Int. J. Human-Computer Studies 79 (2015) 51–65

Contents lists available at ScienceDirect

Int. J. Human-Computer Studies

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

Brief Lags in Interrupted Sequential Performance: Evaluating a Model and Model Evaluation Method ☆

Erik M. Altmann [a], J. Gregory Trafton [b]

[a] Department of Psychology, Michigan State University, East Lansing, MI
[b] Naval Research Laboratory, Washington, DC

ARTICLE INFO

Article history:
Received 6 March 2014
Received in revised form
10 December 2014
Accepted 27 December 2014
Communicated by E. Motta
Available online 5 January 2015

Keywords:
Task interruption
Sequence errors
Cognitive modeling
Goodness-of-fit testing

ABSTRACT

We examined effects of adding brief (1 second) lags between trials in a task designed to study errors in interrupted sequential performance. These randomly occurring lags could act as short breaks and improve performance or as short interruptions and impair performance. The lags improved placekeeping accuracy, and to interpret this effect we developed a cognitive model of placekeeping operations, which accounts for the effect in terms of the lag making memory for recent performance more distinct. Self-report data suggest that rehearsal was the dominant strategy for maintaining placekeeping information during interruptions, and we incorporate a rehearsal mechanism in the model. To evaluate the model we developed a simple new goodness-of-fit test based on analysis of variance that offers an inferential basis for rejecting models that do not accommodate effects of experimental manipulations.

1. Introduction

1.1. Background

Many everyday tasks have two important characteristics that interact to elevate the chances of a performance error. One is sequential constraints: A set of steps has to be performed in some prescribed order and an error occurs when a step is skipped or repeated. For example, in the medical domain, one might forget to record a dose of medication in a log (a skipped step), which could then lead to administering a second dose (a repeated step). Sequential constraints are common in medicine, equipment maintenance, computer programming and technical support, data analysis, legal analysis, accounting, and many other home and workplace environments. Sequential constraints also play a role in such basic cognitive processes as language production, event counting, serial recall, and problem solving. To perform correctly under sequential constraints, the cognitive system has to keep track of where it is in the sequence and select the correct next step when one step is complete, a process we refer to as *placekeeping*.

The second characteristic is the possibility of interruption: In the middle of a task the phone might ring, an email might arrive, or a glitch or subgoal of some kind might arise in the primary task. Experience suggests that interruptions like this often lead to "where was I?" moments afterwards, and in fact interruptions generate substantial performance costs at the point where the interrupted task is resumed (e.g., Altmann and Trafton, 2007; Hodgetts and Jones, 2006; Monk et al., 2008).

That said, errors in sequential performance can be a challenge to study, both in general and after interruptions, because they are relatively infrequent in most tasks that it makes sense to have people perform. In routine procedures like making coffee, for example, error rates in one study reached only 4% even in the condition where interruptions were timed to be most disruptive (Botvinick and Bylsma, 2005). To obtain enough errors to analyze, researchers have variously studied neurological patients (Cooper et al., 2005) and used diary methods to expand the temporal window during which errors can occur (Reason, 1990). In laboratory tasks, a common approach is to structure the task environment to increase memory load. This can be done by including "post-completion" steps (Li et al., 2008), which are difficult to remember to begin with (Byrne and Bovair, 1997), or by including an ongoing task that makes it easy to forget to return to the interrupted task (Dodhia and Dismukes, 2009). Perhaps the most common device is to eliminate any cues in the task display that could tell participants where they were in the task sequence (e.g., Brumby et al., 2013; Gray, 2000; Trafton et al., 2011).

In recent work we developed a new task to study errors in interrupted sequential performance (Altmann et al., 2014). As in other interruption tasks there are no external placekeeping cues, but we also designed the stimulus materials and decision rules to generate enough perceptual and cognitive load that placekeeping

☆This research was supported by grants from the Office of Naval Research, N000140910093 and N000141310247 to the first author and N0001412RX20082 and N0001411WX30014 to the second author.

E-mail addresses: ema@msu.edu (E.M. Altmann), greg.trafton@nrl.navy.mil (J.G. Trafton).

http://dx.doi.org/10.1016/j.ijhcs.2014.12.007
1071-5819/

operations have to compete with task steps for system cycles, and to generate enough variability from trial to trial that processing does not become routine. The task is also continuous, producing many opportunities for error and many opportunities to interrupt participants between steps of the primary task.

The error data generated by this task are rich enough to be analyzed as a function of multiple experimental factors and interactions (Altmann et al., 2014). For example, interruption effects are substantial, but there are also enough errors on trials not preceded by interruptions to shed some light on placekeeping under baseline conditions. Errors also form gradients as a function of the "offset" of the incorrect step from the correct step within the sequence, and the shapes of these gradients interact with interruption effects. All told, the empirical patterns are complex enough to provide strong constraints on a theory of the underlying mechanisms.

1.2. Present study

In the present study we address an interrelated set of applied, theoretical, and methodological goals concerning interrupted sequential performance. The applied question is whether slowing people down a little can improve placekeeping accuracy. There is considerable evidence that people can trade speed for accuracy strategically (e.g., Wickelgren, 1977), and there is evidence from interruptions research in particular that linking errors to a high time cost improves accuracy (Brumby et al., 2013). Of interest here is whether a lower bound on the time between events—not an upper bound on time to respond, as in deadline procedures, but a brief lockout period in which there is no processing to be done—has the side effect of improving accuracy. To address the question we added brief (1 second) lags randomly between trials of our task, and compared performance on trials preceded by a lag with trials preceded immediately by another trial.

We also wanted to investigate rehearsal as a placekeeping strategy during interruptions. Rehearsal is a core strategy in memory procedures (e.g., Baddeley et al., 1975; Reitman, 1974), but beyond an earlier study of ours (Trafton et al., 2003) there seems to be little research evaluating the empirical prevalence of rehearsal in context of task interruption. Here we include a self-report measure asking participants to indicate, after the experimental session, if they used any strategies to keep their place in the interrupted task.

Our theoretical goal is to develop a cognitive model of placekeeping mechanisms that explains the effect of brief lags and the role for rehearsal if we find evidence for it, and that accounts for the complex empirical patterns in data from our task more generally. As we suggested above, placekeeping seems to be a general capability expressed in many different tasks, so such a model could inform our understanding of errors in many different contexts.

The basic theoretical premise in our model is that placekeeping involves two interacting memory systems, one that stores episodic information about what steps were recently performed, and another that stores a long-term associative representation of the task sequence. When the cognitive system has finished performing one step in a sequence, it selects the next step by first remembering what step it just performed, then using that memory to index into the associative representation of the task sequence to find that step's successor. Skipped or repeated steps arise from errors in these two retrieval operations.

In context of this basic theoretical framework, several cognitive mechanisms could lead to improved accuracy after brief lags, each by sharpening memory for the most recently performed step and thereby improving accuracy in looking up the next step. Cowan (1999) proposed that an item does not decay as long as it remains in the focus of attention. One possible illustration of this mechanism is that participants in discrimination learning tasks hold tightly to their most recent hypothesis over a long series of trials if given no feedback to update it (Frankel et al., 1970). In context of our lag condition, if information about the most recently performed step remains in the focus of attention during the lag, then it will not decay—even as information about earlier steps that is not in the focus of attention does decay. Thus, after a lag, the most recently performed step will be more active in memory in relation to earlier steps, leading to more accurate selection of the next step.

Another mechanism is a "strengthening" process that has played a role in previous models of goal-directed performance (Altmann and Gray, 2008; Altmann and Trafton, 2002; Trafton et al., 2011). Strengthening hypothetically takes some time but could be deployed during a brief temporal lag to maintain the activation of relevant control information. A related construct is the attentional refresh process found in some models of working memory (Barrouillet et al., 2004; Oberauer and Lewandowsky, 2011). Strengthening and attentional refreshing are more active and strategic whereas Cowan's (1999) mechanism is more passive and structural, but each mechanism points to the same outcome, which is improved accuracy after a brief lag.

There is also some reason to expect the opposite outcome. In previous work with our task, interruptions as brief as 2.7 seconds reduced accuracy (Altmann et al., 2014), and 1 second is not that much shorter than 2.7 seconds. Moreover, there is evidence that unpredictable onset of events impairs placekeeping. Using an event-counting task, Carlson and Cassenti (2004) found higher error rates when the timing between event onsets was random than when it was rhythmic, for a given average time between events. In our task, if placekeeping operations are triggered by completion of a step, then an unpredictable lag between that step and the next could increase the chance of an anticipatory error. In this case a successful model would have to spell out the timing and coordination of the underlying control operations in detail.

Finally, our methodological goal is to develop and evaluate a simple method for testing whether a model adequately accounts for effects of experimental manipulations. The method involves fitting the model to the data from each individual participant, to generate a distribution of model-data residuals across participants for each cell of the experimental design. If these distributions cluster around zero in all cells of the design, this would indicate that the model is able to track all the experimental effects. If the distributions differ significantly from zero in at least some cells, this would indicate that the model was unable to track a specific experimental main effect or interaction—the conditions of which should help us identify the underlying theoretical problem. The decision rule for testing model-data residuals comprises a set of F ratios derived in part from the analysis of variance (ANOVA) applied to the empirical data.

In sum, our goals in this study are as follows. Empirically, we would like to investigate the effect of brief lags between trials that could function either as short breaks that help performance or as short interruptions that hinder it. We would also like to examine the role of rehearsal as a strategy for maintaining placekeeping information during interruptions. At a theoretical level, we would like to develop a cognitive model of placekeeping, focusing in this study on explaining effects of brief lags and rehearsal. At a methodological level, we would like to demonstrate a simple procedure for testing model fit and inferentially rejecting models that include incorrect assumptions.

In the remaining sections we present the experiment (Section 2), then describe the model (Section 3), and then describe our goodness-of-fit test and apply it to different model versions (Section 4). In the General Discussion (Section 5) we discuss the external validity of our task, relate our goodness-of-fit test to Bayesian methods, and discuss limitations of our modeling approach. In the Appendix we describe the model mathematics and assumptions in detail.

2. Experiment

2.1. Overview of experimental task and design

Here we describe key characteristics of the task (first reported by Altmann et al., 2014) and describe the design for the experiment we report here. Remaining details about task materials and procedure are given in Section 2.2 describing the method.

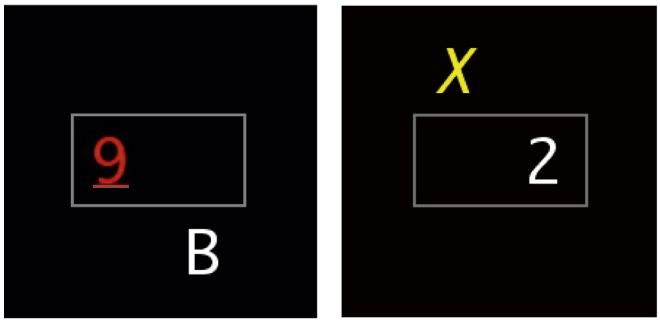
Participants perform a sequence of seven different forced-choice subtasks, or *steps*, in a prescribed order, and start the sequence over again when they reach the end. To make the sequence itself easy to remember, the prescribed order of steps is defined by an acronym—the word UNRAVEL—with each letter of the acronym representing a mnemonic for one of the seven choice rules. Participants cycle through the UNRAVEL sequence about 45 times in an experimental session, so there are about $45 \times 7 = 315$ trials per session, where a *trial* is a performed step. After every sixth trial on average, performance is interrupted by a simple typing task in which a randomized letter sequence is presented visually and the participant has to type the sequence correctly into a box.

Fig. 1a shows two examples of the kind of stimulus presented to the participant on each UNRAVEL trial. Each stimulus contains a randomly selected letter and a randomly selected digit. One character or the other has a randomly selected font style (underline or italics), one character or the other has a randomly selected color (red or yellow), and one character or the other is randomly placed outside the gray outline box (above or below). The box itself is a fixture that appears in the same location on every trial. A new stimulus is generated for each trial.

Fig. 1b shows the choice rules for each step. The choice for the *U* step is whether the font style in the stimulus is underline or italic; for the *N* step is whether the letter is near to or far from the start of the alphabet (the candidate letters are *A*, *B*, *U*, and *X*); for the *R* step is whether the color is red or yellow; for the *A* step is whether the character outside the box is above or below; for the *V* step is whether the letter is a vowel or a consonant; for the *E* step is whether the digit is even or odd; and for the *L* step is whether the digit is less than or more than 5 (the candidate digits exclude 5). In each case, the letter for the step mnemonically identifies one of the two candidate responses— *u* for underline, *n* for near to, *r* for red, *a* for above, *v* for vowel, *e* for even, and *l* for less-than. There are 14 candidate responses, each of which maps to one step, so from any actual response we can identify which step the participant thought was correct on that trial and can therefore code sequence errors. The stimulus is constructed such that any choice rule can be applied on any trial, so information about which step is correct to perform on the current trial has to be maintained in memory.

The main measure of interest is the frequency of *sequence errors* on UNRAVEL trials. A sequence error occurs when the participant skips or repeats one or more steps of the sequence. A sequence error is coded with respect to the previous trial. For example, if on three successive trials the participant performs *U*, *R*, and *A*, respectively, the *R* would represent a sequence error because the

a



b

Step	Candidate responses		Choice rules	Responses to sample stimuli: Stimulus 1	Stimulus 2
U	u	i	character is Underlined or in Italics	u	i
N	n	f	letter is Near to or Far from start of alphabet	n	f
R	r	y	character is Red or Yellow	r	y
A	a	b	character is Above or Below the box	b	a
V	v	c	letter is Vowel or Consonant	c	c
E	e	o	digit is Even or Odd	o	e
L	l	m	digit is Less than or More than 5	m	l

c

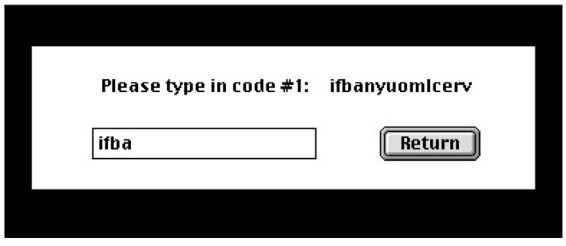


Fig. 1. (a) Two sample stimuli for the UNRAVEL task (the 9 is red and the *X* is yellow). (b) Response mappings for the UNRAVEL task, and responses for the two sample stimuli in (a). (c) Sample stimulus for the interrupting task, after the participant has started to type the "code".

participant skipped *N*, but the *A* would represent correct performance because *A* follows *R* in the sequence.

Fig. 1c shows a sample interruption stimulus. There is a string of 14 letters, which the participant is in the middle of typing. After typing all 14 the participant presses the Return key and a second set is presented. After typing the second set of 14 the participant presses Return again and that ends the interruption. If a string is incorrect when the participant presses Return, the box is emptied and the participant must type the string again. The 14 letters are a random permutation of the 14 candidate responses for the UNRAVEL task (see Fig. 1b), the aim being to interfere with participants' ability to use the keyboard as external memory to remember their place in the UNRAVEL sequence.

An interruption occurs every six trials, on average. The number of trials between one interruption and the next is determined by summing two terms: a constant 3 and a sample from an exponential distribution with mean 3. Because of the exponential term, there is a flat hazard function for interruption occurrence after the third trial, so the timing of interruption occurrence is unpredictable. Because the UNRAVEL sequence has seven steps, the average of six trials between interruptions means that across the experimental session interruptions are evenly distributed across the steps of the task sequence.

The experimental manipulation in the present study involves the *timing* of events. After a trial there is a .5 probability of a 1 second lag occurring during which the computer monitor goes blank and no task-related processing is called for. After the lag, the next event—the next UNRAVEL trial, or an interruption—occurs as it usually would. The *lag* condition includes trials preceded by a lag, and, in the case of trials preceded by interruptions, those where the interruption was preceded by a lag. The *nolag* condition includes all other trials.

The experimental design is 2 (timing) $\times$ 6 (position) $\times$ 6 (offset), with all factors within participants. The factor *timing* (lag, nolag) indicates whether there is a 1 second lag before a trial (lag) or not (nolag). The factor *position* (1–6) is the serial position of the trial after the interruption. (Every run of trials after an interruption has positions 1, 2, and 3, because of the constant 3 we referred to above, but for positions 4 and beyond the number of observations drops off sharply because of the exponential term, so we stop at position 6.) Finally, the factor *offset* (-3, -2, -1, $+1$, $+2$, $+3$) is the distance, in steps, of a performed step from the correct step when a sequence error occurs. For example, after an *R* trial, the correct next step is *A*, and if the step performed instead is *U*, *N*, *R*, *V*, *E*, or *L*, this is a -3, -2, -1, $+1$, $+2$, or $+3$ error, respectively.

2.2. Method

2.2.1. Participants

Participants were members of the Michigan State University community, who received either credit toward a course requirement or $10. Data from 150 participants were included in the study. Data from an additional eight participants were excluded because the participant's accuracy in each case did not meet a threshold, as described in Section 2.2.3.

2.2.2. Materials

In Section 2.1 we gave an overview of the experimental task; here we give a more formal definition of how the stimulus is constructed on each trial. For reference, Fig. 1a shows two sample stimuli. Each stimulus has five *attributes* each with four candidate *values* (in parentheses): letter (*A*, *B*, *U*, *X*), digit (1, 2, 8, 9), font style (left-underline, left-italic, right-underline, right-italic), color (left-red, left-yellow, right-red, right-yellow), and height (left-above, left-below, right-above, right-below). The "left" and "right" elements of the candidate values for font style, color, and height refer to the left and right character positions, and the "above" and "below" elements are relative to the box, which is a fixed feature of the stimulus. On a given trial, the left-right order of the letter and digit attributes is first randomly determined. Then, for each attribute, one value from the set of four candidates is randomly sampled, subject to the constraint that a value cannot repeat between trials.

2.2.3. Procedure

Participants were tested individually in sessions lasting about 45 minutes. A session began with the participant being introduced to each step of the UNRAVEL step sequence. The introduction emphasized the acronym, showing how the choice rule for each step in turn corresponded to a constituent letter. Once all the steps had been introduced, a screen appeared showing the letters spelling out the word and summarizing the choice rules for each step (essentially Fig. 1b). After this, to ensure that participants understood both the choice rules and the sequential structure of the task, the computer administered 16 practice trials during which it required the participant to make the correct response on each trial before the participant could move on. This 16-trial sequence was interrupted twice, to illustrate for participants how they should pick up after an interruption where they had left off. The experimenter remained present during this period to provide answers if necessary. A sheet of paper with the choice rules for the UNRAVEL sequence remained visible to the side of the computer throughout the session.

In preparation for the experimental phase of the session, participants were reminded to "please try to keep your place in the UNRAVEL sequence," and to "please try to pick up in the sequence where you left off" after an interruption.

The experimental phase consisted of 4 blocks, each with 12 interruptions and thus about $13 \times 6 = 78$ trials (there were 13 runs of trials per block because there was one run before the first interruption). During this phase the computer accepted any of the 14 candidate responses in Fig. 1b as the response for a trial. No feedback was given after individual trials about whether the response was correct or not. After each trial, there was a 50% chance of a lag occurring.

After each block the participant was given his or her score, computed as the percentage of trials in that block for which the step and response were both correct. If the score was above 90% the participant was asked to go faster. If the score was below 70% the participant was asked to be more accurate and that block was excluded from analysis (2 cases). A participant was replaced if he or she scored below 70% on two or more blocks (8 cases). We also checked that a participant's accuracy on the post-interruption trial was significantly above chance; no participant failed only this additional test.

At the end of the experimental session, the experimenter asked the participant, "When you were interrupted by the license codes, did you use any particular strategy or technique to remember where you were in the UNRAVEL sequence?" (The interruption stimuli had been described to participants in the beginning as being "kind of like software license codes.") The data are the experimenter's written summaries of each participant's response. The experimenter was an undergraduate research assistant naive to our theoretical assumptions.

2.3. Results

All our data are posted as supplementary materials at msu.edu/~ema/brieflags.

2.3.1. Errors

The top panel of Fig. 2 shows sequence errors separated by timing (lines) and position (abscissa) and averaged over the offset

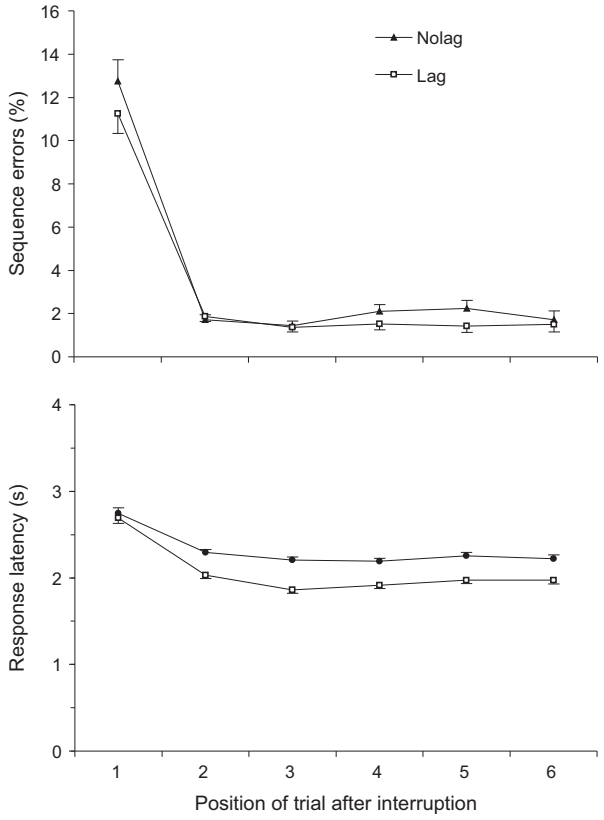


Fig. 2. Sequence errors and response latencies from the experiment, averaged over the offset factor. Error bars are ±1 standard error of the empirical mean.

factor. The effects of interruption were isolated to position 1, and on this basis we consider positions 2 and later as *baseline* trials. For symmetry we often refer to the position 1 trial as the *post-interruption* trial.

Fig. 3 shows sequence errors separated by baseline versus post-interruption (empty vs. filled markers) as well as timing (panels) and offset (abscissa). The markers represent data values, and the lines (dot-dashed, dashed, and solid) represent the best-fitting theoretical values produced by different models we discuss later.

We examined the error data with a 2 (timing) × 6 (position) × 6 (offset) within-participants ANOVA, the results of which appear in Table 1. There was a main effect of timing, with the overall error rate in the lag condition (M=0.525%, SE=0.038) lower than in the nolag condition (M=0.610%, SE=0.042). Timing did not interact significantly with the other factors. There were also main effects of position and offset. The position effect is simply that error rates were higher on the post-interruption trial than on baseline trials. The offset effect, evident in Fig. 3, reflects the decrease in error rate

Table 1
Omnibus Analysis of Variance of Sequence Error Data from the Experiment.

Contrast	df_{effect}	df_{error}	MS_{effect}	MS_{error}	F	p	η_p^2
Timing (T)	1	149	1.95E-03	2.94E-04	6.6	.011*	.043
Position (P)	5	745	8.88E-02	7.01E-04	126.6	.000*	.459
Offset (O)	5	745	3.58E-02	5.00E-04	71.7	.000*	.325
T × P	5	745	4.56E-04	2.81E-04	1.6	.152	.011
T × O	5	745	4.00E-04	3.13E-04	1.3	.271	.009
P × O	25	3725	9.84E-03	3.62E-04	27.2	.000*	.154
T × P × O	25	3725	4.41E-04	3.21E-04	1.4	.101	.009

* p < .05.

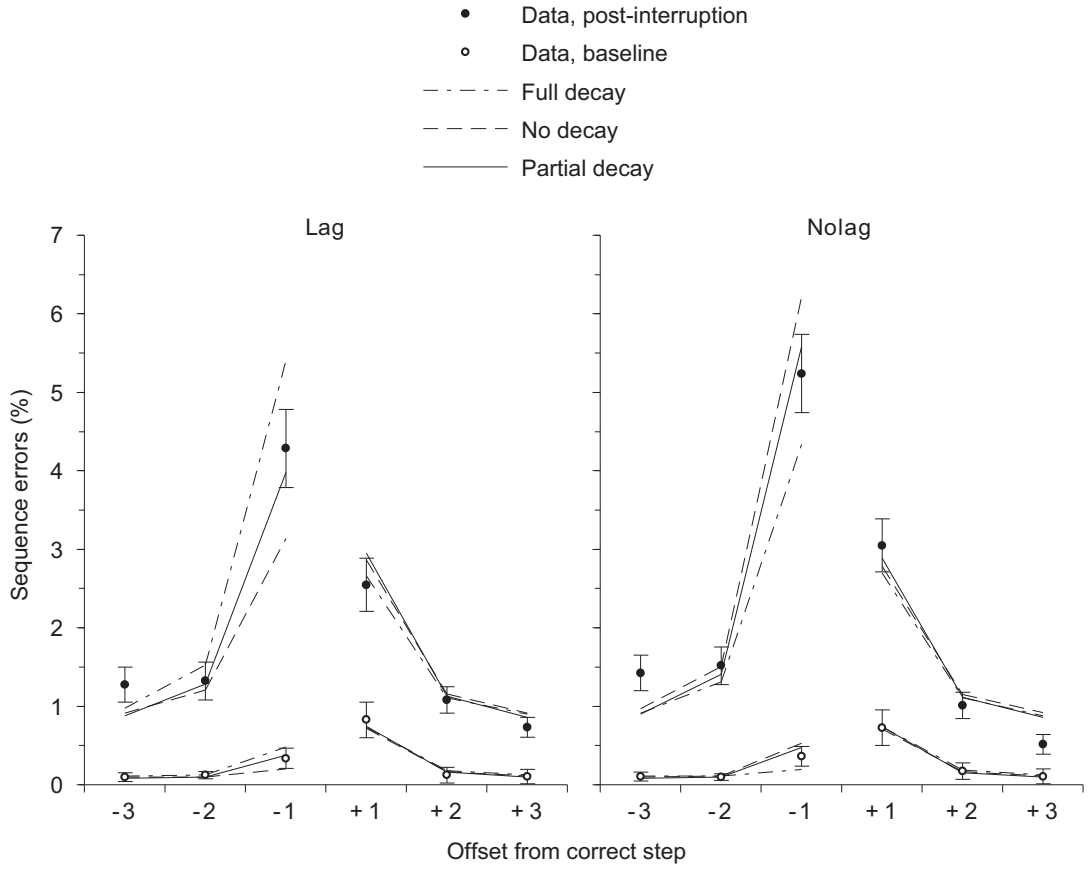


Fig. 3. Empirical sequence error rates (markers) and model fits (lines). Post-interruption is the position 1 trial. Baseline is the average of positions 2–6. Error bars are ±1 standard error of the empirical mean.

with increase in offset from the correct step. Finally, there was a position × offset interaction, also evident in Fig. 3, which reflects the larger difference between post-interruption and baseline error rates at offsets −1 and +1 than at offsets further from the correct step.

2.3.2. Response latencies

The bottom panel of Fig. 2 shows response latencies separated by timing (lines) and position (abscissa). The data are means of participant medians on correct trials. Interruption effects were mainly local to position 1, as for errors.

We examined the latency data with a 2 (timing) × 6 (position) ANOVA. There was a main effect of timing, $F(1, 149)=254.5$, $p<.001$, $\eta_p^2=.631$, with latencies faster in the lag condition ($M=2.075$ s, $SE=0.036$) than in the nolag condition ($M=2.320$ s, $SE=0.033$). There was also a main effect of position, $F(1, 5)=102.3$, $p<.001$, $\eta_p^2=.407$, and a Timing × Position interaction, $F(5, 745)=8.4$, $p<.001$, $\eta_p^2=.053$. The interaction reflects the absence of a timing effect on position 1, $t(149)=1.2$, $p=.217$. There, the lag before trials in the lag condition occurs before the interruption and is therefore temporally distal, which may have attenuated its effects.

Interruption duration was $M=20.424$ seconds ($SE=0.504$); this is the mean of participant medians taken across all 48 interruptions.

2.3.3. Self-reported strategy use

In the strategy self-reports we collected at the end of the session, we coded a response as indicating a rehearsal strategy if it made reference to any form of the word "repeat" or to saying (or singing) something out loud. According to this rubric, most participants used rehearsal—125 of 150, or 83%. To be conservative, we did not code rehearsal if a response made reference only to "remembering" a step or to use of "mnemonic devices," as these characterizations could have indicated some other strategy.

We coded the 125 rehearsal responses on two additional dimensions. One was whether the rehearsed target was the step performed immediately before the interruption (cases that referred to "last" or "previous") or the step to be performed after the interruption (cases that referred to "next"). Of the 125 rehearsal cases, 54 referred to the pre-interruption step, 65 referred to the post-interruption step, and 6 referred to rehearsing both steps at different times.

The second dimension was whether the target of rehearsal was the step itself or some other, derived code. Of the 125 rehearsal cases, 115 referred to the step itself. Of the remaining 10 cases, 8 made reference to numbering or otherwise identifying the position of steps in the sequence and rehearsing the number or position rather than the step itself.

2.4. Discussion

Trials preceded by a lag were performed more accurately than trials that were not. The effect was small (0.085%), but it was significant, so the results suggest that in a task environment with especially costly errors, one approach to improving performance is to slow people down with a brief, forced pause between events. Response time decreased by about 300 msec after lags, which was not enough to offset the lag itself, so the net effect of a lag was to slow performance.

One open question is whether lags would have the same effect if their frequency or predictability were different. If the mechanisms that take advantage of brief lags are under strategic control, then lags might be more effective if they occur predictably after every response and less effective if they are rare. For example, there is evidence from the task-switching domain that the system's propensity to make use of short preparatory intervals depends on whether the length of the interval is randomized or constant (Altmann, 2004). If lag had been a between-participants variable the error effect might have been larger.

From the self-report strategy data, it seems that rehearsal was the dominant way to maintain placekeeping information during interruptions. However, as interruptions go, they were relatively frequent in our task. In environments where interruptions are less frequent or somehow more surprising, strategies like rehearsal may not be ready to hand, in which case we would expect interruption effects to be larger.

The strategy data also indicate that rehearsal was flexible, with different participants rehearsing different target information, and some participants recoding steps into numerical or positional information. The latter finding suggests that even if the interrupted task were less distinctly verbal, people may nonetheless recode task elements verbally so as to be able to bring rehearsal to bear—unless the task materials made this difficult, in which case we would again expect interruption effects to be larger.

3. The remember-advance model

3.1. Model overview

Here we describe our model at a functional level suitable for interpreting the effects of our timing manipulation. The model comprises a set of equations that determine the activation levels and retrieval probabilities of various memory codes hypothetically involved in placekeeping operations. The equations and the associated theoretical assumptions are described in detail in the Appendix. The model source materials are posted at msu.edu/∼ema/brieflags.

The basic theoretical assumption in the model is that placekeeping operations—the control operations that select the next step when one step is complete—involve two interacting memory systems. One of these is an episodic memory of information left over from recent performance. The other is an associative representation of the task sequence stored in long-term memory. Placekeeping involves a *remember* stage in which the system consults the episodic memory for recent performance to determine the most recently performed step, followed by an *advance* stage in which the memory retrieved during the remember stage is used to "look up" the next step in the representation of the task sequence stored in long-term memory.

In terms of the first memory system, previous work suggests that performance of even fine-grained tasks—including simple, two-alternative forced-choice tasks like those performed on each trial of the UNRAVEL task—involves *control codes* stored in episodic memory that serve to organize operations like stimulus interpretation and response selection (Altmann, 2013; Altmann and Gray, 2008). In context of the UNRAVEL task, we assume that a control code is generated for each trial and includes the step to perform on that trial. After the trial, the code lingers in episodic memory. These codes decay over time, creating a ranking in which the most recent code is the most active, the next most recent is the next most active, and so on.

In terms of the second memory system, we assume that a task sequence is represented as an associative chain, with each step linked (only) to its successor. These links convey spreading activation from whichever step is in the focus of mental attention. Thus, after the remember stage retrieves a control code from episodic memory, the system focuses on this code, which makes the step for that code an activation *source*. Activation then spreads from the source step to its successors in the task sequence. This spreading activation attenuates with each additional link that it spreads,

creating a ranking of future steps in terms of activation, with the next step the most active, the step after that less active, and so on.

We refer to old control codes as *predecessors* and to steps in the task sequence following a given step as *successors*. For example, $pred_1$ is the most recent predecessor, representing the just-completed step, and $succ_1$ is the retrieved predecessor's immediate successor. In general, $pred_d$ is the dth predecessor and $succ_d$ is the dth successor, where d measures *distance* in number of performed trials for predecessors and number of steps in the task sequence for successors.

Correct performance occurs when the remember stage returns $pred_1$ and the advance stage returns $succ_1$, so the probability of correct performance depends on the retrieval probabilities for $pred_1$ and $succ_1$ versus other codes. The remember and advance stages each retrieve the most active code (predecessor and successor, respectively) when they execute, so retrieval probabilities are determined by the activation levels of codes relative to one another. $pred_1$ is the most active predecessor, on average, because predecessors decay as they age and $pred_1$ is youngest. $succ_1$ is the most active successor, on average, because spreading activation attenuates as it spreads, and $succ_1$ is closest to the activation source. Thus, usually the system generates correct performance.

A sequence error occurs when an older predecessor is retrieved in place of $pred_1$ or a more distant successor is retrieved in place of $succ_1$. For example, if the remember stage returns $pred_2$ instead of $pred_1$, and if the advance stage then returns $succ_1$ as it usually does, the just-performed step will be repeated, representing a sequence error with offset -1.

The reason $pred_2$ could be retrieved in place of $pred_1$ in this scenario is that activation levels are noisy: the activation of a code fluctuates about its mean, independently for each code and from moment to moment. This *activation noise* means that even though $pred_1$ is more active than $pred_2$ on average, it will occasionally be less active. Intrusions specifically of $pred_2$ on $pred_1$ are central to the model's account of the effect of brief lags, as we describe next.

3.2. Explaining the effect of the timing manipulation

The model's basic prediction for the timing effect in our experiment rests on two characteristics of how predecessor activation decays (as defined by Eq. (A1) in the Appendix). The first is that decay is negatively accelerating, such that younger predecessors decay faster than older ones. This characteristic predicts that during a lag, $pred_1$ should decay faster than $pred_2$, such that after a lag there is increased probability of an intrusion by $pred_2$ and thus an error at offset -1. The second characteristic is that decay continues without a lower bound—that is, although predecessors decay more slowly as they age, they nonetheless decay indefinitely. This characteristic predicts that predecessors older than $pred_2$ will be too decayed to show much of an effect if the activation of $pred_1$ changes. The two characteristics together predict that error rates should be higher in the lag condition than in the nolag condition, with the difference registering mainly at offset -1 because most intrusions will be from $pred_2$.

This basic prediction was incorrect, in that error rates were lower in the lag condition than in the nolag condition. This discrepancy is valuable for purposes of evaluating the goodness-of-fit test we describe in Section 4, because it identifies a theoretical problem with the model that the test should be able to detect and ideally help characterize.

We refer to this incorrect version of the model as the *full decay* version, because $pred_1$ decays throughout the lag. To explain the actual outcome of the experiment we developed three other versions. The *no decay* version represents Cowan's (1999) idea that information in the focus of mental attention does not decay. We assume that $pred_1$ stays in the mental focus of attention during a lag, because $pred_1$ governed performance on the pre-lag trial and there is no required processing during the lag that would displace it. In the no decay version, then, $pred_1$ does not decay during the lag while $pred_2$ does, such that accuracy should be higher after a lag.

In the *partial decay* version of the model, we weaken the assumption that information in the focus of attention is completely protected against decay, supposing only that information is somewhat protected against decay. In this version we added a free parameter to represent the amount by which $pred_1$ does decay during the lag. This version can also be interpreted as incorporating a strengthening or attentional refresh process that executes in the time available during the lag. The effect of such a process would be to give an activation boost to $pred_1$ that offsets some but not all of the decay that $pred_1$ would otherwise undergo during the lag.

Adding a parameter necessarily gives a model more flexibility to fit the data, so the partial decay version might pass our goodness-of-fit test simply because it has an extra parameter, not because the extra parameter reflects a mechanism necessary to explain the data. To control for this possibility we developed the *extra-parameter control* version of the model in which we added a different free parameter to the full decay model. The parameter we added governs the amount of activation noise, which affects accuracy more broadly than offset -1, so estimating it separately for each timing condition should help absorb the main effect of the timing manipulation. If this version nonetheless fails our goodness-of-fit test and the partial decay version passes, this would reinforce the conclusion that the effect of the lag is to improve memory specifically for $pred_1$.

4. Testing the model

4.1. The goodness-of-fit test

In the previous section we described four different versions of our model. Here we test each against the data, but first describe the goodness -of-fit test itself.

The test evaluates whether a model fit to individual participant data leaves systematic variance due to experimental factors unexplained. The test builds on the ANOVA design used to examine the empirical data. In our design the three experimental factors were timing, position, and offset. To test model fit, we added a fourth factor called *fit*, with levels *data*, meaning the empirical values, and *model*, meaning the best-fitting model values. The fit factor was coded as a within-participants variable. Of interest are any interactions of the fit factor with the experimental factors, which would indicate that the model fits the data differently at different levels of the experimental factor(s).

The inferential element of the test comprises a set of F ratios for interactions of the fit factor with experimental factors. These F ratios are formed using the error variance in the data, not the error variance pooled across the two levels of fit. The model contributes no error variance of its own, so a pooled error term is necessarily attenuated, the more so the better the model fit. The empirical error term, in contrast, provides a scale against which to measure the extent to which model-data residuals differ from 0 across cells of the design. A significant F for an interaction of experimental factors with fit, based on this error term, indicates that the amount by which model-data residuals differ from 0 is large relative to between-participants variability.

In the following sections we demonstrate the test by applying it to each of the four model versions. The test outcomes appear in Table 2, where information from the empirical ANOVA in Table 1 is replicated four times, once for each model. Each row shows a contrast from Table 1 now including the fit factor. The F ratio for

Table 2
Goodness-of-fit Analysis of Variance for the Four Models, with Degrees of Freedom (*df*) and MS_{error} from Table 1 (Same for Each Model).

Contrast	df_{effect}	df_{error}	MS_{effect}	MS_{error}	*F*	*p*	η_p^2
Full decay model							
Timing (T) × Fit (F)	1	149	3.63E-03	2.94E-04	12.4	.001*	.077
Position (P) × F	5	745	6.68E-05	7.01E-04	0.1	.993	.001
Offset (O) × F	5	745	3.89E-04	5.00E-04	0.8	.565	.005
T × P × F	5	745	6.04E-04	2.81E-04	2.1	.058~	.014
T × O × F	5	745	1.29E-03	3.13E-04	4.1	.001*	.027
P × O × F	25	3725	1.78E-04	3.62E-04	0.5	.984	.003
T × P × O × F	25	3725	4.07E-04	3.21E-04	1.3	.166	.008
No decay model							
T × F	1	149	3.87E-04	2.94E-04	1.3	.253	.009
P × F	5	745	6.68E-05	7.01E-04	0.1	.993	.001
O × F	5	745	3.24E-04	5.00E-04	0.6	.662	.004
T × P × F	5	745	4.17E-04	2.81E-04	1.5	.193	.010
T × O × F	5	745	1.51E-03	3.13E-04	4.8	.000*	.031
P × O × F	25	3725	1.98E-04	3.62E-04	0.5	.967	.004
T × P × O × F	25	3725	5.33E-04	3.21E-04	1.7	.021*	.011
Partial decay model							
T × F	1	149	1.16E-04	2.94E-04	0.4	.531	.003
P × F	5	745	6.22E-05	7.01E-04	0.1	.994	.001
O × F	5	745	2.72E-04	5.00E-04	0.5	.743	.004
T × P × F	5	745	8.07E-05	2.81E-04	0.3	.920	.002
T × O × F	5	745	2.51E-04	3.13E-04	0.8	.548	.005
P × O × F	25	3725	1.78E-04	3.62E-04	0.5	.984	.003
T × P × O × F	25	3725	2.08E-04	3.21E-04	0.6	.908	.004
Extra-parameter control model							
T × F	1	149	2.03E-04	2.94E-04	0.7	.407	.005
P × F	5	745	7.08E-05	7.01E-04	0.1	.992	.001
O × F	5	745	3.98E-04	5.00E-04	0.8	.553	.005
T × P × F	5	745	9.08E-05	2.81E-04	0.3	.899	.002
T × O × F	5	745	8.23E-04	3.13E-04	2.6	.023*	.017
P × O × F	25	3725	2.03E-04	3.62E-04	0.6	.962	.004
T × P × O × F	25	3725	2.53E-04	3.21E-04	0.8	.759	.005

* $p < .05$.
~ $05 < p < .10$.

each contrast is formed from the MS_{effect}, MS_{error}, and degrees of freedom on that row. The MS_{error} and degrees of freedom are repeated from Table 1 and are the same for each model.

To preview the test outcomes, three of the models produce significant interactions with the fit factor and can therefore be rejected as not fully accounting for effects of the experimental factors. One model does account for the data, with *F*s < 1 for all contrasts.

4.2. Testing the full decay version

In the full decay version of the model, we suppose that $pred_1$ decays relative to $pred_2$ during a lag, predicting a higher error rate in the lag condition specifically at offset -1. In Fig. 3, the fits of the full decay model are shown as dot-dashed lines (the data are shown as markers). The figure shows a substantial model-data misfit at offset -1, most noticeably for the post-interruption trial (filled markers). For the post-interruption trial, comparing across panels, the model predicts that -1 errors should have been more frequent in the lag condition (left panel) than in the nolag condition (right panel), whereas the empirical pattern was the opposite.

The goodness-of-fit test showed two significant interactions (Table 2). The Timing × Fit interaction indicates that the model predicted the wrong main effect of timing. Specifically, the interaction means that the model fit the two timing conditions differently, predicting too many errors in the lag condition and too few in the nolag condition. The misfit is distributed across the two conditions like this as a function of the fitting process maximizing model likelihood.

The Timing × Offset × Fit interaction means that the misfit across timing conditions was worse at some offsets than others. An analysis on offset -1 showed a significant Timing × Fit interaction, $F(1, 149) = 19.6$, $p < .001$, $\eta_p^2 = .116$, indicating that the model predicted the wrong effect of timing specifically at that offset. An analysis on offsets excluding -1 showed no significant interactions, all $p > .164$ and $\eta_p^2 < .013$, indicating that at these offsets the model fit was acceptable.

Thus, the goodness-of-fit test offers an inferential basis for saying that this version of the model predicts too many errors after a lag specifically at offset -1. The theoretical implication is that $pred_1$ does not decay during the lag like other predecessors.

4.3. Testing the no decay version

In the no decay version of the model we suppose that $pred_1$ is protected against decay during the lag because it remains in the focus of attention, predicting a lower error rate in the lag condition specifically at offset -1. In Fig. 3, the fits of the no decay model are shown as dashed lines. The figure again shows a substantial model-data misfit at offset -1, most noticeably for the post-interruption trial. For this trial, comparing across panels, the model now predicts the correct direction of the timing effect but over-predicts the size of the effect.

The goodness-of-fit test now shows no significant Timing × Fit interaction (Table 2), because the model predicted the correct direction of the timing effect. However, the test again shows a significant Timing × Offset × Fit interaction (Table 2), because the model over-predicts the size of the effect at offset -1. An analysis on offset -1 showed a significant Timing × Fit interaction, $F(1, 149) = 14.5$, $p < .001$, $\eta_p^2 = .089$, but an analysis on offsets excluding -1 showed no significant interactions, *F*s < 1, indicating that at these offsets the fit was acceptable.

Thus, the goodness-of-fit test offers an inferential basis for saying that this version of the model predicts too few errors after a lag specifically at offset -1. The theoretical implication is that $pred_1$ is not fully protected against decay during the lag.

4.4. Testing the partial decay version

In the partial decay version of the model we suppose that $pred_1$ is partially protected against decay during the lag. This model has an extra parameter *Y* that represents the extent of this protection. Activation in our model is based on timing parameters (see Eq. (A2) in the Appendix), so *Y* is coded as a delay from onset of the lag until $pred_1$ starts decaying.

In Fig. 3, the fits of the partial decay model are shown as solid lines. The fit is now better than for the previous models, and the goodness-of-fit test yields no basis to reject the model, with *F*s < 1 for all interactions involving fit (Table 2). The mean estimated value of *Y* was 0.657 seconds (Table A1), meaning that whatever process protects or boosts the activation of $pred_1$ during the lag is equivalent to delaying decay of $pred_1$ by two-thirds of a second from the start of the lag.

4.5. Testing the extra-parameter control version

The partial decay model would have fit better than the full decay and no decay models in any case because it had an extra free parameter. To evaluate whether simply adding a parameter allowed the partial decay model to pass our goodness-of-fit test, we added a different extra parameter to the full decay model. The parameter we added governs the amount of activation noise—that is, the standard deviation of moment-to-moment fluctuations of activation levels of codes in memory. We estimated two activation noise parameters, one each for the lag and nolag conditions,

instead of one such parameter for both conditions as in the other models.

Activation noise is an effective choice for our purposes here because it is not selective for any one predecessor or successor (see Eqs. (A3) and (A5) in the Appendix) and therefore affects model predictions more generally than at offset −1. Because of this general effect, estimating activation noise separately for each timing condition should help absorb the main effect of timing and thus weaken the Timing × Fit interaction that was significant for the full decay model—without necessarily weakening the Timing × Offset × Fit interaction, precisely because activation noise affects accuracy more generally than offset −1.

Consistent with this analysis, the goodness-of-fit test for this version of the model shows no Timing × Fit interaction ($F < 1$) but continues to show a significant Timing × Offset × Fit interaction (Table 2). (The model values are not shown in Fig. 3.) The theoretical implication is that it was not simply adding a parameter that allowed the partial decay model to pass our test—it was adding a parameter that targeted the model-data misfit specifically at offset −1.

4.6. Discussion

The winning model was the partial decay version, in which $pred_1$ decays somewhat during the lag but less than $pred_2$, such that the error rate at offset −1 is lower after a lag than after no lag. The goodness-of-fit test helped us converge on this account, rejecting models that did not fit the data and localizing the misfit to a level of an experimental factor (offset −1) where performance was hypothetically most sensitive to the activation of $pred_1$. The test then further rejected a model that did not single out $pred_1$ for special processing but did accommodate the main effect of the timing manipulation with an extra parameter, indicating that not just any extra parameter will do and that the one we added to the partial decay model addressed an underlying theoretical problem with the model.

In practical terms, the test gave us a clear signal for when to stop tinkering with the model. In contrast, descriptive measures of fit such as R^2 and root mean squared deviation (rmsd) offer no practical decision rule that could have guided the theoretical convergence we described above. For example, the three model versions plotted in Fig. 3 all have high R^2 values—.98, .92, and .97 for the full decay, no decay, and partial decay versions, respectively—and low rmsd values—0.35%, 0.37%, and 0.19%, respectively—so by some standard are all acceptable. Similarly, the error bars in Fig. 3 offer no clear sense of whether model-data misfits are large enough to be systematic. For instance, the solid lines representing the winning model do not pass through the error bars at offset −3, which might be a signal that the model is incorrect. However, the goodness-of-fit test says this misfit is not large enough relative to noise in the data to warrant further model changes.

From a hypothesis-testing perspective, a weakness of our method is that accepting a model depends on accepting a null effect. However, accepting the null is characteristic of goodness-of-fit testing generally (see, e.g., D'Agostino and Stephens, 1986), so although the use of ANOVA for goodness-of-fit testing is novel, it fits within a larger analytical tradition. Moreover, even in context of hypothesis testing, null effects are meaningful when power to detect an effect is high. Although we have not developed a formal approach to power analysis using our method, we have shown here that the method can inferentially reject at least some incorrect theoretical assumptions.

The main limitation of our approach is that it requires a model that can be fit to participant-level data to generate distributions of model-data residuals. This requirement means that each participant must generate enough error data to constrain a model, which means that not all tasks will work. The requirement may also be difficult to meet unless the model is represented as closed-form equations, for which maximum likelihood estimation using nonlinear optimization methods is computationally tractable. Stochastic cognitive simulations have advantages over closed form models, as we discuss below in Section 5.4, but we know of no practical techniques for fitting such models to hundreds of data sets using maximum likelihood estimation. In Section 5.3 we discuss the benefits and limitations of our test in relation to a more common method of model evaluation involving Bayesian statistics.

5. General discussion

We found that brief lags occurring randomly between trials had a small but significantly positive effect on accuracy on the post-lag trial, rather than functioning as brief interruptions that disrupted performance. We also found that rehearsal was the dominant strategy for maintaining placekeeping information during interruptions—at least in our task, where the task materials were verbal and where interruptions were frequent enough that people could be expected to keep rehearsal mechanisms at the ready.

Our model explains the effect of brief lags in terms of memory processes: Forcing people to slow down a little with a brief lockout period between trials made memory for immediate past performance a little more distinct, which led to fewer repetitions of the last step. However, the model also implies that the benefits of slowing performance are subject to theoretical constraints. In the version of the model that fit the data, $pred_1$ was protected against decay not for the full interval of the lag, but only for the period estimated by the Y parameter (Y=0.657 s, Section 4.4). One interesting possibility is that Y represents an architectural limit on the time for which an item in the focus of attention is effectively protected from decay. If so, then the estimate of Y would be the optimal lag between events, in the sense that lags up to that length would add to the distinctiveness of $pred_1$ but longer lags would not. Testing this possibility would require a parametric manipulation of lag length and estimation of Y for each level. If there is an underlying architectural limit, then estimates of Y should increase with lag length to that limit and then level off, with performance accuracy following a parallel pattern.

More generally, the memory distinctiveness account of the lag effect suggests that episodic memory processes might mediate other kinds of speed-accuracy tradeoffs as well. For example, when people slow down strategically in order to be more careful—as opposed to slowing down because the procedure makes them, as here—the effect on accuracy may simply reflect the effect of interfering control information in memory having a chance to decay a little more, rather than any reallocation of cognitive resources.

The model also captures effects of the other manipulations in our experiment, as indicated by our goodness-of-fit test. The gradients across the offset factor (Fig. 3; see also Altmann et al., 2014) arise because older predecessors and more distant successors have less activation and thus are less likely to intrude on the memory retrievals in the remember and advance stages, respectively. The large effect of interruptions (in Fig. 3, filled vs. empty markers) arises both because rehearsal takes some time to set up after interruption onset, allowing $pred_1$ to decay relative to older predecessors, and because the activation spreading to successors decays during interruptions (see discussion of the E and W_{post} parameters, respectively, in Section A.1 of the Appendix). In ongoing work we are probing these mechanisms in more detail by fitting the model to data from a range of interruption durations.

An open question is whether the processes that kept $pred_1$ active while older codes decayed were passive and structural (the protected focus construct) or active and strategic (strengthening

and/or attentional refreshing). Manipulating the frequency or predictability of lags might yield some evidence on this point. A structural account would seem to predict little to no change in the effect of brief lags, whereas a strategic account might predict a larger effect of brief lags the more predictable they are, as the system would then have a chance to adapt.

Below we address four other issues arising from this study: the external validity of the UNRAVEL task (Section 5.1), the seeming fine line between brief lags that help performance and brief interruptions that hinder it (Section 5.2), the relationship of our goodness-of-fit test to Bayesian model selection (Section 5.3), and limitations of our model (Section 5.4).

5.1. External validity of the UNRAVEL task

The UNRAVEL task is abstract in that the stimulus materials and decision rules bear little resemblance to any task that people perform in the workplace or at home. And yet, it represents control operations common to a diverse set of tasks. For example, language production requires that words be produced in the correct order, and research in this domain has examined sequence errors at the level of anticipation and perseveration errors (Dell et al., 1997), which map to "+" and "−" levels of offset, respectively, in our task. Event counting (Carlson and Cassenti, 2004) plays a role in everyday activities ranging from counting patrons in a restaurant (to remain with capacity limits) to counting repetitions of an exercise during a physical training regimen. Errors in event counting can be interpreted as skipping or repeating steps in the sequence of positive integers. Finally, serial recall involves reproducing a randomized sequence correctly from memory, but positional confusions at test can nonetheless be interpreted as skipping or repeating steps in the sequence of positional or context codes representing serial position (e.g., Anderson and Matessa, 1997; Brown et al., 2000).

These domains differ in many ways, but they all share constraints represented in our task. All involve chaining along some mental representation in a prescribed order, without repeating or omitting elements; all are susceptible to interruptions, to the extent they play a role in daily life; and all involve verbal materials that make them amenable to rehearsal. Our task was designed to represent these common features under conditions that produce high enough error rates to examine as a function of theoretically relevant factors like offset and thus support development of tightly constrained models. Preliminary evidence suggests that placekeeping mechanisms are relatively task-independent, in that sequence error rates do not seem to vary with difficulty of individual task steps (Altmann et al., 2014), so there is reason to think the model we developed here may generalize. Indeed, in serial recall, where errors have been analyzed extensively, the data show gradients similar to those in our task in which errors are more frequent the closer the item is to its correct output position (e.g., Brown et al., 2000).

Nonetheless, validation of laboratory tasks is an important step, and in recent work we found that performance on our task predicts a measure of general cognitive ability (Hambrick and Altmann, in press), in line with proposals that general ability involves systematic sequential processing of subgoals (Carpenter et al., 1990; Duncan, 2010). Similar validation would be useful for other tasks used in interruptions research also. As we noted earlier, such tasks often lack external placekeeping cues, leaving open the question of whether they predict performance on interfaces designed to have the proper cues. Moreover, cover stories such as doughnut production (Brumby et al., 2013) or financial management (Trafton et al., 2011) may add face validity, but whether they really add external validity is an open question. A recent study of the effect of interruptions on the quality of written essays (Foroughi et al., in press) is the relatively rare case in which interruption manipulations are performed directly on a criterion task.

5.2. Brief Lags versus brief Interruptions

The effect of the timing manipulation in our experiment was opposite the effect of brief interruptions in our previous work with the UNRAVEL task (Altmann et al., 2014). In that study, participants typed only 2 letters per interruption, instead of 28 as here, and the mean duration of interruptions was only 2.7 seconds, which is not that much longer than the 1-second lag between trials here. The question, then, is why the lag here helped rather than hindered performance. One difference is that lags were more frequent, occurring every second trial on average rather than every six trials.

However, what distinguished the effect of the 1 s lags here from the effect of the 2.7 s interruptions in Altmann et al. (2014) was probably neither the length nor frequency of the events, but that the lags were unfilled whereas the interruptions were filled. Monk et al. (2008, Experiment 3) compared unfilled and filled interruptions as short as 3 s and found that resumption lags were longer after filled interruptions, presumably because the filler task interfered with rehearsal of information related to the interrupted task. Based on our model, we would further predict that error rates would increase specifically for offset −1. In the full decay version of the model, $pred_1$ decays during the lag just like any other predecessor—as it would if there were a filler task that displaced $pred_1$ from the focus of attention and/or prevented $pred_1$ rehearsal. As $pred_1$ decays, $pred_2$ becomes more likely to intrude on the remember stage on the post-interruption trial, predicting an increase in −1 errors. In future work it would be useful to test this prediction by contrasting unfilled with filled interruptions using a task like UNRAVEL that involves placekeeping and affords measurement of the offset factor.

5.3. The goodness-of-fit test in relation to Bayesian model selection

A method often advocated for selecting between competing models is the Bayesian Information Criterion, or BIC (e.g., Wagenmakers, 2007). The BIC has at least two attractive characteristics. First, it is based on likelihoods, which have to be computed anyway in the course of maximum likelihood estimation. Second, it includes a method for adjusting the likelihood of a model based on the number of free model parameters, which is useful when comparing models with different numbers of parameters as we did here.

One limitation of the BIC approach is that it represents no information about the functional form of parameters. For example, parameters such as *Y* in the partial decay model (Section 4.4) and the extra activation noise parameter in the extra-parameter control model (Section 4.5) are treated as adding exactly the same amount of flexibility to the model—when, as we have shown, these two parameters had different effects on the model's ability to account for systematic variance in the data.

To demonstrate this limitation, we used the BIC approach to compare the full decay and partial decay models, which have five and six parameters, respectively. The log likelihoods for the two models, summed across participants, are −2542 for the full decay model and −2474 for the partial decay model. Thus, the log likelihoods favor the partial decay model (−2542 < −2474), as does our goodness-of-fit test. However, the corresponding BIC values, which are adjusted for number of parameters, are 8291 for the full decay model and 8797 for the partial decay model. For BIC values, lower is better, [1] so these values favor the full decay model (8291 < 8797).

[1] $BIC(M) = -2ln(Likelihood) + klnn$, where M is a model, *Likelihood* is the maximum likelihood of model M fit to a given participant's data, k is the number of free parameters, and n is the number of observations (Wagenmakers, 2007). In our application, $k=5$ for the full decay model and $k=6$ for the partial decay model, and $n=72$ for both (2 timing conditions x 6 offsets x 6 positions).

Nonetheless, we would say that the full decay model includes an incorrect theoretical assumption, which is that $pred_1$ decays like any other predecessor during an unfilled lag. We could be wrong, of course, but there is no reason to think that the BIC approach is right, because it ignores the functional form of the model's parameters whereas our approach tests the functional form directly against the data. The BIC approach also summarizes model fit in one number, whereas our approach yields a set of contrasts (Table 2) that help pinpoint the locus of the misfit. In several senses, then, our method seems to factor in more information.

A second limitation of the BIC approach is that it is structured to select the best among a set of models, rather than testing a given model inferentially against data. When there is only one model, and the question is whether it provides an adequate account of the data, the BIC approach offers little useful information. That said, when there are multiple models that pass an inferential goodness-of-fit test like ours, the BIC approach might be a useful complementary method to select the winning model heuristically.

5.4. Limitations of the model

Here we address two limitations of our model. The first concerns its scope with respect to strategic variability in task performance. The second concerns the tradeoffs associated with a closed-form representation compared with a computational cognitive simulation.

5.4.1. Strategic variability in placekeeping during interruptions

The data we reported in Section 2.3.3 indicate that participants used various strategies to maintain their place in the task sequence during interruptions. The predominant strategy was rehearsal, but there was variation in what kinds of codes participants rehearsed. Roughly half of participants reported rehearsing the step they had just completed (a predecessor, in our parlance) whereas the other half reported rehearsing the step to be performed after the interruption (a successor). Our model represents only the predecessor case, which is the more basic of the two in that to retrieve a successor to rehearse the system must first retrieve a predecessor. More importantly, the difference between predecessor and successor rehearsal should be absorbed by differences in estimates of model parameters, as we discuss in Section A.1 of the Appendix.

That said, sequence errors could arise from other mechanisms that we have not represented in our model. One possibility is a kind of reality monitoring failure affecting rehearsal during interruptions (Trafton et al., 2011). As the system rehearses, it could lose track of whether it is rehearsing the last step or the next step, and thus repeat or skip a step depending on whether it mistakes the last step as the next step or vice versa. This mechanism can account for $+1$ and -1 errors, but whether it can account for the full error gradients across the offset factor is a question we leave for future work.

5.4.2. Closed-form versus simulation models

Closed-form models like the one we used here are relatively straightforward to fit to participant-level data, even if they are non-linear. These participant-level fits in turn played an important role here in generating the distributions of residuals we used to test our model, and could in principle be used to characterize individual differences in terms of cognitive parameters rather than latent statistical variables, an approach we hope to pursue in future work.

However, closed-form models are limited in that they cannot tractably represent all dynamic processes that might be of interest. For example, we did not represent the possibility of second-order errors, in which the wrong predecessor is retrieved on one trial, leading to creation of an out-of-sequence control code on that trial that then changes the distribution of retrieval probabilities across different steps of the task sequence on the next trial. Representing all the necessary contingencies mathematically was simply not tractable, so we made a simplifying assumption (described in more detail in Section A.2.3 of the Appendix) that for any given trial, performance leading up to that trial was error-free. In a stochastic cognitive simulation, no particular effort would have been necessary to represent second order errors, which would instead be a natural product of the operation of the underlying memory processes.

In future work it may be useful to take a hybrid approach in which simulation experiments are used to evaluate whether mechanisms like second-order errors influence performance enough to be detectable behaviorally. Such an approach might offer additional leverage as we move toward more complete models of interrupted task performance that factor in perceptual placekeeping cues, hierarchical task structures, partial orderings, differential error costs, and other characteristics of everyday sequential performance.

Appendix

Here we describe the remember-advance model in detail. Section A.1 gives an overview of the model parameters and the mechanisms they represent, Section A.2 describes the model equations and associated theoretical assumptions, and Section A.3 describes the model-fitting procedure and parameter values.

A.1. Overview of model parameters

The model comprises a set of equations that characterize activation levels of predecessors and successors, map those to retrieval probabilities, and map those in turn to probabilities of sequence errors at different levels of the offset factor. We fit the model to each participant's data by estimating the values of a set of free parameters that affect the values of the different equations. The full decay and no decay versions of the model each have the same five parameters, and the partial decay and extra-parameter control models each have an additional sixth. Here we describe each parameter in terms of the cognitive mechanisms and any associated assumptions it represents. Estimated values for each parameter appear in Table A1.

The first parameter, E, concerns rehearsal, which was the dominant strategy used to maintain placekeeping information during interruptions (Section 2.3.3). We assume the following operating principles for rehearsal. Rehearsal does not begin

Table A1
Mean Best-Fitting Parameter Values for the Four Model Versions.

Parameter	Full decay version	No decay version	Partial decay version	Extra-parameter control version
E	5.133	3.624	3.867	4.246
s	0.046	0.055	0.056	0.050
W_{post}	1.169	0.387	0.375	0.315
W_{base}	0.412	0.509	0.511	0.455
g	0.295	0.316	0.341	0.317
Y			0.657	
s_{lag}				0.046

Note. E and Y are in seconds, W_{post} and W_{base} are in units of activation, and g, s, and s_{lag} are dimensionless.

immediately with interruption onset, but only after an interval E during which the system attends to the interrupting stimulus and/or activates the rehearsal machinery. After this interval, rehearsal begins by retrieving a predecessor as the rehearsal target. Rehearsal then maintains the retrieved predecessor in an active state through the rest of the interruption, while unrehearsed predecessors continue to decay. Thus, at the end of the interruption the rehearsed predecessor is much more active than other predecessors and therefore has high probability—we assume 1.0—of being the product of the remember stage on the post-interruption trial.

Under these operating principles, the interval E determines the probability of a correct step selection on the post-interruption trial. That is, after E seconds into the interruption, $pred_1$ will have decayed relative to $pred_2$ (and older predecessors), and that amount of decay will determine the probability of $pred_1$ entering the rehearsal process and therefore ultimately guiding selection of the step for the post-interruption trial. The role of the interval E in determining accuracy on the post-interruption trial is consistent with simulation studies suggesting that critical rehearsal processes occur early in an interruption (Salvucci et al., 2009).

The second parameter, s, governs the amount of activation noise. s is related by a transformation to the standard deviation of the moment-to-moment fluctuations in activation levels, so a larger s means more noise. For the extra-parameter control model (Section 4.5) we estimated this parameter separately for the lag and nolag conditions.

The third and fourth parameters, W_{post} and W_{base}, represent the amount by which successors are primed by spreading activation. W_{post} is for the post-interruption trial and W_{base} is for baseline trials. The two are distinct because priming accumulates over time when a source of activation is available, and decays over time when it is not (Anderson and Pirolli, 1984). We assume that during interruptions the focus of attention is elsewhere, such that priming decays. Thus, W_{post} should generally be smaller than W_{base}.

The W_{post} parameter should accommodate at least some of the variability in self-reported strategy use identified in Section 2.2.3. Most of the variability was in terms of what code participants reported rehearsing, with roughly half indicating the last completed step (a predecessor) and the other half the step to be performed after the interruption (a successor). In the model we only represent the predecessor case, which is the more basic of the two. That is, to retrieve a successor to rehearse, the system must first retrieve a predecessor, so the E parameter will affect accuracy the same way in both cases. However, if the system then immediately retrieves a successor as a rehearsal target, instead of rehearsing the predecessor and waiting until the position 1 trial to perform the advance stage, activation spreading to successors will be less decayed. Less decay of spreading activation is represented by a relatively smaller shortfall of W_{post} relative to W_{base}, as we describe in Section A.2.2.

The fifth parameter, g, is the proportion of spreading activation reaching a step that is passed on to that step's successor. Because spreading activation attenuates as it spreads, g should take on values less than 1.

The partial decay model (Section 4.4) has a sixth parameter Y that represents the interval between the start of the lag and the moment at which $pred_1$ starts decaying. This parameter in effect creates a continuum between the full decay and no decay models, where $Y=0$ for the full decay model and $Y=1$ s (the duration of the lag) for the no decay model.

In the model equations discussed in the next section, E plays a role in Eq. (A2), s in Eqs. (A3) and (A5), W_{post} and W_{base} as variants of W in Eq. (A4), and g in Eq. (A4).

A.2. Model equations

Here we describe the model equations—those affecting activation and retrieval probability of predecessors, those affecting activation and retrieval probability of successors, and those that map retrieval probabilities to sequence error probabilities, followed by modifications that implement the no decay and partial decay models.

A.2.1. Predecessor activation and retrieval probability

The activation of $pred_d$—the control code that governed performance of the dth preceding trial—is

$$A(t_d) = -0.5ln(t_d), \tag{A1}$$

where t_d is the age of $pred_d$ and 0.5 is the decay rate. Eq. (A1) is a simplified representation of base-level activation in ACT-R (Anderson and Lebiere, 1998). (The full-blown representation, in which an item's activation is boosted each time the item is retrieved, is not well suited to representing the activation of control codes, which have to decay for functional reasons regardless of how often they are retrieved; Altmann and Gray, 2008.) The decay rate of 0.5 is a standard value that has emerged across ACT-R models (Anderson, 2007).

The decay function specified by Eq. (A1) has two important characteristics that are the basis for interpreting the effects of the timing manipulation in our experiment. The first is that decay negatively accelerates, with greater decay per unit time for younger predecessors. Thus, as predecessors decay together as a set, $pred_1$ loses activation relative to older predecessors and thus becomes less likely to be retrieved, increasing the chance of a retrieval error.

The second characteristic is that the function has no asymptote, such that predecessors decay indefinitely. Thus, as a predecessor ages, its retrieval probability becomes vanishingly small, which has the functional effect of reducing proactive interference in episodic memory for control codes (Altmann and Gray, 2008). For our purposes, the critical implication is that intrusions on $pred_1$ come mainly from $pred_2$, which mainly generate errors at offset -1 (but see Section A.2.3). Thus, factors that affect how and whether $pred_1$ decays should mainly affect the -1 error rate.

Several factors play a role in determining t_d in Eq. (A1), and thus the activation of $pred_d$. One is the response latency, R, to perform a trial. A second is the lag L, if any, between the response and the next stimulus onset; on average, half of all trials are followed by a lag. A third factor is the interruption duration, I, which affects the age of most codes from before an interruption (though not all, as we describe below).

A fourth factor is rehearsal of a predecessor during the interruption. Rehearsal improves accuracy on the post-interruption trial, where the rehearsed predecessor is the target of the remember stage, but *impairs* accuracy on the baseline trials that follow, where the rehearsed predecessor is a distractor that could intrude on the target of the remember stage. We represent both effects in terms of predecessor age, which modulates predecessor activation. Of interest is the age of the rehearsed predecessor when the critical retrieval of that predecessor occurs.

On the post-interruption trial this critical retrieval occurs when rehearsal starts, E seconds into the interruption (see Section A.1). This retrieval is the critical one under the assumption that whatever predecessor enters rehearsal is also then the product of the remember stage on the post-interruption trial (Section A.1). If rehearsal is as prevalent as our strategy data suggest (Section 2.3.3), then estimates of E should generally be less than I, the full interruption duration.

On baseline trials, meaning trials at position 2 or later, the critical retrieval occurs at the start of the trial. The correct retrieval target on baseline trials is always the predecessor from the preceding position, not the predecessor rehearsed during the interruption, so the rehearsed predecessor is a distractor rather than a target. We assume that rehearsal served to offset decay of the rehearsed predecessor during the interruption, leaving that code as active as it would have been had the interruption never

happened. We also assume, for tractability and only for baseline trials, that the rehearsed predecessor is the one from the pre-interruption trial (we elaborate on this assumption in Section A.2.3). Thus, on baseline trials, the age of the pre-interruption predecessor is simply the age of the post-interruption predecessor plus the time between trials.

Taking these various factors into account, the age t_d of $pred_d$ in the full decay model described in Section 4 is

$$t_{d,\ \mathrm{nolag}} = \begin{cases} R+(d-1)(R+L/2)+E & \text{for position} = 1 \\ R+(d-1)(R+L/2) & \text{for position} \geq 2 \text{ and } d \leq \text{position} \\ R+(d-1)(R+L/2)+I & \text{for position} \geq 2 \text{ and } d > \text{position} \end{cases}$$

$$t_{d,\ \mathrm{lag}} = t_{d,\ \mathrm{nolag}} + L \tag{A2}$$

where *position* is the serial position of a trial after an interruption, with position 1 being the post-interruption trial and positions 2 and later being baseline trials. $L=1$ is the lag, and the $L/2$ terms reflect the fact that half of all trials between $pred_{d>1}$ and $pred_1$ were followed by a lag. R is the response latency on trials and I is the interruption duration (Section A.3 gives means for R and I and describes how they were computed). The no decay and partial decay models are implemented in terms of adjustments to Eq. (A2), as described in Section A.2.4.

Eq. (A2) defines t_d separately for the nolag and lag conditions. The lag case, presented second, is defined in terms of the nolag case. That is, $t_{d,\ \mathrm{lag}}$ is simply L seconds greater than $t_{d,\ \mathrm{nolag}}$, to account for the lag that occurred between the current trial and the previous trial (or, for the post-interruption trial, between the interruption and the pre-interruption trial).

The nolag case is defined by three clauses, each of which applies to a separate set of predecessors. The first clause applies to all predecessors when the current trial is the post-interruption trial (position = 1). On this trial, $pred_1$ is from the pre-interruption trial, so is $R+E$ seconds old, and $pred_2$ is from the trial before that, so is older than $pred_1$ by the average time between trials, which is $R+L/2$ because the lag occurs after half of all trials on average.

The second clause of the nolag case applies to predecessors that follow the interruption ($d <$ position) and the predecessor from the pre-interruption trial ($d =$ position), when the current trial is a baseline trial (position ≥ 2). For example, for position = 2, $pred_1$ is from the post-interruption trial and $pred_2$ is from the pre-interruption trial, and by assumption $pred_2$ was rehearsed during the interruption. Thus, the age of $pred_1$ is R, and by assumption $pred_2$ is older than this by the average time between trials, $R+L/2$, not by the duration of the interruption, I.

The third clause of the nolag case applies to the predecessors not covered by the second clause, namely those preceding the pre-interruption trial ($d >$ position), when the current trial is a baseline trial (position ≥ 2). For example, for position = 2, $pred_3$ is from the pre-pre-interruption trial, and its age is $R+2(R+L/2)+I = 3R+L+I$, where $3R$ is the total response latency for 3 consecutive trials, L is the average time due to lags following the pre-pre-interruption and pre-interruption trials (2 opportunities × .5 probability for each), and I is the interruption duration.

Under Eq. (A2), the predecessor from the pre-interruption trial will generally get younger from position 1 to position 2, instead of older, because for position 2 we assume the pre-interruption predecessor was rehearsed during the interruption, and represent the effect of this rehearsal in terms of reduced age. This manipulation of age loads the age construct with multiple meanings, but is a tractable way to include some representation of rehearsal in a closed-form model without extra machinery or free parameters. To make our assumptions about predecessor age and rehearsal as transparent as possible, we have posted worked examples with the supplementary materials at msu.edu/~ema/brieflags.

Eqs. (A1) and (A2) together specify the activation levels of all predecessors, which we can then use to derive retrieval probabilities during the remember stage. The probability $u(d)$ of retrieving the dth predecessor is

$$u(d) = \frac{e^{A(t_d)/s}}{\sum_{i=1}^{D} e^{A(t_i)/s}}, \tag{A3}$$

where A is activation from Eq. (A1) and $s = \sqrt{6}\,\sigma/\pi$, where σ is the standard deviation of activation noise (Anderson and Lebiere, 1998). This equation normalizes the activation of any given target by the total activation of all candidates, while amplifying activation differences with exponentiation. Greater values of s reduce the amplifying effect of the exponentiation.

In sum, then, the remember stage retrieves the predecessor that is most active at that moment. Eq. (A1) ranks predecessors by their mean activation levels, with $pred_1$ on top, $pred_2$ next, and so on. Eq. (A2) adjusts the age of predecessors, and thus the differences between them in the activation ranking—without changing the rank ordering itself—to reflect timing parameters of the procedure. Eq. (A3) factors in the effect of activation noise, which makes $u(1) < 1$. Because $pred_2$ is second in the ranking and older predecessors continue to decay, assumptions about whether predecessors decay during a lag after a trial will affect primarily the rate at which $pred_2$ intrudes and thus the rate of errors at offset -1.

A.2.2. Successor activation and retrieval probability

Successors are defined relative to the predecessor retrieved during the remember stage. The step coded in the retrieved predecessor enters the system's focus of attention, and then spreads activation to the steps that follow it in the task sequence, with the most activation spreading to the immediate successor and progressively less spreading to each later successor. We treat the step sequence as a looped chain, with each step connected to its successor by a one-directional associative link, and the last step (L) connected to the first (U).

To formalize spreading activation we assume that each step passes on a proportion g of the activation that reaches it (Anderson and Pirolli, 1984). The activation spreading to $succ_d$ is

$$B(d) = Wg^{d-1}, \tag{A4}$$

where W is the amount of activation spreading from the source (in effect the retrieved predecessor), g is the proportion of activation passed on from one link to the next, and d is the number of steps from the source to $succ_d$ in the UNRAVEL sequence.

Activation that has spread to a successor decays over time when the activation source is "turned off," as we assume happens during an interruption. To represent this decay, we assume that the decay rate at a successor is proportional to the amount of activation that has spread to that successor, so that decay can be absorbed into the W parameter. That is, after a decay period, the activation of a successor is what it would have been without that decay period but with a smaller W. Under this assumption, decay of spreading activation during an interruption can be represented with a smaller value of W for the post-interruption trial (as long as interruption effects do not spill over past the post-interruption trial, which they largely do not in this task; see Fig. 2). Accordingly, the model has two source activation parameters, W_{post} for the post-interruption trial (position 1) and W_{base} for baseline trials (positions 2 and later).

The probability $v(d)$ of retrieving the dth successor is

$$v(d) = \frac{e^{B(d)/s}}{\sum_{i=1}^{D} e^{B(i)/s}}, \tag{A5}$$

where B is from Eq. (A4) and s is the same activation noise parameter as in Eq. (A3). Eq. (A5) transforms the activation

ranking produced by Eq. (A4) into retrieval probabilities just as Eq. (A3) transforms the activation ranking produced by Eq. (A1).

A.2.3. Predicted sequence error rates

Eqs. (A3) and (A5) define the retrieval probabilities of predecessors and successors, respectively. It remains to transform these retrieval probabilities to predicted error proportions at different levels of the offset factor.

The probability of selecting the step at a given offset (including the correct step) is the sum of the probabilities of different possible "paths" through predecessor/successor pairs. For example, the most likely path to selection of the correct next step is through retrieval of $pred_1$ followed by retrieval of $succ_1$. Alternatively, the remember stage may retrieve $pred_2$ and the advance stage may then retrieve $succ_2$, with the second retrieval canceling the error in the first.

Each step of the UNRAVEL sequence is represented once as a predecessor and once as a successor, so that in principle any step can be the product of either stage and contribute to errors at any offset. Thus, the index variable d for $pred_d$ and $succ_d$ ranges from 1 to $D=7$. To illustrate, if the system has just performed step U, then $pred_1$ would be U and $pred_7$ would be L. If $pred_1=U$ is retrieved during the remember stage, then $succ_1$ would be N and $succ_7$ would be U. Larger values of d "wrap" in the mapping to the offset of the behavioral error. For example, $pred_1$ followed by $succ_4$ generates a $+3$ error, but $pred_1$ followed by $succ_5$ generates a -3 error, by $succ_6$ a -2 error, and by $succ_7$ a -1 error.

An important simplifying assumption concerning predecessors is that the system has performed the last D steps correctly. As we noted in Section 5.4, ideally we would have represented the possibility of second-order errors in which a sequence error occurred on a previous trial and the control code from that trial is retrieved in error during the remember stage on the current trial. However, we found that in a closed-form model it was not tractable to include all such contingencies, so we represent only the most likely scenario as context for performance on the current trial, namely that previous trials were performed correctly.

Under these operating principles, the probability of selecting the correct next step is the sum of the probabilities of the different possible predecessor/successor pairs—that is,

$$p(0) = \sum_{d=1}^{D} u(d)v(d), \tag{A6}$$

where 0 signifies the correct next step (i.e., 0 steps skipped) and d is the distance skipped backward in the remember stage and forward in the advance stage. The summation over D terms reflects the fact that each step is represented once as a predecessor and once as a successor.

The probability of selecting an incorrect step—that is, the probability of a behavioral sequence error—follows the same logic. For example, the most probable path to a $+1$ error is through $pred_1$ and $succ_2$, but another path is through $pred_2$ and $succ_3$. The total probability is $p(+1)=u(1)v(2)+u(2)v(3)+\ldots+u(D)v(1)$, where the last term is for the path through $pred_D$ and $succ_1$ (e.g., if the just-performed step was U, then $pred_D$ is N and its $succ_1$ is R). Similarly, the total probability of a -1 error is $p(-1)=u(2)v(1)+u(3)v(2)+\ldots+u(1)v(D)$. Generalizing the algebra, the probability of a $+n$ error is

$$p(+n) = \sum_{d=1}^{D} u(d)v\left[f(n,d,D)\right] \tag{A7}$$

and the probability of a -n error is

$$p(-n) = \sum_{d=1}^{D} u\left[f(n,d,D)\right]v(d), \tag{A8}$$

where $f(n,d,D)=1+\left[(d-1+n) \bmod D\right]$. The probabilities $p(-3)$, $p(-2)$, $p(-1)$, $p(+1)$, $p(+2)$, and $p(+3)$ are the model values for the proportion of sequence errors at offsets -3, -2, -1, $+1$, $+2$, and $+3$, respectively.

Although the model includes all possible paths for all offsets, the most probable paths are those through either $pred_1$ or $succ_1$, and these drive the model's account of empirical data. For example, if the remember stage yields $pred_2$, the advance stage is most likely to yield $succ_1$, so the most likely behavioral outcome is a -1 error, as we have addressed at length. Thus, in discussing the different model variants in Section 4, which effectively differed in the probability of $pred_2$ intruding after a lag, we focused on predicted changes in the error rate at offset -1 because those are most likely.

A.2.4. Adjustments for the no decay and partial decay models

The no decay and partial decay versions of the model involve adjustments to Eq. (A2). In the no decay version (Section 4.3), we delayed by time L the moment at which $pred_1$ starts to decay, so it remains fully activated during the lag. Two adjustments were necessary for Eq. (A2). First, $pred_1$ in the lag condition is L seconds younger here than in Eq. (A2), which is to say that $t_{1,\,\mathrm{lag}} = t_{1,\,\mathrm{lag,\ Eq.\ 2}} - L = t_{1,\,\mathrm{nolag,\ Eq.\ 2}}$. Second, for each $d>1$, $pred_d$ in each timing condition is $L/2$ seconds younger here than in Eq. (A2), or $t_d = t_{d,\,\mathrm{Eq.\ 2}} - L/2$, because half the time the trial governed by $pred_d$ was followed immediately by a lag, during which $pred_d$ did not decay.

In the partial decay version (Section 4.4), $pred_1$ decays during the lag by an amount determined by a new free parameter Y that represents the delay in onset of decay (in seconds). There were again two adjustments necessary for Eq. (A2). First, in the lag condition, $pred_1$ is Y seconds younger in this model than in Eq. (A2), or $t_{1,\,\mathrm{lag}} = t_{1,\,\mathrm{lag,\ Eq.\ 2}} - Y$. Second, for each $d>1$, $pred_d$ in each timing condition is $Y/2$ seconds younger here than in Eq. (A2), or $t_d = t_{d,\,\mathrm{Eq.\ 2}} - Y/2$, because half the time the trial governed by $pred_d$ was followed immediately by a lag, during which $pred_d$ started to decay after Y seconds.

A.3. Modeling Procedure and parameter estimates

We used maximum likelihood estimation to fit the model to the data. The likelihood function was the binomial distribution,

$$Likelihood = \binom{n}{k} p^k (1-p)^{n-k}, \tag{9}$$

where k is the number of sequence errors, n is the number of trials, and p is the probability of an error predicted by Eqs. (A7) or (A8).

We fit each model to each participant's data. To do this we estimated five parameters (E, s, W_{post}, W_{base}, and g) from 72 observations, one per cell of the 2 (timing) × 6 (position) × 6 (offset) design. For two of the models we estimated a sixth parameter (either Y or a second s).

The estimation procedure was the non-linear optimization method in the Solver add-in in Microsoft Excel. We used Excel for Mac 2004, running under the Rosetta emulation included with Mac OS 10.6, because in the version of Excel current at time of writing (Excel for Mac 2011) the Solver does not work correctly when called from a loop in Visual Basic.

Table A1 gives mean estimated values of free parameters for the four models. The s parameter (both s parameters, in the model with two) was constrained to take values no smaller than 0.001; without this bound the estimation process would in a few cases generate $s=0$, for which the model is undefined. The other parameters were constrained to take only non-negative values. The seed values for the free parameters for each model for each individual participant were the maximum-likelihood estimates derived by fitting that model to the aggregate data.

The response latency parameter R ($M=2.093$ s) was bound for each participant with the mean of $5\times2=10$ median response latencies on correct trials for that participant, one for each of

positions 2–6 in each timing condition. The interruption duration parameter I (M=20.424 s) was bound for each participant with the median of all that participant's interruption durations.

References

Altmann, E.M., 2004. The preparation effect in task switching: Carryover of SOA. Memory & Cognition 32, 153–163.

Altmann, E.M., 2013. Fine-grain episodic memory processes in cognitive control. Zeitschrift für Psychologie 221, 23–32.

Altmann, E.M., Gray, W.D., 2008. An integrated model of cognitive control in task switching. Psychological Review 115, 602–639.

Altmann, E.M., Trafton, J.G., 2002. Memory for goals: An activation-based model. Cognitive Science 26, 39–83.

Altmann, E.M., Trafton, J.G., 2007. Timecourse of recovery from task interruption: Data and a model. Psychonomic Bulletin & Review 14, 1079–1084.

Altmann, E.M., Trafton, J.G., Hambrick, D.Z., 2014. Momentary interruptions can derail the train of thought. Journal of Experimental Psychology: General 143, 215–226.

Anderson, J.R., 2007. How can the human mind occur in the physical universe?. Oxford University Press, New York.

Anderson, J.R., Lebiere, C., 1998. The atomic components of thought. Erlbaum, Hillsdale, NJ.

Anderson, J.R., Matessa, M., 1997. A production system theory of serial memory. Psychological Review 104, 728–748.

Anderson, J.R., Pirolli, P., 1984. Spread of activation. Journal of Experimental Psychology: Learning, Memory, and Cognition 10, 791–798.

Baddeley, A.D., Thomson, N., Buchanan, M., 1975. Word length and the structure of short-term memory. Journal of Verbal Learning and Verbal Behavior 14, 575–589.

Barrouillet, P., Bernardin, S., Camos, V., 2004. Time constraints and resource sharing in adults' working memory spans. Journal of Experimental Psychology: General 133, 83–100.

Botvinick, M.M., Bylsma, L.M., 2005. Distraction and action slips in an everyday task: Evidence for a dynamic representation of task context. Psychonomic Bulletin & Review 12, 1011–1017.

Brown, G.D.A., Preece, T., Hulme, C., 2000. Oscillator-based memory for serial order. Psychological Review 107, 127–181.

Brumby, D.P., Cox, A.L., Back, J., Gould, S.J.J., 2013. Recovering from an interruption: Investigating speed-accuracy trade-offs in task resumption behavior. Journal of Experimental Psychology: Applied 19, 95–107.

Byrne, M.D., Bovair, S., 1997. A working memory model of a common procedural error. Cognitive Science 21, 31–61.

Carlson, R.A., Cassenti, D.N., 2004. Intentional control of event counting. Journal of Experimental Psychology: Learning, Memory, and Cognition 30, 1235–1251.

Carpenter, P.A., Just, M.A., Shell, P., 1990. What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. Psychological Review 97, 404–431.

Cooper, R.P., Schwartz, M.F., Yule, P., Shallice, T., 2005. The simulation of action disorganisation in complex activities of daily living. Cognitive Neuropsychology 22, 959–1004.

Cowan, N., 1999. An embedded-process model of working memory. In: Miyake, A., Shah, P. (Eds.), Models of working memory: Mechanisms of active maintenance and executive control. Cambridge University Press, Cambridge, pp. 62–101.

D'Agostino, R.B., Stephens, M.A., 1986. Goodness-of-fit techniques. Marcel Dekker, New York.

Dell, G.S., Burger, L.K., Swec, W.R., 1997. Language production and serial order: A functional analysis and a model. Psychological Review 104, 123–147.

Dodhia, R.M., Dismukes, K., 2009. Interruptions create prospective memory tasks. Applied Cognitive Psychology 23, 73–89.

Duncan, J., 2010. How intelligence happens. Yale University Press, New Haven.

Foroughi, C. K., Werner, N. E., Nelson, E. T., & Boehm-Davis, D. A. (in press). Do interruptions affect quality of work? *Human Factors*.

Frankel, F., Levine, M., Karpf, D., 1970. Human discrimination learning: A test of the blank-trials assumption. Journal of Experimental Psychology 85, 342–348.

Gray, W.D., 2000. The nature and processing of errors in interactive behavior. Cognitive Science 24, 205–248.

Hambrick, D. Z., & Altmann, E. M. (in press). The role of placekeeping ability in fluid intelligence. *Psychonomic Bulletin & Review*.

Hodgetts, H.M., Jones, D.M., 2006. Interruption of the Tower of London task: Support for a goal activation approach. Journal of Experimental Psychology: General 135, 103–115.

Li, S.Y.W., Blandford, A., Cairns, P., Young, R.M., 2008. The effect of interruptions on postcompletion and other procedural errors: An account based on the activation-based goal memory model. Journal of Experimental Psychology: Applied 14, 314–328.

Monk, C.A., Trafton, J.G., Boehm-Davis, D.A., 2008. The effect of interruption duration and demand on resuming suspended goals. Journal of Experimental Psychology: Applied 14, 299–313.

Oberauer, K., Lewandowsky, S., 2011. Modeling working memory: A computational implementation of the Time-Based Resource-Sharing theory. Psychonomic Bulletin & Review 18, 10–45.

Reason, J., 1990. Human Error. Cambridge University Press, New York, NY.

Reitman, J.S., 1974. Without surreptitious rehearsal, information in short-term memory decays. Journal of Verbal Learning and Verbal Behavior 13, 365–377.

Salvucci, D., Monk, C.A., Trafton, J.G., 2009. A process-model account of task interruption and resumption: When does encoding of the problem state occur? *Proceedings of the Human Factors and Ergonomics Society 53rd Annual Meeting*. Human Factors and Ergonomics Society, Santa Monica, pp. 799–803.

Trafton, J.G., Altmann, E.M., Brock, D.P., Mintz, F.E., 2003. Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. International Journal of Human-Computer Studies 58, 583–603.

Trafton, J.G., Altmann, E.M., Ratwani, R., 2011. A memory for goals model of sequence errors. Cognitive Systems Research 12, 134–143.

Wagenmakers, E.-J., 2007. A practical solution to the pervasive problems of p values. Psychonomic Bulletin & Review 14, 779–804.

Wickelgren, W.A., 1977. Speed-accuracy tradeoff and information processing dynamics. Acta Psychologica 41, 67–85.