all frames. Recall was prompted by blue question marks appearing one-by-one in each frame in random order. Probes remained on the screen until a response was given, or until the maximum response time of 5 seconds per frame was reached. After recall of all three frames, feedback ("x out of 3 correct") was given. The blank-screen inter-trial interval was 2.5 seconds. A representative (albeit short) trial sequence is shown in Figure 1.

In sum, the experiment comprised a total of 3 (number of updated frames: 1 vs. 2 vs. 3) $\times$ 2 (removal-cue time: 200 vs. 1500 ms) conditions. There were 28 trials in total (plus 4 practice trials), with an average of 9 updating steps per trial, yielding approximately 252 updating steps, or 42 per condition. Each trial took approximately 40 seconds, and the experiment took about 20 minutes.

Results

Response times below 300 ms, and more than 3 standard deviations from the individual mean were discarded. Updating response time data are shown in Figure 2.

A two-way repeated measures ANOVA on updating response times yielded a significant main effect of the number of updated frames, $F(2, 136) = 64.30$, $MSE = 0.03$, $p < .001$, $\eta_p^2 = 0.49$, a significant main effect of removal-cue time, $F(1, 68) = 281.68$, $MSE = 0.02$, $p < .001$, $\eta_p^2 = 0.81$, as well as a significant interaction, $F(2, 136) = 109.55$, $MSE = 0.01$, $p < .001$, $\eta_p^2 = 0.62$. Planned contrasts showed that on average it took significantly longer to update two frames compared to one (1.44 seconds vs. 1.26 seconds;$F(1, 68) = 146.29$, $MSE = 0.02$, $p < .001$) and that with 1 or 2 frames, updating took significantly longer with short as compared to long removal-cue time (1.53 vs. 1.17 seconds; $F(1, 68) = 320.14$, $MSE = 0.03$, $p < .001$. However, updating three frames was relatively quick (1.23 seconds), and removal-cue time had a negligible impact on updating time when all three frames were updated. While the removal-cue time effect was

statistically significant (arguably due to the large sample size and power; 1.26 vs. 1.20 seconds; $F(1, 68) = 13.59$, $MSE = 0.01$, $p < .001$), it was much smaller with three frames than with one or two frames (53 vs. 362 ms).[1]

Before we discuss these results, we turn to a potential alternative account of the data. Because of the need to equate the retention interval across conditions, our removal-cue time is fully confounded with the blank-screen interval preceding each update: When removal-cue time is long (1500 ms) then the blank-screen interval is short (500 ms) and vice versa (200 ms and 1800 ms, respectively). One could hence argue that the response time differences we find could be influenced by some sort of short-term consolidation process. In particular, items updated after a short removal-cue interval may be more difficult to update because they have had more time to consolidate.

A straightforward way to investigate this is to compare the data pattern in cases where the updated frame (on updating step n) was also updated on step $n - 1$ with cases where the updated frame was not updated on the previous step. In the former case, an item's memory representation may still be consolidating when the updating cue appears, but in the latter case, consolidation should be complete by that time (or at least, the additional 1300 ms should not matter much anymore).

To this end, we classified all 1-frame updates depending on whether or not the same frame was also updated on the previous step (which could have been a 1-, 2-, or 3-frame update), and then ran a 2×2 repeated-measures ANOVA with the factors removal-cue time (short vs. long) and frame-switch (yes/no). The analysis returned a main effect of removal-cue time, $F(1, 68) = 18.79$, $MSE = 0.02$, $p < .001$, $\eta_p^2 = 0.22$, and a main effect of frame-switch, $F(1, 68) = 246.66$, $MSE = 0.04$, $p < .001$, $\eta_p^2 = 0.78$, but no interaction, $F(1, 68) = 1.85$, $MSE = 0.02$, $p = 0.18$, $\eta_p^2 = 0.03$.

This demonstrates two things: (1) Updating the same frame twice in a row is quicker than updating a frame that was not updated on the immediately preceding step.

One can argue that this may be because an item that has not been consolidated fully is easier to update, but one could also argue that updating the same frame twice in a row requires less attentional switching between frames. With our data, we can not distinguish between these two possibilities. More importantly, (2) the effect of removal-cue time is not affected by a frame-switch or the associated difference in presumed consolidation time. We can hence conclude that our main result is not affected by the confound of removal-cue time and "consolidation time." We now move on to discussing this effect of prime interest.

Discussion

The results of Experiment 1 support our hypothesis that partial updating of a memory set involves a process of active removal. "Outsourcing" this removal process by cueing the to-be-updated frames prior to presentation of the new items sped up partial updating substantially. This time saving of roughly 300-400 ms can be interpreted as the time that is required to remove information from working memory.

In contrast, the opportunity to remove items from memory before presentation of new memoranda brought no substantial advantage when the entire memory set was updated. This finding supports our notion that memory can be cleared almost instantly (more evidence for very rapid forgetting of entire memory sets can be found in ?, ?, ?, ?), and that hence updating of an entire memory set does not require the time-consuming selective removal process.

The speed-up induced by the long removal cue did not increase with the number of to-be-updated items. This unexpected observation seems to imply that removing one item from WM takes as long as removing two items. There are two possible explanations. One is that removal of multiple items can occur in parallel. While this is a theoretical possibility, it is at odds with the SOB framework, in which items must be retrieved individually to be removed by anti-learning.

The second explanation is that people use the removal-cue time only to remove one item, even when two items are about to be updated. This might be an efficient strategy because item-specific removal is likely to require the focus of attention, and switching the focus of attention to a new item takes time (cf. ?, ?, ?). Thus, the most efficient use of removal-cue time might be to focus on the first to-be-removed item and remove it, then wait. As soon as the new item(s) are presented, one can immediately be encoded in the currently focused frame. Only then would the focus move on to the second to-be-updated item (if there is one).

One may wonder whether the reported effects depend on serial position. The SOB framework would predict that—because removal of an item requires its retrieval—the item that is quickest to be retrieved should also be quickest to be removed. In the case of 1-frame updates, this would hence mean that the last and/or first item should be updated fastest. With 2-frame updates, however, the predictions are somewhat less clear.

Figure X summarizes the serial position data (there was no interaction with removal-cue time, hence we present the data collapsed across removal-cue conditions). These data demonstrate that with 1-frame updates, the third frame is the quickest to be updated, which is in line with the removal notion. However, the removal notion did not predict that updating frame 1 takes the same time as updating frame 2. The 2-frame updates show that updating is quickest with the last two frames, followed by the first two frames, and updating frames 1 and 3 concurrently is slowest. This suggests that in the case of 2-frame updates, retrieval speed of the individual items may not be the only relevant factor. While the first and last item should be retrieved quickest, updating both at the same time may be complicated by the fact that the items are disjunct.

At first glance, the serial-position data might seem relatively uninformative. However, they strongly support the notion of Kessler & Oberauer (2012) that updating times are strongly determined by task switching. Assuming that, without substantial

pre-cued removal time, participants scan the items from the beginning of the list to the end, starting in a maintenance mode (M). Hence, if the first item on the list is not updated, people might refresh that representation, then move to the next item. As soon as they encounter an item that requires updating, they switch to updating mode and actively remove that item from WM (R). Updating frames 1 and 3 would hence require 3 switches (M-R-M-R), updating frames 1 and 2 (M-R-M) two switches, updating frames 2 and 3 (M-R) one switch. In the case of 1-frame updates, updating frame 1 and updating frame 2 both require two switches (M-R-M), but updating frame 3 only requires one switch (M-R).

(All 2-frame updates are also slower than the 3-frame update.)

In Experiment 2, we demonstrate that from this removal task, one can calculate a reliable estimate of removal speed, and we consider this a measure of WM updating ability. The aim of Experiment 2 was to investigate individual differences in removal speed. As reviewed in the Introduction, previous research has emphasized the importance of WM updating as a predictor of higher cognitive functions. This view might suggest a correlation between a WM updating measure and WM capacity, in particular if WM capacity is considered "executive attention" (?, ?). In contrast, ? (?) identified a substitution component of WM updating that did not covary with WM capacity. Our prediction was hence that removal speed and WM capacity would be unrelated.

Experiment 2

Experiment 2 used the removal paradigm from the first study and additionally measured WM capacity in a large-scale individual-differences study.

Methods

Participants

Participants ($N = 203$) were sampled as in the first study. None had participated in Experiment 1. Based on various outlier criteria (see below), 15 participants were excluded from the analyses, yielding a final sample of $N = 188$ (142 female, mean age 20.7 years, range 18-42).

Apparatus, Stimuli, Design, and Procedure

Participants performed the updating task described in Experiment 1. The only modification was that the condition with three updates (i.e., updating the entire memory set) was omitted because this condition did not show any noteworthy effects of pre-cueing time. Hence the experiment comprised a total of 2 (number of updated frames: 1 vs. 2) $\times$ 2 (removal-cue time: 200 vs. 1500 ms) conditions. There were 28 trials in total (plus 4 practice trials), with an average of 9 updating steps per trial, yielding approximately 252 updating steps, or 63 per condition.

Additionally, participants completed a WM capacity task battery (?, ?). This task battery comprises four WM tasks: an operation span task (OS), a sentence span task (SS), a memory updating task (MU) and a spatial short-term memory task (SSTM). We describe the tasks only briefly because we used the default settings described in detail in ? (?).

The two complex-span tasks arguably are the most widely used tasks to measure WM capacity (?, ?). They require participants to memorize a set of items for serial recall, while interleaving the memoranda with a secondary processing task such as judging the correctness of equations (OS) or sentences (SS). The two complex-span tasks used memory set sizes of 4 to 8 consonants, and a single arithmetic equation with operands ranging from 1-10 (OS) or a short sentence (SS) in between each pair of memoranda. The

MU task was a "standard" WM updating task—measuring mainly generic WM abilities as discussed in the Introduction. It involved encoding a set of between 3 and 5 digits, presented in individual frames on the screen, which were then repeatedly updated before a final cued recall. Updating was prompted by a series of between 2 and 6 successive cues for arithmetic operations (ranging from -7 to +7) presented randomly in individual frames. The final task (SSTM) required participants to memorize and then reproduce a spatial pattern of between 2 and 6 dots, presented sequentially in a 10×10 grid. Scores on these 4 tasks can be combined into a single latent score, which has been shown to be a valid and reliable estimate of WM capacity (?, ?, ?, ?, ?).

Results

Participants scoring more than 3 standard deviations from the mean on any of the WM capacity tasks or the recall portion of the updating task were excluded from analyses.

Updating task

As in Experiment 1, recall accuracy was very high ($M = 0.95; SE = 0.003$).

Response time results from the removal-updating task replicated the pattern found in Experiment 1. These results are shown in Figure 3.

A two-way repeated measures ANOVA on updating response times yielded a significant main effect of the number of updated frames, $F(1, 187) = 617.79$, $MSE = 0.01$, $p < .001$, $\eta_p^2 = 0.768$, a significant main effect of removal-cue time, $F(1, 187) = 766.78$, $MSE = 0.02$, $p < .001$, $\eta_p^2 = 0.804$, and no interaction, $F < 1$. It took significantly longer to update two frames compared to one (1.30 seconds vs. 1.10 seconds). Updating took significantly longer with short as compared to long removal-cues (1.34 vs. 1.06 seconds).

An obvious measure of removal speed would be the difference between the means of the short and long removal-cue conditions. However, to calculate an estimate of removal speed that is not confounded with general processing speed (which itself tends to correlate

with WM capacity; ?, ?, ?, but see ?, ?, we calculated our removal measure as a proportional gain score: $Removal = (mean(shortcue) - mean(longcue))/mean(shortcue)$.

In an important second step, we tested whether this removal score was statistically reliable. We calculated the score's split-half reliability, which (after Spearman-Brown correction) was $\rho = .76$ and hence sufficiently high to permit further analysis of this score's relation with WM capacity.

WM capacity battery

Table 1 gives descriptive data of the WM capacity tasks. From the four primary task scores, we calculated a global WM capacity score.

Correlational analyses

As expected, all four WM capacity scores correlated highly significantly with each other, with r ranging from 0.24 (OS and SSTM) to 0.69 (OS and SS). The global WM capacity score also correlated substantially with recall accuracy in the updating task ($r = 0.45$, $p < .001$). We then calculated the correlation between the global WM capacity score and the removal score. This correlation was virtually zero, $r = 0.03$; $p > .1$. Moreover, our removal score did not correlate with any of the four WM tasks individually ($r = 0.07$, $p > .1$ for the MU task, and all $r < 0.03$ for the remaining three tasks, $p > .1$).

Discussion

The aim of Experiment 2 was to investigate the relation between removal speed and WM capacity. WM capacity has been termed the "engine of cognition" (?, ?); it accounts for roughly half the variance in general fluid intelligence (?, ?). Previous research has demonstrated that WM updating task performance is strongly related to performance on WM capacity tasks (?, ?). We have argued that the reason for this is that "standard" WM updating tasks mainly measure WM capacity, whereas only the removal process identified

in the present experiments is unique to memory updating. Our results indicate that there is no relation between this more specific measure of WM updating and WM capacity.

General Discussion

In this article we have introduced a novel measure of WM updating. Traditional WM updating tasks arguably measure general WM processes in addition to updating, whereas it is the removal of information from WM that is specific and unique to WM updating. We demonstrated that giving people preparation time to remove information from WM speeds up updating when new information is subsequently presented, but only when a subset of the memory set is updated. Updating an entire memory set does not benefit (much) from preparation time, arguably because—in line with the predictions derived from SOB (?, ?, ?, ?)—the time-consuming removal process only applies to individual items, whereas an entire memory set can be removed by instant resetting of the weight matrix. Our notion of removal by unlearning item-position associations is a specific incarnation of the more general idea advanced by ? (?), who suggested that partial updating of memory sets involves "dismantling" or "unbinding" the old representations.

Whereas our research is guided by the SOB model, and our data support the implementation of removal into SOB (?, ?), it is important to note that other researchers have provided independent evidence for the existence of an active and attention-demanding removal process. For example, ? (?, ?) have shown that directed forgetting of an item (1) slows down responses on an unrelated secondary task for up to 2.6 seconds, and (2) impairs incidental memory for a subsequent distractor, in particular when the directed forgetting of the studied item is successful.

In addition to the experimental evidence for removal of outdated material from WM, we found that the speed of removal can be reliably measured, and that it does not covary with WM capacity. The independence of removal speed from WM capacity

confirms and specifies the earlier finding that the ability to substitute information in WM is unrelated to WM capacity (?, ?). This is a noteworthy finding because WM capacity has been found to predict a broad range of cognitive abilities from categorization (?, ?) to metaphor comprehension (?, ?). The removal measure is therefore a highly specific index of WM updating ability. This result further suggests that the previously reported predictive link between WM updating and fluid intelligence (?, ?, ?) may in fact rest solely on the strong relation between fluid intelligence and WM capacity (?, ?).

This should not, however, imply that removal is unimportant. On the contrary, we assume removal to be crucial to maintaining a functional WM system that can efficiently focus on relevant information. Further research is needed to ascertain whether removal in itself predicts some intelligence-related variance (cf. ?, ?), or whether it relates to other executive functions (e.g., inhibition or filtering of information) or real-world updating on longer time-scales (?, ?).

References

Author Note

Preparation of this paper was facilitated by a Discovery Grant from the Australian Research Council to Stephan Lewandowsky and Klaus Oberauer, as well as an Australian Professorial Fellowship to Stephan Lewandowsky, and an Australian Postdoctoral Fellowship to Ullrich Ecker. Address correspondence to the first author at the School of Psychology (M304), University of Western Australia, Crawley, W.A. 6009, Australia. Electronic mail may be sent to ullrich.ecker@uwa.edu.au or stephan.lewandowsky@uwa.edu.au. The lab's web site is located at www.cogsciwa.com. We thank Charles Hanich for research assistance.

Footnotes

[1]The astute reader may notice that this latter contrast has a much lower p-value than the within-subject confidence intervals in Figure 2 suggest. This is because the confidence intervals are based on data that was normalized across all conditions (cf. ?, ?). To the extent that measures do not correlate perfectly across conditions—in our case in the partial-update vs. full-update conditions—, taking out individual differences across conditions will lead to a situation where the remaining variability (from which the confidence intervals are calculated) still contains a proportion of true individual differences. The contrast, however, will only take into account individual variability in the difference between the two conditions contrasted.

Table 1

Descriptive Statistics of Performance on the Working Memory Tasks in Experiment 2

Task	Mean	SE	Range
OS	0.72	0.008	0.37 – 0.95
OS_{pt}	0.92	0.004	0.75 – 1.00
SS	0.69	0.009	0.33 – 0.96
SS_{pt}	0.92	0.003	0.75 – 1.00
MU	0.65	0.012	0.20 – 0.97
SSTM	0.84	0.004	0.70 – 0.97

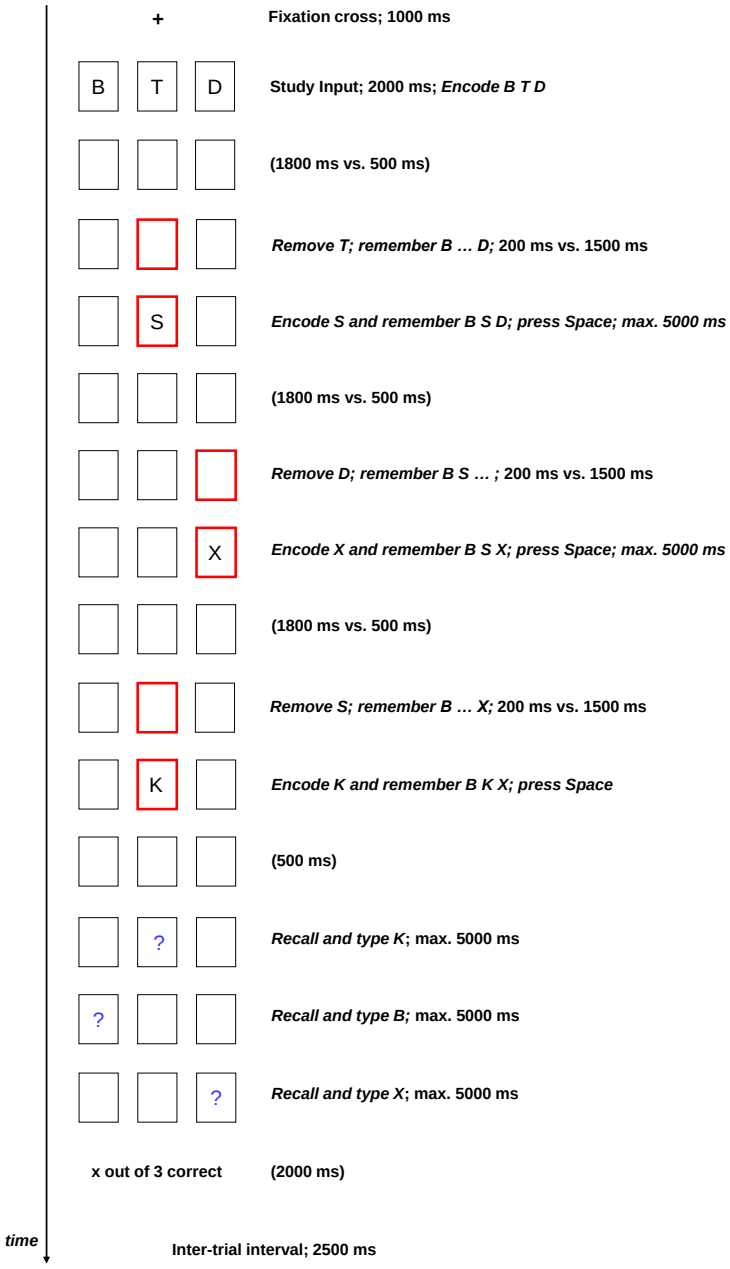
Legend. OS, Operation Span; SS, Sentence Span; $_{pt}$ denotes processing tasks; MU, Memory Updating; SSTM, Spatial Short-Term Memory; SE, Standard Error of the Mean.

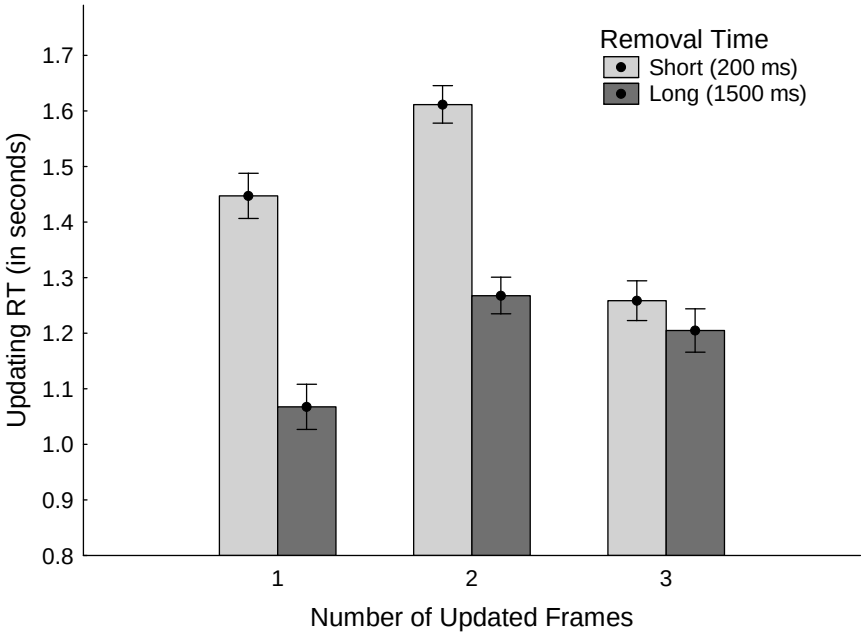
Figure Captions

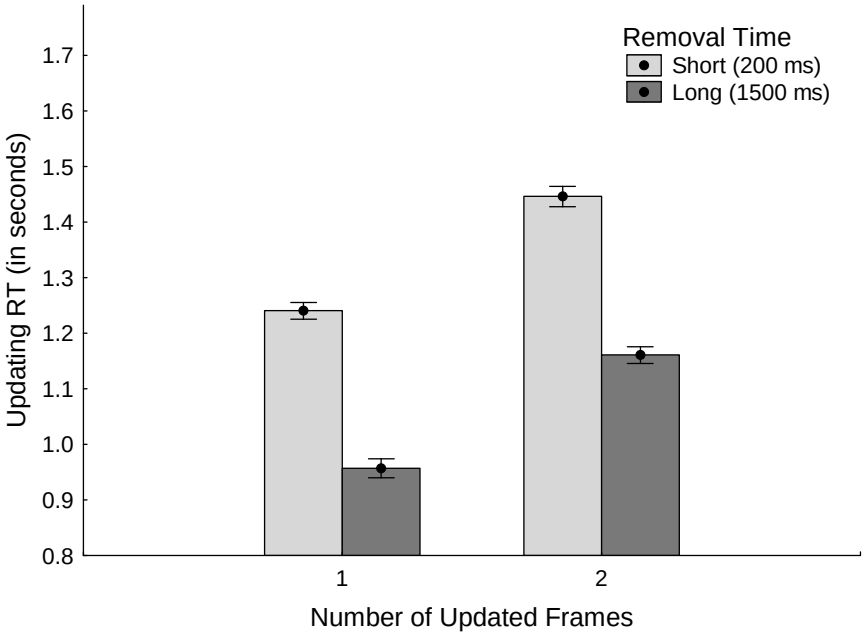
Figure 1. A trial sequence from Experiment 1. The example trial features 4 updating steps. Across trials, the number of updating steps ranged from 1 to 21, with a 10% termination probability after each step; this yielded a mean number of approximately 9 updating steps per trial. Note that the duration of the empty-frames interval after each updating step was determined by the length of the subsequent removal-cue interval: If the removal-cue interval was short (200 ms), the empty-frames interval was long (1800 ms), if the removal-cue interval was long (1500 ms), the empty-frames interval was short (500 ms), and hence the retention interval between updating steps was constant at 2000 ms.

Figure 2. Updating response times from Experiment 1. Vertical bars denote 95% within-subject confidence intervals (?, ?).

Figure 3. Updating response times from Experiment 2. Vertical bars denote 95% within-subject confidence intervals (?, ?).


+
Fixation cross; 1000 ms
B T D
Study Input; 2000 ms; Encode B T D
(1800 ms vs. 500 ms)
Remove T; remember B … D; 200 ms vs. 1500 ms
S
Encode S and remember B S D; press Space; max. 5000 ms
(1800 ms vs. 500 ms)
Remove D; remember B S … ; 200 ms vs. 1500 ms
X
Encode X and remember B S X; press Space; max. 5000 ms
(1800 ms vs. 500 ms)
Remove S; remember B … X; 200 ms vs. 1500 ms
K
Encode K and remember B K X; press Space
(500 ms)
?
Recall and type K; max. 5000 ms
?
Recall and type B; max. 5000 ms
?
Recall and type X; max. 5000 ms
x out of 3 correct
(2000 ms)
time
Inter-trial interval; 2500 ms


Removal Time
Short (200 ms)
Long (1500 ms)
Updating RT (in seconds)
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
1
2
3
Number of Updated Frames


Removal Time
Short (200 ms)
Long (1500 ms)
Updating RT (in seconds)
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
1
2
Number of Updated Frames