# The Benefit of Removing Information from Working Memory: Increasing Available Cognitive Resources or Reducing Interference?

Chenyu Li, Gidon T. Frischkorn, Hannah Dames & Klaus Oberauer

University of Zurich

**Author Note**

This research was supported by grants from the Swiss National Science Foundation to Klaus Oberauer (project 100014_179002) , Gidon Frischkorn (project PZ00P1_208828), and from the China Scholarship Council to Chenyu Li (No. 202106320040).

**Abstract**

Removing information from working memory is thought to free up capacity and improve the retention of other information. However, whether this benefit arises from reducing interference or freeing up cognitive resources remains unclear. We examined this by comparing removal immediately following encoding an item, or delayed until after other items have been encoded. Interference theories predict that both types of removal should reduce interference and improve memory performance. In contrast, if removal frees up resources, the beneficial effect on memory should be larger the earlier it occurs. Experiment 1 showed that both immediate and delayed removal failed to reduce interference from the to-be-forgotten items but improved memory for item-location bindings of other items still maintained in working memory. In Experiment 2, removal only facilitated item-location bindings for items encoded afterward. These results suggest that removal frees up working memory capacity by increasing available resources rather than by reducing interference.

*Keywords*: Working Memory, Removal, Interference, Cognitive Resources, Memory Measurement Model

## Open Practice

Both experiments were not preregistered. All study materials, primary data, and analysis scripts are publicly available (https://osf.io/9w6pa/).

## Statement of Relevance

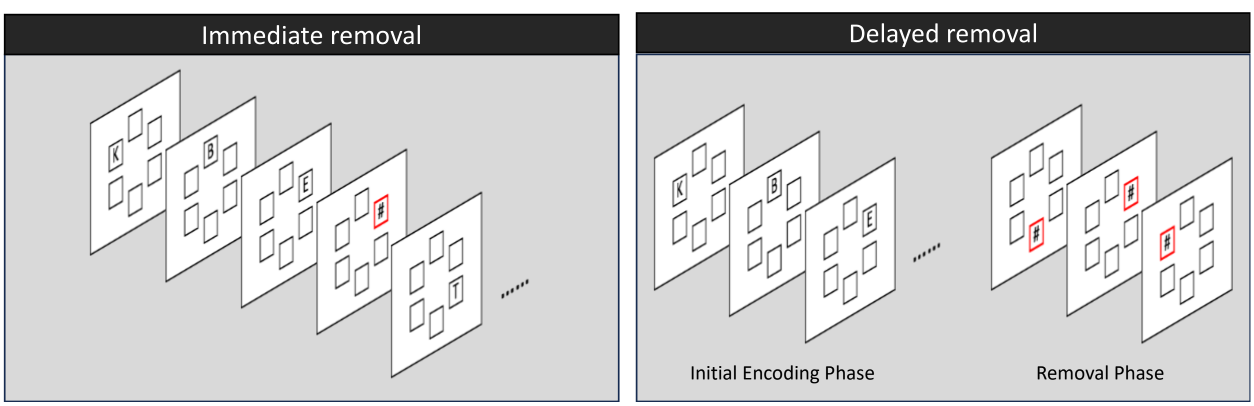
Working memory is a system that holds and processes information for the current task. Due to its limited capacity, only a small amount of information can be maintained at once. Removal is the process of freeing up capacity in working memory by eliminating irrelevant information. Ideally, it should benefit the memory of other information in working memory. However, the benefit of removal only appears when items are removed immediately after encoding and disappears when items are removed with a delay. This study addressed the discrepancy and clarified the benefit of removal through two experiments involving both types of removal. We found that both types of removal failed to fully remove irrelevant items from working memory but still benefited the binding memory of subsequently encoded information. The benefit of removal is limited to items encoded subsequently. This explains the discrepancy between the two types of removal in previous studies: Only in immediate removal but not delayed removal, removal is followed by encoding of further information.

**The Benefit of Removing Information from Working Memory: Increasing Available Cognitive Resources or Reducing Interference?**

Humans continuously interact with the environment. During these interactions, we constantly update the information in our working memory (WM) to keep it relevant to the task at hand. Updating information in WM consists of two sub-processes: removal of outdated information and encoding of new information (Ecker et al., 2014). Removal of information from WM is critical for updating as it should free up WM capacity for new information to be encoded into WM (Lewis-Peacock et al., 2018; Oberauer et al., 2012; Taylor et al., 2023). Successfully removing irrelevant information should enhance memory for the remaining items. Such a benefit was observed in studies where information was removed immediately after encoding (*immediate removal*, shown in left panel of Fig. 1), but not in studies where information was removed with a delay (*delayed removal*, shown in right panel of Fig. 1). The purpose of this study is to elucidate why immediate removal improves memory for the remaining information in WM whereas delayed removal does not.

**Fig. 1**

*Illustrations of varied removal conditions from Li et al. (2024). A frame with a red border and a "#" means that the letter previously presented in that frame should be forgotten.* ***Immediate removal****: Items were cued to be forgotten immediately after encoding them;* ***Delayed removal****: After all items have been encoded, some of them were cued as to be forgotten.*




There are two main perspectives on how the removal frees up WM capacity. According to interference theories of WM (Farrell & Lewandowsky, 2002; Oberauer et al., 2012), representations in

WM interfere with each other, leading to decreased performance when retrieving information. Removal of irrelevant information from WM diminishes interference and thereby can improve the retrieval performance of other information held in WM (Lewis-Peacock et al., 2018; Oberauer et al., 2012). By contrast, according to resource theories of WM (Adam et al., 2017; Schneegans & Bays, 2017; Sewell et al., 2014; Smith et al., 2016), maintaining information in WM requires allocating part of a limited resource to representations in WM. That resource can be released by removing some representations from WM and then reallocated to other information to be maintained in WM, thereby enhancing their memory. According to both perspectives removal should facilitate the retrieval of other information in WM. However, this benefit was only observed in the studies where information was removed immediately after encoding it (Dames & Oberauer, 2022; Oberauer, 2018). When the information was removed with a delay, the benefit disappeared (Dames et al., 2023; Li et al., 2024; Oberauer, 2018, Experiment 2).

One possible explanation for this discrepancy is that delayed removal is less effective than immediate removal. Results from Li et al. (2024) suggest that delayed removal still leaves substantial information of the to-be-forgotten (TBF) items in WM. Using a Memory Measurement Model ($M^3$, Oberauer & Lewandowsky, 2019), Li et al. (2024) measured the memory strength of both to-be-maintained (TBM) items and TBF items after removal. The $M^3$ estimates strength of item memory (i.e., which items were held active) and binding memory (i.e., to which the context items were bound to) separately. In their study, item memory of TBF items was fully retained in WM after removal, but binding memory was weakened. Therefore, when the TBF items were tested through their contexts, they were less accessible than the TBM items. However, the remaining unbound representations of items in WM still interfered with the TBM items in WM, which could explain the lack of a removal benefit. In contrast to these findings, in studies using immediate removal, there is evidence that not only binding memory but also item memory of TBF items is partially removed from WM (Dames & Oberauer, 2022). These findings could explain why the retrieval performance of other information held in WM benefits from removal. Therefore, one possibility is that information can be removed more successfully immediately after encoding rather than after a delay. Hence, only immediate removal substantially reduces interference of TBF information with other information held in WM, thereby improving retrieval performance.

Another explanation for the differences between immediate and delayed removal could be methodological variations in the existing experiments using immediate and delayed removal. In experiments employing immediate removal, most removed items are followed by new items that are encoded after the removal. In experiments investigating delayed removal, the removal cues are presented after the entire memory set has been encoded, and hence, no new items are encoded after delayed removal. According to interference theories of WM, the presence of items after removal should not make a difference, because interference occurs at retrieval: The interfering effect of one item affects all other items regardless of whether they were encoded before or after (Farrell et al., 2016; Oberauer et al., 2012). By contrast, in some resource theories of WM, removal frees up cognitive resources from WM, which can only be allocated to subsequently encoded items. In the sample size model (Sewell et al., 2014; Smith et al., 2016), and in the neural population model (Schneegans & Bays, 2017), the maintenance resource is conceptualized as a large population of information coding units that can be assigned to represent the information in a given stimulus that needs to be maintained in WM. Each unit codes a noisy sample of the relevant information from the stimulus (e.g., the orientation of an arrow, or the identity of a letter). The more units are assigned to code the same relevant information redundantly, the more precisely that information can be read out because averaging across more units cancels more of the noise. In this kind of resource model, when an item is removed from WM, units assigned to it are released, and could be added to the set of units used to sample information from earlier presented stimuli, thereby increasing the precision for that stimulus. However, assigning the released unit to a previously encoded stimulus does not yield such a benefit. At this point the previous stimulus is no longer perceived, and therefore the units re-assigned to it cannot take independent samples of the stimulus information. Therefore, they cannot contribute to making the memory representation of a preceding stimulus more precise. Hence, re-assigning freed-up resource units to previously encoded items yields no benefit. The same argument holds for models assuming that WM capacity is limited by a discrete number of place holders for items (Adam et al., 2017; Zhang & Luck, 2008).[1] When an item is removed, its placeholder becomes available. It can be used to encode a subsequently presented item, but it cannot be used to improve the representation of a previously presented item that has not already received a placeholder. Hence, according to resource theories, the benefit of removal applies exclusively to subsequently encoded items. Because in previous experiments using delayed removal no items were encoded after removal, the benefit of removal was observed only in studies using immediate removal.

In the following experiments, we aimed to investigate whether the benefit of removal arises due to the reduction of interference or increase of available maintenance resources. In Experiment 1, we included both immediate and delayed removal conditions. In both conditions, items could be encoded either before or after removal. If items in WM can only be removed with immediate removal, TBF items should be more successfully removed with immediate removal rather than delayed removal, and removal benefits should only be observed for immediate removal. By contrast, if removal only benefits items encoded afterward, the benefit of removal should be observed for both immediate and delayed removal because both forms of removal are followed by subsequently encoded items. In Experiment 2, we experimentally separated items encoded before removal from items encoded after removal and examined whether removal only benefits subsequently encoded items.

## Experiment 1

### Method

#### *Participants*

A total of 60 native English speakers, aged 18 to 35, were recruited online through Prolific. Participants took part in the experiment after providing informed consent and were free to withdraw at any time. We aimed to recruit 60 participants for this experiment based on previous experience with these types of experiments. We used Bayesian statistics for the data analysis. It implies that the sample size

[1] The discrete-capacity model is a special case of the sample-size model with very few units, and each item receives either one of them ore none (Schneegans et al., 2020)

could have been increased (i.e., ten participants at a time) in case of inconclusive evidence (Schönbrodt & Wagenmakers, 2018).

### ***Design***

There were two baseline conditions (set-size 5 and set-size 8) plus four removal conditions in this experiment. In the set-size 8 baseline condition, eight letters were displayed in each trial without being cued for removal. The set-size 5 baseline condition was similar to the set-size 8 condition, but three randomly selected letters were replaced by "#", so that only five letters were displayed. In each removal condition eight letters were displayed. A random three of them were cued as to be forgotten; some immediately after their presentation, and others later during the presentation sequence. The number of letters being removed immediately ranged from 0 to 3, whereas the other items were removed with a delay. This results in four removal conditions differing in how the TBF items were divided among immediate and delayed removal. We name each removal condition as "N (immediate removal) - N (delayed removal)". For example, the condition 2-1 indicates a condition consisting of two immediate removal and one delayed removal.

### ***Procedure***

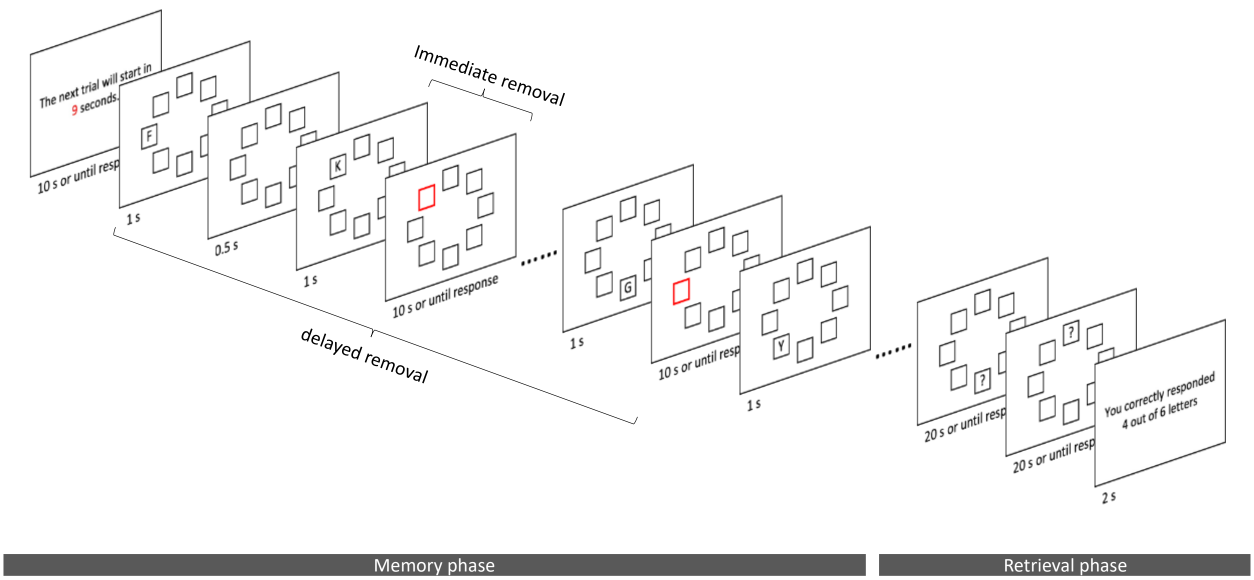
The experiments were programmed using the free online study builder jsPsych (Kuroki, 2021; Leeuw et al., 2023) and hosted on a JATOS server (Lange et al., 2015). The procedure for each trial is illustrated in Fig. 2. Each trial was started by pressing the space bar or waiting for a 10-second countdown to finish. Eight frames with thin black outlines arranged on a circle were displayed on the screen. Letters appeared sequentially in clockwise order within these frames, starting at the 9 o'clock position. Each letter was displayed for 1 second, with a 0.5-second interval between them. Following the offset of a letter, one of the frames could be highlighted in bold and red, cueing participants to remove the letter in that frame from WM. The frame could either be the frame in which the last letter has been presented (immediate removal) or another frame in which a letter has been presented before (delayed removal). Participants were instructed that the letter corresponding with the cued frame would not be tested during the retrieval phase and could be removed from WM. There were always three letters to be removed in each trial in the removal conditions. Each step of removal lasted until participants pressed the "Space bar" or for a maximum of 20 s. The next letter would be displayed after a 0.5-second interval.

A memory test followed after the memory encoding phase. All frames with TBM letters were tested in a random order. A question mark was displayed in the tested frame, and participants had a maximum of 20 s to recall the correct letter by typing it into the frame. Participants received feedback (e.g., "You correctly recalled 3 out of 5 letters") for 2 s after completing all the tests in one trial.

We assigned 8 trials to each condition, which were randomly intermixed. Participants completed one trial for each condition to practice. After completing half of the experiment (i.e., 24 trials), participants were given a 30-second break to mitigate exhaustion for the remainder of the experiment. The experiment lasted approximately 40 min for each participant.

**Fig. 2**

*Illustration of the experimental procedure used in Experiment 1*




### *Data analysis*

We only report the results from the response choices because the results from the reaction time data were not relevant to the purpose of this study. Results from the reaction time data for both Experiment 1 and 2 can be accessed in the supplementary materials.

**Accuracy and TBF intrusion analysis.** We analyzed the proportion of correct responses, and the proportion of responses in which TBF items were erroneously reported, using Bayesian generalized liner mixed models assuming a Binomial data distribution predicted by a linear model through a logistic link function. For the regression coefficients, we used normal priors with a mean of 0 and SD of 1. For the standard deviations of random effects, we used weakly informative half Student t-priors with three degrees of freedom and a scaling parameter of 2 on the positive real line.

In addition, to determine whether the TBF items were reported by chance or still partially maintained in WM, we compared the proportion of responses for TBF items with those for not-presented lures (NPL), which are items that were not presented in the current trial. In this model we directly estimated the probability of responding to each response category in each condition. However, it would be unfair to compare the TBF items and NPL directly because the number of response candidates differed: three for the TBF items and 18 for the NPL. Therefore, prior to computing BF, the prior and posterior probability distributions for each response category were divided by the number of candidates in that category. This results in probability estimates for recalling an individual letter of the alphabet, given that it is in the TBF category, or given that it is in the NPL category. From these prior and posterior probability distributions we computed the Savage-Dickey density ratio for the difference between the TBF items and NPL.

**Memory measurement model.** We applied the $M^3$ to estimate the memory strength of the TBM items, as well as the effectiveness of removing TBF items. The $M^3$ is a computational measurement model framework for WM (Oberauer & Lewandowsky, 2019). It describes distinct processes in WM that lead to correct responses and various types of errors in a WM task. In this model, the probability of producing different types of responses depends on the competition among them. The response candidates receive different degrees of activation reflecting the strength of evidence for them on the basis of information from WM. Response candidates with higher activation have a higher probability of winning the competition and being selected. Based on these assumptions, memory strength for both TBM and TBF items was estimated using the frequencies of different responses.

The responses in this study were classified into four categories: A response is considered *correct* if items were correctly recalled in their respective position. When an item from the memory set was recalled in another position, this is referred to as *transposition*. Although we did not test the position displaying TBF items, these items could still be occasionally recalled when we tested the positions for TBM items; we coded recall of a *TBF item* as a separate response category because it informs the model about the remaining strength of TBF items in memory. The last response category is the *NPL*, reflecting selection of an item that was not presented in the current trial. According to the $M^3$, each response in the four categories receives memory activation as given by the following model equations:

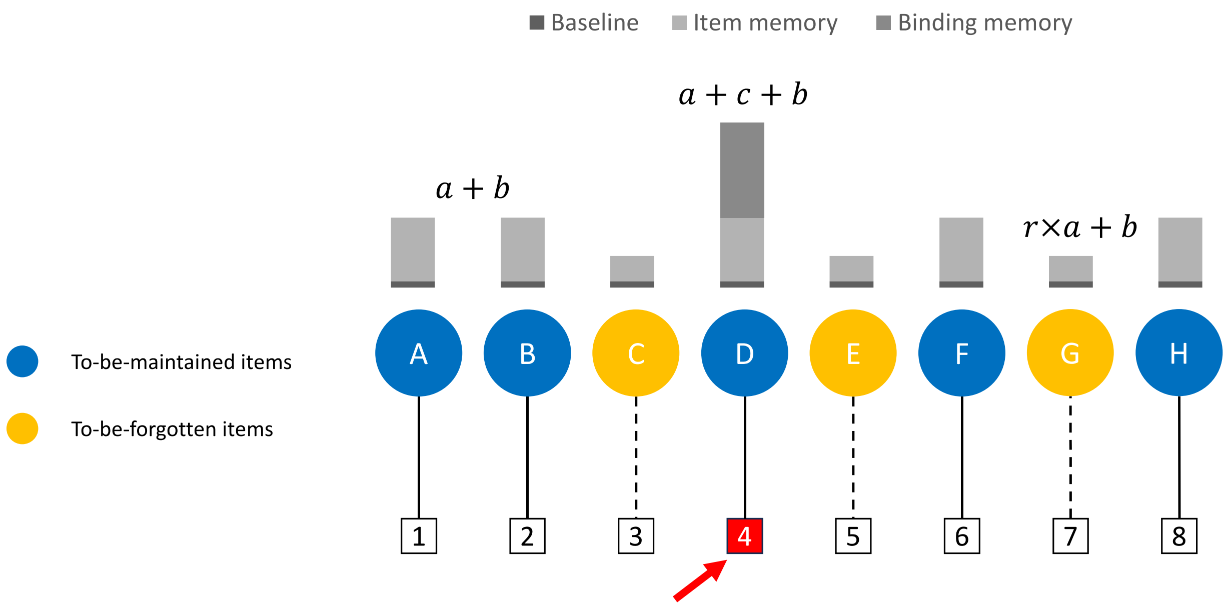
$$\begin{aligned} A(correct) &= a + c + b \\ A(transposition) &= a + b \\ A(TBF\ item) &= r * a + b \\ A(NPL) &= b \end{aligned} \tag{1}$$

Parameter $a$ represents the memory strength for the item itself (*item memory*, i.e., the identity of the presented letters), whereas parameter $c$ represents the memory strength for the item-context relation (*binding memory*, i.e., the bindings between the letters and their corresponding positions). The *correct* letter, as well as letters categorized as *transposition* errors, receive activation from item memory, because these letters are maintained active in WM with activation $a$. In addition, both of them, like every other response candidate (i.e., every letter in the alphabet), receives the baseline activation $b$.

When a position is tested, this position serves as a retrieval cue to reactivate the letter bound to it. The amount of activation generated from the position cue, received only by the correct letter, reflects the strength of binding, $c$. Since TBF letters were encoded in the same way as TBM letters, we assumed that they would receive the same activation during encoding. Because the positions of TBF items are never tested, recalled TBF items must come from another than the tested position, and hence they are not bound to the tested position. Therefore, their activation only reflects item memory $a$. However, the activation of the TBF letters should be weakened to some extent when participants are asked to forget them. We introduce the proportional removal parameter $r$, ranging from 0 to 1, to represent the residual proportion of TBF item-memory strength compared to the TBM items. The item-memory strength of TBF items is represented by $r \times a$. Finally, as the letters in the NPL category are not presented in WM at all, they receive no activation from memory. A simple example for describing the logic underlying the $M^3$ framework is illustrated in Fig. 3.

**Fig. 3**

*Illustration of an example for the logic of the $M^3$ framework. The example describes the memory activation of each item in a trial of a removal condition when position 4 is cued for participants to retrieve. In this trial, items C, E, and G were cued to-be-forgotten. Parameter b represents the baseline activation of every letter. Parameters $a$ and $c$ represent the item memory and binding memory, respectively, Parameter $r$ represents the residual proportion of item memory for to-be-forgotten items after removal.*




The chance of choosing each response candidate depends on its level of activation relative to that of all other recall candidates. We used the exponentiated version of Luce's choice rule to translate them into probabilities:

$$p(i) = \frac{N_i \times \exp\,(A(i))}{\sum_{j=1}^{n} N_j \times \exp\,(A(j))} \tag{2}$$

In the equation, $A(i)$ represents the activation value of response category $i$. The probability of retrieving an individual candidate is calculated from its exponentiated activation value, normalized by the sum of exponentiated activation values over all candidates (here: $n$ letters of the alphabet). To obtain the probability of retrieving a candidate from the response category $i$ we multiply the exponentiated activation values for candidates in that category by the number of response candidates $N_i$ in the category.

We fit the $M^3$ using a multinomial distribution to describe the frequency distribution over response categories. Data from trials where participants skipped responding or did not respond within the maximum time were excluded from this analysis, because they don't contribute to the frequency distribution. Each parameter in the $M^3$ was estimated separately for each experimental condition. We used moderately informative normal priors with a mean of 3 and SD of 1 for the $a$ and $c$ parameters. Parameters $r$ was initially estimated on the real line and then transformed by the logistic function to restrict the value between 0 and 1. We used logistic priors with a mean of 0 and standard deviation of 1 for them, resulting in the priors being uniformly distributed between 0 and 1 after transformation. Parameter $b$ was set to 0 for scaling purposes.

All our models were fitted using the R package brms (Bürkner, 2017), including random participant effects on all estimated parameters (Oberauer, 2022). We sampled through eight independent Markov chains with 10,000 iterations each (including 2,500 warm-up each). We computed Bayes Factors (BFs) through the Savage-Dickey density ratio (Wetzels et al., 2010) between the estimated posterior distribution and the prior distribution using bayestestR package (Makowski et al., 2019). We considered a $BF_{10}$ larger than 3 to provide evidence in support of an effect, whereas a $BF_{10}$ smaller than 0.33 provides evidence against an effect (Kass & Raftery, 1995). However, the Savage-Dickey Density Ratio approach does not support model comparison of non-nested models. Thus, we compared different $M^3$ by estimating BFs through bridge sampling (Gronau et al., 2020), repeating the process ten times for each comparison to assess the stability of BF estimation. For these, we report the median value of the 10 BFs. Again, we regard a $BF_{10}$ larger than 3 as strong enough evidence to accept an alternative model.

## Results

### *Accuracy*

The accuracy across all conditions is shown in the left panel of Fig. 4. The accuracy in the set-size 5 condition was higher than in the set-size 8 condition ($BF_{10} = 2.18\times10^7$), indicating a set-size effect common for tests of WM. To assess the benefit of removal, we compared the accuracy across the four removal conditions with both the larger (set-size 8) baseline and the smaller (set-size 5) baseline. Without benefit from removal, performance in the removal conditions should be as bad as in the set-size 8 condition. A maximal benefit of complete removal would bring accuracy in the removal conditions to the level of the set-size 5 condition. Hence, the benefit of removal increases as accuracy approaches the set-size 5 condition, and decreases as accuracy nears the set-size 8 condition. We found strong evidence for the accuracy of the condition with three immediate removals (i.e., 3-0) being equal to accuracy in the set-size 5 condition ($BF_{10} = 0.21$) and higher than in the set-size 8 condition ($BF_{10} = 9.87\times10^6$), indicating a virtually perfect removal benefit. However, accuracy decreased with the increase of the number of delayed removals. The accuracy in the condition with three delayed removals (i.e., 0-3) was lower than in the set-size 5 ($BF_{10} = 3.29\times10^3$), and lower than in the condition with three immediate removals (i.e., 3-0), but still higher than that in the set-size 8 condition ($BF_{10} = 7.86$). These results show that both immediate removal and delayed removal facilitated the retrieval of the TBM items, with the facilitation effect from immediate removal being larger than that from delayed removal.

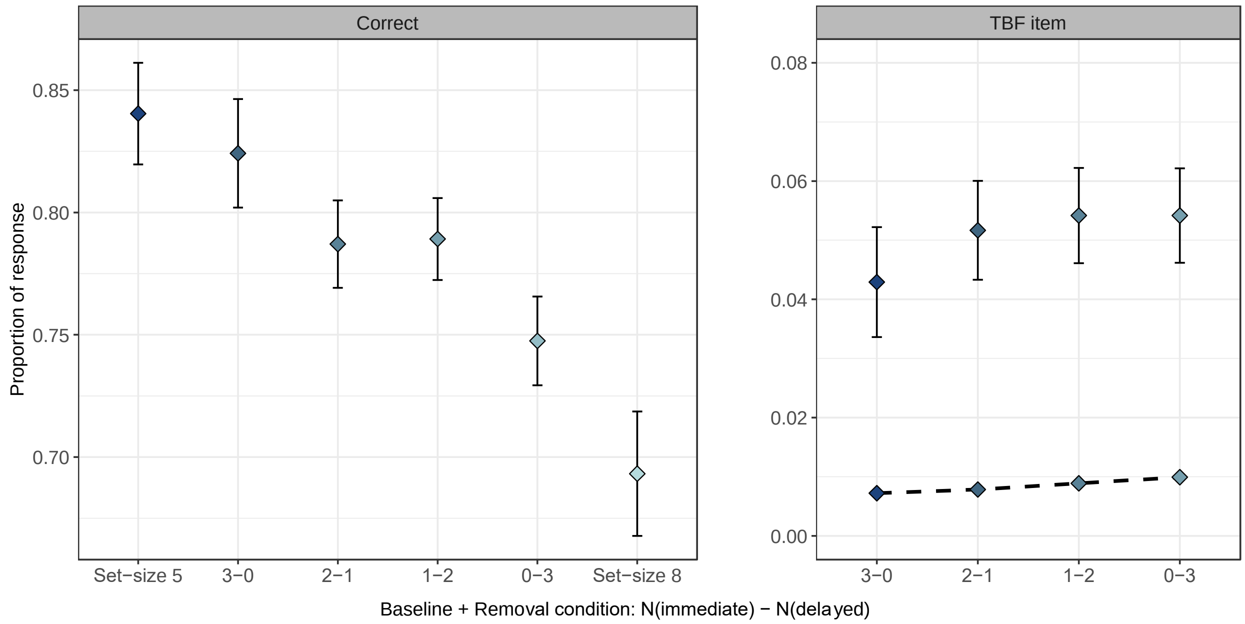
***Proportion of TBF items***

The proportion of responses for TBF items across all the removal conditions is shown in the right panel of Fig. 4. We first tested whether the TBF items were still retained in WM or were recalled merely by chance. We compared the proportion of responding TBF items and NPL to the control group, taking into account the number of response options in each category. There was overwhelming evidence supporting that a TBF letter had a higher probability of being recalled than an NPL letter ($BF_{10} = 5.52 \times10^{13}$), suggesting that the TBF items were still to some extent retained in WM after removal.

To examine whether the TBF items were removed more successfully with immediate removal than delayed removal and therefore led to higher accuracy, we compared the proportion of TBF items that were erroneously recalled across the four removal conditions. There was strong evidence against a difference between each pair of the removal conditions (BFs < 0.11), except for the condition with three immediate removals, for which some pair-wise comparisons had ambiguous BFs (BFs < 0.49).

**Fig. 4**

*Proportion of correct responses and of error responses reporting a TBF item across conditions in Experiment 1*




*Note*. Error bars represent the 95% confidence within-subject interval. The dashed lines in the right panel represent the recall probability of the TBF items expected if they were chosen with the same probability as NPLs, correcting for the number of letters in each category.

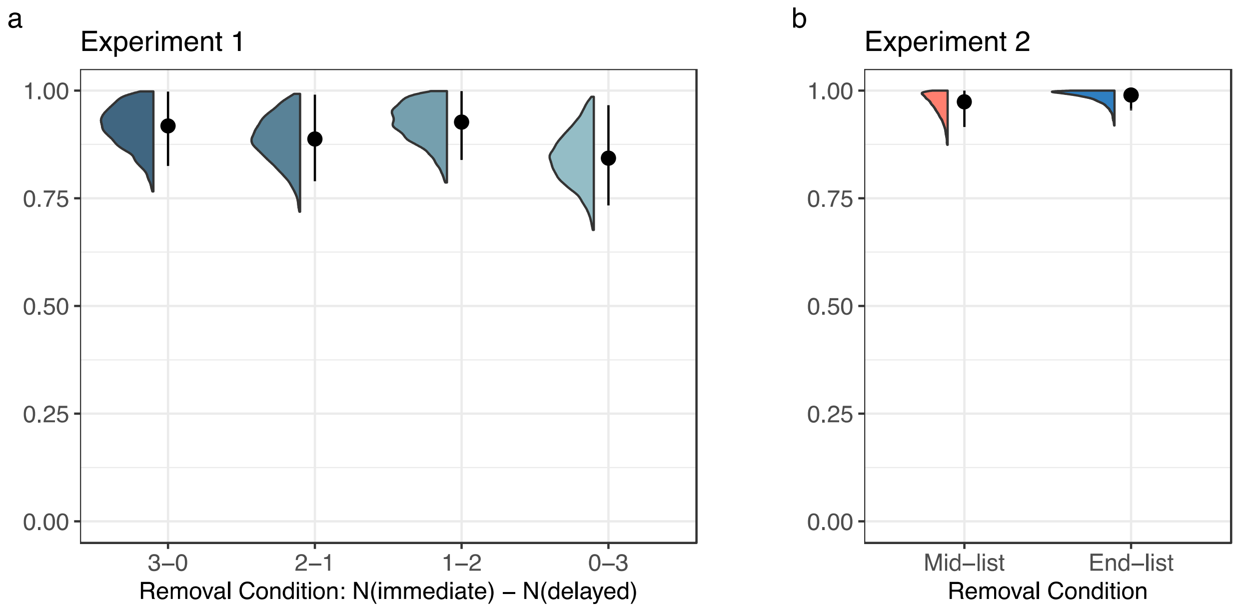
### ***Memory Measurement Model***

To further investigate which factor drives the benefit of removal, we fit the $M^3$ to the data. Results for all model parameters, the proportion of responses for each category, as well as predicted results based on the best $M^3$ (see below) of each experiment, are given in the supplementary material.

**Removal of TBF items.** Parameter $r$ represents the residual proportion of TBF items in the model. Estimates of it for each removal condition are shown on Fig. 5a. The posterior distribution indicated that the TBF items was barely removed from WM in each condition, with similar results across all conditions (BFs < 0.33). We examined whether the TBF items were completely retained in WM with the same strength as the TBM items by comparing the original model with a constrained model that fixed parameter $r$ to 1. There was overwhelming evidence against the constrained model ($BF_{median} = 2.03\times10^{-5}$), indicating that the TBF items were slightly removed in each removal conditions.

**Fig. 5**

*Parameter estimates of the proportion of remaining item-memory strength for to-be-forgotten items (parameter $r$) in Experiment 1 and 2*

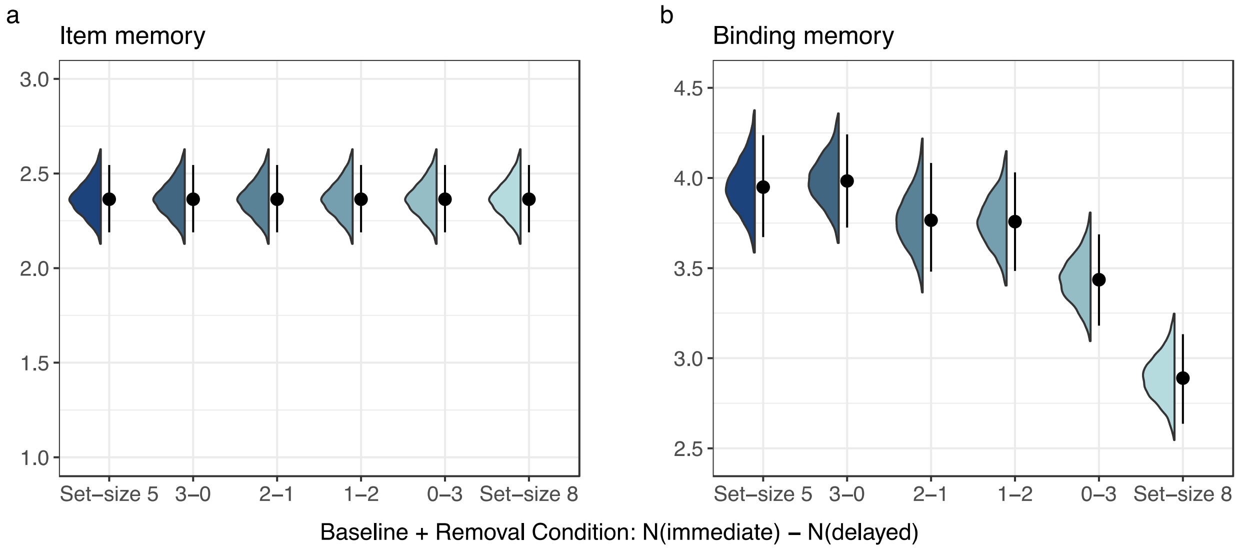



*Note*. The posterior mean (point) and 95% highest density interval (solid line range) of the parameter estimates are shown in black. The distributions illustrate the 99% posterior distribution of estimated parameters from the $M^3$.

**Memory of TBM items.** Subsequentially, we tested the binding hypothesis of WM capacity, which states that only binding memory but not item memory is affected by the number of items held in WM (Oberauer, 2019). For this, we compared the original model with a constrained model that set parameter $a$ to the same value in all conditions. The results favored the constrained model over the original model ($BF_{median}$ = 3.54 $\times 10^{12}$), thus supporting the binding hypothesis. Therefore, in the following analysis, we only compared binding memory (represented by parameter $c$) across conditions.

**Fig. 6**

*Parameter estimates of item memory (parameter $a$) and binding memory (parameter $c$) in Experiment 1*




*Note.* The posterior mean (point) and 95% highest density interval (solid line range) of the parameter estimates are shown in black. The distributions illustrate the 99% posterior distribution of estimated parameters from the $M^3$.

The parameter estimates of parameters $a$ and $c$ for each condition in the constrained model are depicted in Fig. 6. Estimates of parameter $c$ clearly show that the binding memory of TBM items in the set-size 5 condition was stronger than in the set-size 8 condition ($BF_{10} = 5.36 \times 10^{15}$), suggesting a set-size effect on binding memory. In addition, the more items are removed immediately, the stronger binding memory was, and the more it approached that of set-size 5. The binding memory of TBM items in the condition with three immediate removals (i.e., 3-0) was as strong as the set-size 5 condition ($BF_{10} = 0.15$) and stronger than the set-size 8 condition ($BF_{10} = 4.38 \times 10^{16}$). By contrast, in the condition with three delayed removals (i.e., 0-3), it was still stronger than the set-size 8 condition ($BF_{10} = 7.48$) but weaker than the set-size 5 condition ($BF_{10} = 3.77$). Overall, these results suggest that both immediate removal and delayed removal enhance the binding memory of TBM items, with immediate removal showing greater improvement than delayed removal.

## Discussion

In Experiment 1, we investigated the effectiveness of removing TBF items and the facilitation of memory for TBM items with immediate removal and delayed removal. We found that in both conditions, removal improved the memory of TBM items, although it did not reduce item-memory strength for TBF items in WM. However, we noticed that the benefit of removal in the conditions with more immediate removal was larger than in the conditions with more delayed removal. One possibility is that removal only benefits subsequently encoded items, but not the items encoded previously. In Experiment 2 we tested this hypothesis by separating the benefit of removal for previously encoded items from that for subsequently encoded items. We aimed to determine if only the subsequently encoded items benefit from removal.

# Experiment 2

## Method

### *Participant*

A total of 120 native English speakers were recruited online via Prolific, aged between 18 and 35. Participants took part in the experiment after providing informed consent and were free to withdraw at any time. Seven participants were excluded from the analysis due to technical issues preventing their data from being transferred to the server. Three participants reported in the post-experiment survey that they used aids such as pen and paper to remember stimuli during the task, and were therefore excluded from the analysis. The data from the remaining 110 participants were analyzed. The sample size in Experiment 2 was twice that of Experiment 1. Because the responses in each trial of Experiment 2 were divided into two stages (details provided below), this larger sample size resulted in the same number of observations per stage as in Experiment 1.

### *Procedure and Design*

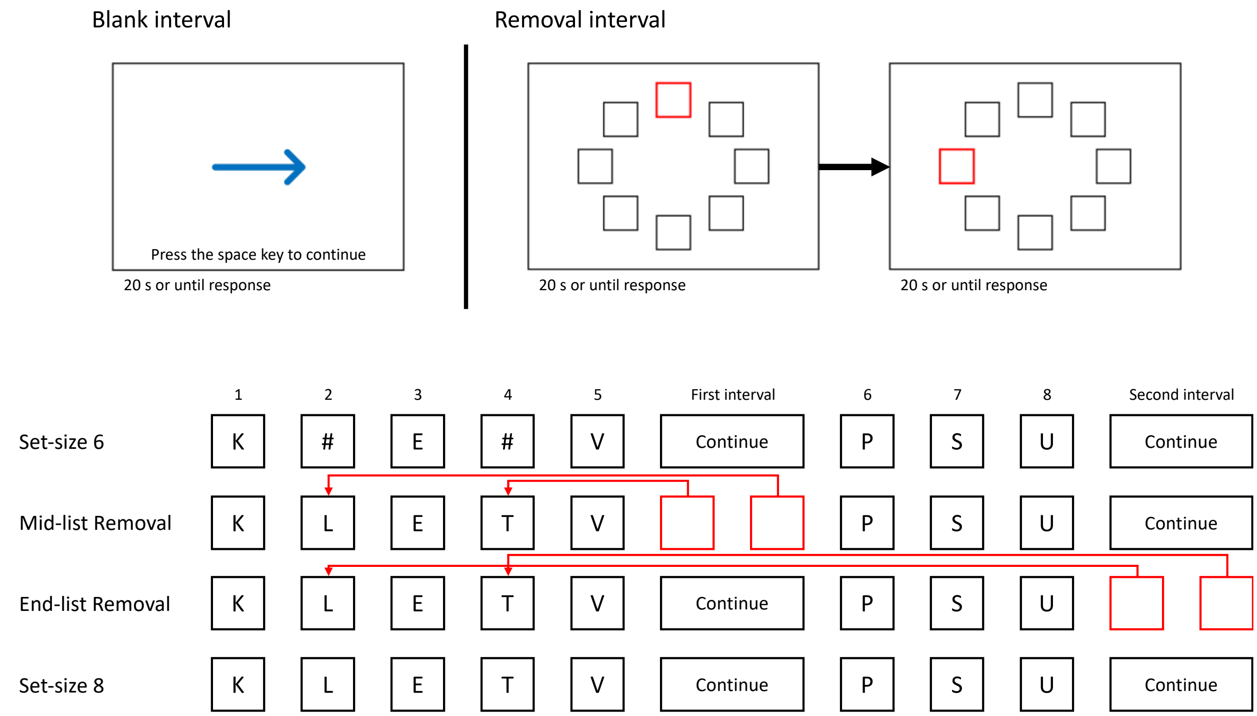
The procedure of each trial in Experiment 2 was similar to Experiment 1, with a change to the scheduling of removal cues. The encoding of letters was separated into two stages: stage 1 for the first five letters and stage 2 for the last three letters. Each stage was followed either by a blank interval or a removal interval (shown in the upper panel of Fig. 7). In the blank interval, only an arrow appeared on the screen, instructing participants to continue. In contrast, during the removal interval, two out of the first five positions were sequentially cued as to-be-forgotten. This could happen either after stage 1 or after stage 2. Participants were asked to forget the letters that corresponded to the cued positions during the removal interval. The blank interval, and each removal step in a removal interval, lasted a maximum of 20 s or until participants pressed the space bar.

There were two baseline conditions (i.e., *set-size 6* and *set-size 8*) and two removal conditions. A simple example for each condition is shown in the lower panel of Fig. 7. There were always five letters presented in stage 1 and three letters presented in stage 2 in each condition except the set-size 6 condition, where only three letters were presented in the first five positions, with the remaining two positions filled with a "#" symbol. In both baseline conditions, the two intervals were always blank intervals. The set-size 8 was the baseline condition for assessing the accuracy expected if no items were removed in the removal conditions, whereas set-size 6 served to assess the accuracy expected if both TBF items were perfectly removed. In the two removal conditions, two letters encoded in stage 1 were to be removed either in the first interval (*mid-list removal*) or in the second interval (*end-list removal*). As a result, letters encoded in stage 2 were items encoded after removal (subsequently encoded items) in the mid-list removal condition, but were items encoded before removal (previously encoded items) in the end-list removal condition. In each removal condition the interval not including removal cues was a blank interval.

There were 12 trials for each condition, which were mixed randomly in the experiment. Participants completed one practice trial for each condition before starting the experiment and had a 30-second break available after every 12 trials during the experiment. Each participant took approximately 45 min to complete the experiment.

**Fig. 7**

*Illustration of two types of intervals (upper panel) and a simple example for each condition (lower panel) in Experiment 2*




## Results

### *Accuracy*

The accuracy across stages and conditions is depicted in Fig. 8. We assessed the benefit of removal in each stage by comparing the accuracy in the two removal conditions with the two baseline conditions. There was no difference in accuracy between the two removal conditions for the items encoded in stage 1 ($BF_{10} = 0.12$). Both accuracies were poorer than in the set-size 6 condition (BFs > $3.15 \times 10^6$) but were the same as in the set-size 8 condition (BFs < 0.08). These results indicated that the items encoded in stage 1 did not benefit from removing TBF items.

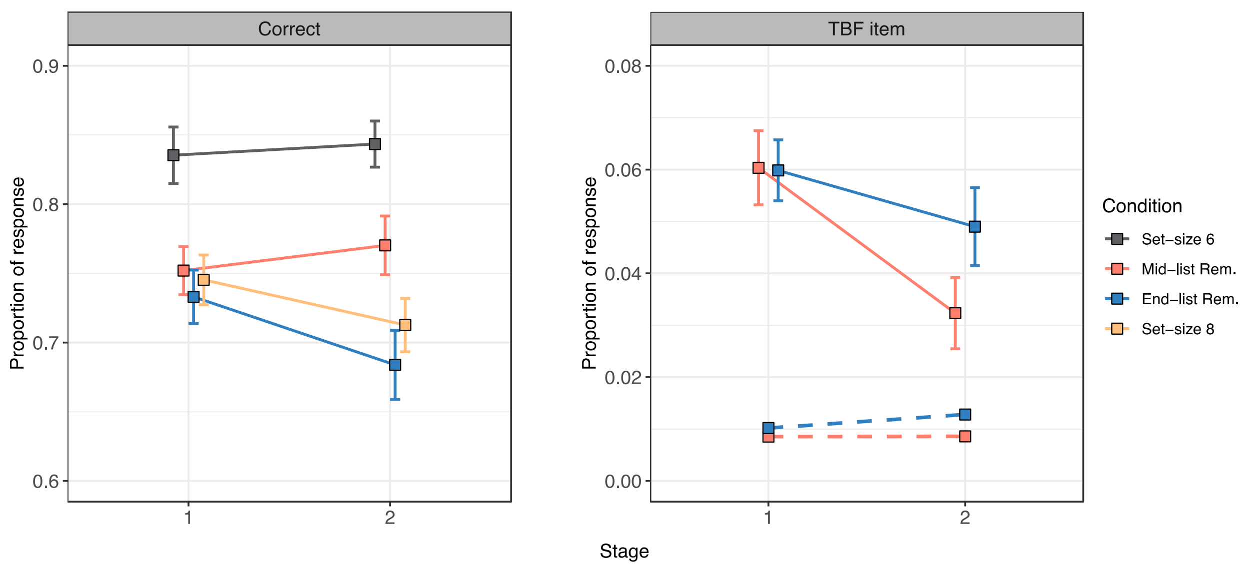
However, the accuracy in stage 2 differed between the two removal conditions. In the mid-list removal condition, stage 2 memory was better than in the end-list removal condition ($BF_{10} = 1.35 \times 10^3$). It was also better than the set-size 8 condition ($BF_{10} = 102.40$) but worse than that in the set-size 6 condition ($BF_{10} = 2.59 \times 10^3$). The accuracy in the end-list removal condition did not differ from the set-size 8 condition ($BF_{10}$ = 0.17) and was also worse than the set-size 6 condition ($BF_{10}$ = 1.36×$10^7$). These results show that only the items encoded after removal benefit from removing information from WM.

***Proportion of TBF items***

The proportion of TBF items erroneously recalled in the two stages of the two removal conditions is depicted in Fig. 8. Although only items encoded in stage 1 were cued to be forgotten, the TBF items were observed in both tested positions of stages 1 and 2. In stage 1, there was no difference between the two removal conditions ($BF_{10}$ = 0.08), whereas in stage 2, TBF items were reported less in the mid-list removal condition than in the end-list removal condition ($BF_{10}$ = 21.81). We tested whether the TBF items were still retained in WM by comparing the probabilities of recalling an individual letter from the TBF category and the NPL category. There was overwhelming evidence supporting that recall probability was higher for TBF letters than for NPL letters (BFs > 3.42 ×$10^{13}$) in each stage and in each condition, indicating that the TBF items were still retained in WM after removal.

**Fig. 8**

*Proportion of correct responses and of error responses reporting a TBF item by stages and conditions in Experiment 2*




*Note*. Error bars represent the 95% confidence within-subject interval.

***Memory Measurement Model***

To differentiate between the benefits of removal for items encoded previously and subsequently, we assigned different parameters to represent the memory of items in stages 1 and 2. Parameters $a_1$ and $c_1$ represent the item and binding memory, respectively, for the items encoded in stage 1, while parameters $a_2$ and $c_2$ represent those in stage 2. Since the TBF items were always encoded in stage 1, their memory activation is represented by $r \times a_1$. The rest of the model was almost the same as in Experiment 1. Details of the $M^3$ for Experiment 2 can be found in the supplementary materials.

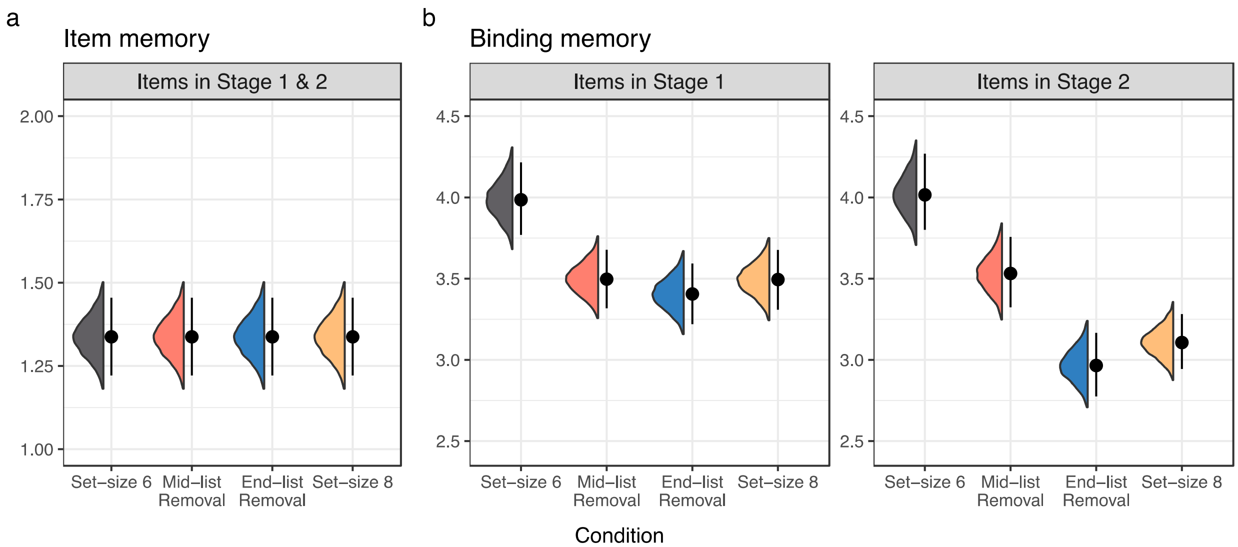
**Removal of TBF items.** We measured the residual proportion of TBF after removal using parameter $r$. Estimates of it in both removal conditions are displayed on Fig. 5b. As Experiment 1, the posterior distribution in both conditions suggested that item memory for the TBF items was barely removed from WM. We tested if these items were completely retained in WM in both conditions by comparing the original model with the model with fixed parameter $r$ = 1. The result was in favor of the constrained model ($BF_{median} = 5.48\times10^{22}$), indicating that the TBF items were completely retained in WM in both conditions.

**Memory of TBM items.** We tested the binding hypothesis, stating that item memory was the same in all conditions, regardless of the number of items held in WM, following the same steps as in Experiment 1. The results favored the model supporting this hypothesis ($BF_{median} = 5.42\times10^{39}$). Hence, we only compared binding memory across conditions.

The parameter estimates of parameters $a$ and $c$ across stages for each condition in the constrained model are depicted in Fig. 9. The binding memory strengths of the items encoded in stages 1 and 2 are represented by parameters $c_1$ and $c_2$, respectively. If binding memory benefits from removal in the two removal conditions, it should be stronger than in the set-size 8 condition and trend towards the set-size 6 condition. For the items in stage 1, the binding memory in both removal conditions was weaker than the set-size 6 condition (BFs > 240) and did not differ from the set-size 8 condition (in the mid-list removal condition: $BF_{10} = 0.16$; whereas in the end-list removal condition: $BF_{10} = 0.37$). In stage 2, the binding memory was still lower in both removal conditions compared to the set-size 6 condition (BFs > 7.57). However, in the mid-list removal condition, binding memory was stronger than in the set-size 8 condition ($BF_{10}$ = 6.18). That was not the case in the end-list removal condition ($BF_{10}$ = 0.17). These results showed that only the items in stage 2 of the mid-list removal condition benefited from removal, implying that removing information from WM benefits only the subsequently encoded items and not the items that have been encoded previously.

**Fig. 9**

*Parameter estimates of item memory (parameters $a_1$ and $a_2$) and binding memory (parameter $c_1$ and $c_2$) in Experiment 2*




*Note*. The posterior mean (point) and 95% highest density interval (solid line range) of the parameter estimates are shown in black. The distributions illustrate the 99% posterior distribution of estimated parameters from the $M^3$.

## General Discussion

The present study investigates how removal benefits other information in WM. Previous studies have shown that removal only benefits memory when items are removed immediately after encoding, but not when removal happened after complete encoding (Dames et al., 2023; Dames & Oberauer, 2022; Li et al., 2024; Oberauer, 2018). We found that immediate and delayed removal do not differ in their effectiveness of removing information from WM, but in the facilitation for the memory of items still maintained in WM. In detail, removal only benefits items encoded afterwards, not those encoded before. With more items encoded after removal, a larger benefit should be observed. This finding explains why immediate but not delayed removal yielded performance benefits in previous studies. Because in those experiments applied delayed removal (Dames et al., 2023; Li et al., 2024; Oberauer, 2018, In Experiment 2), no new items were encoded after removal. The benefit of removal has not been observed.

Our findings support resource (Adam et al., 2017; Schneegans & Bays, 2017; Smith et al., 2016) but not interference theories (Farrell et al., 2016; Oberauer et al., 2012). Yet, resource theories cannot completely account for all our results. An odd finding in both Experiment 1 and 2 is that the benefit of removal was observed alongside the failure to remove TBF items. If TBF items were not removed, how would the subsequently encoded items benefit from their removal? In refined analyses, we found that these results reflect two different dimensions of memory strength: item memory and binding memory. With respect to item memory, the TBF items were not removed; they still interfered with other items maintained in WM, resulting in them being occasionally recalled in any of the tested positions. With respect to binding memory, although we did not measure binding memory of TBF items in this study, previous studies showed that binding memory of TBF items can be mostly removed from WM (Dames et al., 2023; Li et al., 2024). The removal benefit we observed here was an improvement in the binding memory for the items to be held in WM, particularly for the subsequently encoded items. Therefore, our findings suggest that the removal of information from WM frees up resources for binding, which can then be reallocated to bind subsequently encoded items to their context.

Together with previous findings, our results converge on the following conclusions: removal of information from WM unbinds items from their contexts (Dames et al., 2023; Li et al., 2024) and frees up cognitive resources for binding. This resource can be reallocated to encode new items but cannot be used to strengthen the memory of representation that has already been held in WM before. In this process, item memory cannot be removed from WM. Therefore, representations of item identity will still interfere with other items maintained in WM. As a consequence, items encoded before removal do not benefit from removal because they are still interfered with by the TBF items and cannot benefit from receiving more of the WM maintenance resource. By contrast, the subsequently encoded items are also interfered with by the TBF items but gain from added resources, and therefore benefit from removal.

**References**


Adam, K. C. S., Vogel, E. K., & Awh, E. (2017). Clear evidence for item limits in visual working memory. *Cognitive Psychology*, *97*, 79–97. https://doi.org/10.1016/j.cogpsych.2017.07.001

Bürkner, P.-C. (2017). brms: An R Package for Bayesian Multilevel Models Using Stan. *Journal of Statistical Software*, *80*, 1–28. https://doi.org/10.18637/jss.v080.i01

Dames, H., Li, C., Frischkorn, G. T., & Oberauer, K. (2023). *Removing Information from Working Memory with a Delay: Effective but not Beneficial*. https://doi.org/10.31234/osf.io/jqc69

Dames, H., & Oberauer, K. (2022). Directed forgetting in working memory. *Journal of Experimental Psychology: General*, *151*(12), 2990–3008. https://doi.org/10.1037/xge0001256

Ecker, U. K. H., Lewandowsky, S., & Oberauer, K. (2014). Removal of information from working memory: A specific updating process. *Journal of Memory and Language*, *74*, 77–90. https://doi.org/10.1016/j.jml.2013.09.003

Farrell, S., & Lewandowsky, S. (2002). An endogenous distributed model of ordering in serial recall. *Psychonomic Bulletin & Review*, *9*(1), 59–79. https://doi.org/10.3758/BF03196257

Farrell, S., Oberauer, K., Greaves, M., Pasiecznik, K., Lewandowsky, S., & Jarrold, C. (2016). A test of interference versus decay in working memory: Varying distraction within lists in a complex span task. *Journal of Memory and Language*, *90*, 66–87. https://doi.org/10.1016/j.jml.2016.03.010

Gronau, Q. F., Singmann, H., & Wagenmakers, E.-J. (2020). bridgesampling: An R Package for Estimating Normalizing Constants. *Journal of Statistical Software*, *92*, 1–29. https://doi.org/10.18637/jss.v092.i10

Kass, R. E., & Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, *90*(430), 773–795. https://doi.org/10.1080/01621459.1995.10476572

Kuroki, D. (2021). A new jsPsych plugin for psychophysics, providing accurate display duration and stimulus onset asynchrony. *Behavior Research Methods*, *53*(1). https://doi.org/10.3758/s13428-020-01445-w

Lange, K., Kühn, S., & Filevich, E. (2015). "Just Another Tool for Online Studies” (JATOS): An Easy Solution for Setup and Management of Web Servers Supporting Online Studies. *PLOS ONE*, *10*(6), e0130834. https://doi.org/10.1371/journal.pone.0130834

Leeuw, J. R. de, Gilbert, R. A., & Luchterhandt, B. (2023). jsPsych: Enabling an Open-Source Collaborative Ecosystem of Behavioral Experiments. *Journal of Open Source Software*, *8*(85), 5351. https://doi.org/10.21105/joss.05351

Lewis-Peacock, J. A., Kessler, Y., & Oberauer, K. (2018). The removal of information from working memory. *Annals of the New York Academy of Sciences*, *1424*(1), 33–44. https://doi.org/10.1111/nyas.13714

Li, C., Frischkorn, G. T., & Oberauer, K. (2024). *Removal of Information from Working Memory: Time Course and Consequences*. OSF. https://doi.org/10.31234/osf.io/qt3f9

Makowski, D., Ben-Shachar, M. S., & Lüdecke, D. (2019). bayestestR: Describing Effects and their Uncertainty, Existence and Significance within the Bayesian Framework. *Journal of Open Source Software*, *4*(40), 1541. https://doi.org/10.21105/joss.01541

Oberauer, K. (2018). Removal of irrelevant information from working memory: Sometimes fast, sometimes slow, and sometimes not at all. *Annals of the New York Academy of Sciences*, *1424*(1), 239–255. https://doi.org/10.1111/nyas.13603

Oberauer, K. (2019). Working Memory Capacity Limits Memory for Bindings. *Journal of Cognition*, *2*(1), Article 1. https://doi.org/10.5334/joc.86

Oberauer, K. (2022). The Importance of Random Slopes in Mixed Models for Bayesian Hypothesis Testing. *Psychological Science*, *33*(4), 648–665. https://doi.org/10.1177/09567976211046884

Oberauer, K., & Lewandowsky, S. (2019). Simple measurement models for complex working-memory tasks. *Psychological Review*, *126*(6), 880–932. https://doi.org/10.1037/rev0000159

Oberauer, K., Lewandowsky, S., Farrell, S., Jarrold, C., & Greaves, M. (2012). Modeling working memory: An interference model of complex span. *Psychonomic Bulletin and Review*, *19*(5), 779–819. https://doi.org/10.3758/s13423-012-0272-4

Schneegans, S., & Bays, P. M. (2017). Neural Architecture for Feature Binding in Visual Working Memory. *The Journal of Neuroscience*, *37*(14), 3913–3925. https://doi.org/10.1523/JNEUROSCI.3493-16.2017

Schneegans, S., Taylor, R., & Bays, P. M. (2020). Stochastic sampling provides a unifying account of visual working memory limits. *Proceedings of the National Academy of Sciences*, *117*(34), 20959–20968. https://doi.org/10.1073/pnas.2004306117

Schönbrodt, F. D., & Wagenmakers, E.-J. (2018). Bayes factor design analysis: Planning for compelling evidence. *Psychonomic Bulletin & Review*, *25*(1), 128–142. https://doi.org/10.3758/s13423-017-1230-y

Sewell, D. K., Lilburn, S. D., & Smith, P. L. (2014). An information capacity limitation of visual short-term memory. *Journal of Experimental Psychology: Human Perception and Performance*, *40*(6), 2214–2242. https://doi.org/10.1037/a0037744

Smith, P. L., Lilburn, S. D., Corbett, E. A., Sewell, D. K., & Kyllingsbæk, S. (2016). The attention-weighted sample-size model of visual short-term memory: Attention capture predicts resource allocation and memory load. *Cognitive Psychology*, *89*, 71–105. https://doi.org/10.1016/j.cogpsych.2016.07.002

Taylor, R., Tomić, I., Aagten-Murphy, D., & Bays, P. M. (2023). Working memory is updated by reallocation of resources from obsolete to new items. *Attention, Perception, & Psychophysics*, *85*(5), 1437–1451. https://doi.org/10.3758/s13414-022-02584-2

Wetzels, R., Grasman, R. P. P. P., & Wagenmakers, E.-J. (2010). An encompassing prior generalization of the Savage–Dickey density ratio. *Computational Statistics & Data Analysis*, *54*(9), 2094–2102. https://doi.org/10.1016/j.csda.2010.03.016

Zhang, W., & Luck, S. J. (2008). Discrete fixed-resolution representations in visual working memory. *Nature*, *453*(7192), 233–235. https://doi.org/10.1038/nature06860

# Supplementary materials

## Reaction time results

We excluded all RTs that were faster than 0.2 s or exceeded 10 s. For each person, we removed all RTs that exceeded their mean in each removal type by 3 intraindividual SDs. The RT data was analyzed using Bayesian Linear Mixed Models, assuming a lognormal data distribution predicted by a linear model with an identity link function. For the regression coefficients, we used Cauchy priors with a mean of 0 and SD of 0.354. For the standard deviations of random effects, we used weakly informative priors on the positive real line (half student t-prior with three degrees of freedom and a scaling parameter of 2).

### *Experiment 1*

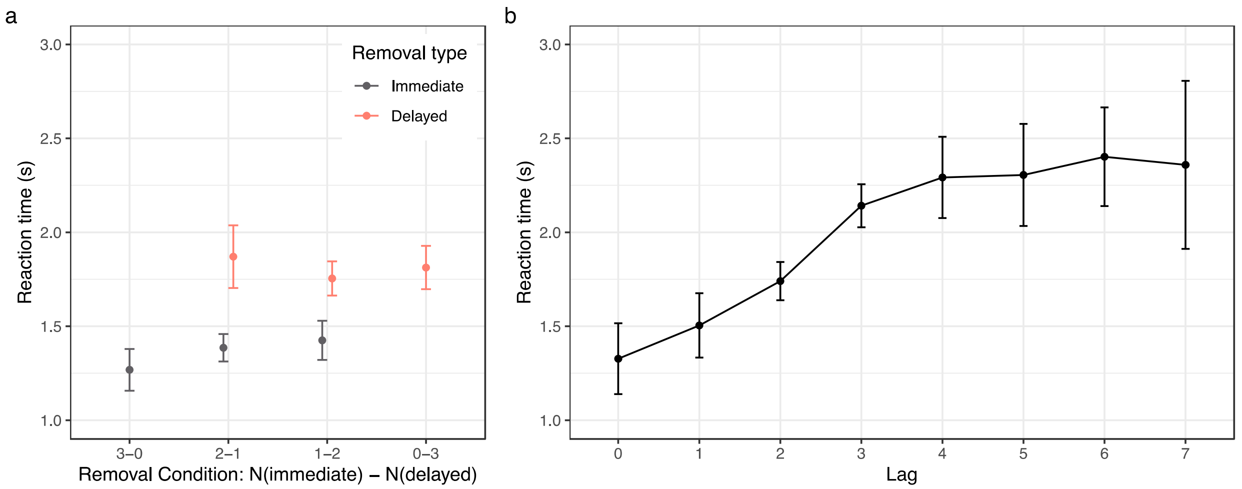
Mean RTs for both immediate removal and delayed removal across all removal condition are shown on Figure 1a. The RTs for the delayed removal in each condition are the same (BFs < 0.05). Similarly, the RTs for the immediate removal in each condition are the same (BFs < 0.22) with an exception between the 3-0 and 1-3 condition where the difference was inconclusive ($BF_{10}$ = 1.11). In the two conditions that involve both immediate removal and delayed removal, the RTs for immediate removal were faster than those for delayed removal in both conditions (BFs > 583.54). To test whether the difference in RTs between delayed removal and immediate removal was driven by the shifting of attention to another position than the one in which an item had just been encoded, we assessed the lag of TBF items from their display to being cued as TBF. Figure 1b displays RTs as a function of the removal lag. RTs at lag 0 are the RTs from immediate removal, and RTs at other lags are the RTs from delayed removal. There was overwhelming evidence supporting a lag effect ($BF_{10} = 1.80\times10^7$), suggesting that the RTs of removal increase with an extended lag period.

### *Experiment 2*

We compared the RTs of removal between mid-list and end-list removal. The mean RTs of removal were 2.14 s in the mid-list removal condition and 2.08 s in the end-list removal condition. There was weak evidence against a difference between them ($BF_{10} = 0.22$), indicating that the mid-list removal took about the same time as the end-list removal.

**Figure 1**

*Mean reaction times in Experiment 1. (a) Reaction times across all conditions vary with different removals and (b) Reaction times as a function of the removal lag. Data in lag 0 was from immediate removal, and data in the other lag was from delayed removal.*




*Note*. Error bars represent the 95% confidence within-subject interval.

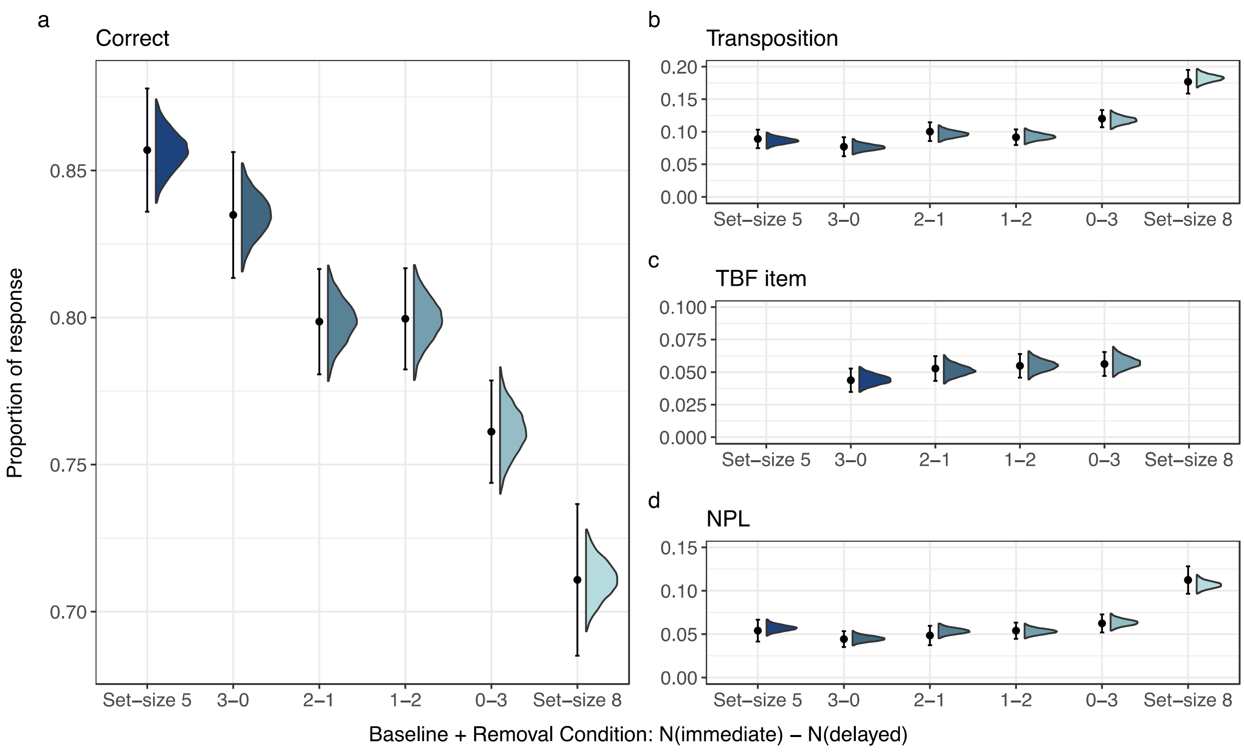
## Memory Measurement Models

### *Experiment 1*

The proportion of each response category and their predicted results based on the best-fitted $M^3$ are depicted in Figure 2. The results presented here were slightly different from those in the results section, as the data for trials in which participants did not respond within the maximum response time or skipped a response were excluded.

**Figure 2**

*Proportion of responses for each category and their predicted results based on the best fitted $M^3$ in Experiment 1. (a) Accuracy of the to-be-maintained items; (b) Proportion of responding to-be-maintained items in different positions; (c) Proportion of responding to-be-forgotten items; and (d) Proportion of responses for the not-presented lures.*




*Note*. The mean proportion of each response (point) and its 95% confidence interval (solid line range) are shown in black. The densities next to the lines range illustrate the 99% posterior distribution of estimated results for each response category from the best fitted $M^3$. The results were computed based on the data excluding the trials in which participants did not respond within the maximum response time or skipped a response.

***Experiment 2***

The model equations of the $M^3$ for Experiment 2 are as follows:

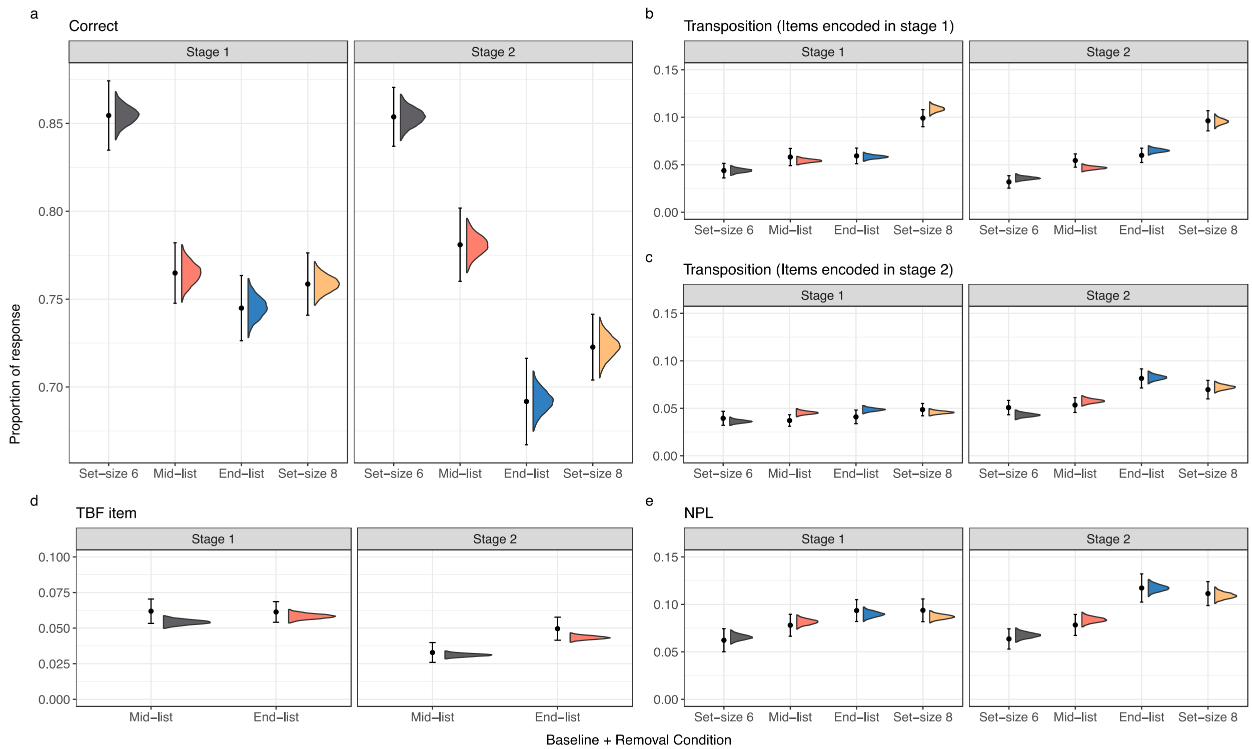
$$\begin{aligned} A(correct) &= (1-p)\times(a_1+c_1)+p\times(a_2+c_2)+g+b \\ A(transposition_{stage1}) &= a_1+(1-p)\times g+b \\ A(transposition_{stage2}) &= a_2+p\times g+b \\ A(TBF\ item) &= r\times a_1+(1-p)\times g+b \\ A(NPL) &= b \end{aligned} \tag{1}$$

Index $p$ represents the test positions. For positions in stage 1, $p$ = 0, and for positions in stage 2, $p$ = 1. With the switch of this index, the activation of correct response candidates was estimated by the equation $a_1 + c_1$ for items encoded in stage 1, and by equation $a_2 + c_2$ for items encoded in stage 2. The transposition error in Experiment 2 can be observed for items encoded in both stages 1 and 2. They were used to estimate parameters $a_1$ and $a_2$ separately. In addition, we introduced a new parameter $g$ to represent the strength of binding of each item to its group, defined by the encoding stage. That is, we assume that the first five items are bound to group 1, and the last three items are bound to group 2. The group is used as an additional retrieval cue at test. Therefore, each response candidate that was an item in group 1 receives activation from the group context when an item from group 1 is tested, and each response candidate that was an item in group 1 receives group-context activation when an item from group 2 is tested. For example, TBF items were always encoded at stage 1. Therefore, response candidates matching a TBF item receive activation from the group context whenever an item from stage 1 is tested.

The proportion of each response category and their predicted results based on the best-fitted $M^3$ are depicted in Figure 3. The results presented here were slightly different from those in the results section, as the data for trials in which participants did not respond within the maximum response time or skipped a response were excluded.

**Figure 3**

*Proportion of responses for each category and their predicted results across stage based on the best fitted M³ in Experiment 2. (a) Accuracy of the to-be-maintained items; (b) Proportion of responding items encoded in stage 1 in different positions; (c) Proportion of responding items encoded in stage 2 in different positions; (d) Proportion of responding to-be-forgotten items; and (e) Proportion of responses for the not-presented lures.*




*Note.* The mean proportion of each response category (point) and its 95% confidence interval (solid line range) are shown in black. The densities next to the lines range illustrate the 99% posterior distribution of estimated results for each response category from the best fitted M³. The results were computed based on the data excluding the trials in which participants did not respond within the maximum response time or skipped a response.

**The benefit of removal for items encoded before and after removal in Experiment 1**

To examine whether the removal only benefits subsequently encoded items or not, we counted how many removal steps occurred for each TBM item before and after they were encoded, separately by the types of removal. We predicted the retrieval success of each item by the number of removal steps preceding it and the number of removal steps following it. The regression coefficient for the number of removals that preceded encoding of the item was used to estimate the benefit of removal for subsequently encoded items. The regression coefficient for the number of removal that followed encoding of the item was used to estimate the benefit of removal of previously encoded items. We fitted the data with Bayesian Generalized Mixed Models, like the models for accuracy data, using the same setting of priors. The model specification used for all analyses is specified in Equation 2:

$$\begin{gathered} p(correct) = logit^{-1}\left(b_0 + b_1 \times N_{immediate} + b_2 \times N_{delayed} + b_3 \times position\right) \\ n \sim Binomial(k, p(correct)) \end{gathered} \quad (2)$$

The $N$ variables were entered as continuous variables in this model, representing the number of removal steps for immediate and delayed removal. In the model estimating the benefit of removal for subsequently encoded items, the $N$ variables represent the number of removal steps that occurred after the items were encoded. In the model estimating the benefit of removal for previously encoded items, they represent the number of removal steps that occurred before the items were encoded. The position variable represents the serial position of the item in the list, and was contrast coded.

The results show that accuracy of the subsequently encoded items increased with more steps of removal before encoding, both for immediate ($b$ = 0.45, 95% CI = [0.32, 0.57], $BF_{10}$ = $6.29\times10^5$) and delayed removal ($b$ = 0.44, 95% CI = [0.31, 0.57], $BF_{10}$ = $2.85\times10^5$). However, the accuracy of the previously encoded items decreased with more steps of removal after encoding, both for immediate ($b$ = -0.25, 95% CI = [-0.38, -0.12], $BF_{10}$ = 77.27) and delayed removal ($b$ = -0.44, 95% CI = [-0.57, -0.30], $BF_{10}$ = $5.98\times10^4$). These results indicated that both immediate and delayed removal benefit the accuracy of subsequently encoded items, but not the accuracy of previously encoded items.