Cogn Comput
DOI 10.1007/s12559-014-9294-8

Is Attentional Refreshing in Working Memory Sequential? A Computational Modeling Approach

Sophie Portrat · Benoît Lemaire

Received: 22 July 2013 / Accepted: 7 July 2014

Abstract Short-term memorization of items while performing a concurrent distracting task requires maintenance processes. The time-based resource-sharing model of working memory (Barrouillet et al. in Psychol Rev 118:175–192, 2011) and its computational version TBRS* (Oberauer and Lewandowsky in Psychon Bull Rev 18:10–45, 2011) proposed that items are refreshed when attention is not captured by the distracting activity. However, these models are unable to account for human performance on the last items when temporal constraints are substantial. The present study presents an analytic approach and computational simulations showing that the sequentiality of the domain-general attentional refreshing mechanism is responsible for the discrepancy between humans and model. It is suggested that the focus of attention could be flexible. The implementation of a computational model based on this solution provides a much better fit to human data. Outcomes are discussed in reference to contemporary works on the phonological loop as well as in reference to other computational models of short-term memory.

Keywords Working memory · Computational model · Time-based resource-sharing model · TBRS* · Refreshing · Serial recall

S. Portrat · B. Lemaire
Univ. Grenoble Alpes, LPNC, 38040 Grenoble, France
e-mail: Benoit.Lemaire@upmf-grenoble.fr

S. Portrat · B. Lemaire
CNRS, LPNC UMR5105, 38040 Grenoble, France

S. Portrat (✉)
Laboratoire de Psychologie et NeuroCognition, Bat. Sciences de l'Homme et Mathématiques, Université Pierre Mendès-France, BP 47, 38040 Grenoble Cedex 9, France
e-mail: Sophie.Portrat@upmf-grenoble.fr

Introduction

Working memory (WM) is a cognitive system devoted to the short-term maintenance of information while concurrent processing activity is simultaneously performed. According to a contemporary and influential functional conception of WM, items stored in WM (e.g., digits of a phone number) are very likely to be forgotten when another task is required (e.g., searching for a piece of paper). Effective memorization of these items requires them to be refreshed when attention is not directed at anything else. This functional conception is presented in the TBRS model of WM (TBRS for time-based resource-sharing model [4, 5]), and in its computational implementation, TBRS* [30], that makes it possible to study the evolution of the activation of the items to be memorized. The goal of the present paper is to specify the functional characteristic of the attentional refreshing mechanism at hand in WM. To this end, we present a challenge to the TBRS* model and suggest one way to rescue it.

An analytic study of this computational implementation of the TBRS model is presented in order to investigate the time course of the attentional refreshing mechanism involved in human WM. The ground truth, with which the model performance is compared, is a set of data collected from participants performing a WM task. It is worth noting from now that we are concerned with refreshing, not rehearsal. Indeed, as we will show, the characteristics of this refreshing mechanism departed from the well-known and well-documented phonological loop proposed in Baddeley's multi-component model of WM [2]. The analytic study as well as the failure of the model to fit the data led us to propose a new mechanism of refreshing.

TBRS aims at describing the functioning of WM and the time course of the cognitive mechanisms at hand when

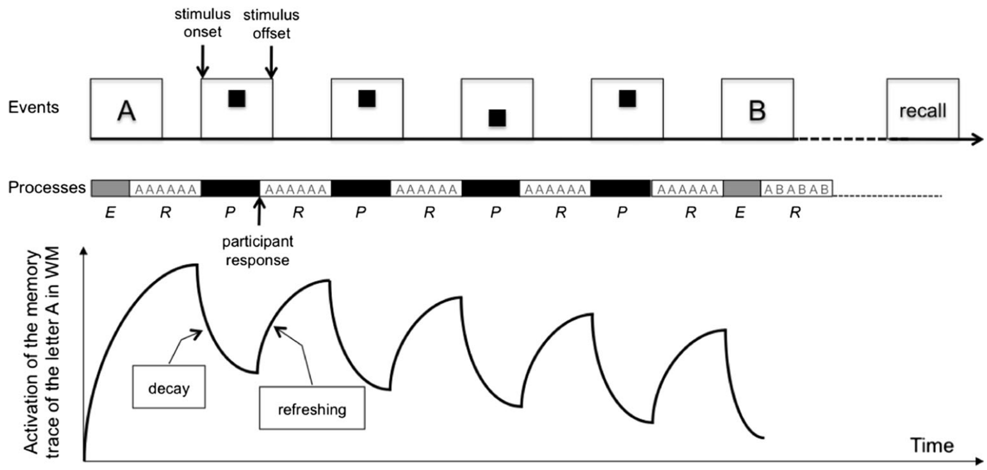


Fig. 1 *Top panel*, "Events," presents the time line of the events occurring at the beginning of a complex span task, the *letters A* and *B* are the first two letters that have to be memorized for further recall and the *four black squares* constitute the stimuli of the concurrent processing task that intervenes after each letter. The processing task illustrated here is the one used in the experiment presented later in the paper (cf. Sect. 2. Experiment), which consisted of a location judgment task: Participants had to press keys as fast as possible without sacrificing accuracy as a function of the position of the *square* on screen (e.g., right key when square is on the upper side of screen). The *middle panel shows* the time course of the WM processes postulated by TBRS. *Gray*, *white* and *black bars* represent encoding (*E*), refreshing (*R*) and processing (*P*) times, respectively. It is worth noting that the duration of the processing process (*P*) does not coincide with the duration of the presentation of the stimulus on screen. Indeed, participants take about 390 ms on average to respond that a square is in the *upper* or the *lower* part of the screen while the squares actually stay on screen for 520 ms. The *bottom panel* presents the supposed evolution of the level of activation of the memory traces of a letter through the task. When the participant is engaged in the processing activity, memory trace decays while it benefits from refreshing, enhancing its activation, during free pauses occurring in between successive processing

participants are performing WM complex span tasks. These consist of dual tasks in which participants have to memorize items while concurrently performing a distracting activity. Participants are thus presented with memory items for further recall (e.g., letters, words), each item being followed by a concurrent processing task, the nature of which is variable (e.g., reading, spatial judgment, parity judgment). Basically, the TBRS model supposes that after being encoded, memory traces of the items to be maintained are preserved vivid through attentional focusing but fade away as soon as they leave the focus of attention. This is exactly the case when a concurrent processing activity occurs because of a central bottleneck [33] that allows only one elementary cognitive step to take place at a time. Hence, during intervening activities, memory traces suffer from a time-related decay but they can be refreshed through attentional focusing during short pauses that would be freed while concurrent processing is running. According to this functional conception, within a complex span task (Fig. 1), WM performance depends on the cognitive load of the processing task that corresponds to the proportion of time during which controlled attention is captured by the concurrent activity and hence unavailable for refreshing memory traces. In other words, the more the distracting task captures attention, the less the memory traces can be refreshed and hence the poorer the recall performances are. Several experimental pieces of evidence have been collected in favor of this theoretical model of WM via behavioral studies in adults and children (e.g., [4, 5, 21, 34, 38]).

TBRS*, the computational implementation of TBRS, is fed with behavioral data collected from complex span tasks. When designing TBRS*, Oberauer and Lewandowsky [30] had to make specific representational choices that were not described in the verbal theory. They chose a two-layer connectionist network. One layer is composed of nodes representing the items to be memorized and the other layer encodes the sequential position of items. Each position is coded by a subset of position units and two adjacent positions share common units with a proportion (P). Memorizing is modeled as a process of connecting positions with items, by Hebbian learning. The strength of the increase of a connection weight (w) depends on a strength parameter (η), and it is bound by an asymptote L.

$$\Delta w = (L - w)\eta \tag{1}$$

The strength η depends on the time t devoted to encoding as well as a stochastic parameter r modeling human variability:

$$\eta = 1 - e^{-rt} \tag{2}$$

Table 1 Overview of (1) TBRS* free parameters and their default values, their values used in the parameter estimation and the values used in our models, and (2) the temporal characteristics of the WM task used as ground truth benchmark [5]

	TBRS* (original)	Parameter estimation		Our models	
		Lower values	Higher values	Tr = .01	AFS4
(1) Free parameters of the models					
Position marker overlap (P)	.3	.15	.45	.3	.3
Criterion for encoding (τ_E)	.95 (.5 in Sim. 4 with children)	.92	.98	.95	.95
Processing rate (R)	6 (5 in Sim. 2)	4	8	6	6
SD of processing rate (s)	1	.5	1.5	1	1
Decay rate (D)	.5 (.31 and .35 in Sim. 5 and 6 with a Brown Peterson paradigm, .4 in Sim. 4 with children)	.35	.65	.5	.5
Refreshing duration (Tr, in s)	.08	.01, .02, .04, .07	1	.01	.08
Threshold for retrieval (θ)	.1 (.05 in Sim. 6)	.05	.15	.1	.1
Noise (σ)	.02 (.05 in Sim. 7)	.005	.035	.02	.02
Attentional focus size	1	N/A	N/A	1	4

	Pace		
	Fast	Medium	Slow
(2) Temporal characteristics of the WM task			
Mean reaction time (Ta in ms)			
4 squares	393	388	400
8 squares	396	389	395
Time (ms) allowed for each processing step			
4 squares	790	990	1,190
8 squares	790	990	1,190

with $r = N(R, \sigma^2)$. Usually, $R = 6$, $\sigma = 1$ and $t = 500$ ms [30]. Table 1 of the present document sums up these default values.

For instance, if the sequence to be memorized is *ABCD*, *A* is first encoded which results in strengthening the links between item *A* and the nodes coding for position 1.

When attention is captured by another task, those link values w decrease according to an exponential function:

$$w(t) = w_0(t)\, \mathrm{e}^{-Dt} \tag{3}$$

D is usually set at 1/2 (see Table 1).

When attention is redirected toward the memory task, a refreshing process takes place and leads to an increase of the w values. Each atomic refreshing follows Eq. 2 but with a much shorter time t, usually 80 ms, compared with the initial encoding (500 ms). Following a classical loop conception, all items are successively considered, starting with the first one and this process cycles until a new activity requires attention.

To pursue our example, when *B* is encoded, activation values between the node representing *B* and the nodes representing positions 2 are strengthened (while in the meantime, the activation values of *A* are decreased). If there is time for refreshing, it is alternately done between *A* and *B* (see the last block of the processes panel of Fig. 1 in which *A* and *B* are refreshed in turns).

When it is time to recall items, the one most connected with the position at hand is recalled. To avoid recalling that item later, all its connections to the position layer are decreased by Hebbian anti-learning.

Methods

In order to increase our understanding of the fine-grained WM mechanisms and especially of this attentional refreshing mechanism, we were interested in the ability of the model to reproduce the serial position curves (SPC). SPC are the curves which display memory accuracy as a function of the serial position of the item in a list to be recalled and, hence, provide much more information than the raw recall performance. That issue has been investigated by Oberauer and Lewandowsky [30, pp. 28–29]. However, the ground truth data underlying their comparison between behavioral and computational performances were not properly controlled for time. Indeed, they reported

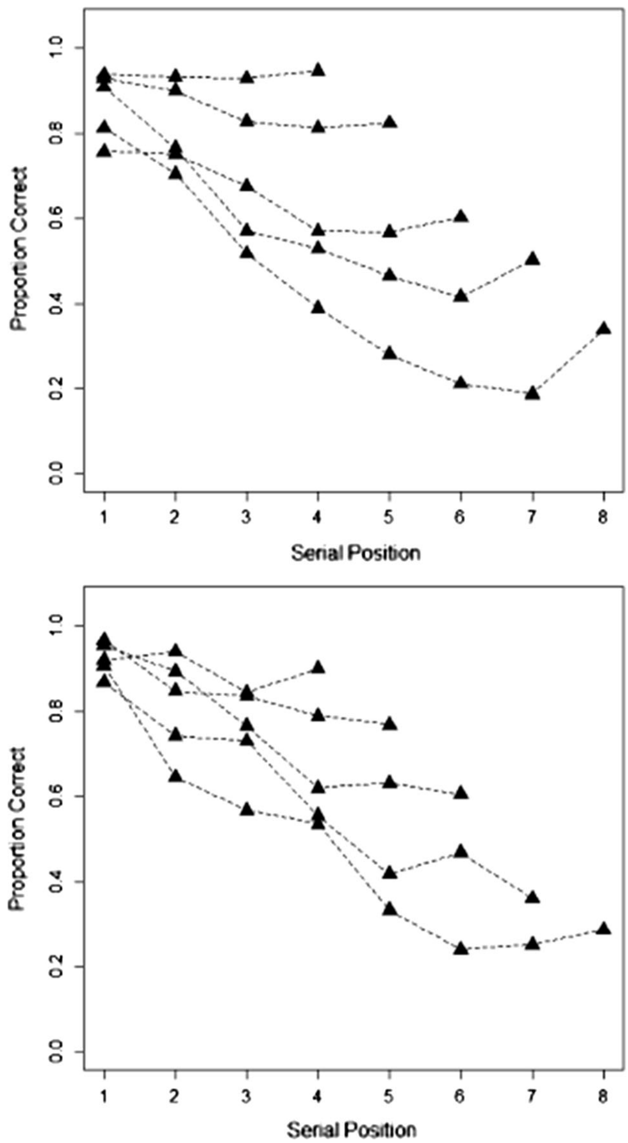


Fig. 2 Serial position curves for an operation span task (*top panel*) and a sentence span task (*bottom panel*). Participants had to memorize various list lengths of consonants while verifying a single operation or sentence (true or false indicated by key-pressing) after each consonant. In each graph, the *different curves* refer to different lengths of memory list (from 4 to 8 letters to be maintained). Adapted from Oberauer and Lewandowsky [30, Fig. 12]

an experiment involving two complex span tasks that were run in their own laboratory: In their reading span and operation span tasks, participants had to memorize series of 4–8 letters while verifying, by pressing keys, either a sentence (e.g., "A hammer is a tool"—[true]) or an arithmetic operation (e.g., $3 + 2 = 5$—[true]), respectively (Lewandowsky et al. [27], Exp. 2, the SPC of these tasks are presented in Fig. 2).

Even if the complex span tasks were computer-paced, as acknowledged by the authors themselves, the time courses of the various operations intervening in between memory items were not controlled. However, the particularity of the paradigm used to test the TBRS model, and thus underlying TBRS*, is that the intervening task is divided into individual steps, the duration of which is controlled and measured. It follows that it is possible to know how much time participants effectively spent on the task and how much time they spent on refreshing, critical controls that are not possible in the aforementioned reading and operation span tasks. Moreover and as a consequence, neither the free time nor the number of operations nor the duration of each operation was varied, which is a pity because, on the one hand, they are crucial factors within the TBRS model, and, on the other hand, it is important to test models on a variety of situations.

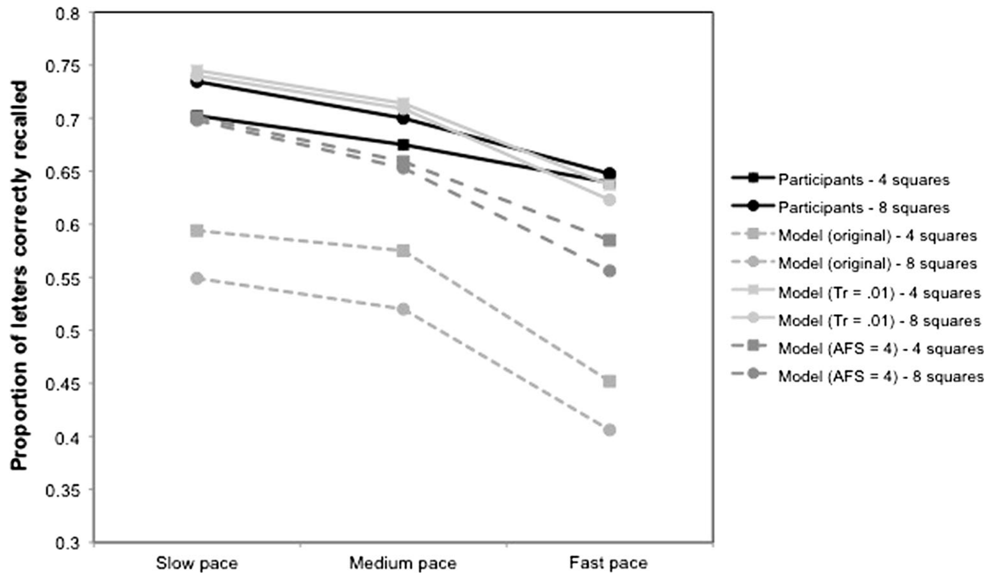
Despite that, the authors claimed that there is qualitative agreement between the model and the participants' performances which both show, for list length 7, extensive primacy and limited recency effects. It is worth noting from now that the claimed classical SPC effects produced by TBRS* were actually neither strictly nor systematically observed in the behavioral data presented by Oberauer and Lewandowsky [30, p. 29]. For example, the bottom panel of Fig. 2 does not show any recency effect for any list lengths, except for 8-item lists for which the recency effect is very limited. In the next section, the ground truth data to which the model predictions will be compared are presented. Then, an analytic study of the computational implementation of the TBRS model will allow us to improve TBRS* by proposing a new mechanism of refreshing.

Ground Truth Data: Behavioral Experiment

To test TBRS* thoroughly and eliminate the aforementioned problem of the behavioral task used for comparison, we analyzed the SPC of a published experiment in which all the temporal parameters were carefully controlled, including for each step of the intervening task [5, pp. 1309–1313].

This experiment involved a complex span task paradigm similar to the one presented above (see Fig. 1). Thirty participants in the experiment were presented with lists composed of seven memory items (letters) for further recall. Each memory item was followed by a concurrent processing activity consisting of a location judgment task. For this task, a square appeared either in the upper or in the lower part of the computer screen and participants had to indicate, by a binary key press, which part contained the square. Either four or eight location judgment tasks intervened after each letter following a fast, medium or slow pace (790, 990 or 1,190 ms per square, respectively). Participants had to read aloud each letter to be maintained and press the keys corresponding to the position of the

Fig. 3 Proportion of letters correctly recalled for the three pace conditions (*slow*, *medium*, *fast*) and the two numbers of operations within the processing tasks (4 or 8). *Black lines* represent human participants' performance. *Gray lines* represent the model performance with the default parameters (Oberauer and Lewandowsky [30]; Model [original]), when the duration of the atomic refreshing step is set at 10 ms (Model [Tr = .01]) and when the model disposes of a flexible focus of attention whose size is set at 4 (Model [AFS = 4])



squares (e.g., the right key if the square appeared in the upper part of the screen) as fast as possible without sacrificing accuracy. Finally, they were asked to recall letters in forward order by typing letters one by one on the computer keyboard, a method that makes it possible to be confident about the shape of the SPC. In addition to the recall performance that was computed as the percentage of letters recalled in correct position, the mean response time (in ms) as well as the mean accuracy on the location judgment task was recorded. Overall, results were in line with the TBRS predictions. First, the typical pace effect was observed with faster pace resulting in lower recall performance. However, neither the effect of the number of squares nor the interaction between the two factors was significant (Fig. 3, solid black lines).

The solid black lines of Fig. 4 present the unpublished SPC of this experiment that constitutes the benchmark ground truth data for the following analysis.[1] At first glance, the shapes of the SPC are characteristic of the primacy and recency effects usually observed in short-term memory paradigm [29]. The *U* shapes of the curves are well explained by the TBRS conception of WM functioning. On the one hand, given that the first encoded items benefit from many more refreshing possibilities, their memory traces are activated much more when recall is required. On the other hand, the last items of the list, being recently encoded, do not suffer much from decay and hence could be easily retrieved at recall. However, the next simulation and our analytic demonstration will show that having been recently encoded is not sufficient for the last items to show the recency effect: Minimal refreshing is required.

[1] Because the number of squares condition (4 and 8 squares) did not have any effect on recall performances, the three SPC represent data for the three pace conditions (slow, medium, fast) irrespective of the number of squares.

Computational Investigations: Simulation

TBRS*, whose core functions are described above, is a non-deterministic model, with many random choices throughout all the simulation. Therefore, any simulation has to be run several times in order to obtain an average result. We ran many times the MATLAB code accompanying Oberauer and Lewandowsky's paper and found that the number of runs they suggest, 600 (200 simulated subjects $\times$ 3 runs each), is probably too low. Indeed, when their simulations are launched several times, using 600 runs, results vary. To illustrate this point, the left curves of Fig. 5 display one original simulation with 600 runs (Oberauer and Lewandowsky [30], simulation 1, p. 22, with 4 processing steps after each memory item, a value of 0.3 s for the duration of each processing step and a free time following each processing step set at 0.6 s), as well as two other simulations also with 600 runs but with different initial seeds of the random number generator, all selected randomly. It is obvious that the curves are quite different, especially on the bottom side of the *U* shape. Oberauer and Lewandowsky were therefore lucky to obtain such a nice curve. The right panel of Fig. 5 shows the three identical simulations, with the same seeds of the random generator, but with 6,000 runs each. Curves are now similar. The initial value of the seed therefore does not matter as long as there are enough runs. All our simulations are thus based on 6,000 runs, except for parameter estimation which tolerates rougher results.

We therefore simulated our experiment, with the default parameters indicated by Oberauer and Lewandowsky [30]

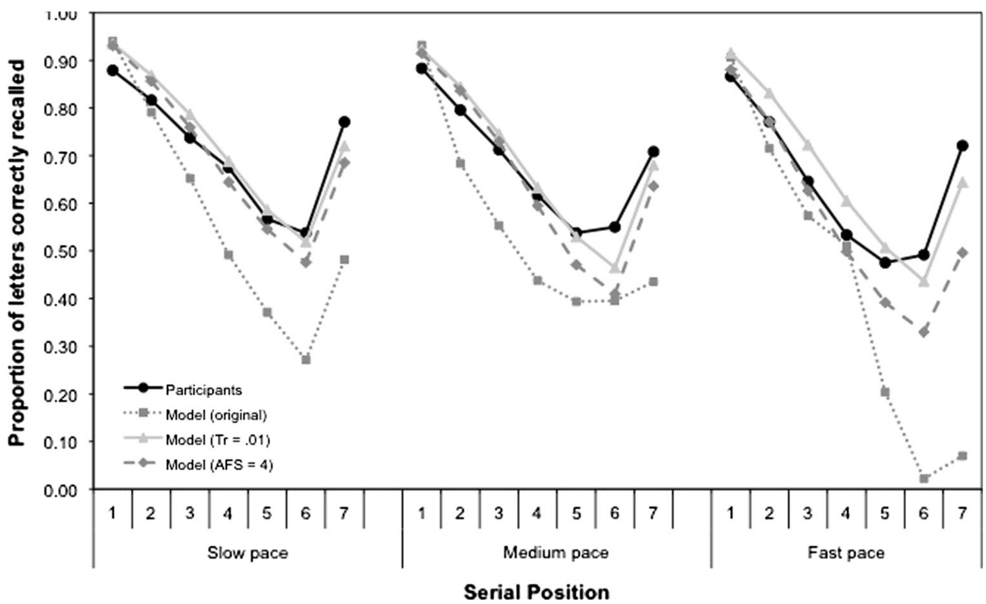
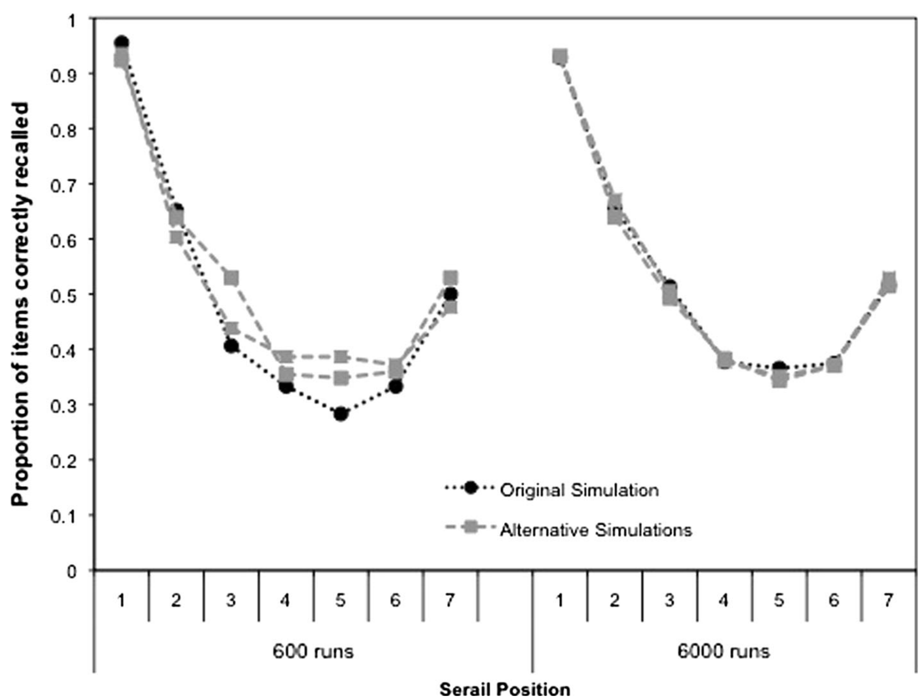


Fig. 4 Serial position curves produced by the model and the participants for the three pace conditions (*slow*, *medium*, *fast*). *Black lines* represent human participants' performance. *Gray lines* represent the model performance with the default parameters (Oberauer and Lewandowsky [30]; Model [original]), when the duration of the atomic refreshing step is set at 10 ms (Model [Tr = .01]) and when the model disposes of a flexible focus of attention whose size is set at 4 (Model [AFS = 4])

Fig. 5 *Left curves* serial position curves based on 600 runs with three different random seeds (7 items, 4 operations, mean operation duration = 0.5 s, free time = 0.6 s, default values for all other parameters). *Right curves* serial position curves based on identical simulations but with 6,000 runs instead



and with the temporal parameters of the task presented in Barrouillet et al. [5, Table 1, p. 1312]. All the parameter values are presented in Table 1 of the present document. While the critical pace effect seems to be replicated by the model, the main problem is that the model performs much worse than participants (Fig. 3, Model [original] data, light gray dotted lines). Although a quantitative evaluation of TBRS* is thought to be premature by its authors [30, p. 40], a scrutiny of the SPC reveals precisely the qualitative difficulty encountered by TBRS* in predicting strictly controlled behavioral performances.

The low performance of the model obviously appears through the SPC presented in Fig. 4 (Model [original] data, light gray dotted lines). While the *U* shape of the SPC seems generally reproduced, the model encountered difficulties for the most time-constrained condition (fast pace) for which it dramatically failed in recalling the last items of the lists. As we will show, the analysis of the SPC predicted by the original TBRS* model compared to the SPC observed in participants' behavior made it possible to delineate the possible reasons for such a discrepancy.

First of all, there might be other parameter values that would not lead to a discrepancy between model and experimental data in the fast pace condition. To check that possibility, we performed a systematic search in the TBRS* parameter space and computed the root mean square error (RMSE) between experimental SPC and the model SPC [25] on all conditions (6 conditions × 7 positions = 42 data on both sides). TBRS* is based on several parameters but some of them are correlated. For instance, the mean encoding rate R and the refreshing duration Tr both operate on the same lever through a criterion for implementing the accumulator model of response time: $\tau_R = 1 - e^{-R \cdot \mathrm{Tr}}$. However, we still performed a systematic search considering three values for each parameter: the default value, a smaller and a higher value, except for Tr, the duration of a single refreshing operation, for which we considered five values because this parameter plays a major role in the refreshing mechanism. The parameter values that were considered are presented in Table 1.

This gave $5 \times 3^7 = 10{,}935$ RMSE computations, each one based on 600 runs for each of the 6 conditions, which means about 40 million runs. We sorted the results and found some parameter combinations which provide a good fit to the experimental data. However, they were based on low values of Tr, the refreshing duration parameter. Most of them corresponded to a value of 10 ms to refresh a single item, which is not cognitively plausible. There were also good parameter combinations with Tr values of 40 ms, but with other values than the default values for some parameters: a low decay rate (.35), a high processing rate (8) or a high standard deviation of the processing rate (1.5). No good parameter combinations were obtained with Tr values higher than 40 ms. All these changes tend to maintain the last items vivid in the fast pace condition. However, these solutions are not satisfactory because they introduce big changes in the default values which showed a remarkable stability in the eight simulations performed by Oberauer and Lewandowsky [30]. It would be better to solve that issue by working on a single parameter.

One reason why the original model produced a performance that did not fit could be that the sequential refreshing process as implemented by Oberauer and Lewandowsky [30] within TBRS* creates important differences between the first items and the last items in the refreshing sequences, especially for substantial temporal constraints. The key point is that, following a loop conception akin to the phonological loop mechanism [2], items are refreshed successively, starting with the first one, until

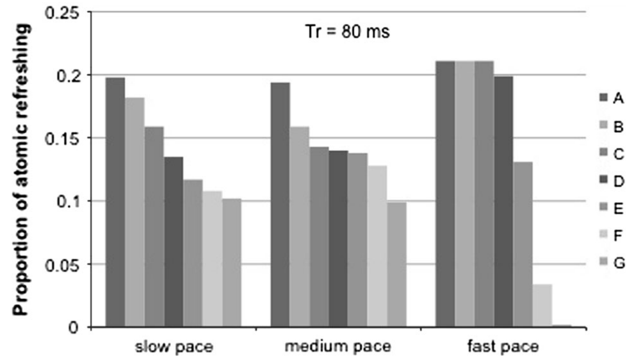


Fig. 6 Proportion of atomic refreshing for each item of a seven-item list (*A*, *B*, *C*, *D*, *E*, *F* and *G*) computed during a refreshing period occurring after the 7th item for the *slow*, *medium* and *fast* pace conditions

there is no time left. For example, a refreshing sequence could be *ABCABCA* if there are three items and only seven refreshing operations possible within the available time. It is obvious that, with this mechanism, first items are always refreshed equally or more than the last items (*A* is refreshed three time, whereas *C* is refreshed twice in our example). This might explain why the bad fit concerns mainly the last items especially when time constraints are important, i.e., when the time available for refreshing is short (Fig. 4, fast pace). In fact, when there is time for many refreshing turns, the amount of atomic refreshing that could benefit the first and the last items tends to be equal, whereas, when the free time is highly limited, the last items could not even be refreshed at all. We studied in more detail this particular mechanism.

Let Ta be the amount of time that is available for refreshing after each processing step. If a single refreshing step takes Tr seconds, the number of refreshing steps that can occur after each processing step is Ta/Tr. For instance, in the fast pace condition, Ta = 390 ms. With Tr = 80 ms, it means that 4.8 refreshing steps can occur. If the memorandum is composed of three letters at that time (e.g., *ABC*), *A* will be refreshed twice and *C* will be refreshed only once.[2]

Let us now compute the proportion of refreshing of a given item across all items to be maintained. Suppose we have only two items to refresh, *A* and *B*. The proportion of refreshing steps devoted to the first item depends on the moment at which refreshing is stopped. It is 100 % if there is time for only one item to be refreshed (*A*), it is 1/2 if two items can be refreshed (*AB*), it is 2/3 if three items can be

[2] For sake of clarity, it is worth noting that in the present case, *B* would have been refreshed once plus once more during a fraction of time of about 70 ms (corresponding to the remaining time available after having refreshed the *ABCA* sequence).

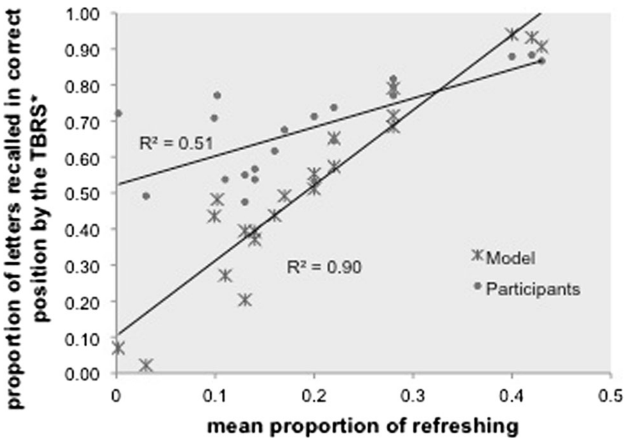


Fig. 7 Proportion of letters recalled in correct order by the model and the participants regressed against the estimated mean proportion of atomic refreshing

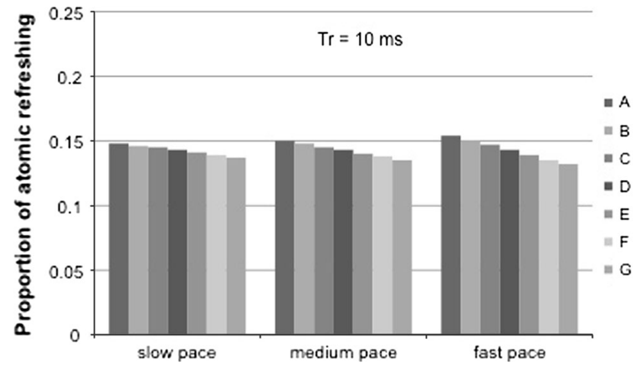


Fig. 8 Proportion of atomic refreshing for each item of a seven-item list (*A*, *B*, *C*, *D*, *E*, *F* and *G*) computed during a refreshing period occurring after the 7th item for the *slow*, *medium* and *fast* pace conditions with an atomic refreshing step duration (Tr) of 10 ms

refreshed (*ABA*), it is then 2/4, 3/5, 3/6, etc. If n items could be considered given the current free time, that value is the number of times *A* is refreshed divided by *n*. Let us call $r_2^i(n)$ the proportion of refreshing of an item at position *i* in a sequence of length *s*, depending on n, the number of atomic refreshing steps. If the sequence contains two items, the proportion of refreshing of the first item is:

$$r_2^1(n) = \frac{\left\lfloor \frac{n+1}{2} \right\rfloor}{n}$$

which generates the sequence 1/1, 1/2, 2/3, 2/4, 3/5…

For the second item, that value is similar except that the numerator is shifted:

$$r_2^2(n) = \frac{\left\lfloor \frac{n}{2} \right\rfloor}{n}$$

which generates the sequence 0/1, 1/2, 1/3, 2/4, 2/5…

The general formula for two items is therefore:

$$r_2^i(n) = \frac{\left\lfloor \frac{n+2-i}{2} \right\rfloor}{n}$$

where *i* is the position of the item in the sequence.

This equation can be applied to more than two items. Given a sequence of size *s* to refresh, the proportion of refreshing of an item *i* according to the number of atomic refreshing steps *n* is:

$$r_s^i(n) = \frac{\left\lfloor \frac{s+n-i}{s} \right\rfloor}{n}$$

The number of items that can be refreshed within a given refreshing period is actually not a fixed value, because the time it takes to refresh a single item, Tr, is a stochastic variable which follows a normal distribution with mean *R* and standard deviation σ, as indicated previously (Eq. 1).

Given the total duration of a refreshing period, we therefore computed the probability that an item is refreshed according to its position in the list, by multiplying the proportion of refreshing r by the probability of stopping the process, for each position of refreshing. We performed those computations for each of the three pace conditions. Figure 6 shows the results for a 7-item-long sequence in which the proportion of atomic refreshing for each item of the list is computed during a refreshing period after the 7th item. As expected, there is a strong relationship between the position in the sequence and the probability of being refreshed. In addition, the fast condition appears much different from the other two: The last items are refreshed much less. The last one is hardly refreshed at all.

Hence, the proportion of atomic refreshing steps devoted to a particular item to be maintained across all the list seems to be a crucial determinant of the TBRS* performances. The recall performance produced by the computational model and observed on the experimental data was regressed on these proportions of atomic refreshing values. As shown in Fig. 7, 95 % of the variance observed in the serial position memory performance of the computational model TBRS* can be explained by the variation of the proportion of refreshing of the letters that varied as a function of their position in the list. This does not correspond to what is observed for humans. First of all, it is worth noting that the R^2 value falls to .51. Moreover and most importantly, while the last item would never have been refreshed in the fast pace condition with such a TBRS*-like refreshing mechanism, it is, however, recalled 72 % of the time by the participants.

This analytic study clearly shows that a sequential refreshing mechanism in which item #1 is always refreshed first introduces a strong bias in favor of the first items. This is the reason for the poor fit to our experimental data which does not show a catastrophic degree of forgetting of the last items. That analytic study enlightens the results of the parameter estimation: A low value of Tr gives good predictions because if a single refreshing step takes less time, more items would be refreshed in a restricted period of time and the last items would be more likely to be refreshed.

Individual Refreshing Takes Less Time

Therefore, our first approach would be to consider that atomic refreshing might take less time than what was initially proposed by Oberauer and Lewandowsky [30]. Figure 8 shows the new proportion of atomic refreshing with Tr = 10 ms instead of 80 ms. First, the distribution appears much more uniform across list positions. Most importantly, the difference in the shape of the distribution across the pace conditions that we alleged to be responsible for the SPC that did not fit between the model and the participants at fast pace no longer exists. The last items in the fast condition are now refreshed much more often.

When TBRS* is run with a refreshing duration of 10 ms, its performance gives a very good fit, reproducing the pace effect (Fig. 3, Model [Tr = .01] data, solid gray lines) and providing quasi-perfect SPC whatever the pace condition (Fig. 4, Model [Tr = .01] data, solid gray lines). The problem is that a value of 10 ms is not psychologically plausible. We should therefore turn to another way to improve the model.

Attentional Focus Size is Larger

One other, more plausible, way to minimize the discrepancy between the behavioral and the computational SPC is to consider that more than one item is refreshed at the same time. This point of view is not part of the original TBRS verbal theory but is not at odds with its conception of WM functioning. Indeed, following Pashler [33], TBRS postulates that a central bottleneck allows only one controlled elementary process to take place at a time (e.g., [5]). However, this does not mean that each single elementary process cannot handle several pieces of information at the same time. This idea is described by Cowan and collaborators who proposed that the focus of attention is flexible and can be zoomed-into concentrate on a single particular item (especially in the case of a task involving interference) or zoomed-out to manage with a maximum of about 4 items (e.g., [14, 16]).

We therefore implemented this idea: Instead of a purely sequential refreshing mechanism handling one item at a time during an average duration of Tr ms, up to four items can be refreshed at the same time. The duration of each refreshing step is still set at Tr ms but the strength of the links between the items and their position units (parameter η in Eq. 2 in [30, p. 16]) is therefore divided by the number of items simultaneously handled, since the attention involved in such controlled processes is a limited resource[3] (e.g., [4, 19]).

This mechanism should tend to minimize the difference in the amount of refreshing operations over items. Last items are less penalized because they have more chance to be inside the attentional focus. The implementation of this new refreshing mechanism within TBRS*, with its default parameters (including Tr = 80 ms), produced better results with an attentional focus size set to four (Figs. 3, 4, Model [AFS4], gray dotted lines). When compared to the experimental data, the RMSE is .0127 which is the best of all models based on a single parameter change, except when the refreshing duration is set to lower values, which leads to a similar RMSE.[4] In particular, no models were better than ours using the standard refreshing duration of 80 ms, as defined by Oberauer and Lewandowsky [30]. These results suggested that when refreshing is possible (i.e., when attention is not captured by the concurrent processing activity), several item-position links could be simultaneously strengthened. Let's say, for instance, that at the beginning of a list, only two items have been encoded. The refreshing process is then applied to these two items simultaneously during Tr ms with η divided by two. Now, let us move on toward the end of the memory list and consider that six items have already been encoded. For instance, if *ABCDEF* have been encoded and have to be refreshed and the attentional focus has a size of four, *ABCD* are first refreshed at the same time with a strength η divided by four during Tr ms, then *EFAB*, then *CDEF* again and so on until there is no time left for refreshing (we will discuss later this manner of implementing the sequential refreshing window). Hence, following Cowan [13], the focus of attention is thought to be a limited-capacity resource that can be flexibly allocated from one to four elements as a function of the external constraints. Table 1 sums up the difference between our model and the original TBRS* model.

[3] Refreshing in TBRS* is based on Hebbian association learning. With this mechanism, it is not possible to associate n items with n positions simultaneously so that the first item is associated with the first position, the second item to the second position, and so on. So, even if the refreshing is conceptually simultaneous across multiple item-position associations, at a computational level, refreshing still occurs sequentially, one item-position association at a time, but in such a way that four subsequent item-position associations are grouped together, and assigned a single duration Tr.

[4] Other simulations showed that an improvement over the original model has already appeared with an attentional focus size of two and is increasing slightly with a size of three, then four.

Discussion

The present study evaluates for the first time a recent computational model of WM, TBRS* [30], about its abilities to produce SPC and proposes that the time course of the attentional refreshing mechanism would not be as simple as initially implemented. The SPC of a published behavioral study that were precisely time-controlled [5] were analyzed to propose a proper qualitative evaluation of the performance of TBRS*. The comparison between the model and the participants' behavior showed that TBRS* in its first instantiation encounters difficulties with a scrutiny of the SPC produced by participants in this strictly controlled task. Indeed, we did not find any combination of parameters that would make it possible to replicate the experimental data, with a limited number of changes compared with the values obtained by Oberauer and Lewandowsky in several simulations. Through an analytic approach of the computational model, we suggested that the temporal and sequential characteristics of the refreshing mechanism implemented in TBRS* are, at least in part, responsible for this discrepancy and proposed alternatives. To fit behavioral data, the model needs either to be provided with a time parameter value that does not have any psychological plausibility (10 ms for refreshing a single item) or supplemented by a new characteristic of the focus of attention: its flexibility. No single parameter change in the original TBRS* model provided a better fit than our new model including a flexible focus of attention.

The present work has several practical and theoretical implications for future computational and behavioral research aiming at studying WM functioning. Indeed, it is worth noting that the TBRS model is very influential nowadays in cognitive psychology literature since it proposes a groundbreaking conception of the interplay between maintenance and processing activities in WM [3]. It includes an attentional refreshing mechanism through which the decaying memory traces could be maintained vivid for immediate recall. As we will see below, this refreshing mechanism departs from the well-known phonological loop.

On a practical level, the alternatives proposed here to improve the TBRS* model, which happened to fail on temporally constrained situations, do not require an additional free parameter. It is always a problem to improve models by adding free parameters which do not have real theoretical counterparts. TBRS* already contains several free parameters which cannot be easily justified. For example, the level of noise is set at .02 and the percentage of common units for successive positions is set at 30 %. We therefore paid attention to not adding a free parameter whose value would only depend on fitting our own behavioral data. The size of the adaptive attentional focus is set as flexible from 1 to 4 in reference to another well-documented model of WM proposed by Cowan et al., the embedded-processes model of WM (e.g., [12–16]).

On a theoretical level, our analytic study showed that the sequentiality of the refreshing mechanism is acceptable as long as there is enough time to refresh items, so that each one is refreshed at least once. The longer the free time, the more uniform the refreshing. However, when time constraints become substantial, as in the fast pace condition of the present experimental study, there could be no time to refresh the last items and the performance of the model falls for these items. This catastrophic degree of forgetting of the last items of a list does not occur in human data. Hence, we have shown that any solution has to tend toward a more uniform distribution of the probabilities of refreshing for each item.

First of all, decreasing the duration of atomic refreshing has such an effect. As highlighted by Lewandowsky and Oberauer [30, p. 43], attentional refreshing seems to be a very rapid process. Accordingly, a recent behavioral study evaluating the effect of the amount of memory items on the time needed to perform a concurrent processing task suggested that it takes about 40 or 50 ms to refresh an item within WM [39]. A second solution proposed in the present paper is to consider that the attentional focus is flexible. However, the precise functioning of the implemented multi-size attentional refreshing mechanism is an open question. In the current version of the model, the focus of attention could grasp items four by four. If the number of items to be refreshed is not a multiple of four, the last items are in the same focus of attention as the first ones. For example, if the following sequence "*ABCDEFG*" has to be refreshed, the attention will focus successively on *ABCD*, *EFGA*, *BCDE*, *FGAB*, etc. Another possible implementation could be to avoid grouping last and first items, or to group differently (*ABCD*, *BCDE*, *CDEF*, etc.). It is worth noting that our solution does not make any assumption on the fact that the items within the attentional focus are processed serially or in parallel, which is a recurrent problem in the literature [37]. This refreshing mechanism could also be managed within a different architecture. Instead of representing items by single units, some models represent them in a distributed way. Storing sequences in this kind of architecture is a classical issue, although it is usually considered for modeling long-term memory [23] or initial item encoding in WM [24]. However, it could also be possible to consider this approach for modeling the refreshing iterations. For instance, Snaider and Franklin [35] proposed a new mechanism to handle sequences in an associative memory, by storing elements by pairs (*AB*, *BC*, *CD*, etc.) which interestingly brings up to date the obsolete chaining approach [22] whose limits are avoided by the auto-associative retrieval mechanism.

Another improvement alternative, which could be more in line with Cowan's conception of the attentional flexibility, would be to adapt the attentional focus size to the cognitive constraints imposed by the task. It is possible that humans adapt their focus of attention according to the time they are allotted. For instance, if the pace is slow and there is thus time for refreshing, a size of one could be appropriate. However, under severe temporal constraints, the focus of attention could be extended. We plan to implement and test these ideas in the near future. Indeed, the existence of an adaptive maintenance strategy that depends on the temporal constraints imposed by the task to be performed has been shown in a behavioral immediate serial recall study. In this experiment, participants repeated only the most recently presented item when memory items were presented at fast pace (1 s/word) while they rehearse several items in a row when the temporal pressure was lowered (5 s/word, Tan and Ward [36]). A third solution that would lead to a more uniform distribution of refreshing probability could be to refresh items in a random way. If the choice of the next item to refresh is made randomly, the distribution of probabilities of refreshing would tend toward a uniform distribution. However, a strictly random refreshing mechanism would lead to an over-uniformization of the proportion of atomic refreshing that would probably weaken the well-known primacy effect. A solution could therefore be to mix a sequential refreshing and a random refreshing. Anderson et al. [1] used such a kind of hybrid maintenance mechanism in the context of the ACT-R framework. Within a short-term serial recall task, two processes compete to refresh items, one that is sequential and one that could randomly avoid refreshing a given item. However, the aforementioned study by Tan and Ward [36] has shown that the overt rehearsal strategy, which is the one simulated in the ACT-R framework, used by participants for verbal material is mostly a "full cumulative" forward order strategy consisting in rehearsing in the correct order all the items presented to date. The results also showed that the overt rehearsal strategy varied across the position of the rehearsal set. Indeed, the "full cumulative" rehearsal strategy decreased toward the end of the 6-item list (i.e., after the presentation of the fifth or the sixth item), while "fixed" rehearsal strategy that consists in rehearsing only the most recently presented item increased. The "partial cumulative" strategy (e.g., rehearsing in order only some items) was seldom used, other strategies were marginal and, finally, random overt rehearsal was actually not reported.

Several other memory models exist in the literature are able to produce the SPC observed in human performance. However, none of them are suitable to specify the time course of the attentional refreshing mechanism. For example, the SIMPLE (scale-independent memory, perception and learning) model of Brown et al. [6] proposed that the probability of retrieving an item depends on a logarithmic temporal dimension according to which the recent items, that are much more psychologically distinctive from each other than the previous items of the list, are easily retrieved. It thus predicts the recency effect. However, regardless of the SPC, the temporal distinctiveness notion of the SIMPLE model also predicts that the longer the time separating two items' encoding, the better their recall. The memory performance of the human participants that serve as the ground truth benchmark for the present study does not reflect this effect. Indeed, while the inter-letters interval increased from 3,960 ms on average in the 4-squares condition to 7,920 ms in the 8-squares condition, the effect of the square conditions in the percentage of letters recalled in correct position was not significant (67 vs. 69 % in the 4- and the 8-squares conditions, respectively, $F(1, 29) = 1.97$, $p > .10$, see Barrouillet et al. [5, p. 1312]. Other computational models of memory (e.g., Brown et al. [7]; [20, 26, 32] can reproduce realistic SPC in an immediate serial recall task. However, it is necessary to go further for two reasons. First of all, they have often been studied in the short-term memory tradition that does not take into consideration the interplay between the maintenance mechanisms and the concurrent processing activities. In the present study, this point is crucial since a particularity occurs in the recency effect when the temporal constraints imposed by the concurrent processing activities are important. It is worth noting that there is a major contender to TBRS*, also able to produce SPC and to deal with complex span tasks, namely SOB-CS [31]. Contrary to TBRS, forgetting in WM, in this model, is mainly due to interferences. However, the material of our task was precisely designed to reduce as much as possible interferences between memoranda (verbal material) and distractors (visual material [38]). We do not deny the existence of interference and believe that the future of complex span modeling lies in hybrid models based on both interference and decay. The second reason is that our goal is not to look for the best model able to fit our SPC behavioral data but rather to evaluate TBRS*, as it is an influential contemporary framework that underpins a growing body of behavioral as well as emergent computational studies.

Finally, the present work questions the interplay between the attentional refreshing mechanism studied here and the phonological rehearsal described in Baddeley's model of WM. Within TBRS*, the nature of the items to be maintained does not matter. However, it is worth noting that the SPC presented by Oberauer and Lewandowsky [30], as well as the one we reanalyzed here, imply verbal material. Indeed, as highlighted by Baddeley et al. [3, p. 62], while the complex span can be thought, as in TBRS [5], "[…] to reflect the capacity to prevent the decay of the memory traces through rehearsal, this rehearsal does not necessarily refer to subvocal rehearsal, but simply "keeping in mind" the item by intermittently focusing attention on the fading traces." Evidence for the existence of two systems of maintenance in verbal WM has been recently accumulated [9–11, 28]. The first one is an articulatory rehearsal as described in Baddeley's model (e.g., [2]) and consists of the vocal or subvocal repetition of the item to be maintained within a loop and is therefore a domain-specific system devoted to verbal material. The second one is the refreshing mechanism implemented in the computational TBRS* model, described by Cowan et al. [17] as a covert retrieval, and viewed as a general attention-based system. However, that does not mean that verbal items could only be maintained by the domain-specific phonological loop system. Indeed, a reduction of the available attention by a concurrent non-verbal task (e.g., the location judgment task) reduced recall performance on verbal items (e.g., [5, 28]). Hence, these two mechanisms operating independently on the maintenance of verbal information had been suggested to affect different features of the memoranda that result from various levels of encoding [9]. In the reanalyzed behavioral experiment, the participants most probably used the two systems to rehearse the letters to be maintained while performing the non-verbal key-pressing concurrent task (see also Vergauwe et al. [39]). It is nonetheless worth noting that, despite an overall superiority of recall performance for verbal information (letters or digits) as opposed to visual (locations of dots), the classical *U* shape SPC is observed for both materials [18]. Given that the current version of TBRS* does not implement any specific verbal rehearsal mechanism that would reflect the phonological loop, the present results have to be tempered. As a consequence, the next step in our investigation of the qualitative performance of TBRS* is to simulate visuospatial WM performance that would only be underpinned by an attentional refreshing mechanism. Moreover, it would be of particular interest to add to a TBRS-like architecture an implementation of the phonological loop that would be inspired by the Burgess and Hitch [8] model of the articulatory loop or by the Primacy Model proposed by Page and Norris [32].

Conclusion

In summary, we reported a study that qualitatively challenges the computational implementation of the time-based resource-sharing model of WM, TBRS* [30]. Using the behavioral SPC of a strictly controlled study as the benchmark ground truth for comparison, we suggested, through an analytic approach, that the temporal and the sequential characteristics of the implemented attentional refreshing mechanism are questionable. An alternative that had already been proposed in the verbal TBRS [5, p. 178] receives here computational support: Supplement the model with a flexible focus of attention that can handle up to four items at a time. Although further investigations are obviously needed, the present results provide promising perspective to better specify the characteristics of this domain-general attentional maintenance mechanism involved in WM that really seem to depart from the well-known domain-specific phonological loop proposed in Baddeley's multi-component model of WM [2].

Acknowledgments We thank anonymous reviewers of a previous version of this paper whose comments helped us to improve the manuscript, as well as the "Région Rhône-Alpes" for its financial support.

References

1. Anderson JR, Bothell D, Lebiere C, Matessa M. An integrated theory of list memory. J Mem Lang. 1998;38:341–80.
2. Baddeley AD. Working memory. Oxford: Clarendon Press; 1986.
3. Baddeley AD, Eysenck M, Anderson MC. Memory. Hove: Psychology Press; 2009.
4. Barrouillet P, Bernardin S, Portrat S, Vergauwe E, Camos V. Time and cognitive load in working memory. J Exp Psychol Learn Mem Cogn. 2007;33:570–85.
5. Barrouillet P, Portrat S, Camos V. On the law relating processing to storage in working memory. Psychol Rev. 2011;118:175–92.
6. Brown GDA, Neath I, Chater N. A temporal ratio model of memory. Psychol Rev. 2007;114:539–76.
7. Brown GDA, Preece T, Hulme C. Oscillator-based memory for serial order. Psychol Rev. 2000;107(1):127–81.
8. Burgess N, Hitch GJ. Towards a network model of the articulatory loop. J Mem Lang. 1992;31:429–60.
9. Camos V, Lagner P, Barrouillet P. Two maintenance mechanisms of verbal information in working memory. J Mem Lang. 2009;61:457–69.
10. Camos V, Mora G, Barrouillet P. Phonological similarity effect in complex span task. Q J Exp Psychol. 2013;66:1927–50.
11. Camos V, Mora G, Oberauer K. Adaptive choice between articulatory rehearsal and attentional refreshing in verbal working memory. Mem Cogn. 2011;39:231–44.
12. Chen Z. Attentional focus, processing load, and stroop interference. Percept Psychophys. 2003;65:888–900.
13. Cowan N. An embedded-process model of working memory. In: Miyake A, Shah P, editors. Models of working memory: mechanisms of active maintenance and executive control. Cambridge: Cambridge University Press; 1999. p. 62–101.
14. Cowan N. Working memory capacity. Hove: Psychology Press; 2005.
15. Cowan N, Elliott EM, Saults JS, Morey CC, Mattox S, Hismjatullina A, Conway ARA. On the capacity of attention: its estimation and its role in working memory and cognitive aptitudes. Cogn Psychol. 2005;51:42–100.
16. Cowan N, Morey CC, Chen Z, Bunting M. What does estimates of working memory capacity tell us? In: Osaka N, Logie R, D'Esposito M, editors. The cognitive neuroscience of working memory. Oxford: Oxford University Press; 2007. p. 43–58.
17. Cowan N, Saults JS, Elliott EM. The search for what is fundamental in the development of working memory. Adv Child Dev Behav. 2002;29:1–49.
18. Davis LC, Rane S, Hiscock M. Serial recall of visuospatial and verbal information with and without material-specific

interference: implications for contemporary models of working memory. Memory. 2013;21:778–97.
19. Engle RW, Kane MJ, Tuholski SW. Individual differences in working memory capacity and what they tell us about controlled attention, general fluid intelligence, and functions of the prefrontal cortex. In: Miyake A, Shah P, editors. Models of working memory: mechanisms of active maintenance and executive control. Cambridge: Cambridge University Press; 1999. p. 102–34.
20. Farrell S, Lewandowsky S. An endogenous model of ordering in serial recall. Psychon Bull Rev. 2002;9:59–79.
21. Gaillard V, Barrouillet P, Jarrold C, Camos V. Developmental differences in working memory: Where do they come from? J Exp Child Psychol. 2011;110:469–79.
22. Henson RNA, Norris DG, Page MPA, Baddeley AD. Unchained memory: error patterns rule out chaining models of immediate serial recall. Q J Exp Psychol. 1996;49A:80–115.
23. Kanerva P. Sparse distributed memory. Cambridge: The MIT Press; 1988.
24. Lewandowsky S, Farrell S. Computational models of working memory. In: Nadel L, Chalmers D, Culicover P, Goldstone R, French B, editors. Encyclopedia of cognitive science. London: Macmillan; 2002. p. 578–83.
25. Lewandowsky S, Farrell S. Computational modeling in cognition: principles and practice. Thousand Oaks, CA: Sage publication; 2010.
26. Lewandowsky S, Murdock BB. Memory for serial order. Psychol Rev. 1989;96:25–57.
27. Lewandowsky S, Oberauer K, Yang LX, Ecker UKH. A working memory test battery for MATLAB. Behav Res Methods. 2010;42:571–85.
28. Mora G, Camos V. Two Systems of Maintenance in Verbal Working Memory: evidence from the Word Length Effect. PLoS ONE. 2013;8:8.
29. Murdock BB Jr. Effect of a subsidiary task on short-term memory. Br J Psychol. 1962;56:413–9.
30. Oberauer K, Lewandowsky S. Modeling working memory: a computational implementation of the time-based resource-sharing theory. Psychon Bull Rev. 2011;18:10–45.
31. Oberauer K, Lewandowsky S, Farrell S, Jarrold C, Greaves M. Modeling working memory: an interference model of complex span. Psychon Bull Rev. 2012;19:779–819.
32. Page MPA, Norris D. The primacy model: a new model of immediate serial recall. Psychol Rev. 1998;105:761–81.
33. Pashler H. The psychology of attention. Cambridge, MA: MIT Press; 1998.
34. Portrat S, Barrouillet P, Camos V. Time-related decay or interference: based forgetting in working memory? J Exp Psychol Learn Mem Cogn. 2008;34(6):1561–4.
35. Snaider J, Franklin S. Extended sparse distributed memory and sequence storage. Cogn Comput. 2012;4:172–80.
36. Tan L, Ward G. Rehearsal in immediate serial recall. Psychon Bull Rev. 2008;15:535–42.
37. Townsend JT, Wenger MJ. The serial–parallel dilemma: a case study in a linkage of theory and method. Psychon Bull Rev. 2004;11:391–418.
38. Vergauwe E, Barrouillet P, Camos V. Do mental processes share a domain-general resource? Psychol Sci. 2010;21:384–90.
39. Vergauwe E, Camos V, Barrouillet P. The effect of storage on processing: How is information maintained in working memory? J Exp Psychol Learn Mem Cogn. 2014;40(4):1072–95.