ISSN: 0278-7393

2021, Vol. 47, No. 9, 1403–1438 https://doi.org/10.1037/xlm0001052

Previously Retrieved Items Contribute to Memory for Serial Order

Dakota R. B. Lindsey¹ and Gordon D. Logan² ¹Department of Psychology, University of South Alabama ²Department of Psychology, Vanderbilt University

It has long been understood that associations can form between items that are paired (Ebbinghaus, 1885), but it is commonly assumed that previously retrieved items are not used when remembering items in serial order. We present a series of experiments that test this assumption, using a serial learning procedure inspired by Ebenholtz (1963). In this procedure, participants practiced recalling ordered lists of letters, and the order of the letters was manipulated. Half of the lists were scrambled such that the serial positions and relative positions of the letters were inconsistent over practice. The other half of the lists were spun (e.g., ABCDEF \rightarrow FABCDE), making the serial positions inconsistent but preserving the relative positions of the letters over practice. In Experiment 1, participants learned to recall spun lists more accurately than the scrambled lists with practice. In Experiments 2 and 3, participants recalled new lists more accurately when they shared the relative order of previously learned spun lists. In Experiments 4 and 5, the influences of motor and perceptual representations were removed and shown to have little impact on the advantage for spun lists. Experiment 6 extended our findings to the more traditional Hebb (1961) learning procedure. The results of our experiments indicate that the commonly held assumption is incorrect-previously retrieved items can contribute to memory for serial order. Previously retrieved items best serve serial memory when there is ample opportunity to strengthen itemto-item associations.

Keywords: associative chaining, Hebb effect, serial learning, serial order, serial recall

The problem of serial order (Lashley, 1951) has been a central issue in psychology for more than a century (Ebbinghaus, 1885). Its relevance extends multiple domains, from memory (Henson, 1998; Logan, 2021) to skilled performance (Logan, 2018) to speech (Dell, 1986). One of the most contentious debates—particularly in the memory domain—concerns whether previously retrieved items aid in the reproduction of a sequence. This debate ultimately concerns the nature of retrieval cues used in serially ordered behavior—can an item contribute to the retrieval cue, and if it can, when can it do so?

The purpose of the present work was to seek direct evidence that previous items contribute to the retrieval cue in serial memory. We will present a series of experiments that combine *serial learning* with serial recall. Participants are asked to recall short lists of letters in their presented order, and we manipulate the repeated features of the lists to observe how they influence the rate of learning. Our experiments focus on the relative order of items as a feature that might affect learning rate. Using a variant of the *spin list procedure* (Ebenholtz, 1963; Kahana et al., 2010; Lindsey & Logan, 2019), we compare the rate of learning *spun lists*, in which the relative order of list items is consistent over practice, with the rate of learning *scrambled lists*, in which relative order is not consistent. If an item can contribute to the retrieval cue, then learning should benefit from the consistent relative order of items in spun lists. Our experiments are designed to rule out alternative explanations and to link our findings to more commonly used tasks (i.e., serial recall and Hebb repetition procedures).

Although we do not explicitly test any theory of serial memory, a primary goal of this article is to inform future modeling efforts, so we will frequently discuss the ability of certain theories or theoretical mechanisms to capture our data. In these discussions, we categorize theories of serial memory by the cues they assume to contribute to retrieval. Likewise, we categorize the cues by whether or not they incorporate features of previously retrieved items. Retrieval cues that incorporate previously retrieved items are *item-dependent retrieval cues*, and those that do not are *item-independent retrieval cues*. Any theory that assumes the use of item-dependent retrieval cues whether in addition to or in the absence of item-independent retrieval cues—is an *item-dependent retrieval theory*. Any theory

This article was published Online First October 28, 2021. Dakota R. B. Lindsey () https://orcid.org/0000-0002-4852-4570

This research was supported by National Eye Institute Grant R01 EY025275. The data presented in this article were collected for Dakota R. B. Lindsey's dissertation under the direction of Gordon D. Logan. We are grateful to Jeff Annis and the members of the dissertation committee—Lisa Fazio, Sean Polyn, Duane Watson, and Geoff Woodman—for providing helpful comments while conducting this research. We also thank Geoff Ward, Simon Dennis, and Simon Fischer-Baum for their valuable commentary throughout the review process. The data and analysis files are available on the Open Science Framework (https://osf.io/yeku2/).

Correspondence concerning this article should be addressed to Dakota R. B. Lindsey, Department of Psychology, University of South Alabama, 307 North University Boulevard, Mobile, AL 36688, United States. Email: drlindsey@southalabama.edu

that strictly assumes the use of item-independent retrieval cues is an *item-independent retrieval theory*.

At their simplest, item-dependent retrieval theories use associative chaining (Dennis, 2009; Ebbinghaus, 1885; Lewandowsky & Murdock, 1989; Solway et al., 2012): Pairs of items are bound by itemto-item associations, and individual items-specifically the one most recently retrieved-serve as retrieval cues. More complex item-dependent retrieval theories incorporate multiple items into the retrieval cue (sometimes in addition to other contextual elements that are present at the time of retrieval), establishing a running history of all the items retrieved so far (Botvinick & Plaut, 2006; Logan, 2021; Murdock, 1995).¹ Item-independent retrieval theories have used various types of item-independent cue: Some cue memory with an abstract marker of serial position (e.g., Conrad, 1965; Farrell & Lewandowsky, 2013; Henson, 1998), others cue memory with an index of time or temporal context (e.g., Brown et al., 2000; Brown et al., 2007), and still others cue memory with a representation of the sequence that activates all list items in parallel (Bryden, 1967; Estes, 1972; Page & Norris, 1998). Many of these theories assume that an item is suppressed after it is retrieved, so it competes less on subsequent retrieval attempts. These theories allow previous items to influence recall indirectly by removing retrieved items from the competition; however, they are united in the assumption that previously retrieved items do not contribute to retrieval cues directly.

The predictions of item-independent theories are frequently supported in "standard" serial recall tasks, in which lists are generated from a small set of items (a *closed set*), the positions of list items are randomized, and the length of the lists is short (Farrell et al., 2013; Osth & Dennis, 2015). The predictions of item-dependent retrieval theories are often not supported by these tasks (but see Fischer-Baum & McCloskey, 2015), and they are widely disregarded as a result. However, item-dependent theories garner more support in "nonstandard" serial recall tasks (those that use longer lists of items, or those that use an *open set*—a nonrepeating set of items) and serial learning tasks (Kahana et al., 2010; Solway et al., 2012).

This discrepancy seems to imply that different mechanisms are tapped in serial recall and serial learning tasks: Item-independent retrieval cues are used in novel recall scenarios (serial recall), and item-dependent retrieval cues are used in highly practiced recall scenarios (serial learning). However, some theories—for example, the item-independent retrieval theories proposed by Farrell (2012) and by Page and Norris (2009)—have successfully captured key phenomena in both serial recall and serial learning tasks. The two types of task are ultimately measuring the same serial memory system, and current evidence seems to suggest that this system uses item-independent cues.

There is growing acceptance of the idea that item-dependent cues contribute to serial memory retrieval in addition to item-independent cues (Caplan, 2015; Fischer-Baum & McCloskey, 2015; Kahana et al., 2010; but see Farrell et al., 2013; Osth & Dennis, 2015). The evidence for item-dependent cues is not isolated to nonstandard serial recall tasks and serial learning tasks. For example, in a standard serial recall task, Fischer-Baum and McCloskey (2015) found that an item involved in a *prior-list intrusion*—a type of error in which an item is incorrectly recalled from a previously seen list—is sometimes reported adjacent to an item it was previously presented alongside. They interpreted this as evidence

that the coding of an item's list position incorporates the surrounding list items.

The current experiments sought direct evidence of item-dependent cues by adapting the spin list procedure to serial recall. The original spin list procedure, created by Ebenholtz (1963), compared the rate of learning in spun lists (in which letters preserved their relative positions from trial to trial, but not their serial positions; ABCDEF \rightarrow FABCDE) with *same lists* (in which letters preserved their relative positions and serial positions; ABCDEF \rightarrow ABCDEF; also see Kahana et al., 2010) to determine whether serial learning depends on serial position cues. Serial position is consistent in same lists and inconsistent in spun lists. Same lists are learned faster, so serial memory depends on consistencies in serial position and uses item-independent position codes as cues for retrieval (Ebenholtz, 1963).

Kahana et al. (2010) emphasized that even though learning was slower in spun lists, it was still substantial. They attributed this improvement to the strengthening of item-to-item associations: The relative positions of items were consistent, so associations between items had the opportunity to strengthen. However, the original spin list procedure does not have an adequate control list to which learning on spun lists can be compared, so it is not clear how much of the learning observed in their spun lists can be attributed to item-dependent cues.

The current experiments use Ebenholtz's logic to design a better test of item-dependent cues. We compare the rate of learning spun lists to *scrambled lists* that lack consistency in both serial position and relative position. If people are slower to learn scrambled lists than spin lists, then serial memory is sensitive to relative position, and the retrieval cue is item-dependent.

Experiment 1

Experiment 1 aims to establish the foundational effect of the paper: better learning for spun lists. We compare performance on spun lists to performance on scrambled lists. The only structural difference between these list types is relative item order: In spun lists it is consistent, and in scrambled lists it is not. This difference is key—consistent relative order can be thought to strengthen the association between a cue and an item (from the perspective of associative chains of single items; Lewandowsky & Farrell, 2008; Solway et al., 2012) or to generate multiple similar cues (or contextual states) with which an item can be retrieved (from the perspective of complex cues with shared runs of items; Logan, 2021; Murdock, 1995). If item-dependent retrieval cues are used by the serial memory system, then the consistent relative order of items in spun lists should lead to more accurate retrieval of items in those

¹ There is variation in how item-dependent cues are used to retrieve the next response. In some theories, the cue retrieves the next item via association (Lewandowsky & Murdock, 1989; Murdock 1995; Solway et al., 2012). Other theories treat the cue as a context representation and retrieve the next item via the similarity between the current context and the context in which the next item was encoded (Logan, 2021; also see Polyn et al., 2009). This variation can lead to important differences in predictions among theories that use item-dependent cues. For example, in association-based retrieval, an item can only retrieve an item with which it has established an association (by presenting the items together in space or time). In similarity-based retrieval, any item can be retrieved if the appropriate context can be reinstated.

lists. Owing to the novelty of the presented lists and the possibility that other sources of information (e.g., inconsistent position-based cues) may interfere at retrieval, we expect this advantage to emerge with practice as a learning advantage for spun lists. If spun lists are not learned any more quickly than scrambled lists, then item-dependent retrieval cues are not used by the serial memory system.

Method

The current study was reviewed and approved by the Institutional Review Board at Vanderbilt University.

Participants

We tested 24 participants, consistent with previous experiments that used a very similar procedure to study serial order in copy typing (Lindsey & Logan, 2019). The participants were native English speakers between the ages of 18 and 35 who reported normal or corrected-to-normal vision and could type at least 40 words per minute (WPM). In the typing test, participants typed 72.52 WPM and made no errors on 93.15% of the words. The participants were tested in 1.5 hr timeslots and received \$18 or course credit as compensation.

Apparatus and Stimuli

We used E-prime 2.0 (Psychology Software Tools, 2012) to present stimuli, record responses, and record the accuracy and timing of responses. The task was administered on ASUS M32BF desktop computers with BenQ XL2411Z flat screen monitors. Responses were taken from standard QWERTY keyboards with rubber dome switches. Only the letter keys and spacebar were enabled.

Participants remembered lists of six lowercase letters. The lists were created by selecting letters from one of a few closed sets. The sets were generated by sampling randomly from the alphabet without replacement, thereby ensuring that there was no overlap among them. The letters a and e were removed from selection to reduce the chance of producing word-like lists. We generated six lists for each of the sets, ensuring that each letter occupied each position in the list once.

 Table 1

 Construction of Letter Sequences in Experiment 1

For half of the sets, we laterally shifted each of the six letters to produce spun lists (Ebenholtz, 1963; Kahana et al., 2010). In spun sets, letters were moved to new serial positions in each list, but they appeared adjacent to the same letters. Serial positions were inconsistent, but relative positions were consistent. For the other half of the sets, we inserted the letters into a balanced 6×6 Latin square to produce scrambled lists. The balanced Latin square ensured that each letter appeared in each serial position, and it ensured that each letter was preceded by and followed by every other letter in the set. Serial positions and relative positions were inconsistent. Both of the order manipulations generated six lists for each set. The ordering manipulation was done within subject, so spun lists and scrambled lists were practiced by every participant. Table 1 shows example lists that a participant could have seen in this experiment.

Procedure

Prior to beginning the memory task, we asked participants to type a paragraph about Border Collies (Crump & Logan, 2013) to measure their copy typing speed and accuracy. The instructions for the memory task informed participants that a list of letters would briefly be presented on the screen, that they should memorize the list because they would be asked to type it, that they should type as quickly and accurately as possible, that they should guess if they could not remember the letter that occupied a position, and that mistakes could not be corrected because the backspace key was disabled.

Participants completed many study-test trials. At the beginning of each trial, a fixation cross was presented in the center of the screen for 500 ms. The fixation cross disappeared and was replaced by the list of letters. The letters appeared simultaneously in a single row with no spaces (e.g., ABCDEF), and they remained on screen for 500 ms. A 500-ms blank screen separated the end of the list presentation and the response screen. The text "Response:," presented in the center of the screen, cued participants to type the letters they could remember from the most recently presented list. Letter keys that they pressed were echoed on the screen under the response cue, so participants could see their prior responses.

List			Spu	n Set			List			Scram	oled Set		
First half													
List 1	t	р	g	u	b	k	List 7	q	f	W	m	s	1
List 2	р	g	ů	b	k	t	List 8	ŕ	m	q	1	W	s
List 3	g	u	b	k	t	р	List 9	m	1	ŕ	S	q	W
List 4	u	b	k	t	р	g	List 10	1	S	m	W	f	q
List 5	b	k	t	р	g	u	List 11	s	W	1	q	m	f
List 6	k	t	р	g	ŭ	b	List 12	W	q	S	ŕ	1	m
Second half			1	e					1				
List 1	y	Z	с	n	r	d	List 7	Х	h	0	i	i	v
List 2	z	с	n	r	d	y	List 8	h	i	Х	v	0	i
List 3	с	n	r	d	y	z	List 9	i	v	h	i	Х	Ő
List 4	n	r	d	y	z	с	List 10	v	i	i	0	h	х
List 5	r	d	y	z	с	n	List 11	i	0	v	Х	i	h
List 6	d	У	Z	с	n	r	List 12	0	х	j	h	v	i

Note. Each participant received a different random selection of letters. When presented on the computer screen, the spaces between letters were removed.

Participants submitted their responses by pressing the spacebar, which cleared the screen and initiated the next trial. A 1-s blank screen separated the spacebar press and the fixation cross on the next trial.

In each half of the experiment, participants saw only 12 lists six from one of the spun list sets and six from one of the scrambled list sets—and each was presented 40 times. Participants were not explicitly informed about the switch between letter sets in the first and second halves of the experiment. Lists were practiced in repetition blocks: No list could be presented again until all of the other lists had been presented. Within a repetition block, lists were selected randomly without replacement. Participants completed a total of 960 trials—480 in each half of the experiment. Self-paced breaks were allowed every 240 trials (six repetition blocks; approximately 20 minutes between breaks).

After finishing the memory task, we asked participants what they thought of the task, what strategies they used to remember the letters, and whether they noticed any patterns in the order that the letters appeared. We categorized participants based on their responses on this survey: those who detected that some of the lists had consistently ordered items, and those who did not.

Analyses

We recorded accuracy and latency measures on each trial, and we report three measures of performance: error rate (the percentage or proportion of erroneous responses, averaged over the six list positions), initiation time (IT; the latency from the onset of the response cue to the first letter response in ms), and interresponse time (IRT; the average latency between letter responses in ms). Errors were scored using strict scoring; a response was scored as correct when the reported item was in the most recently presented list and it was reported in its original serial position. We excluded responses that took longer than 3 s from calculations of average IT and IRT. To conduct statistical analyses, we calculated averages for each of the dependent measures for each participant, list type, and block in an experiment. We computed the difference in error rate between spun and scrambled lists, averaged over the all of experimental blocks, and then we checked for outliers using the Tukey (1977) interquartile range method. We planned to use robust statistical procedures if any outliers were detected. No outliers were detected in this experiment, so we used traditional, nonrobust procedures.

We focus our discussion on error rate. The timing measures are presented primarily to diagnose whether improvements in error rate are the result of intentional slowing (i.e., a speed-accuracy tradeoff). IT and IRT are presented separately because they represent different types of slowing. Longer IT may reflect additional rehearsal after the onset of the response cue or the creation of a response plan (Miller et al., 1960; Rosenbaum et al., 1983). Longer IRT may reflect posterror slowing or in-the-moment adjustments to the response plan (Rosenbaum et al., 1984).

If item-dependent cues are used, we should see a spun list advantage in error rate. We tested for this advantage in two ways. First, we tested the overall difference between spun and scrambled lists with an ANOVA main effect analysis of list type. The main effect analyses were accompanied by t-tests comparing first block performance on the two list types to check for any preexperimental differences resulting from accidents of randomization. A list typeby-block interaction test could have been used to check for a difference in learning while also accounting for preexperimental differences. However, the significance of this test is too dependent upon how quickly performance in the two list types diverges (Lindsey & Logan, 2019).

Second, we fit power functions to the data and checked for differences in the rate of learning. Often, the improvement of performance over practice follows a power function, and the parameters of this power function can be theoretically meaningful (e.g., Logan, 1988). A major draw to this method is that it reduces the large amount of trial data (40 blocks of trials) into a few meaningful parameters. The functions we fit to the data followed the form:

$$Y = a + bX^{-c} \tag{1}$$

Y is a performance measure (error rate, IT, or IRT), *X* is the experiment block, and *a*, *b*, and *c* are parameters that determine how performance changes as block changes. The *a* parameter reflects asymptotic performance, the *b* parameter reflects how much performance can improve, and together they control the starting point of performance. The *c* parameter is the learning rate—it reflects how quickly performance improves from the starting point to peak performance, with higher *c* values indicating superior learning. We had no a priori reason to expect a starting point difference between the list types, so only the *c* parameter was allowed to differ between list type and participant, and we used a paired-samples *t* test to compare the average *c* values. Additional information about the fitting procedure can be found in Appendix A.

To better understand what information was being learned in the experiment, we analyzed changes in omission, misorder, and intrusion errors over practice. An omission was committed when no response was given for a particular position. Participants were not provided a way to leave "blank" responses in the middle of their response, so the number of omissions quantified the difference in length between the response string and target string. A misorder was committed when an item from the most recent list was reported in an incorrect position (i.e., not the serial position in which the item was presented in the most recent list). An intrusion was committed when an item that was not in the most recent list was reported. We repeated the main effect analyses on list type for each of the error type counts. If item-dependent cues are used, we should see differences in the number of misorders committed (perhaps in addition to advantages in other error types), with fewer misorders being committed in spun lists.

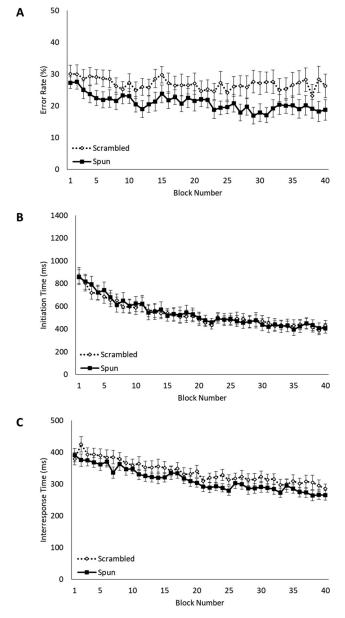
Finally, we were interested in whether detection of the order manipulation affected the spun list advantage. We computed the advantage for each participant by subtracting their average scrambled list error rate from their average spun list error rate. We conducted an independent samples t test on the advantage scores using detection of structural differences as the grouping variable.

Results

Error Rate

We averaged performance in the two halves of the experiment because they were identical in structure (e.g., block 1 spun list IT in the first half was averaged with block 1 spun list IT in the second half). Mean error rate is plotted over block in of Figure 1A.

Experiment 1: Mean Error Rate (A), Initiation Time (IT; B), and Interresponse Time (IRT; C) for Each List Type as a Function of the Presentation Number of the List (Block Number)



Note. The bars are standard errors of the means.

Result tables are in Appendix B: *F* tests obtained from the 2 (list type) \times 40 (repetition block) ANOVA (for mean error rate, IT, and IRT) are in Table B1; *t* tests are in Table B2; and *F* tests of error types are in Table B3. For convenience, test statistics, error terms, and *p* values are printed in the text.

Participants recalled spun lists more accurately with practice (error rate decreased 8.6% from block 1 to block 40), supported by a significant simple main effect of presentation number on error rate (F[39, 897] = 2.898, MSE = .005, p < .001). Learning in scrambled lists

was marginally significant (error rate decreased 3.8%; *F*[39, 897] = 1.364, *MSE* = .005, *p* = .070). Error rates were lower for spun lists than for scrambled lists (21.0% vs. 26.9%), supported by a significant main effect of list type, *F*(1, 23) = 7.924, *MSE* = .211, *p* = .010. This difference was not a consequence of an initial difference in performance because first block performance was not significantly different between the list types (spun: 27.3%, scrambled: 30.1%; *t*[23] = -1.080, *SE* = .026, *p* = .291). Participants learned the spun lists more quickly than the scrambled lists.

Error Rate: Error Types

Figure 2 shows the rates of omissions, intrusions, and misorders. In scrambled lists, error rate decreased with practice because of a reduction in the number of intrusions, F(39, 897) = 3.210, MSE = 3.830, p < .001—the frequency of omissions and misorders did not change over practice. In spun lists, the frequency of misorders and intrusions decreased with practice (misorders: F[39, 897] = 1.895, MSE = 19.135, p = .001; intrusions: F[39, 897] = 2.300, MSE = 3.830, p < .001). Spun lists and scrambled lists differed only in the frequency of misorders: list items were reported in incorrect positions less often in spun lists (spun: 13.5%, scrambled: 18.3%; F[1, 23] = 13.433, MSE = 430.985, p = .001).

Error Rate: Power Function Analysis

Power functions did not fit participant data well—average R^2 across the 24 participants was .299 (RMSD = .81). We suspect the issue was that we had very few trials per list type per block—only 12 trials per participant—and thus a large amount of variability in the means of each block. In spite of the poor overall fits, the parameter estimates obtained from the power functions were useful for assessing within-subject differences in learning between the list types. The shared *b* parameter, averaging over the estimates for individual participants, was 32.4; the power functions estimated that, on average, 32.4% of responses were erroneous at the start of the experiment. The *c* parameter—the learning rate—was significantly higher for spun lists in terms of error rate (spun: .181, scrambled: .089; t[23] = 2.807, SE = .033, p = .010).

Initiation Time

Mean IT is shown in Figure 1B. Participants initiated retrieval more quickly over practice in both types of list (spun: F[39, 897] = 29.198, MSE = 11,820, p < .001; scrambled: F[39, 897] = 25.068, MSE = 11,820, p < .001). Initiation time was not significantly different between the list types (spun: 537.1 ms, scrambled: 531.7 ms; F[1, 23] = .013, MSE = 1.037,908, p = .909).

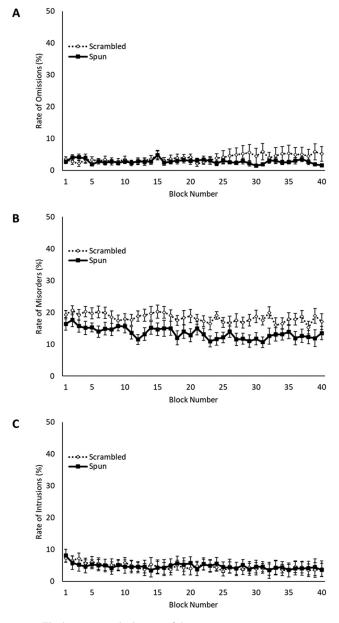
Initiation Time: Power Function Analysis

Average R^2 was .547 (RMSD = 12.752), and the average shared *b* parameter was 811.6. Learning rate was not significantly different between the two list types in IT (spun: .304, scrambled: .296; *t* [23] = .224, *SE* = .034, *p* = .825).

Interresponse Time

Mean IRT is shown in Figure 1C. Participants recalled the letters in both types of list more quickly with practice (spun: F[39, 897] = 16.000, MSE = 1,889, p < .001; scrambled: F[39, 897] = 14.416, MSE = 1,889, p < .001). Retrieval of letters was numerically but not

Experiment 1: Rate of Omissions (A), Misorders (B), and Intrusions (C) for Each List Type as a Function of the Presentation Number of the List (Block Number)



Note. The bars are standard errors of the means.

significantly faster in spun lists than in scrambled lists (spun: 313.2 ms, scrambled: 338.4 ms; F(1, 23) = 2.184, MSE = 140,065, p = .153).

Interresponse Time: Power Function Analysis

Average R^2 was .504 (RMSD = 4.961), and the average shared *b* parameter was 288.1. The learning rate for spun lists was numerically higher in IRT, but this difference did not reach significance (spun: .228, scrambled: .178; *t*[23] = 1.285, *SE* = .039, *p* = .212).

Detection

The postexperiment interviews revealed that 17 of the 24 participants (70.8%) noticed that there was additional structure in some of the lists. The remaining seven participants (29.2%) did not detect any structural differences in the lists. The spun list advantage did not depend on detection of structural differences in the list (advantage with detection: 5.9%; advantage without detection: 5.9%), t(22) = .001, SE = .047, p = .999.

Discussion

The foundational effect of the article was obtained: With practice, participants recalled spun lists more accurately than scrambled lists. This accuracy advantage arose because of a decreased tendency to commit order errors in spun lists. The accuracy advantage does not reflect a speed-accuracy tradeoff, because spun lists were not recalled more slowly than scrambled lists. The current results support the use of item-dependent retrieval cues.

The learning we observed in scrambled lists is somewhat surprising. Even in the absence of consistent order, people are capable of improving at lists that repeat the same items. Scrambled list improvement may reflect *item-set learning*—participants may have performed better because they learned the items that made up the lists (remembering an item is a prerequisite to remembering an item in the correct position)—because lists were drawn from closed sets. It may also reflect the formation of list-level representations, or *listlevel learning*, because each scrambled list was presented and recalled multiple times.

We will discuss the possible sources of task improvement in more detail later in the general discussion. For now, we will simply point out that item-set learning and list-level learning likely occurs in both list types, and this highlights the merits of the current method. Unlike Kahana et al. (2010), we are able to distinguish learning that results from item-dependent cues from other sources of learning. Consistent relative order allows item-dependent cues to be effective in spun lists and, consequently, produces *additional* learning in spun lists, above and beyond the learning observed in scrambled lists.

Experiment 2

Experiments 2 adds a transfer test to the basic serial learning procedure introduced in Experiment 1. Participants were trained on a subset of the spun and scrambled lists and transferred to a different subset of lists. Training lists and test lists were sampled from the same item sets, so test lists were simply novel reorderings of the items seen during the training period. If item-dependent cues are used, then new spun lists in the test portion should be easier to recall than new scrambled lists. The relative order of items in test period spun lists is the same as the relative order learned in training period lists can transfer to the test period lists. The same cannot be said for test period scrambled lists, which do not preserve order of training portion scrambled lists.

Table 2	
Construction of Letter Sequences in	Experiment 2

List			Spur	n Set			List			Scramb	oled Set		
Training													
List 1	t	р	g	u	b	k	List 9	q	f	W	m	s	1
List 2	р	g	ů	b	k	t	List 10	f	m	q	1	W	s
List 3	g	ŭ	b	k	t	р	List 11	m	1	f	S	q	W
List 4	u	b	k	t	р	g	List 12	1	s	m	W	f	q
List 5	у	Z	с	n	r	d	List 13	Х	h	0	i	j	v
List 6	Z	с	n	r	d	у	List 14	h	i	х	v	0	j
List 7	с	n	r	d	у	Z	List 15	i	v	h	j	х	0
List 8	n	r	d	у	Z	с	List 16	v	j	i	0	h	х
Test				-					-				
List 1	b	k	t	р	g	u	List 5	s	W	1	q	m	f
List 2	k	t	р	g	ŭ	b	List 6	W	q	s	f	1	m
List 3	r	d	ý	Z	с	n	List 7	j	0	v	х	i	h
List 4	d	v	Z	с	n	r	List 8	0 0	х	i	h	v	i

Note. The same letters were shown in the training and test periods. Participants practiced 4 lists from each set and tested on the remaining 2 lists from each set.

Method

Participants

Twenty-four participants were recruited for this experiment. People who participated in Experiment 1 were excluded from participation in this experiment. On average, participants typed 70.95 WPM and made no errors on 93.10% of the words in the typing test. Participants were tested in 1 hr timeslots and received \$12 or course credit as compensation.

Apparatus and Stimuli

Across Experiments 2 and 3, we manipulate the number of training period and test period lists. In Experiment 2, participants practiced four lists from each set in the training period, and then tested on the remaining two lists from each set in the test period. For each set, lists 1 through 4 became training lists, and lists 5 and 6 became test lists. As a consequence of this method of selecting lists, the trained spun lists were consecutive spins of the item set (e.g., ABCDEF, BCDEFA, CDEFAB, and DEFABC). Participants practiced 24 lists—16 in the training period and eight in the test period. Table 2 presents an example set of lists.

Procedure

Training period lists were practiced 40 times each, and test period lists were practiced six times each. In both the training and test periods, lists were presented in repetition blocks. Participants completed 688 trials altogether (640 training + 48 test). Unlike Experiment 1, all four letter sets were presented in each block— the letter sets did not change halfway through the experiment. Participants were not explicitly informed that there was a training and test period or that the lists would change during the experiment. Self-paced breaks were offered every 80 trials.

Analyses

A learning advantage was assessed in the same manner as Experiment 1. Transfer was also assessed in two ways (using the ANOVA and tests on power function parameters). We conducted a main effect analysis of list type in the test period to assess overall differences in transfer. This analysis aggregated performance from all of the test period blocks, but it did not include any training period blocks. We also conducted two simple main effect analyses for each list type. One compared performance in the test period blocks to performance in (an equal number of) blocks at the *beginning* of the training period, to test for the presence of transfer (different test block performance is evidence of transfer). The other similarly compared performance in test blocks to performance in blocks at the *end* of the training period, to test for the completeness of transfer (a lack of difference in test block performance is evidence that transfer was perfect). In the text, we only report the result of the transfer presence test when there is a significant difference between the list types in the test period, and we only report the result of the transfer completeness test when the transfer presence test is significant.

Power functions (Equation 1) were fit to the test period blocks separately from the training period blocks. The starting point of performance in test blocks may differ due to transfer differences, so the *b* parameter was allowed to differ between list types in the test portion. We conducted two paired-samples *t* tests: one comparing the average *c* values (to assess different rates of learning the new lists) and one comparing the average *b* values (to assess different initial performance on the new lists). Higher *b* values and higher *c* values indicate superior transfer. The test on the *b* parameter, which assesses differences in initial test period performance, is more diagnostic of transfer differences.²

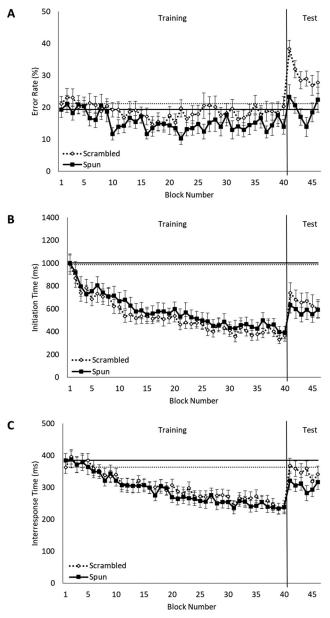
Results

Error Rate: Training

Mean error rate for each experiment block is shown in Figure 3A. Result tables (Tables B4, B5, and B6; divided in the same way as Experiment 1) are in Appendix B. Error rates reduced significantly for

² The *c* parameter test can give misleading conclusions about transfer differences. Consider a situation where no learning occurs in the test period, yet new spun lists are recalled more accurately because of an initial test period advantage. The *b* parameter test would correctly indicate a transfer difference, but the *c* parameter test would not (the *c* parameters would be nearly identical for both list types—both would be close to zero).

Experiment 2: Mean Error Rate (A), Initiation Time (IT; B), and Interresponse Time (IRT; C) for Each List Type in Each of the 46 (40 Training + Six Test) Experiment Blocks



Note. The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

both list types in the training period (spun: 5.5% reduction from block 1 to block 40, F[39, 897] = 2.514, MSE = .007, p < .001; scrambled: 1.0% reduction, F[39, 897] = 1.479, MSE = .007, p = .031). Error rate was lower for spun lists during training (spun: 15.4%, scrambled: 18.7%; F[1, 23] = 4.470, MSE = .119, p = .046), despite no significant initial difference in performance between the two list types (spun: 19.4%, scrambled: 21.2%; t[23] = -.574, SE = .032, p = .571). The

5.9% error rate advantage for spun lists observed in Experiment 1 shrank to 3.3% in this experiment. The change in experimental design may have caused this reduction. Compared with Experiment 1, participants saw fewer spun lists from each set per block (four lists vs. six), and there was more spacing between repetitions of the same list on average (16 lists between each repetition vs. 12).

Error Rate: Transfer

New spun lists were recalled more accurately than new scrambled lists (19.3% errors vs. 30.4%), supported by a significant main effect of list type in the test period, F(1, 23) = 22.480, MSE = .040, p < .001. The simple main effect analyses revealed negative transfer in error rate for scrambled lists (test: 30.4%, training start: 21.5%; F[1, 23] = 44.179, MSE = .007, p < .001) and no transfer in error rate for spun lists (test: 19.3%, training start: 19.4%; F[1, 23] = .014, MSE = .007, p = .906).

Error Rate: Error Types

Error rates split by error category are shown in Figure 4. Reductions in all error types underpinned the improvement in spun lists, whereas reductions in omissions and intrusions underpinned scrambled list improvement. The training and test differences in error rate were carried by misorder errors: Only the number of misorders differed between the list types (training: F[1, 23] = 5.538, MSE = 147.665, p = .028; test: F[1, 23] = 16.906, MSE = 17.873, p < .001).

Error Rate: Power Function Analysis

For the error rate training functions, average R^2 was .221 (RMSD = .98), and the average shared *b* parameter was 25.7. Learning rate *c* was significantly faster for spun lists in the training period (spun: .219, scrambled: .137; t[23] = 2.427, SE = .034, p = .023). For the test functions, average R^2 was .508 (RMSD = 2.44). In the test period, the starting point of error rate was lower for spun lists (spun: 25.1; scrambled: 38.2), supported by a significant difference in the *b* parameter between list types, t(23) = -4.385, SE = .030, p < .001.

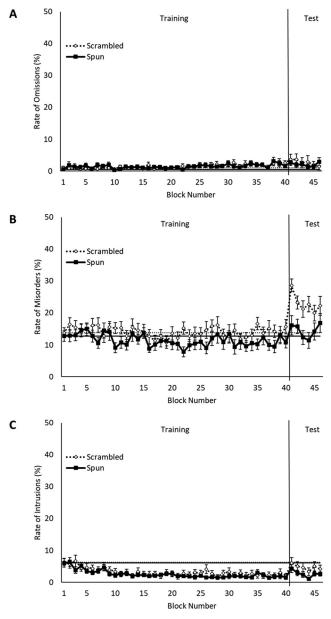
Initiation Time

Mean IT for each block is shown in Figure 3B. In the training period, participants initiated retrieval of spun lists more slowly than scrambled lists, but this difference was not significant (spun: 577.5 ms, scrambled: 527.1 ms; F[1, 23] = 1.278, MSE = 954,735, p = .270). In the test period, participants initiated retrieval of new spun lists more quickly than new scrambled lists, but this difference was also not significant (spun: 585.8 ms, scrambled: 660.6 ms; F[1, 23] = 2.049, MSE = 196,948, p = .166).

Initiation Time: Power Function Analysis

For the training functions, average R^2 was .516 (RMSD = 16.519), and the average shared *b* parameter was 958.4. Average R^2 was .421 (RMSD = 39.363) for the test functions. Training period learning rates did not differ significantly between the list types (spun: .342, scrambled: .369; t[23] = -.633, SE = .042, p = .533). However, the learning on spun lists had greater transfer: the test period starting point for IT was significantly lower for spun lists (spun: 508.2, scrambled: 646.8; t[23] = -2.554, SE = 54.279, p = .018).

Experiment 2: Rate of Omissions (A), Misorders (B), and Intrusions (C) for Each List Type as a Function of the Presentation Number of the List (Block Number)



Note. The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

Interresponse Time

Figure 3C displays mean IRT for each block. During training, the letters in spun lists were retrieved faster numerically but not significantly than the letters in scrambled lists (spun: 289.8 ms, scrambled: 299.7 ms; F[1, 23] = .427, MSE = 46,920, p = .520). Letters in new spun lists were recalled significantly more quickly than letters in new

scrambled lists (spun: 305.1 ms, scrambled: 348.7 ms; F[1, 23] = 4.916, MSE = 27,887, p = .037). Underlying the test period difference was significant positive, yet imperfect, transfer for spun lists (presence of transfer: F[1, 23] = 18.761, MSE = 17,703, p < .001; completeness of transfer: F[1, 23] = 25.333, MSE = 11,766, p < .001) and no significant transfer for scrambled lists (presence of transfer: F[1, 23] = 3.002, MSE = 17,703, p = .097).

Interresponse Time: Power Function Analysis

For the training functions, average R^2 was .484 (RMSD = 5.969), and the average shared *b* parameter was 293.7. Average R^2 was .425 (RMSD = 17.101) for the test functions. Training period learning rate was numerically but not significantly faster in IRT (spun: .360, scrambled: .314; t[23] = .893, SE = 24.317, p = .381). The starting point for IRT was numerically but not significantly lower for spun lists in the test period (spun: 187.1, scrambled: 231.1; t[23] = -1.741, SE = 25.270, p = .095).

Detection

The postexperiment survey revealed that 12 of the 24 participants (50.0%) noticed structural differences among the lists. The accuracy advantage was numerically higher for participants who detected structural differences in the lists (6.5% advantage with detection; 2.2% advantage without detection), but the difference between the groups was not significant, t(22) = 1.37, SE = .032., p = .185)

Discussion

We replicated the main findings of Experiment 1: Recall accuracy improved on both types of list with practice, accuracy improved more for spun lists than for scrambled lists, and the accuracy advantage was a consequence of participants being better able to remember spun list items in their correct serial positions. New to Experiment 2, we found that the accuracy advantage for spun lists transfers to unpracticed lists, and we found that this transfer advantage was also a consequence of participants being better at remembering new spun list items in their correct serial positions. The spun list learning advantage observed in the training period and the spun list transfer advantage observed in the test period are both evidence that the serial memory system can use itemdependent retrieval cues. The lack of positive transfer in spun lists also highlights the importance of other information to serial memory, such as serial position and list-level representations, that changed in the test period.

Neither the learning advantage observed during training nor the transfer advantage observed at test reflect a speed-accuracy tradeoff. IT and IRT were not significantly slower for spun lists in either portion of the experiment.

Experiment 3

In Experiment 3, we replicated the basic transfer of training design, but participants trained on two lists from a set and tested on the remaining four. Changing the number of trained lists per set from six (Experiment 1) to 4 (Experiment 2) caused a reduction in the spun list advantage, so we expect the training period advantage to reduce even further. The current experiment also allows us to

see how the transfer advantage is affected by a reduction in the number of trained lists.

Method

Participants

Twenty-four participants were recruited for this experiment, none of whom participated in Experiments 1 or 2. On average, participants typed 67.74 WPM and made no errors on 92.88% of the words in the typing test. Participants were tested in 1 hr timeslots and received \$12 or course credit as compensation.

Apparatus and Stimuli

Like Experiment 2, we selected a subset of the lists to be trained lists. Lists 1 and 2 in each set were selected to be trained lists, and lists 3–6 in each set became test lists. Trained spun lists were always consecutive spins of the item set (e.g., ABCDEF and BCDEFA). In total, there were eight training lists and 16 test lists. Table 3 presents an example set of lists.

Procedure

Participants completed 416 trials (320 training + 96 test). This experiment was otherwise identical to Experiment 2 in procedure.

Analyses

The analyses are the same as those conducted for Experiment 2.

Results

Error Rate: Training

Mean error rates are shown in Figure 5A, and the rates of different error types are shown in Figure 6. Error rates reduced for both types of list with practice (spun: 7.3% reduction from block 1 to block 40; scrambled: 7.1% reduction). The reduction in spun list error rate was attributable to a reduction in misorders and intrusions, whereas the reduction in scrambled list error rate was attributable to a reduction in omissions and intrusions. Error rates were numerically but not significantly lower for spun lists during training (spun: 17.4%, scrambled: 19.6%; F[1, 23] = 1.378, MSE = .166, p = .252).

Table 3

Construction of Letter Sequences in Experiment 3

Training period error rates were 2.2% lower for spun lists on average, compared with 5.9% in Experiment 1 and 3.3% in Experiment 2.

Error Rate: Transfer

Error rates were significantly lower for spun lists in the test period (spun: 22.3%, scrambled: 30.3%; F[1, 23] = 21.796, MSE = .021, p < .001). The spun list transfer advantage reduced from 11.1% in Experiment 2 to 8.0% in Experiment 3. Like Experiment 2, the transfer advantage reflected negative transfer for scrambled lists (F[1, 23] = 11.166, MSE = .028, p = .003) and net-zero transfer for spun lists, F(1, 23) = .003, MSE = .028, p = .954, and it was the result of participants committing fewer misorders in new spun lists, F(1, 23) = 16.337, MSE = 44.388, p = .001.

Error Rate: Power Function Analysis

For the training functions, average R^2 was .184 (RMSD = 1.17), and the average shared *b* parameter was 29.4. Learning rates did not differ significantly in the training period (spun: .234, scrambled: .195; t[23] = .957, SE = .040, p = .349). For the test functions, average R^2 was .435 (RMSD = 1.92). The starting point of error rate was significantly lower for spun lists (spun: 27.5, scrambled: 34.2; t[23] = -2.855, SE = .024, p = .009).

Initiation Time

Mean IT is shown in Figure 5B. IT reduced for both list types with practice, but spun and scrambled lists did not differ in IT in the training period (spun: 645.2 ms, scrambled: 627.4 ms; F[1, 23] = .173, MSE = 874,796, p = .682) or test period (spun: 661.7 ms, scrambled: 715.4 ms; F[1, 23] = 2.730, MSE = 75,986, p = .112).

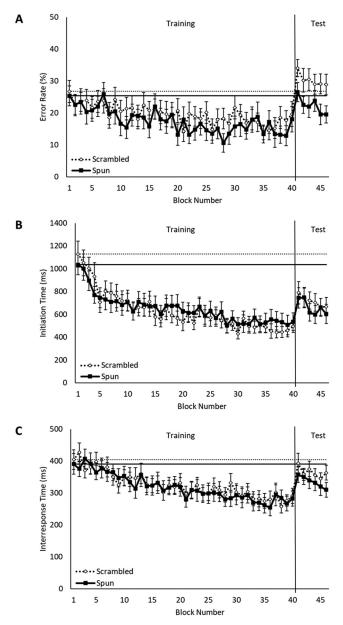
Initiation Time: Power Function Analysis

For the training functions, average R^2 was .386 (RMSD = 23.252), and the average shared *b* parameter was 1,023.5. Learning rates did not differ significantly in the training period (spun: .310, scrambled: .320; t[23] = -.338, SE = .031, p = .738). Average R^2 was .400 (RMSD = 35.860) for the test functions. Neither the starting points (spun: 664.8, scrambled: 676.6; t[23] = -.378, SE = 31.459, p =

List			Spu	n Set			List			Scramb	oled Set		
Training													
List 1	t	р	g	u	b	k	List 5	q	f	W	m	S	1
List 2	р	g	u	b	k	t	List 6	f	m	q	1	W	S
List 3	y	Z	с	n	r	d	List 7	Х	h	0	i	j	v
List 4	Z	с	n	r	d	у	List 8	h	i	х	v	0	j
Test						-							-
List 1	g	u	b	k	t	р	List 9	m	1	f	s	q	W
List 2	u	b	k	t	р	g	List 10	1	s	m	W	f	q
List 3	b	k	t	р	g	ū	List 11	s	W	1	q	m	f
List 4	k	t	р	g	ŭ	b	List 12	W	q	s	f	1	m
List 5	с	n	r	d	у	Z	List 13	i	v	h	j	х	0
List 6	n	r	d	у	Z	с	List 14	v	j	i	0	h	х
List 7	r	d	У	Z	с	n	List 15	j	o	v	х	i	h
List 8	d	У	Z	с	n	r	List 16	0	х	j	h	v	i

Note. The same letters were shown in the training and test periods. Participants practiced 2 lists from each set and tested on the remaining 4 lists from each set.

Experiment 3: Mean Error Rate (A), Initiation Time (IT; B), and Interresponse Time (IRT; C) for Each List Type in Each of the 46 (40 Training + Six Test) Experiment Blocks



Note. The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

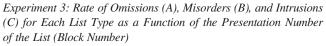
.709) nor the learning rates (spun: .269, scrambled: .160; t[23] = 1.928, SE = .057, p = .066) differed significantly in the test period.

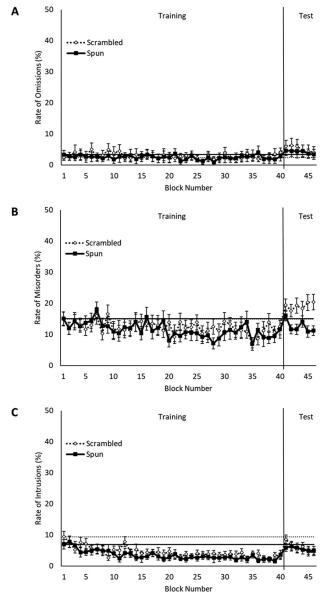
Interresponse Time

Figure 5C displays mean IRT. IRT improved for both list types with practice, but the lists types did not differ in the training period (spun: 317.4 ms, scrambled: 324.9 ms; F[1, 23] = .173, MSE =

153,765, p = .682). IRT was lower for spun lists in the test period (spun: 334.8 ms, scrambled: 365.0 ms; F[1, 23] = 4.822, MSE = 13,612, p = .038). There was positive transfer for both lists (spun: F[1, 23] = 17.924, MSE = 9,873, p < .001; scrambled: F[1, 23] = 5.487, MSE = 9,873, p = .028), so the transfer difference reflects greater positive transfer for spun lists than scrambled lists.

Figure 6





Note. The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

Interresponse Time: Power Function Analysis

For the training functions, average R^2 was .331 (RMSD = 8.104), and the average shared *b* parameter was 304.1. Learning rate was numerically but not significantly higher for spun lists in the training period (spun: .325; scrambled: .233; *t*[23] = 1.778, *SE* = .052, *p* = .089). Average R^2 was .421 (RMSD = 13.342) for the test functions. The starting point for IRT was significantly lower for spun lists in the test period (spun: 216.1; scrambled: 252.1; *t*[23] = -2.434, *SE* = 14.780, *p* = .023).

Detection

The postexperiment survey revealed that 13 of the 24 participants (54.2%) noticed structural differences among the lists. Contrary to previous experiments, the accuracy advantage was significantly higher for participants who detected structural differences in the lists (N = 13; 6.7% advantage) versus those who did not (N = 11; -.2% advantage), t(22) = 2.17, SE = .032, p = .041. Further investigation revealed that there was a spun list advantage in the training period only for those who detected structural differences (7.6% advantage with detection, 3.8% disadvantage without), and there was a test period advantage regardless of detection (10.0% advantage with detection, 5.3% advantage without).

Discussion

Further reducing the number of trained spun lists led to a reduction in the spun list advantage, both during initial learning and during transfer. Compared with previous experiments, Experiment 3 gave fewer opportunities to learn the relative positions of items in spun lists (just 2 opportunities per block), and there was less inconsistency in the serial positions of the items (each item was presented in only 2 serial positions during training). Participants therefore had less opportunity to strengthen item-dependent cues, and the relative benefit of using these cues was diminished.

Although the advantage for spun lists reduced, it was not eradicated: Misorders decreased in trained spun lists but not in trained scrambled lists, and new spun lists were recalled more accurately as a result of fewer misorders being committed in those lists. Participants were still using item-dependent cues in this experiment, even though these cues conferred less of a benefit.

Table 4

Construction of Letter Sequences in Experiment 4

Experiment 4: Separating Memory and Perception

From stimulus to response, the lists in our experiments are represented in multiple different forms: first as iconic representations in perception, then as abstract representations in memory, and finally as motor commands in the motor system. These representations are coupled in the previous experiments—each letter has one associated visual representation and one associated response—so it has not been possible to pin the spun list advantage on any one of these systems. In Experiment 4, we decouple perceptual and memory representations to test whether the spun list advantage is isolated to perception.

In our experiments, the letters were presented simultaneously and very briefly, so some errors—particularly those early in the experiment—may reflect failures to properly encode the list. Spun lists placed letters in consistent relative spatial positions, and this consistency may have promoted perceptual learning, similar to that which is observed in other serial learning paradigms (e.g., the *serial* reaction time task; Abrahamse et al., 2010). Importantly, the relative spatial locations of letters are more consistent in spun lists than in scrambled lists, so perceptual learning alone predicts more accurate encoding, and thus more accurate retrieval of the spun lists.

Perceptual learning can be eliminated by changing aspects of the stimulus. Several letters in the English alphabet have visually distinct uppercase and lowercase forms (e.g., r and R), so we manipulated letter case. Participants trained on lists rendered in one letter case (e.g., all lowercase) and then tested on letters rendered in the other letter case (e.g., all uppercase). If item-dependent cues are isolated to perception, then the spun list advantage should disappear after changing letter case; if they are used outside of perception, then the advantage should persist.

Method

Participants

Twenty-four participants were recruited for this experiment. People who participated in Experiments 1–3 were excluded from selection, but all other selection criteria were the same as previous experiments. On average, participants typed 72.95 WPM and made no errors on 93.82% of the words in the typing

List			Spu	n Set			List		Scram	bled Set			
Training													
List 1	q	g	b	n	r	t	List 7	d	u	j	h	m	У
List 2	g	b	n	r	t	q	List 8	u	h	d	у	j	m
List 3	b	n	r	t	q	g	List 9	h	У	u	m	d	j
List 4	n	r	t	q	g	b	List 10	у	m	h	j	u	d
List 5	r	t	q	g	b	n	List 11	m	j	у	d	h	u
List 6	t	q	g	b	n	r	List 12	j	d	m	u	у	h
Test								U U				·	
List 1	Q	G	В	Ν	R	Т	List 7	D	U	J	Н	М	Y
List 2	G	В	Ν	R	Т	Q	List 8	U	Н	D	Y	J	Μ
List 3	В	Ν	R	Т	Q	Ğ	List 9	Н	Y	U	М	D	J
List 4	Ν	R	Т	Q	Ğ	В	List 10	Y	М	Н	J	U	D
List 5	R	Т	Q	Ğ	В	Ν	List 11	Μ	J	Y	D	Н	U
List 6	Т	Q	Ĝ	В	Ν	R	List 12	J	D	М	U	Y	Н

Note. The order in which participants saw lowercase or uppercase lists was counterbalanced. Half the participants saw lowercase lists first (shown above), and the other half saw uppercase lists first (not shown).

test. Participants were scheduled in 1 hr timeslots and received \$12 or course credit as compensation.

Apparatus and Stimuli

We trimmed the pool of selectable letters to those with more distinct lowercase and uppercase forms. The final pool of 12 letters was b, d, g, h, j, m, n, q, r, t, u, and y. Six of these letters were randomly selected to be the letters of the spun set, and the remaining six letters become the scrambled set. Example lists are shown in Table 4.

Procedure

Participants practiced the same 12 lists 50 times over the course of the experiment–40 times in the training period and 10 times in the test period. For half the participants (12) the first 40 exposures were in lowercase, and the last 10 exposures were in uppercase. For the other half, the first 40 exposures were in lowercase, and the last 10 exposures were in lowercase, and the last 10 exposures were in lowercase, because the Shift and Caps Lock keys were disabled. Participants typed the lowercase and uppercase lists in the same way, so we could observe the effects of changing perception in isolation. Participants completed 600 (480 training and 120 test) trials and had the opportunity to take self-paced breaks every 120 trials.

Analyses

The order in which participants were exposed to the two letter cases was included as a between-subjects factor in the ANOVA analyses of accuracy and latency. However, the main effect of order was not significant for any of the dependent measures. As a result, the reported analyses average over group, and group was not included as a factor on analyses of specific error types. Simple main effect analyses of transfer included all 10 test blocks and an equal number of training blocks. All other analyses were identical to those used in Experiments 2 and 3.

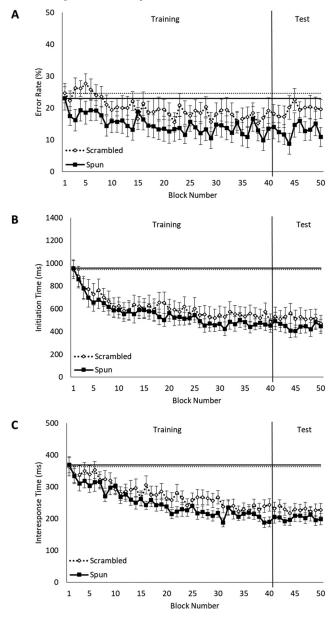
Results

Error Rate

Mean error rate averaging over participants and groups is shown in Figure 7A, and error type graphs are shown in Figure 8. Result tables (Tables B10, B11, and B12) can be found in Appendix B. Error rates reduced in spun and scrambled lists (spun: 9.5% reduction from block 1 to block 40; scrambled: 5.4% reduction). This improvement reflected reductions in misorders in spun lists, and reductions of both misorders and intrusions in scrambled lists. Error rates were lower for spun lists during training (spun: 14.9%; scrambled: 19.9%; F(1, 22) = 8.007, MSE = .154, p = .010) despite no initial difference in error rate (spun: 23.0%; scrambled: 24.7%; t[23] = -.471, SE = .034, p = .642). This training period advantage persisted into the test period after changing letter case (spun: 13.0%; scrambled: 19.5%; F(1, 22) = 19.861, MSE = .026, p < 1000.001). Transfer was positive for both list types, and transfer was perfect for spun lists. Both the training and test advantage arose because participants committed fewer misorders in spun lists (training: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 111.035, p = .003; test: F[1, 23] = 10.980, MSE = 10.98023] = 31.424, *MSE* = 21.239, *p* < .001).

Figure 7

Experiment 4: Mean Error Rate (A), Initiation Time (IT; B), and Interresponse Time (IRT; C) for Each List Type in Each of the 50 (40 Training + 10 Test) Experiment Blocks

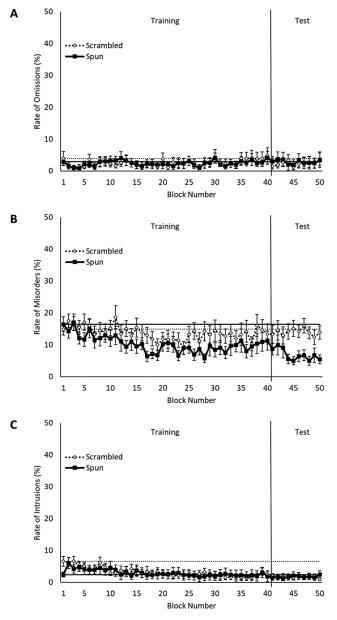


Note. The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

Error Rate: Power Function Analysis

Average R^2 was .268 (RMSD = .97) and .239 (RMSD = 1.76) for fits to the training data and test data, respectively. The average shared *b* parameter for the training period was 28.4. Training period learning rate (spun: .369; scrambled: .162; t[23] = 3.075, SE = .067, p = .005) and test period starting point (spun: 14.9; scrambled: 30.0; t[23] = -3.546, SE = .017, p = .003) were superior for spun lists.

Experiment 4: Rate of Omissions (A), Misorders (B), and Intrusions (C) for Each List Type as a Function of the Presentation Number of the List (Block Number)



Note. One participant was excluded from each of the error rate means (see text for details). The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

Initiation Time

Mean IT averaging over participants and groups is shown in Figure 7B. IT decreased in both list types with practice, but there was no significant difference in IT between them during the training or test period.

Initiation Time: Power Function Analysis

Average R^2 was .408 (RMSD = 16.528) and .317 (RMSD = 24.174) for fits to the training data and test data, respectively. The average shared *b* parameter for the training period was 851.4. None of the IT tests reached significance.

Interresponse Time

Mean IRT averaging over participants and groups is shown in Figure 7C. There was no significant advantage for spun lists in the training period or test period.

Interresponse Time: Power Function Analysis

Average R^2 was .443 (RMSD = 6.130) and .368 (RMSD = 8.737) for fits to the training data and test data, respectively. The average shared *b* parameter for the training period was 257.2. None of the power function parameters were significantly different between the list types.

Detection

The postexperiment interviews revealed that 12 participants (50%) noticed additional structure in some lists. Detection did not confer a significantly greater error rate advantage for spun lists (detection: 6.9% advantage; no detection: 2.1% advantage), t(22) = .931, SE = .051, p = .362.

Discussion

The main findings of previous experiments replicated: Spun lists were learned more quickly, and the advantage for spun lists was isolated to misorder errors. Additionally, this advantage persisted into the test period after letter case was changed. Changing the perceptual representations of the lists did not remove the spun list advantage, so item-dependent cues are not isolated to perception. They must be used in a later processing stage—either in memory, or in the motor system.

Experiment 5: Separating Memory and Motor Programs

In a previous series of experiments (Lindsey & Logan, 2019), we compared copy typing speed in spun and scrambled lists, and we found that people typed the spun lists more quickly with practice. We concluded that participants used item-dependent cues (specifically, single item cues³) to retrieve the appropriate sequence of keystrokes. The current experiments build upon the logic and experimental procedure that we previously used—including the typing of responses. The spun list advantage we observed in the current study may have arisen simply because item-dependent cues were used when typing the memorized letters.

In Experiment 5, we decouple memory and motor representations to test whether the spun list advantage is isolated to action. We manipulated response method: Participants practiced recalling spun and scrambled lists using one response method—either typing the responses or speaking the responses into a microphone—and then

³ In one of the experiments in Lindsey & Logan (2019), participants practiced just one list in the spun set and then were tested on all of the lists in the set. Positive transfer was observed, and transfer was equal for all new lists in the set. If more complex item-dependent cues were used, then transfer would have differed among the new lists.

they were tested on the same lists using a different response method (e.g., speaking the lists if they typed the lists during training). If item-dependent cues only support action, then the spun list advantage should disappear after changing the response method (from manual to oral, or vise-versa). If item-dependent cues are not isolated to the motor system, then the advantage should persist.

Method

Participants

Twenty-four participants were recruited for this experiment. People who participated in Experiments 1–4 were excluded from selection, but all other selection criteria were the same as previous experiments. On average, participants typed 79.06 WPM and made no errors on 93.44% of the words in the typing test. Participants were scheduled in 1.5 hr timeslots and received \$18 or course credit as compensation.

Apparatus and Stimuli

On spoken trials, participants spoke responses into a standing microphone. Participants were allowed to adjust the position and height of the microphone to their needs, but we ensured that the microphone did not obstruct their view of the screen or their ability to use the keyboard.

We exerted more control over the phonological characteristics of the letters in this experiment. We omitted all vowel letters, including 'y,' to prevent the formation of nonsense syllables. We omitted 'w' because it has multiple syllables when pronounced alone. We omitted all letters with the "-ee" sound, excluding 'z,' to lessen the likelihood of phonological confusion errors. The final pool of 12 letters was f, h, j, k, l, m, n, q, r, s, x, and z. For each participant, six of these letters were randomly selected to be the letters of the spun set, and the remaining six letters became the scrambled set. Example lists are shown in Table 5.

Procedure

Half the participants (12) practiced typing each of the 12 lists 40 times in the training period and then practiced speaking each of the lists 10 times in the test period. There was a short break between the training and test period, during which these participants were instructed to speak the letters they remember into a microphone instead of typing them when the "Response:" screen appeared. Like before, they pressed the spacebar to finish the trial and move on to the next trial. The other half of participants practiced speaking the 12 lists during the training period and practiced

typing the lists during the test period. All participants completed 600 trials (480 training and 120 test), and self-paced breaks were provided every 120 trials.

Analyses

Spoken trials were scored manually. We listened to each of the sound files, transcribed the letter responses, and then scored accuracy by comparing the transcriptions to the letters presented on each trial. Owing to difficulties in detecting the start and end times of each utterance in the sound files, IT and IRT were not distinguished in this experiment. These timing measures were replaced by a measure of the total time taken on each trial, measured from the onset of the response screen to the spacebar press. The time taken on a trial was only included in averages if it was less than 10 s. Two participants took longer than 10 seconds on every trial in one or more blocks. As a result, these two participants had missing time taken data, and we excluded these participants from all graphs and analyses of time taken.

The order in which participants were exposed to the two response orders was included as a between-subjects factor in the ANOVA analyses. It did not have a significant effect on error rate, but it did affect the time taken to report the lists, F(1, 20) = 4.722, MSE = 33,435,559, p = .042, with participants who switched from typing to speech being faster than those who switched from speech to typing. In the tables, all reported analyses average over group, and analyses of specific error types do not include group as a factor. However, in the text detailing analyses of time taken, we specify whether the grouping variable interacted with an effect.

Outlier analyses on error rate detected one outlier: a participant who made 35.5% more errors on spun lists than scrambled lists. This participant was excluded from the figures but was not excluded from the analyses reported in Tables B13, B14, and B15 (found in Appendix B). We conducted additional Wilcoxon signed rank tests on differences in error rate that were robust to the detected outlier. The results of these analyses are shown in Table B16. The Wilcoxon test indicated that there was a significant training advantage for spun lists (Z = 2.100, SE = 34.996, p = .036), whereas the ANOVA did not, F(1, 22) = 2.079, MSE = 1.441, p = .163. The error rate means used by the ANOVA were skewed negatively by the outlier, so our discussion of the error rate results will focus on the Wilcoxon tests whenever possible.

Results

Error Rate: Training

Mean error rate for each experiment block is shown in Figure 9A, and rates of each error type are shown in Figure 10. Accuracy

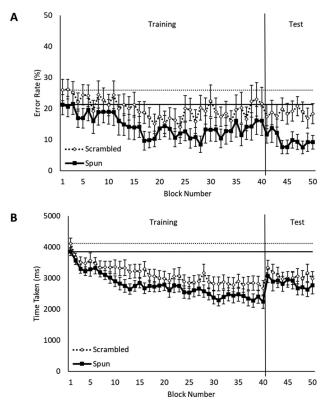
 Table 5

 Construction of Letter Sequences in Experiment 5

construc	поп ој Це	ner segu	chees in L	aperimen											
List		Spun Set					List		Scrambled Set						
List 1	q	r	s	f	n	j	List 7	k	1	х	h	m	z		
List 2	r	s	f	n	j	q	List 8	1	h	k	Z	х	m		
List 3	s	f	n	j	q	r	List 9	h	Z	1	m	k	х		
List 4	f	n	j	q	r	s	List 10	Z	m	h	х	1	k		
List 5	n	j	q	r	s	f	List 11	m	х	Z	k	h	1		
List 6	j	q	r	s	f	n	List 12	х	k	m	1	Z	h		

Note. This experiment included a training period and a test period, but the same letters were used in both. Only the response method differed between the two periods.

Experiment 5: Mean Error Rate (A) and Time Taken (B) for Each List Type in Each of the 50 (40 Training + 10 Test) Experiment Blocks



Note. One participant was excluded from the error rate means, and 3 participants were excluded from the time taken means (see text for details). The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

improved in both list types with practice (excluding the outlier; spun: 5.2% reduction from block 1 to block 40; scrambled: 4.3% reduction). Misorders and intrusions reduced in both list types, and omissions reduced in scrambled lists. Error rate was lower for spun lists than scrambled lists in the training period (excluding the outlier: 14.5% vs. 20.1%), supported by a significant Wilcoxon signed-ranks test on the error rate difference (Z = 2.100, SE = 34.996, p = .036). Error rate was not significantly lower for spun lists at the start of the experiment (Z = .806, SE = 32.878, p = .420), so the lower overall error rate reflected faster learning in those lists. The error rate advantage for spun lists arose because the rate of misorders was lower in spun lists in the training period (Z = 2.600, SE = 35.000, p = .009).

Error Rate: Transfer

Test period error rate was lower for spun lists than scrambled lists (excluding the outlier: 9.8% vs. 18.6%; Z = 2.771, SE = 35.000, p = .006), and there was positive transfer for both list types. The transfer advantage was attributable to participants committing fewer misorders in spun lists during the test period

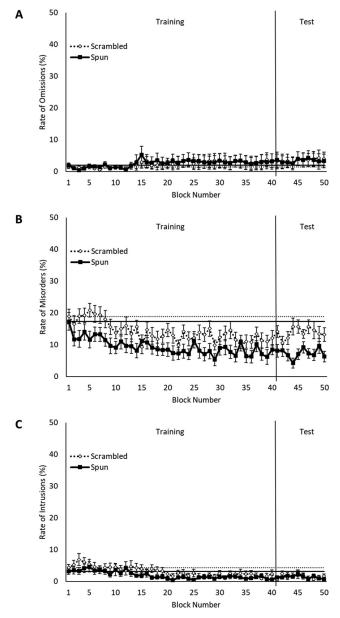
(Z = 3.958, SE = 34.993, p = .006). The spun list advantage persisted after the change in response method.

Error Rate: Power Function Analysis

Average R^2 from fits to training error rate and test error rate were .333 (RMSD = 1.02) and .395 (RMSD = 1.63), respectively. The average shared *b* parameter was 30.1 for training error rate (excluding the outlier). Learning rate was faster for spun lists in the

Figure 10

Experiment 5: Rate of Omissions (A), Misorders (B), and Intrusions (C) for Each List Type as a Function of the Presentation Number of the List (Block Number)



Note. The bars are standard errors of the means. The horizontal lines without error bars show first block performance for each list type (solid: spun; dotted: scrambled).

training period (excluding the outlier: .396 vs. .211; Z = 2.200, SE = 35.000, p = .028). Initial performance (excluding the outlier: 14.5 vs. 20.6; Z = -2.057, SE = 35.000, p = .040) and learning rate (excluding the outlier: .290 vs. .051; Z = 2.029, SE = 34.998, p = .042) were better for spun lists in the test period.

Time Taken

Mean time taken on a trial for each experiment block is shown in Figure 9B. Both list types improved with practice, and improvement was faster for those who typed their responses in the training period. The time taken on spun lists was lower than on scrambled lists, but the difference was only marginally significant (2,755.5 ms vs. 3,083.4 ms; F[1, 20] = 4.280, MSE = 11,048,174, p = .052). The time advantage was numerically higher for participants who typed their responses during training (395.1 ms vs. 260.6 ms), although this difference was not significant. There was a numerical but nonsignificant advantage for spun lists in the test period after the response method was changed (spun: 2,770.5 ms, scrambled: 3,016.9 ms; F[1, 20] = 2.353, MSE = 2,839,515, p = .141).

Time Taken: Power Function Analysis

Average R^2 from fits to time taken during training and time taken during test were .564 (RMSD = 48.240) and .491 (RMSD = 70.184), respectively. The average shared *b* parameter was 4,081.4 for the training period. Learning rate was numerically faster for spun lists in the training period, but the difference was only marginally significant (spun: .177, scrambled: .135; *t*[21] = 1.962, *SE* = .022, *p* = .063). Initial performance (spun *b*: 2,953.1; scrambled *b*: 3,129.8; *t*[21] = -1.025, *SE* = 172.362, *p* = .317) and learning rate (spun *c*: .093; scrambled *c*: .062; *t*[21] = 1.430, *SE* = .022, *p* = .168) were both numerically better for spun lists in the test period, but these differences also did not reach significance.

Detection

The postexperiment interviews revealed that 12 participants (50%) noticed additional structure in some of the lists. The spun list advantage was numerically higher for participants who detected differences in structure (advantage = 6.8%) than for those who did not (advantage = 3.4%), although this difference did not reach significance, t(22) = 1.079, SE = .032, p = .293.

Discussion

Like previous experiments, spun lists were recalled more accurately than scrambled lists, supported by a reduced tendency to commit misorders. Response method had little impact on the spun list advantage: The presence of an accuracy advantage did not depend on the response method, and changing the response method in the middle of the task did not eliminate the accuracy advantage. The item-dependent cues that support performance are not isolated to the motor system—they are being used by a system that is further upstream.

Experiment 4 demonstrated that these cues are also not isolated to perception. The item-dependent cues that produce the spun list advantage are used in memory and incorporate abstract representations of previously retrieved items. Serial memory theories must deal with the spun list advantage directly. It cannot be assumed to arise from processes occurring in other systems (e.g., perception or the motor system).

Although item-dependent cues are not isolated to the motor system, evidence suggests that these cues are used by the motor system. We previously found that typing speed is sensitive to the consistent relative order in spun lists (Lindsey & Logan, 2019). In the current study, we also found a speed advantage for spun lists in the training period, and changing the response method change reduced the speed advantage—both of which are consistent with item-dependent cues being used to speed up response selection. It seems likely that item-dependent cues are used in multiple stages of processing: both in memory and in action (Logan, 2021).

Experiment 6

The previous experiments used a serial learning procedure that was unlike many other procedures used to study serial memory. Like standard serial recall, we used short lists of items, and the items were drawn from closed sets (Farrell et al., 2013). However, like serial learning tasks, participants could not specify when they were omitting an item (Osth & Dennis, 2015). We also presented the list of items simultaneously, which is more commonly used in whole report paradigms than in serial recall.

Our goal in Experiment 6 is to determine whether the spun list advantage is only observed in the particular task setting that we constructed. We adapt the logic of our spun list procedure to a more standard serial memory task—the Hebb (1961) repetition procedure, with sequential presentation of words. Participants saw *control lists* that did not repeat and two types of *Hebb lists* that did. Participants practiced several Hebb lists over the course of the experiment. Some of these Hebb lists were spins of a set of words, and others were scrambled orderings of a set (rearranged by passing the words through a 6×6 balanced Latin square).

An advantage is observed for Hebb lists over control lists (i.e., a *Hebb effect*) when the initial list items in Hebb lists are repeatedly presented in the same serial positions (Hitch et al., 2005; Schwartz & Bryden, 1971). Spinning or scrambling the items causes the items to shift into different serial positions, so one might anticipate that there would be no learning advantage in spun Hebb lists or scrambled Hebb lists. However, in our experiments, every list is repeated exactly (e.g., participants practice both ABCDEF and FABCDE multiple times), so Hebb effects may still be obtained.

We attempt to modulate the Hebb effect by manipulating word order consistency. Spun Hebb lists have greater consistency because items are presented in the same relative position in all lists. If item-dependent cues (which adopt this consistency) contribute to retrieval, then the Hebb effect for spun lists should be larger than the Hebb effect for scrambled lists.

Method

Participants

We recruited 48 participants for a pilot study prior to Experiment 6. It had the same basic design but manipulated the type of Hebb list (spun or scrambled) between subjects. We used this pilot study to inform how many participants should be recruited for Experiment 6. We desired at least 85% power. The Cohen's d effect size for the difference between the spun and scrambled

Table 6		
Construction	of Lists in	Experiment 6

List		5	Spun Hebb (Blocks 1-8	3)		List		Scrat	mbled Hebl	o (Blocks 9	-16)	
List 1	plea	knob	rung	seed	wind	grip	List 1	bolt	harp	cape	kill	burn	lift
List 2	knob	rung	seed	wind	grip	plea	List 2	harp	kill	bolt	lift	cape	burr
List 3	rung	seed	wind	grip	plea	knob	List 3	kill	lift	harp	burn	bolt	cape
List 4	seed	wind	grip	plea	knob	rung	List 4	lift	burn	kill	cape	harp	bolt
List 5	wind	grip	plea	knob	rung	seed	List 5	burn	cape	lift	bolt	kill	harp
List 6	grip	plea	knob	rung	seed	wind	List 6	cape	bolt	burn	harp	lift	kill
List			Spun Contr	ol (Block 1)		List		Scr	ambled Co	ntrol (Bloc	k 9)	
List 1	gain	doll	lump	claw	slit	self	List 1	tale	golf	stew	dust	duck	hind
List 2	doll	claw	gain	self	lump	slit	List 2	golf	dust	tale	hind	stew	ducl
List 3	claw	self	doll	slit	gain	lump	List 3	dust	hind	golf	duck	tale	stew
List 4	self	slit	claw	lump	doll	gain	List 4	hind	duck	dust	stew	golf	tale
List 5	slit	lump	self	gain	claw	doll	List 5	duck	stew	hind	tale	dust	golf
List 6	lump	gain	slit	doll	self	claw	List 6	stew	tale	duck	golf	hind	dust
List 7	card	task	cock	tape	cult	herb	List 7	hill	disc	thud	myth	crow	sten
List 8	task	tape	card	herb	cock	cult	List 8	disc	myth	hill	stem	thud	crov
List 9	tape	herb	task	cult	card	cock	List 9	myth	stem	disc	crow	hill	thud
List 10	herb	cult	tape	cock	task	card	List 10	stem	crow	myth	thud	disc	hill
List 11	cult	cock	herb	card	tape	task	List 11	crow	thud	stem	hill	myth	disc
List 12	cock	card	cult	task	herb	tape	List 12	thud	hill	crow	disc	stem	myt
List			Spun Contr	ol (Block 8)		List		Scra	ambled Cor	ntrol (Block	x 16)	
List 1	wave	ware	moth	hose	kick	shop	List 1	boot	vein	safe	tang	hail	star
List 2	ware	hose	wave	shop	moth	kick	List 2	vein	tang	boot	star	safe	hail
List 3	hose	shop	ware	kick	wave	moth	List 3	tang	star	vein	hail	boot	safe
List 4	shop	kick	hose	moth	ware	wave	List 4	star	hail	tang	safe	vein	boot
List 5	kick	moth	shop	wave	hose	ware	List 5	hail	safe	star	boot	tang	vein
List 6	moth	wave	kick	ware	shop	hose	List 6	safe	boot	hail	vein	star	tang
List 7	heir	heel	mine	hump	band	tent	List 7	drum	mole	arch	hole	fool	tom
List 8	heel	hump	heir	tent	mine	band	List 8	mole	hole	drum	tomb	arch	fool
List 9	hump	tent	heel	band	heir	mine	List 9	hole	tomb	mole	fool	drum	arch
List 10	tent	band	hump	mine	heel	heir	List 10	tomb	fool	hole	arch	mole	drur
List 11	band	mine	tent	heir	hump	heel	List 11	fool	arch	tomb	drum	hole	mol
List 12	mine	heir	band	heel	tent	hump	List 12	arch	drum	fool	mole	tomb	hole

Note. Word sets were randomly assigned to conditions and blocks for each participant. The order in which participants practiced Spun Hebb lists and Scrambled Hebb lists was counterbalanced. Half the participants saw Spun Hebb lists first (shown above), and the other half saw Scrambled Hebb lists first (not shown). Hebb lists were seen in 8 consecutive blocks, and control lists were seen in only 1 block. Control lists in blocks 2–7 and blocks 10–15 are omitted to save space.

Hebb effect was .619. Using this effect size, the power analysis determined that the current study would require at least 22 participants.

We recruited 22 participants for Experiment 6. Participants who participated in Experiments 1–5 or the pilot study were excluded from selection. All other selection criteria were the same as previous experiments. Participants were not given a typing test, so typing speed and accuracy were not obtained for this experiment. Participants were scheduled in 1.5 hr timeslots and received \$18 or course credit as compensation.

Apparatus and Stimuli

We took inspiration from Page et al. (2013) when designing this experiment. Participants performed immediate serial recall of word lists. Lists were formed by reordering a closed set of words, in much the same way that spun and scrambled lists were formed in previous experiments. Hebb lists and control lists had nonoverlapping word sets.

The words used in the task were obtained from the MRC psycholinguistic database (Coltheart, 1981). We obtained a pool of 210 four-letter, one-syllable English words. The pool was divided into 35 sets of six words. To the best of our ability, we matched the 35 word sets on three characteristics: concreteness, familiarity, and frequency. Appendix A shows the full list of the words used (Table A1) and describes the method of dividing the words into sets.

The same word sets were used for all participants. For each participant, one of the sets was randomly selected to be the practice set, one was randomly selected to be the spun Hebb set, one was randomly selected to be the scrambled Hebb set, and the remaining 32 sets became control sets. Like the letter sets in previous experiments, every word set generated six lists. For the practice set, scrambled Hebb set, and control sets, we accomplished this by passing the six words in the set through a 6×6 balanced Latin square. For the spun Hebb set we spun the words to produce six spun lists. Example spun Hebb, scrambled Hebb, and control lists are presented in Table 6.

The six control lists in each control set were presented one time each, in just one block of the experiment. The six Hebb lists in each Hebb set were presented eight times each—once per block for eight consecutive blocks. Scrambled Hebb lists and Spun Hebb list differed from control lists by having more presentations (of both the list and the words within the lists). Spun Hebb lists differed from Scrambled Hebb lists by having consistent relative order. The serial positions of items were inconsistent for all lists in the current experiment.

Procedure

Participants were informed that they would see multiple lists of words and that for each list they would need to remember and type the words in their presented order. We asked participants to type as quickly and accurately as possible, to type "blank" in place of a forgotten word, and to not attempt to correct typing errors because the backspace was disabled. The number keys, shift, and caps lock were disabled as well.

Each trial began with a centrally presented fixation cross, presented for 500 ms, followed by 500 ms blank screen. Six words were then sequentially presented in the center of the screen. Each word was shown for 1 s, followed by a 500-ms blank screen. After presentation of the sixth word, participants were given prompts ("Word 1:," "Word 2:," etc.) to type each word. As they typed, their keystrokes were echoed on the screen below the prompt. They pressed the enter key to move to the next word, which cleared the previously typed word from the screen. Pressing enter while the sixth prompt ("Word 6":) was on screen ended the trial. A 500-ms blank screen was shown between trials.

Participants first completed six practice trials, wherein they recalled each of the six lists in the practice set. Afterward, participants completed 16 blocks of trials. In eight of those blocks, participants recalled control lists and spun Hebb lists. In the other eight blocks, they recalled control lists and scrambled Hebb lists. Half the participants saw spun Hebb lists in the first half of the experiment and scrambled Hebb lists in the second half of the experiment. The other half saw the opposite order.

Two unique sets of control lists (12 lists total) were randomly assigned to each block. Each list in these sets was presented one time in the block to which they were assigned. These lists were not presented in any other block, so they were only presented one time each during the experiment. Each list in a Hebb set was presented in eight consecutive blocks, so each Hebb list was practiced eight times during the experiment.

There were 18 trials in each block. Hebb lists were presented in six of these trials, and control lists were presented in 12 of them. The pattern of Hebb and control lists was the same for every block: one list from the one of the control sets, one list from the other control set, then one list from the Hebb set (Hebb, 1961; Page et al., 2013⁴). Lists were randomly selected from their respective sets without replacement. Table 7 illustrates how the presentation of Hebb and control lists were structured in the blocks.

Self-paced breaks were given after every fourth block. There were roughly 20 minutes of trials between breaks. Each participant completed a total of 192 trials.

Analyses

We measured error rate and time taken like Experiment 5. A response was scored as correct if the word was typed correctly and recalled in its original serial position. All other responses were scored as incorrect. Time taken was measured from the onset of the "Response:" cue to the Enter press at the end of the trial.

For analyses, we defined four different list types: spun Hebb lists, control lists presented in spun Hebb list blocks (henceforth called *spun control* lists), scrambled Hebb lists, and control lists presented in scrambled Hebb list blocks (henceforth called *scrambled control* lists). Mean error rate and time taken were calculated for each participant, list type, and experiment block.

The analyses were similar in purpose to previous experiments. We checked whether the order of items in Hebb lists mattered using an ANOVA main effect. We checked for initial differences among the list types using a one-way ANOVA on first-block performance. We used simple main effect analyses to check for learning in each list type (analyzing performance over block) and to check for Hebb effects for each of the Hebb list types (analyzing control vs. Hebb performance). For Hebb effect analyses, spun Hebb lists were compared with spun control lists, and scrambled Hebb lists were compared with scrambled control lists. We used the ANOVA interaction to check whether the magnitude of Hebb effect depended on the type of Hebb list.

We fit power functions (Equation 1) to each list type. Functions were fit separately for each participant but simultaneously for each of the list types. All list types shared the same starting point b. Spun control lists and scrambled control lists shared the same learning rate c. Spun Hebb lists and scrambled Hebb lists each had separate learning rates.

We conducted separate analyses of learning, the Hebb effect, and differences in the Hebb effect for each error type (omission, misorder, and intrusion). Participants could only give six responses per trial, so an omission was any response in which the participant typed "blank" or typed nothing before pressing enter. We conducted the same ANOVA analyses as on aggregate error rate.

The order in which participants recalled spun Hebb lists or scrambled Hebb lists was included in analyses of overall error rate and time taken. Participants who practiced spun Hebb lists first did better in the task overall, F(1, 20) = 4.809, MSE = .382, p = .040; however, group did not interact with any of the manipulated variables. The reported analyses average over order, and order is not included in analyses of specific error types. Outlier analyses were conducted like before—this time on the difference in the Hebb effect between spun Hebb lists and scrambled Hebb lists. No outliers were detected.

Results

Error Rate

Result tables are in Appendix B and are split like previous experiments (aggregate *F* tests in Table B17, *t* tests in Table B18, error type *F* tests in Table B19). Figure 11A shows mean error rate for each list type in each experiment block. Error rates reduced in spun Hebb lists (11.1% reduction from block 1 to block 8) and scrambled Hebb lists (9.7% reduction), but they did not reduce significantly in spun control (1.1% increase in errors) or scrambled control lists (2.5% reduction). As a result, we observed a Hebb effect for both spun Hebb lists (13.1% advantage; *F*[1, 20] = 110.956, *MSE* = .014, p < .001) and scrambled Hebb lists

 $^{^{\}rm 4}\,{\rm Our}$ procedure is most similar to their "short spacing nonoverlapping" condition.

Table 7
The Structured Presentation of Hebb Lists and Control Lists

	Blo	ock 1	Blo	ck 8	Blo	ck 9	Bloc	ck 16
Trial	Set	List	Set	List	Set	List	Set	Lis
Trial 1	C1	1	C15	1	C17	1	C31	1
Trial 2	C2	1	C16	1	C18	1	C32	1
Trial 3	H1	1	H1	1	H2	1	H2	1
Trial 4	C1	2	C15	2	C17	2	C31	2
Trial 5	C2	2	C16	2	C18	2	C32	2
Trial 6	H1	2	H1	2	H2	2	H2	2
Trial 7	C1	3	C15	3	C17	3	C31	3
Trial 8	C2	3	C16	3	C18	3	C32	3
Trial 9	H1	3	H1	3	H2	3	H2	3
Trial 10	C1	4	C15	4	C17	4	C31	4
Trial 11	C2	4	C16	4	C18	4	C32	4
Trial 12	H1	4	H1	4	H2	4	H2	4
Trial 13	C1	5	C15	5	C17	5	C31	5
Trial 14	C2	5	C16	5	C18	5	C32	5
Trial 15	H1	5	H1	5	H2	5	H2	5
Trial 16	C1	6	C15	6	C17	6	C31	6
Trial 17	C2	6	C16	6	C18	6	C32	6
Trial 18	H1	6	H1	6	H2	6	H2	6

Note. C1 = control set 1; H1 = Hebb set 1. There were two Hebb sets (one spun and one scrambled) and 32 control sets. Each block had one Hebb set and two Control sets. Hebb lists were presented every third trial (italicized for emphasis). Lists from control sets intervened Hebb list presentation, and control sets alternated.

(7.7% advantage; F[1, 20] = 38.278, MSE = .014, p < .001). Critically, the Hebb effect was significantly larger for spun Hebb lists than scrambled Hebb lists (5.4% advantage difference; F[1, 20] = 9.444, MSE = .014, p = .006). This advantage for spun lists arose despite no significant difference among the list types at the start of the experiment (F[3, 60] = 1.658, MSE = .061, p = .186).

Error Rate: Error Types

Error type rates are shown in Figure 12. Reductions in omissions and intrusions—but not misorders—produced the Hebb effect for scrambled Hebb lists. Reductions in all error types produced the Hebb effect for spun Hebb lists. There was a larger Hebb effect for spun Hebb lists because participants committed fewer misorders in spun Hebb lists, F(1, 21) = 7.488, MSE = 7.360, p = .012. The rate of intrusions was also lower in spun Hebb lists; however, Figure 12C shows that this difference existed early in the experiment and shrunk with practice.

Error Rate: Power Function Analysis

Average R^2 was .405 (RMSD = 1.46). The average shared *b* parameter was 27.6. Learning rate was faster for spun Hebb lists than spun control lists (.737 vs. .073; t[21] = 8.041, SE = .082, p < .001), faster for scrambled Hebb lists than scrambled control lists (.440 vs. .073; t [21] = 5.564. SE = .066, p < .001), and was faster for spun Hebb lists than scrambled Hebb lists (.737 vs. .440; t[21] = 3.281, SE = .090, p = .004). Like the ANOVA, the power function analyses reveal Hebb effects for both list types and a larger Hebb effect for spun lists.

Time Taken

Figure 11B shows mean time taken for each list type. There was no initial difference in time taken at the start of the experiment, and time taken reduced for all list types with practice. However, Hebb effects were observed in the time taken to recall the lists: time taken was significantly lower for both spun Hebb (1,370.8 ms advantage; F[1, 20] = 161.957, MSE = 1,020,990, p < .001) and scrambled Hebb (902.7 ms advantage; F[1, 20] = 70.236, MSE = 1,020,990, p < .001) lists than their respective control lists. Additionally, the spun Hebb advantage was significantly larger than the scrambled Hebb advantage (468.1 ms advantage difference; F[1, 20] = 9.442, MSE = 1,020,990, p = .006).

Time Taken: Power Function Analysis

Average R^2 was .478 (RMSD = 807.028). The average shared *b* parameter was 5,106.5. Learning rate was faster for spun Hebb lists than spun control lists (.493 vs. .091; *t*[21] = 4.647, *SE* = .087, *p* < .001) and faster for scrambled Hebb lists than scrambled control lists (.434 vs. .091; *t*[21] = 4.065, *SE* = .084, *p* < .001). For time taken, learning rate was not significantly faster for spun Hebb lists than scrambled Hebb lists (.493 vs. .434; *t*[21] = .884, *SE* = .067, *p* = .387).

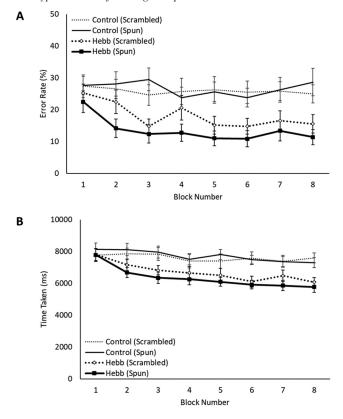
Detection

All 22 participants noticed that some of the lists repeated. Twelve participants (55%) noticed additional structure in some of the repeated lists. We conducted an independent samples *t* test on the difference in the Hebb effect between spun and scrambled Hebb lists. The Hebb effect was not significantly higher for participants who detected differences in structure (N = 12; advantage = 5.4%) than for those who did not (N = 10, advantage = 5.4%), t(20) = .012, SE = .037, p = .991.

Discussion

Hebb lists were recalled more quickly and more accurately than control lists with practice. The words in Hebb lists were presented

Experiment 6: Mean Error Rate (A) and Time Taken (B) for Each List Type in Each of the Eight Experiment Blocks



Note. Hebb lists were presented in each block, and control lists were novel lists that did not repeat. The bars are standard errors of the means.

more frequently than the words in control lists. The Hebb lists themselves also repeated multiple times. Both of these could have contributed to the observed Hebb effect—by promoting item-set learning and list-level learning, respectively.

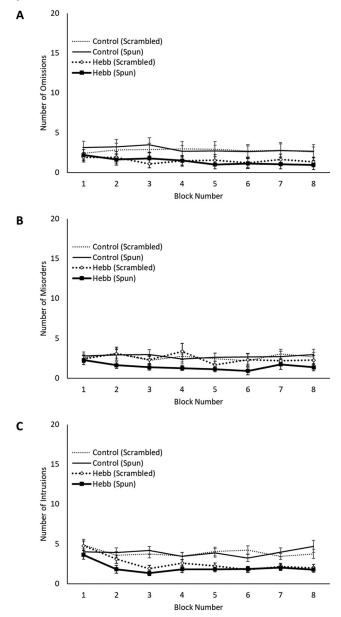
Comparing the magnitude of the Hebb effect between spun and scrambled lists allows us to determine what benefit, if any, itemdependent cues provide above and beyond item-set learning and list-level learning. Critically, the Hebb effect was larger for spun lists than for scrambled lists. When relative order is consistent, item-dependent cues can magnify the Hebb effect.

The results of the current experiment seem to be contrary to the results of other studies that have not found a Hebb effect for spun lists (e.g., Hitch et al., 2005). The key difference, we believe, is that in our experiment each possible spin of a list was repeated multiple times. Previous studies presented spins of a list without repeating the lists verbatim, which caused the starting items to differ on each presentation and likely prevented a Hebb effect from emerging. The Hebb effect seems contingent on the starting items of a list repeating, but the current study demonstrates that consistent relative order can amplify the Hebb effect when this contingency is met.

We made several changes to our experimental design in Experiment 6. We presented lists of words instead of letters. We presented the words sequentially instead of simultaneously. We allowed people to indicate when they were omitting an item. Repetitions of the lists were sparser because of the introduction of control lists. We observed an advantage for spun lists despite these changes. The spun list advantage is not an artifact of the task setting, so the ability to use item-dependent cues is likely a general property of the serial memory system.

Figure 12

Experiment 6: Rate of Omissions (A), Misorders (B), and Intrusions (C) for Each List Type as a Function of the Presentation Number of the List (Block Number)



Note. Hebb lists were presented in each block, and control lists were novel lists that did not repeat. The bars are standard errors of the means.

General Discussion

We presented the results of six experiments that adapted a serial learning procedure we used in previous research (Lindsey & Logan, 2019) to the domain of serial recall. Experiment 1 showed that people learn sequences more quickly when the relative order of items in those sequences is consistent, and Experiments 2 and 3 showed that his learning transfers to new sequences if they share the same relative order. In Experiments 4 and 5, we showed that the spun list advantage is robust to changes in item appearance or response modality. In Experiment 6, we showed that the spun list advantage is obtained in more traditional serial recall task settings. Altogether, these experiments suggest that the ability to use item-dependent retrieval cues is a general property of the serial memory system.

We aggregated the results of the six experiments and subjected them to an ANOVA. The dependent variable was the error rate advantage (or, in Experiment 6, the Hebb effect advantage) for spun lists averaged over each block of an experiment. Experiment number and detection of structural differences between lists types (as indicated by the postexperiment surveys) were included as between-subjects factors. The results of this ANOVA are presented in Table B20 in Appendix B. The magnitude of the spun list advantage did not differ significantly among the experiments, F(5, 130) =.229, MSE = .009, p = .949), and participants made 4.7% fewer errors on spun lists than scrambled lists overall.

Detection of additional structure had an overall positive effect on error rate in spun lists, F(1, 130) = 4.138, MSE = .009, p = .044): The spun list advantage was higher for participants who detected structure (advantage: 6.3%) than for those who did not (advantage: 2.8%). One-sample *t* tests (Table B21, Appendix B) revealed that the advantage for spun lists was significantly greater than 0 whether the participant detected additional structure, t(75) =7.578, SE = .008, p < .001, or not, t(65) = 2.103, SE = .013, p = .039. The learning underlying the spun list advantage can be implicit—reminiscent of the implicit serial learning observed by Hebb (1961) and many others—but explicit recognition of structure can hasten the learning and magnify the advantage.

What Is Learned in Spun and Scrambled Lists?

Throughout the article, we have alluded to four types of learning that might influence task performance: learning associations between items and item-independent representations of position (item-independent cues), learning associations between items and item-dependent representations of position (item-dependent cues), learning the list as a whole (list-level learning), and learning which items belong to a list (item-set learning). To better understand the results, it is useful to discuss how each of these types of learning may have played a role in the current study.

Item-independent cues can improve performance when items are consistently presented in the same serial position. In the current study, items in spun lists and scrambled lists were presented in several different positions. In isolation, these cues cannot explain the improvements observed in either type of list.⁵

Item-dependent cues can improve performance when items are presented in consistent relative positions. Items appeared in consistent relative positions in spun lists but not scrambled lists. Itemdependent cues can explain the improvement in spun lists but not the improvement in scrambled lists. List-level learning—or the formation of hierarchical list-level representations—can occur when the same list is presented multiple times (Yamaguchi & Logan, 2016). List-level learning improves performance by reducing interference among lists (the items in the list are activated more strongly than items in other lists). Spun lists and scrambled lists were each presented multiple times, so list level learning can explain improvements in both list types. Critically, they were presented the same number of times, so list level learning should affect both list types equally.

Item-set learning is facilitated by the use of closed item sets. Closed sets heavily restrict the pool of possible correct responses. As the set is learned, the number of viable competitors for retrieval reduces, and accurate retrieval improves as a result. Closed sets were used for both list types, and the sizes of the sets were equivalent for both list types. Item-set learning likely can explain improvements in both list types, but it cannot explain the difference in improvement.⁶

In summary, list-level learning and item-set learning explain the improvement observed in scrambled lists, and they account for some of the improvement in spun lists. Analysis of the types of errors committed in scrambled lists revealed that list-level learning and item-set learning predominantly led to reductions in intrusions and omissions. Item-dependent cues explain the additional learning advantage for spun lists, and these cues specifically lead to reductions in misorders.

More generally, the fact that participants improved at all in scrambled lists makes it clear that serial memory incorporates more than just previously retrieved item into the retrieval cue. Like others (Caplan, 2015; Fischer-Baum & McCloskey, 2015; Logan, 2021), we think that previously retrieved items are one of potentially many things that can contribute to the retrieval cue. The best model of serial memory may be one that includes item-dependent cues, item-independent cues, and other sources of learning (such as item-set learning or list-level learning).

Theories of Serial Memory: Achieving a Better Balance

When a model fails to capture some aspect of performance, we must determine whether that failure is attributable to the core assumptions of the model—in which case the model is discarded and the core assumptions are changed—or attributable to ancillary assumptions—in which case the model is revised and supporting assumptions are changed or added.

The core assumptions of item-independent theories are rarely questioned, and these theories are frequently revised to capture new

⁵ If lists are chunked, item-independent cues might predict superior overall performance in spun lists. For example, if people remembered a list as three-item chunks, then every item would preserve its within-chunk position when "spun" three positions (e.g., item A occupies the first within-chunk position in both ABC_DEF and DEF_ABC). Past serial memory research has analyzed spikes in latency and accuracy serial position curves to detect chunk boundaries (Farrell, 2012; Henson, 1996). Analysis of serial position curves in the current experiments revealed a latency spike at position 4 in the list, but there was not a corresponding spike in accuracy at this position. We suspect that people were grouping their responses at output, but they were not forming chunked representations that conformed to this grouping.

⁶ It is worth noting that, even though list-level learning and item-set learning are conceptually distinguishable, they cannot be distinguished empirically in the current study because each list exhausts the entire item set.

phenomena. As a result, simple item-independent theories (e.g., Conrad, 1965) have evolved to include additional supporting assumptions like primacy gradients, response suppression, separate stages for phonological selection, subjective grouping, and cumulative matching (Hurlstone et al., 2014).

Associative chaining is widely regarded as a failed explanation of serial memory, so when item-dependent theories fail, their core assumptions are brought under fire. Consequently, few attempts have been made to revise and add supporting assumptions to item-dependent theories (but see Logan, 2021; Solway et al., 2012). This would perhaps be fair treatment if no evidence of item-dependent cues being used in serial memory existed, but this evidence does exist—in the current study and others (e.g., Fischer-Baum & McCloskey, 2015; Kahana et al., 2010).

The importance of supporting assumptions, like the primacy gradient and response suppression, in capturing patterns of responses in serial recall has been stressed by others (e.g., Henson, 1996; Hurlstone et al., 2014). Solway et al. (2012) demonstrated the benefits of adding some of these assumptions to an associative chaining theory. In light of the explanatory power of these assumptions, there needs to be more rigorous evaluation of the failures of item-dependent theories to determine which of them can be corrected with revision and which—if any—truly render these theories untenable.

Serial Recall and Free Recall

Free recall—a list memory task in which participants can report items in any order they wish—is not commonly used to study serial memory. Conceptually, free recall requires only that participants remember the items in the list—they do not need to be remembered in any particular order (i.e., it requires item memory but not order memory; Healy, 1974). However, people seem to code the order of a list even when it is not necessary to do so, as seen by the clear similarities between serial recall and free recall performance on short lists of items (Ward et al., 2010). The implication is that the cognitive machinery underlying free recall and serial recall may not be so different.

Existing models of free recall commonly allow previously retrieved items to contribute to subsequent retrieval attempts (Howard & Kahana, 2002; Polyn et al., 2009; Raaijmakers & Shiffrin, 1980). Logan's (2021) Context Retrieval and Updating (CRU) model took inspiration from these free recall models to capture several key aspects of serial recall performance. "Serial memory" may simply be a particular parameterization of the broader memory system, and recent research—including the current study—suggests that the system can use item-dependent cues.

Beyond Serial Memory

Similar mechanisms are thought to underlie the Hebb effect and word learning (Szmalec et al., 2009), and some successful attempts have been made at capturing memory and speech phenomena with the same model (Brown et al., 2000; Burgess & Hitch, 2006; Page & Norris, 2009; Vousden et al., 2000). Whereas prior research highlights similarities between serial memory and speech production, the current experiments allow us to draw parallels between memory and other skills. Skilled motor tasks, such as playing the piano (Chaffin et al., 2009) and typing (Lindsey & Logan, 2019), are thought to use item-dependent cues. Logan (2021) successfully captured perception, memory, and typing phenomena with the same model. Interestingly, the theories used to describe commonalities among memory and speech do not used item-dependent cues, whereas those that describe commonalities between memory and skilled performance do. It may be that speech and skilled performance have distinct solutions to the problem of serial order—item-independent cues in speech and item-dependent cues in skilled performance—and memory serves as the bridge between them by virtue of using both types of cue. It may also be that speech, memory, and skilled performance all share a common mechanism of serial order, and that mechanism uses both types of cue. This strikes us as a very interesting question to answer moving forward, and the answer to this question may hinge on whether speech uses item-dependent cues like memory and skilled performance.

Conclusions

The current experiments show that lists are recalled more accurately if the relative order of items in those lists is consistent. Consistent relative order allowed item-dependent cues to support recall, and participants were able to leverage these cues to improve their recall accuracy—often without conscious awareness. Previously retrieved items can be used in serial memory retrieval.

References

- Abrahamse, E. L., Jiménez, L., Verwey, W. B., & Clegg, B. A. (2010). Representing serial action and perception. *Psychonomic Bulletin & Review*, 17(5), 603–623. https://doi.org/10.3758/PBR.17.5.603
- Botvinick, M. M., & Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, 113(2), 201–233.
- Brown, G. D., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. *Psychological Review*, 114(3), 539–576.
- Brown, G. D., Preece, T., & Hulme, C. (2000). Oscillator-based memory for serial order. *Psychological Review*, 107(1), 127–181. https://doi.org/ 10.1037/0033-295X.107.1.127
- Bryden, M. P. (1967). A model for the sequential organization of behaviour. Canadian Journal of Psychology/Revue Canadienne de Psychologie, 21(1), 37–56. https://doi.org/10.1037/h0082960
- Burgess, N., & Hitch, G. J. (2006). A revised model of short-term memory and long-term learning of verbal sequences. *Journal of Memory and Language*, 55(4), 627–652. https://doi.org/10.1016/j.jml.2006.08.005
- Caplan, J. B. (2015). Order-memory and association-memory. *Canadian Journal of Experimental Psychology*, 69(3), 221–232. https://doi.org/10.1037/cep0000052
- Chaffin, R., Logan, T. R., & Begosh, K. T. (2009). Performing from memory. In S. Hallam, I. Cross, & M. Thaut (Eds.), *The Oxford handbook of music psychology* (pp. 362–363). Oxford University Press.
- Coltheart, M. (1981). The MRC Psycholinguistic Database. The Quarterly Journal of Experimental Psychology Section A, 33(4), 497–505. https:// doi.org/10.1080/14640748108400805
- Conrad, R. (1965). Order error in immediate recall of sequences. Journal of Verbal Learning and Verbal Behavior, 4(3), 161–169.
- Crump, M. J. C., & Logan, G. D. (2013). Prevention and correction in post-error performance: An ounce of prevention, a pound of cure. *Journal of Experimental Psychology: General*, 142(3), 692–709. https://doi .org/10.1037/a0030014
- Dell, G. S. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review*, 93(3), 283–321. https://doi.org/10 .1037/0033-295X.93.3.283
- Dennis, S. (2009). Can a chaining model account for serial recall? In N. Taatgen & H. van Rijn (Eds.), *Proceedings of the XXXI Annual Conference of the Cognitive Science Society* (pp. 2813–2818). Cognitive Science Society.

- Ebbinghaus, H. (1885). Über das gedächtnis [On memory]. Duncker and Humblot.
- Ebenholtz, S. M. (1963). Serial learning: Position learning and sequential associations. *Journal of Experimental Psychology*, 66(4), 353–362. https://doi.org/10.1037/h0048320
- Estes, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161–190). Winston.
- Farrell, S. (2012). Temporal clustering and sequencing in short-term memory and episodic memory. *Psychological Review*, 119(2), 223–271. https://doi.org/10.1037/a0027371
- Farrell, S., Hurlstone, M. J., & Lewandowsky, S. (2013). Sequential dependencies in recall of sequences: Filling in the blanks. *Memory & Cognition*, 41(6), 938–952. https://doi.org/10.3758/s13421-013-0310-0
- Fischer-Baum, S., & McCloskey, M. (2015). Representation of item position in immediate serial recall: Evidence from intrusion errors. *Journal of Experimental Psychology*, 41(5), 1426–1446. https://doi.org/10.1037/xlm0000102
- Healy, A. F. (1974). Separating item from order information in short-term memory. *Journal of Verbal Learning and Verbal Behavior*, 13(6), 644–655. https://doi.org/10.1016/S0022-5371(74)80052-6
- Hebb, D. O. (1961). Distinctive features of learning in the higher animal. In B. F. Delafresnaye (Ed.), *Brain mechanisms and learning* (pp. 37–46). Blackwell.
- Henson, R. N. A. (1996). Short-term memory for serial order [Unpublished doctoral dissertation]. University of Cambridge.
- Henson, R. N. A. (1998). Short-term memory for serial order: The Start-End Model. Cognitive Psychology, 36(2), 73–137. https://doi.org/10.1006/ cogp.1998.0685
- Hitch, G. J., Fastame, M. C., & Flude, B. (2005). How is the serial order of a verbal sequence coded? Some comparisons between models. *Memory*, 13(3-4), 247–258. https://doi.org/10.1080/09658210344000314
- Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, 46(3), 269–299.
- Hurlstone, M. J., Hitch, G. J., & Baddeley, A. D. (2014). Memory for serial order across domains: An overview of the literature and directions for future research. *Psychological Bulletin*, 140(2), 339–373. https://doi .org/10.1037/a0034221
- Kahana, M. J., Mollison, M. V., & Addis, K. M. (2010). Positional cues in serial learning: The spin-list technique. *Memory & Cognition*, 38(1), 92–101. https://doi.org/10.3758/MC.38.1.92
- Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior* (pp. 112–146). Wiley.
- Lewandowsky, S., & Farrell, S. (2008). Short-term memory: New data and a model. *Psychology of Learning and Motivation*, 49, 1–48. https://doi .org/10.1016/S0079-7421(08)00001-7
- Lewandowsky, S., & Murdock, B. B. (1989). Memory for serial order. Psychological Review, 96(1), 25–57. https://doi.org/10.1037/0033-295X.96.1.25
- Lindsey, D. R. B., & Logan, G. D. (2019). Item-to-item associations in typing: Evidence from spin list sequence learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45(3), 397–416. https://doi.org/10.1037/xlm0000605
- Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, 95(4), 492–527. https://doi.org/10.1037/0033-295X.95.4.492
- Logan, G. D. (2018). Automatic control: How experts act without thinking. Psychological Review, 125(4), 453–485. https://doi.org/10.1037/rev0000100
- Logan, G. D. (2021). Serial order in perception, memory, and action. *Psy-chological Review*, 128(1), 1–44. https://doi.org/10.1037/rev0000253
- Logan, G. D., Ulrich, J. E., & Lindsey, D. R. B. (2016). Different (key) strokes for different folks: How standard and nonstandard typists balance Fitts' law and Hick's law. *Journal of Experimental Psychology: Human Perception and Performance*, 42(12), 2084–2102. https://doi .org/10.1037/xhp0000272

- MathWorks. (2018). MATLAB R2018a. https://www.mathworks.com/ products/new_products/release2018a.html
- Miller, G. A., Galanter, E., & Pribram, K. H. (1960). Plans and the structure of behavior. Henry Holt and Co. https://doi.org/10.1037/10039-000
- Murdock, B. B. (1995). Developing TODAM: Three models for serialorder information. *Memory & Cognition*, 23(5), 631–645. https://doi .org/10.3758/BF03197264
- Osth, A. F., & Dennis, S. (2015). The fill-in effect in serial recall can be obscured by omission errors. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 41(5), 1447–1455. https://doi.org/10 .1037/xlm0000113
- Page, M. P., Cumming, N., Norris, D., McNeil, A. M., & Hitch, G. J. (2013). Repetition-spacing and item-overlap effects in the Hebb repetition task. *Journal of Memory and Language*, 69(4), 506–526. https://doi .org/10.1016/j.jml.2013.07.001
- Page, M. P. A., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review*, 105(4), 761–781. https:// doi.org/10.1037/0033-295X.105.4.761-781
- Page, M. P. A., & Norris, D. (2009). A model linking immediate serial recall, the Hebb repetition effect and the learning of phonological word forms. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 364(1536), 3737–3753. https://doi.org/10 .1098/rstb.2009.0173
- Polyn, S. M., Norman, K. A., & Kahana, M. J. (2009). A context maintenance and retrieval model of organizational processes in free recall. *Psychological Review*, 116(1), 129–156. https://doi.org/10.1037/a0014420
- Psychology Software Tools. (2012). E-Prime 2.0. https://pstnet.com/ products/e-prime/
- Raaijmakers, J. G. W., & Shiffrin, R. (1980). SAM: A theory of probabilistic search of associative memory. In G. Bower (Ed.), *The psychology of learning and motivation* (Vol. 14, pp. 207–262). Academic Press.
- Rosenbaum, D. A., Inhoff, A. W., & Gordon, A. M. (1984). Choosing between movement sequences: A hierarchical editor model. *Journal of Experimental Psychology: General*, 113(3), 372–393. https://doi.org/10 .1037/0096-3445.113.3.372
- Rosenbaum, D. A., Kenny, S. B., & Derr, M. A. (1983). Hierarchical control of rapid movement sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 9(1), 86–102. https://doi.org/10 .1037/0096-1523.9.1.86
- Schwartz, M., & Bryden, M. (1971). Coding factors in the learning of repeated digit sequences. *Journal of Experimental Psychology*, 87(3), 331–334. https://doi.org/10.1037/h0030552
- Solway, A., Murdock, B. B., & Kahana, M. J. (2012). Positional and temporal clustering in serial order memory. *Memory & Cognition*, 40(2), 177–190. https://doi.org/10.3758/s13421-011-0142-8
- Szmalec, A., Duyck, W., Vandierendonck, A., Mata, A. B., & p, M. P. (2009). The Hebb repetition effect as a laboratory analogue of novel word learning. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, 62(3), 435–443. https://doi.org/10.1080/17470210802386375
- Tukey, J. W. (1977). Exploratory data analysis. Addison Wesley.
- Vousden, J. I., Brown, G. D., & Harley, T. A. (2000). Serial control of phonology in speech production: A hierarchical model. *Cognitive Psychology*, 41(2), 101–175. https://doi.org/10.1006/cogp.2000.0739
- Ward, G., Tan, L., & Grenfell-Essam, R. (2010). Examining the relationship between free recall and immediate serial recall: The effects of list length and output order. *Journal of Experimental Psychology: Learning, Mem*ory, and Cognition, 36(5), 1207–1241. https://doi.org/10.1037/a0020122
- Yamaguchi, M., & Logan, G. D. (2016). Pushing typists back on the learning curve: Memory chunking in the hierarchical control of skilled typewriting. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 42(12), 1919–1936. https://doi.org/10.1037/xlm0000288

Appendix A

Supplemental Method

Power Function Fits (Experiments 1–6)

Matching Word Set Characteristics (Experiment 6)

Function parameters were estimated by minimizing the root mean square deviation (RMSD) between the observed data and the function predicted data using the fminsearch simplex optimization routine in MATLAB R2018a (MathWorks, 2018). Fits were done separately for each participant, dependent measure (accuracy and latency), and experiment portion (training and test), but they were done simultaneously for each list type. Simplex was run 50,000 times using random starting parameter values to avoid local minima, and parameter estimates were taken from the iteration that produced the smallest RMSD.

The *a* parameter in Equation 1 was fixed to a constant value for all participants. For error rate, *a* was set to 0 because peak performance is making zero errors. For latency measures, *a* was set to 150, which is a reasonable upper limit for the speed of typing novel strings.⁷ The *b* parameter was estimated by the fitting routine. For training portion data, each participant's *b* parameter was constrained to be the same value for all list types. For test portion data, it was allowed to differ among the list types. The *b* parameter was constrained to take positive values, and for error rate it was restricted to be between 0 and 1. The *c* parameter was also estimated by the fitting routine, and it was always allowed to differ among the list types. The *c* parameter was constrained to take positive values between 0 and 1. We filtered the initial pool of words obtained from the MRC database. We kept words that were between 200 and 700 in concreteness and familiarity ratings, and between 1 and 100 in frequency rating. We then removed proper nouns, offensive words, and words with high frequency until we obtained the final pool of 210 words. This pool was divided into 35 sets of 6 words. Table A1 shows each of the sets and the characteristics of each word in the set.

We attempted to roughly match each of the 35 word sets on average concreteness, familiarity, and frequency. Matching was done with a Matlab script. We set tolerance values on the means and standard deviations of each characteristic. These tolerance values determined the maximum acceptable difference between the means and standard deviations of any two word sets. The program selected words at random until all 35 word sets fell within the tolerance bounds, or until it failed to find a solution after 100,000 attempts. We tried many different tolerance values and settled on the lowest values that would produce a solution. Tolerance values were 10, 10, and 2 for the means of concreteness, familiarity, and frequency, respectively. Tolerance values were 30, 30, and 6 for the standard deviations of concreteness, familiarity, and frequency, respectively.

Table A1

Experiment 6: Word Sets Used in the Hebb Repetition Task

	Set	t 1			Se	t 2			Set	t 3			Se	t 4			Se	t 5	
Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.
plea knob	308 586 532	478 534 423	11 2 3	bolt harp	556 591 581	466 430 521	10 1 20	gain doll	346 588 542	543 503 501	74 10 7	tale golf	352 540 603	499 503 522	21 34 5	card task	565 409 611	543 514 434	26 60 5
rung seed wind grip	552 611 552 490	423 514 592 523	41 63 20	cape kill burn lift	386 490 461	549 548 555	20 63 15 23	lump claw slit self	542 587 520 459	445 487 604	1 6 40	stew dust duck hind	603 550 606 474	522 558 529 416	5 70 9 6	cock tape cult herb	564 349 558	434 567 437 514	35 11 7
M SD	513.2 109.0	510.7 56.7	23.3 24.2	M SD	510.8 79.9	511.5 51.8	22.0 21.5	M SD	507.0 92.3	513.8 54.3	23.0 28.6	M SD	520.8 95.8	504.5 48.2	24.2 25.0	M SD	509.3 104.5	501.5 54.8	24.0 21.2
	Set	t 6			Se	t 7			Set	t 8			Se	t 9			Set	10	
Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.
hill disc thud myth crow stem	588 553 423 334 590 556	585 466 456 514 490 513	72 6 3 35 2 29	wave ware moth hose kick shop	492 368 550 596 485 549	518 394 496 449 563 615	46 1 1 9 16 63	boot vein safe tang hail star	595 553 376 478 502 574	566 496 531 436 440 574	13 25 58 4 10 25	heir heel mine hump band tent	384 579 452 507 590 608	453 524 557 439 555 521	7 9 59 2 53 20	drum mole arch hole fool tomb	602 590 512 485 354 573	506 484 483 545 551 450	11 4 13 58 37 11
M SD	507.3 104.8	504.0 46.2	24.5 27.2	M SD	506.7 79.5	505.8 79.0	22.7 25.9	M SD	513.0 80.2	507.2 60.3	22.5 19.3	M SD	520.0 88.8	508.2 50.6	25.0 24.8	M SD	519.3 93.0	503.2 39.1	22.3 20.8
	Set	11			Set	12			Set	13			Set	14			Set	15	
Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.
nail ease	598 305	563 519	6 42	yawn pole	502 577	567 510	2 18	pump peck	556 432	497 471	11 5	leak pope	472 593	514 489	2 40	belt pear	602 634	550 567	29 6

 $^{^{7}}$ Assuming an average word length of 5 letters (e.g., Logan et al., 2016) and equivalent keystroke timing for each letter in the word, the time to type one word would be 150*5 = 750 ms. This is equivalent to one word per 0.75 s, one word per 0.0125 minutes, and 80 words per minute. This is an above average copy typing speed for participants in the current experiments.

 Table A1 (continued)

1 abie	A1 (cc Set		u)	g. Word Conc. Fam. Freq.				Set 13				Set 14				Set 15			
	Conc.	Fam.	Freq.		_	_	Freq.		Conc.	_	Freq.		Conc.	_		_	Conc.	Fam.	Free
ring	593 451	589 583	47 40	edge	465	569 515	78	pond	623 633	506 519	25 7	moss	575 356	436 545	9 9	trim	388 581	456 539	20 31
inch lace	431 545	383 468	40 7	gown gait	586 381	369	16 8	lamb cell	542	520	65	pour mare	530 549	343 460	16	tube sash	540	385	31
hive	583	386	2	bell	620	543	18	fail	327	568	37	song	514	603	70	save	314	559	62
M	512.5	518.0	24.0	M	521.8	512.2	23.3	М	518.8	513.5	25.0	М	509.8	507.8	24.3	M	509.8	509.3	25.2
SD	115.5	79.0	21.0	SD	89.7	74.4	27.5	SD	118.4	32.2	23.1	SD	87.0	60.5	26.0	SD	128.9	73.0	21.4
	Set	16			Set	17			Set	18			Set	19			Set	20	
Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Free
text	498	511	60	reel	502	456	2	coal	584	513	32	yard	553	522	35	foal	420	338	2
surf	527	526	1	hate	335	552	42	slap	511	521	2	fare	413	482	7	pipe	602	535 519	20 12
cool lice	364 543	567 397	62 2	snow	618 531	615 481	59 21	soup jade	615 570	576 359	16 1	tail dusk	613 455	533 474	24 9	lens pale	573 385	499	58
plum	632	547	1	port tune	464	545	10	hook	525	497	5	draw	433	542	56	male	565 564	588	37
bark	563	518	14	yolk	593	471	1	fast	304	600	78	boar	558	455	1	cave	592	526	9
M	521.2	511.0	23.3	 M	507.2	520.0	22.5		518.2	511.0	22.3	 M	505.7	501.3	22.0	 M	522.7	500.8	23.0
SD	89.2	59.5	29.6	SD	101.7	61.1	23.5	SD	111.7	84.3	29.7	SD	79.6	35.6	20.8	SD	94.7	85.1	20.9
	Set	21			Set	22			Set	23			Set	24			Set	25	
Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq
coat	601	610	43	loop	490	494	21	slip	448	537	19	join	292	544	65	site	408	507	64
sale	364	555	44	shoe	600	569	14	zeal	304	388	8	wrap	457	549	5	pill	610	556	15
germ	464	523	3	hint	312	555	9	camp	571	541	75	mast	576	427	6	gang	492	515	22
crew	523 581	442 433	36 1	neck	587 437	576 452	81 2	bath	600 608	599 540	26 10	dirt	564	571 542	43 1	horn	618 402	498 451	31 1
fawn rust	553	433 484	10	tuck vine	437 601	432 411	2 4	wool fuse	500	340 449	5	weed silk	600 538	482	12	peal dent	402 517	431 480	2
M	514.3	507.8	22.8	M	504.5	509.5	21.8	M	505.2	509.0	23.8	M	504.5	519.2	22.0	M	507.8	501.2	22.5
SD	87.9	68.4	20.3	SD	115.8	68.2	29.8	SD	116.3	76.3	26.2	SD	115.2	54.0	26.0	SD	93.9	35.2	23.4
	Set				Set				Set				Set				Set		
	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.	Word	Conc.	Fam.	Freq.		Conc.	Fam.	Freq.	_	Conc.	Fam.	Free
gulf	449 568	406 516	22 4	vote	389 552	567 520	75 10	watt	350 557	436	2 20	suds	554 242	436	9 41	meal	602 407	603 487	30
toad tree	568 604	613	4 59	scar vest	552 575	529 472	4	nest boil	557 467	521 533	12	sell wage	342 451	585 539	41 56	raid rake	407 597	487 476	10 11
zone	392	456	11	lane	537	525	30	rice	608	548	33	dime	582	586	4	bale	462	386	5
knee	593	599	35	lime	590	447	13	brim	509	430	4	lint	537	479	4	foot	558	583	70
prop	474	444	7	toll	424	485	16	nose	628	584	60	foam	577	462	37	heap	485	519	14
M SD	513.3 87.1	505.7 85.5	23.0 21.0	M SD	511.2 83.8	504.2 44.0	24.7 26.1	M SD	519.8 102.6	508.7 62.3	21.8 21.9	M SD	507.2 93.8	514.5 64.6	25.2 22.4	M SD	518.5 79.3	509.0 78.9	23.3 24.4
	Set	31			Set	32			Set	33			Set	: 34			Set	35	
Word	Conc.		Freq.	Word			Freq.	Word	Conc.		Freq.	Word	Conc.		Freq.	Word	Conc.		Free
cane	590	442	12	veil	560	421	8	plug	558	575	23	milk	670	588	49	skin	614	591	47
salt	594	612	46	leap	389	539	14	womb	538	508	1	hoof	596	401	2	rite	381	400	8
meat	587	589	45	tide	516	504	11	duke	508	457	11	lung	569	546	16	wood	606	574	55
calm	360	547	35	fair	413	573	77	rush	350	546	20	dear	326	536	54	lure	422	451	7
jerk	441	452	2	lark	578	436	2	mate	507	553	21	peel	432	507	3	haze	509	484	7
fowl	532	452	1	beef	637	598	32	fort	580	418	55	dome	517	504	17	bush	585	532	14
М	517.3	515.7	23.5	М	515.5	511.8	24.0	М	506.8	509.5	21.8	М	518.3	513.7	23.5	М	519.5	505.3	23.0
SD	96.8	76.4	21.0	SD	97.1	72.1	27.9	SD		61.1	18.2	SD	123.3	63.1	22.6	SD	99.5	73.8	22.0

Note. conc. = concreteness; fam. = familiarity; freq. = frequency.

Appendix B

Result Tables

Result tables are listed below, split by experiment. Each experiment has three result tables: The first presents F tests for aggregate measures of error rate and latency, the second presents t tests on initial task performance and power function parameters, and the third presents F tests on the rates of omissions, misorders, and intrusions. Subheaders parse each of the tables by specific effects—for example, tests under "Spun Versus Scrambled Performance" indicate whether

there is an advantage for spun lists. Experiment 5 has a fourth table with results from Wilcoxon signed ranks tests; these tests are duplicates of important tests from the other three tables, but the analyses in the fourth table are robust to an outlier that was detected in Experiment 5. Refer to the Method sections in the article for information on how these results were obtained, and refer to Results sections for interpretations of these results.

Experiment 1

Table B1

Experiment 1: ANOVA and Simple Main Effect Analyses for Training Effects

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled performance						
Error rate	7.924	1, 23	0.211	.010	1.000	0.256
IT	0.013	1,23	1,037,908	.909	1.000	0.001
IRT	2.184	1,23	140,065	.153	1.000	0.087
Learning (Spun)						
Error rate	2.898	39, 897	0.005	<.001	0.575	0.112
IT	29.198	39, 897	11,820	<.001	0.273	0.559
IRT	16.000	39, 897	1,889	<.001	0.617	0.410
Learning (Scrambled)						
Error rate	1.364	39, 897	0.005	.070	0.257	0.056
IT	25.068	39, 897	11,820	<.001	0.322	0.522
IRT	14.416	39, 897	1,889	<.001	0.203	0.385

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time. Learning analyses are simple main effects analyses of presentation number that use the *MSE* of the List Type \times Presentation interaction. The simple main effect analyses are not corrected for violations of sphericity, but Huynh-Feldt epsilon values are provided to gauge the severity of these violations.

Table B2

Experiment 1: t Tests for First Block Performance and Power Function Learning Rate

DV	t	df	$M_{ m A-B}$	SE_{A-B}	р	d	BF
Spun vs. Scrambled first block performance							
Error rate	-1.080	23	-0.028	0.026	.291	-0.481	2.766 (N)
IT	-0.300	23	-14.392	47.975	.767	-0.043	4.471 (N)
IRT	0.726	23	11.910	16.404	.475	0.121	3.669 (N)
Spun vs. Scrambled Learning Rate							
Error rate	2.807	23	0.092	0.033	.010	0.687	4.839 (A)
IT	0.224	23	0.008	0.034	.825	0.056	4.553 (N)
IRT	1.285	23	0.050	0.039	.212	0.384	2.243 (N)

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time. Learning rate analyses are conducted on the c parameters obtained from power function fits. For Bayes factors, numbers followed by (A) indicate evidence in favor of the alternative hypothesis, and numbers followed by (N) indicate evidence in favor of the null hypothesis.

(Appendices continue)

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled Performance						
Omission	0.915	1,23	332.318	.349	1.000	0.038
Misorder	13.433	1, 23	430.985	.001	1.000	0.369
Intrusion	0.001	1, 23	325.306	.978	1.000	0.000
Learning (Spun)						
Omission	0.659	39, 897	8.503	.948	0.239	0.028
Misorder	1.895	39, 897	19.135	.001	0.765	0.076
Intrusion	2.300	39, 897	3.830	<.001	0.490	0.091
Learning (Scrambled)						
Omission	1.677	39, 897	8.503	.006	0.037	0.068
Misorder	1.139	39, 897	19.135	.259	0.657	0.047
Intrusion	3.210	39, 897	3.830	<.001	0.210	0.122

Note. DV = dependent variable. Italicized *p* values are nonsignificant after correcting for violations of sphericity with the Huynh-Feldt epsilon value.

Experiment 2

Table B4

Experiment 2: ANOVA and Simple Main Effect Analyses for Training and Transfer Effects

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled Performance (Training)						
Error rate	4.470	1, 23	0.119	.046	1.000	0.163
IT	1.278	1, 23	954,735	.270	1.000	0.053
IRT	0.427	1, 23	46,920	.520	1.000	0.018
Learning in Training (Spun)						
Error rate	2.514	39, 897	0.007	<.001	0.454	0.099
IT	34.098	39, 897	14,721	<.001	0.367	0.597
IRT	21.348	39, 897	2,427	<.001	0.689	0.481
Learning in Training (Scrambled)						
Error rate	1.479	39, 897	0.007	.031	0.535	0.060
IT	36.304	39, 897	14,721	<.001	0.238	0.612
IRT	18.282	39, 897	2,427	<.001	0.636	0.443
Spun vs. Scrambled Performance (Test)						
Error rate	22.480	1, 23	0.040	<.001	1.000	0.494
IT	2.049	1, 23	196,948	.166	1.000	0.082
IRT	4.916	1, 23	27,887	.037	1.000	0.176
Presence of Transfer (Spun)						
Error rate	0.014	1, 23	0.007	.906	1.000	0.001
IT	65.862	1, 23	67,822	<.001	1.000	0.741
IRT	18.761	1, 23	17,703	<.001	1.000	0.449
Presence of Transfer (Scrambled)						
Error rate	44.179	1, 23	0.007	<.001	1.000	0.658
IT	20.560	1, 23	67,822	<.001	1.000	0.472
IRT	3.002	1, 23	17,703	.097	1.000	0.115
Completeness of Transfer (Spun)						
Error rate	11.029	1, 23	0.012	.003	1.000	0.324
IT	21.283	1, 23	74,464	<.001	1.000	0.481
IRT	25.333	1, 23	11,766	<.001	1.000	0.524
Completeness of Transfer (Scrambled)		*	*			
Error rate	77.431	1,23	0.012	<.001	1.000	0.771
IT	74.223	1, 23	74,464	<.001	1.000	0.763
IRT	59.527	1, 23	11,766	<.001	1.000	0.721

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time. Presence and completeness of transfer analyses are simple main effects analyses comparing test performance to initial and final training performance and use the *MSE* of the List Type \times Start Portion and List Type \times End Portion interactions, respectively.

Table	B5
-------	-----------

Experiment 2: t Tests for First Block Performance and Power Function Parameters

DV	t	df	$M_{ ext{A-B}}$	SE_{A-B}	р	d	BF
Spun vs. Scrambled First Block Performance							
Error rate	-0.574	23	-0.018	0.032	.571	-0.157	4.010 (N)
IT	0.266	23	14.026	52.794	.793	0.035	4.510 (N)
IRT	0.893	23	21.715	24.317	.381	0.214	3.253 (N)
Spun vs. Scrambled Learning Rate (Training)							
Error rate	2.427	23	0.082	0.034	.023	0.504	2.388 (A)
IT	-0.633	23	-0.027	0.042	.533	-0.134	3.883 (N)
IRT	0.633	23	0.046	0.072	.533	0.190	3.883 (N)
Spun vs. Scrambled Starting Point (Test)							
Error rate	-4.385	23	-0.131	0.030	<.001	-1.031	136.163 (A)
IT	-2.554	23	-138.606	54.279	.018	-0.331	3.002 (A)
IRT	-1.741	23	-44.005	25.270	.095	-0.393	1.261 (N)
Spun vs. Scrambled Learning Rate (Test)							
Error rate	-0.124	23	-0.007	0.060	.902	-0.026	4.626 (N)
IT	-0.791	23	-0.057	0.071	.437	-0.208	3.512 (N)
IRT	0.328	23	0.023	0.071	.746	0.088	4.435 (N)

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time. Starting point analyses are conducted on the b parameters obtained from power function fits.

Table B6

Experiment 2: ANOVA and Simple Main Effect Analyses for Each Error Type

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled Performance (Training)						
Omission	0.000	1,23	6.012	.993	1.000	0.000
Misorder	5.538	1,23	147.665	.028	1.000	0.194
Intrusion	0.510	1,23	81.716	.482	1.000	0.022
Learning in Training (Spun)						
Omission	1.801	39, 897	1.026	.002	0.192	0.073
Misorder	1.553	39, 897	11.721	.018	0.505	0.063
Intrusion	4.126	39, 897	1.907	<.001	0.717	0.152
Learning in Training (Scrambled)						
Omission	2.141	39, 897	1.026	<.001	0.160	0.085
Misorder	1.119	39, 897	11.721	.286	0.691	0.046
Intrusion	4.219	39, 897	1.907	<.001	0.247	0.155
Spun vs. Scrambled Performance (Test)						
Omission	0.250	1,23	5.007	.622	1.000	0.011
Misorder	16.906	1,23	17.873	<.001	1.000	0.424
Intrusion	3.514	1,23	4.982	.074	1.000	0.133

Note. DV = dependent variable.

(Appendices continue)

Experiment 3

Table B7

Experiment 3: ANOVA and Simple Main Effect Analyses for Training and Transfer Effects

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled Performance (Training)						
Error rate	1.378	1,23	0.166	.252	1.000	0.057
IT	0.173	1,23	874,796	.682	1.000	0.007
IRT	0.173	1,23	156,765	.682	1.000	0.007
Learning in Training (Spun)						
Error rate	2.720	39, 897	0.012	<.001	0.783	0.106
IT	8.004	39, 897	45,358	<.001	0.362	0.258
IRT	7.449	39, 897	5,098	<.001	0.526	0.245
Learning in Training (Scrambled)						
Error rate	1.864	39, 897	0.012	.001	0.604	0.075
IT	15.047	39, 897	45,358	<.001	0.254	0.395
IRT	7.193	39, 897	5,098	<.001	0.707	0.238
Spun vs. Scrambled Performance (Test)						
Error rate	21.796	1,23	0.021	<.001	1.000	0.487
IT	2.730	1,23	75,986	.112	1.000	0.106
IRT	4.822	1,23	13,612	.038	1.000	0.173
Presence of Transfer (Spun)						
Error rate	0.003	1,23	0.028	.954	1.000	0.000
IT	23.665	1,23	123,050	<.001	1.000	0.507
IRT	17.924	1,23	9,873	<.001	1.000	0.438
Presence of Transfer (Scrambled)						
Error rate	11.166	1,23	0.028	.003	1.000	0.327
IT	28.924	1, 23	123,050	<.001	1.000	0.557
IRT	5.487	1, 23	9,873	.028	1.000	0.193
Completeness of Transfer (Spun)						
Error rate	36.448	1,23	0.012	<.001	1.000	0.613
IT	12.745	1,23	93,489	.002	1.000	0.357
IRT	21.065	1,23	12,537	<.001	1.000	0.478
Completeness of Transfer (Scrambled)						
Error rate	117.249	1,23	0.012	<.001	1.000	0.836
IT	48.532	1,23	93,489	<.001	1.000	0.678
IRT	41.222	1, 23	12,537	<.001	1.000	0.642

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time.

Table B8

Experiment 3: t Tests for First Block Performance and Power Function Parameters

DV	t	df	$M_{ ext{A-B}}$	SE_{A-B}	р	d	BF
Spun vs. Scrambled First Block Performance							
Error rate	-0.412	23	-0.014	0.034	.684	-0.086	4.311 (N)
IT	-1.014	23	-92.938	91.649	.321	-0.180	2.938 (N)
IRT	-0.498	23	-13.870	27.824	.623	-0.090	4.161 (N)
Spun vs. Scrambled Learning Rate (Training)							
Error rate	0.957	23	0.038	0.040	.349	0.198	3.087 (N)
IT	-0.338	23	-0.010	0.031	.738	-0.067	4.422 (N)
IRT	1.778	23	0.092	0.052	.089	0.384	1.197 (N)
Spun vs. Scrambled Starting Point (Test)							
Error rate	-2.855	23	-0.067	0.024	.009	-0.484	5.311 (A)
IT	-0.378	23	-11.884	31.459	.709	-0.026	4.364 (N)
IRT	-2.434	23	-35.977	14.780	.023	-0.278	2.418 (A)
Spun vs. Scrambled Learning Rate (Test)							
Error rate	1.861	23	0.077	0.041	.076	0.322	1.060 (N)
IT	1.928	23	0.109	0.057	.066	0.465	1.044 (A)
IRT	0.697	23	0.044	0.063	.492	0.152	3.738 (N)

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time.

Table B9	
Experiment 3: ANOVA and Simple Main Ef	ffect Analyses for Each Error Type

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled Performance (Training)						
Omission	0.425	1,23	8.652	.521	1.000	0.018
Misorder	0.302	1, 23	62.812	.588	1.000	0.013
Intrusion	1.515	1,23	17.873	.231	1.000	0.062
Learning in Training (Spun)		, -				
Omission	1.087	39, 897	0.554	.332	0.206	0.045
Misorder	1.516	39, 897	5.089	.024	0.845	0.062
Intrusion	3.459	39, 897	0.813	<.001	0.267	0.131
Learning in Training (Scrambled)		· · · · · · · · · · · · · · · · · · ·				
Omission	2.111	39.897	0.554	<.001	0.185	0.084
Misorder	0.966	39, 897	5.089	.530	0.678	0.040
Intrusion	4.384	39, 897	0.813	<.001	0.362	0.160
Spun vs. Scrambled Performance (Test)		,,	01012		0.002	01100
Omission	1.919	1,23	9.119	.179	1.000	0.077
Misorder	16.337	1, 23	44.388	.001	1.000	0.415
Intrusion	0.084	1, 23	21.924	.775	1.000	0.004

Note. DV = dependent variable.

Experiment 4

Table B10	
Experiment 4: ANOVA and Simple Main Effect Analyses for Training and Transfer Effects	

DV	F	dfs	MSE	р	3	η_p^2
Letter Case Order						
Error rate	0.039	1, 22	1.447	.845	1.000	0.002
IT	0.107	1, 22	9,721,479	.747	1.000	0.005
IRT	0.028	1, 22	660,432	.869	1.000	0.001
Spun vs. Scrambled Performance (Training)						
Error rate	8.007	1, 22	0.154	.010	1.000	0.267
IT	1.727	1, 22	985,374	.202	1.000	0.073
IRT	3.676	1, 22	110,347	.068	1.000	0.143
Learning in Training (Spun)						
Error rate	2.262	39, 858	0.008	<.001	0.479	0.093
IT	15.058	39, 858	20,424	<.001	0.427	0.406
IRT	16.878	39, 858	2,798	<.001	0.475	0.434
Learning in Training (Scrambled)						
Error rate	2.874	39, 858	0.008	<.001	0.659	0.116
IT	11.626	39, 858	20,424	<.001	0.329	0.346
IRT	13.148	39, 858	2,798	<.001	0.509	0.374
Spun vs. Scrambled Performance (Test)						
Error rate	19.861	1,22	0.026	<.001	1.000	0.474
IT	2.340	1,22	261,965	.140	1.000	0.096
IRT	3.487	1, 22	23,630	.075	1.000	0.137
Presence of Transfer (Spun)		<i>,</i>	,			
Error rate	13.835	1,22	0.023	.001	1.000	0.386
IT	127.814	1, 22	63,301	<.001	1.000	0.853
IRT	181.003	1, 22	8,197	<.001	1.000	0.892
Presence of Transfer (Scrambled)		,	-,			
Error rate	10.956	1,22	0.023	.003	1.000	0.332
IT	172.246	1, 22	63,301	<.001	1.000	0.887
IRT	254.100	1, 22	8,197	<.001	1.000	0.920
Completeness of Transfer (Spun)		,	- ,			
Error rate	0.061	1,22	0.009	.807	1.000	0.003
IT	2.729	1, 22	15,628	.113	1.000	0.110
IRT	1.426	1, 22	2,715	.245	1.000	0.061
Completeness of Transfer (Scrambled)		-,	_,, 10	.2.0	1.000	0.001
Error rate	4.514	1,22	0.009	.045	1.000	0.170
IT	67.908	1, 22	15,628	<.001	1.000	0.755
IRT	45.265	1, 22	2,715	<.001	1.000	0.673

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time.

Experiment 5

Table B11

Experiment 4: t Tests for First Block Performance and Power Function Parameters

DV	t	df	$M_{ ext{A-B}}$	SE_{A-B}	р	d	BF
Spun vs. Scrambled First Block Performance							
Error rate	-0.471	23	-0.016	0.034	.642	-0.098	4.210 (N)
IT	0.258	23	11.882	46.123	.799	0.032	4.519 (N)
IRT	0.223	23	5.783	25.915	.825	0.043	4.554 (N)
Spun vs. Scrambled Learning Rate (Training)							
Error rate	3.075	23	0.207	0.067	.005	0.823	8.206 (A)
IT	1.018	23	0.035	0.035	.160	0.192	2.927 (N)
IRT	1.318	23	0.093	0.071	.100	0.339	2.161 (N)
Spun vs. Scrambled Starting Point (Test)							
Error rate	-3.546	23	-0.060	0.017	.003	-0.397	21.792 (A)
IT	-1.099	23	-70.019	63.726	.283	-0.184	2.717 (N)
IRT	-1.855	23	-27.623	14.893	.076	-0.333	1.070 (N)
Spun vs. Scrambled Learning Rate (Test)							
Error rate	1.118	23	0.054	0.048	.550	0.328	2.668 (N)
IT	0.960	23	0.063	0.065	.347	0.253	3.080 (N)
IRT	0.575	23	0.069	0.120	.571	0.171	4.008 (N)

Note. DV = dependent variable; IT = initiation time; IRT = interresponse time.

Table B12

Experiment 4: ANOVA and Simple Main Effect Analyses for Training and Transfer Effects

DV	F	dfs	MSE	p	з	η_p^2
Spun vs. Scrambled Performance (Training)						
Omission	0.016	1,23	99.692	.901	1.000	0.001
Misorder	4.824	1,23	151.358	.038	1.000	0.173
Intrusion	0.035	1,23	177.922	.854	1.000	0.002
Learning in Training (Spun)						
Omission	1.432	39, 897	1.483	.044	0.157	0.059
Misorder	2.547	39, 897	8.201	<.001	0.346	0.100
Intrusion	2.624	39, 897	1.212	<.001	0.188	0.102
Learning in Training (Scrambled)						
Omission	2.292	39, 897	1.483	<.001	0.130	0.091
Misorder	1.608	39, 897	8.201	.011	0.253	0.065
Intrusion	5.316	39, 897	1.212	<.001	0.213	0.188
Spun vs. Scrambled Performance (Test)						
Omission	0.032	1,23	62.306	.859	1.000	0.001
Misorder	24.236	1,23	33.256	<.001	1.000	0.513
Intrusion	0.002	1, 23	9.858	.966	1.000	0.000

Note. DV = dependent variable. Italicized p values are nonsignificant after correcting for violations of sphericity with the Huynh-Feldt epsilon value.

(Appendices continue)

Experiment 6

Table B13

Experiment 5: ANOVA and Simple Main Effect Analyses for Training and Transfer Effects

DV	F	dfs	MSE	р	3	η_p^2
Response Method Order						
Error rate	2.079	1,22	1.441	.163	1.000	0.086
Time taken	4.722	1,20	33,435,559	.042	1.000	0.191
Spun vs. Scrambled Performance (Training)						
Error rate	2.295	1,22	0.318	.144	1.000	0.094
Time taken	4.280	1,20	11,048,174	.052	1.000	0.176
Learning in Training (Spun)						
Error rate	3.411	39, 858	0.008	<.001	0.272	0.134
Time taken	22.593	39, 780	145,174	<.001	0.214	0.530
Learning in Training (Scrambled)						
Error rate	2.978	39, 858	0.008	<.001	0.176	0.119
Time taken	14.666	39, 780	145,174	<.001	0.240	0.423
Spun vs. Scrambled Performance (Test)						
Error rate	5.833	1,22	0.095	.024	1.000	0.210
Time taken	2.353	1,20	2,839,515	.141	1.000	0.105
Presence of Transfer (Spun)						
Error rate	63.583	1,22	0.013	<.001	1.000	0.743
Time taken	41.961	1,20	749,320	<.001	1.000	0.677
Presence of Transfer (Scrambled)						
Error rate	19.335	1,22	0.013	<.001	1.000	0.468
Time taken	33.561	1,20	749,320	<.001	1.000	0.627
Completeness of Transfer (Spun)						
Error rate	8.826	1,22	0.015	.007	1.000	0.286
Time taken	22.149	1,20	782,445	<.001	1.000	0.525
Completeness of Transfer (Scrambled)						
Error rate	0.176	1,22	0.015	.679	1.000	0.008
Time taken	7.586	1, 20	782,445	.012	1.000	0.275

Note. DV = dependent variable.

Table B14

Experiment 5: t Tests for First Block Performance and Power Function Parameters

DV	t	df	$M_{ m A-B}$	SE_{A-B}	р	d	BF
Spun vs. Scrambled First Block Performance							
Error rate	-0.891	23	-0.037	0.042	.382	-0.236	3.258 (N)
Time taken	-0.765	21	-162.575	212.418	.453	-0.100	3.447 (N)
Spun vs. Scrambled Learning Rate (Training)							
Error rate	2.177	23	0.146	0.067	.040	0.611	1.553 (A)
Time taken	1.962	21	0.043	0.022	.063	0.421	1.119 (A)
Spun vs. Scrambled Starting Point (Test)							
Error rate	-1.424	23	-0.042	0.029	.168	-0.241	1.911 (N)
Time taken	-1.025	21	-176.658	172.362	.317	-0.159	2.811 (N)
Spun vs. Scrambled Learning Rate (Test)							
Error rate	2.786	23	0.229	0.082	.010	0.831	4.647 (A)
Time taken	1.430	21	0.031	0.022	.168	0.348	1.843 (N)

Note. DV = dependent variable.

1435

Table B15

Experiment 5: ANOVA and Sim	ple Main Effect Analyses for	Training and Transfer Effects

DV	F	dfs	MSE	р	3	η_p^2
Spun vs. Scrambled Performance (Training)						
Omission	0.734	1, 23	1.919	.400	1.000	0.031
Misorder	10.980	1, 23	111.035	.003	1.000	0.323
Intrusion	0.948	1, 23	40.636	.340	1.000	0.040
Learning in Training (Spun)						
Omission	3.775	39, 897	0.749	<.001	0.041	0.141
Misorder	2.474	39, 897	8.046	<.001	0.502	0.097
Intrusion	2.321	39, 897	1.413	<.001	0.100	0.092
Learning in Training (Scrambled)						
Omission	3.927	39, 897	0.749	<.001	0.043	0.146
Misorder	3.001	39, 897	8.046	<.001	0.612	0.115
Intrusion	4.078	39, 897	1.413	<.001	0.263	0.151
Spun vs. Scrambled Performance (Test)						
Omission	0.391	1, 23	1.199	.538	1.000	0.017
Misorder	31.424	1, 23	21.239	<.001	1.000	0.577
Intrusion	0.071	1,23	4.935	.792	1.000	0.003

Note. DV = dependent variable. Italicized *p* values are nonsignificant after correcting for violations of sphericity with the Huynh-Feldt epsilon value.

Table B16

DV	Ζ	Ν	SE	р
Spun vs. Scrambled (Training)				
Error rate	2.100	24	34.996	.036
Omission	-0.386	24	34.968	.699
Misorder	2.600	24	35.000	.009
Intrusion	1.004	24	32.861	.315
Spun vs. Scrambled (Test)				
Error rate	2.771	24	35.000	.006
Omission	0.408	23	30.657	.683
Misorder	3.958	24	34.993	.000
Intrusion	-0.198	24	22.782	.843
Spun vs. Scrambled First Block Performance				
Error rate	0.806	24	32.878	.420
Spun vs. Scrambled Learning Rate (Training)				
Error rate	2.200	24	35.000	.028
Spun vs. Scrambled Starting Point (Test)				
Error rate	-2.057	24	35.000	.040
Spun vs. Scrambled Learning Rate (Test)				
Error rate	2.029	24	34.998	.042

Note. DV = dependent variable.

(Appendices continue)

Table B17

Experiment 6: ANOVA and Simple Main Effect Analyses for Training Effects

DV	F	dfs	MSE	р	з	η_p^2
Practice Order (Spun Hebb first vs. Scrambled Hebb first)						
Error rate	4.809	1,20	0.382	.040	1.000	0.194
Time taken	1.733	1,20	59,599,126	.203	1.000	0.080
First Block Performance						
Error rate	1.658	3,60	0.061	.186	0.933	0.077
Time taken	0.569	3,60	1,228,023	.637	1.000	0.028
Hebb Advantage (Spun) vs. Hebb Advantage (Scrambled)						
Error rate	9.444	1,20	0.014	.006	1.000	0.321
Time taken	9.442	1,20	1,020,990	.006	1.000	0.321
Hebb Advantage (Spun)						
Error rate	110.956	1,20	0.014	<.001	1.000	0.847
Time taken	161.957	1,20	1,020,990	<.001	1.000	0.890
Hebb Advantage (Scrambled)						
Error rate	38.278	1,20	0.014	<.001	1.000	0.657
Time taken	70.236	1,20	1,020,990	<.001	1.000	0.778
Learning (Spun Hebb Lists)						
Error rate	5.307	7,140	0.006	<.001	0.913	0.210
Time taken	24.958	7,140	376,621	<.001	0.760	0.555
Learning (Control, Spun Hebb List Blocks)						
Error rate	1.757	7,140	0.006	.101	0.857	0.081
Time taken	6.754	7,140	376,621	<.001	1.000	0.252
Learning (Scrambled Hebb Lists)						
Error rate	6.134	7,140	0.006	<.001	0.847	0.235
Time taken	18.961	7,140	376,621	<.001	0.917	0.487
Learning (Control, Scrambled Hebb List Blocks)						
Error rate	0.290	7,140	0.006	.957	0.846	0.014
Time taken	2.384	7, 140	376,621	.025	0.943	0.106

Note. DV = dependent variable. Hebb Advantage analyses are simple main effects analyses of Hebb list type (repeated vs. Nonrepeated lists) that use the *MSE* of the Order Type \times Hebb type interaction. Learning analyses are simple main effects analyses of presentation number that use the *MSE* of the Order Type \times Hebb Type \times Presentation interaction.

Table B18

Experiment 6: t Tests for First Block Performance and Power Function Learning Rate

DV	t	df	$M_{\mathrm{A-B}}$	SE_{A-B}	р	d	BF
Spun Hebb vs. Scrambled Hebb Learning Rate							
Error rate	3.281	21	0.296	0.090	.004	0.837	11.910 (A)
Time taken	0.884	21	0.059	0.067	.387	0.123	3.162 (N
Hebb vs. Control Learning Rate (Spun)							
Error rate	8.041	21	0.663	0.082	<.001	2.427	193,441 (A)
Time taken	4.647	21	0.402	0.087	<.001	0.612	205.001 (A)
Hebb vs. Control Learning Rate (Scrambled)							
Error rate	5.564	21	0.367	0.066	<.001	1.325	1,427 (A)
Time taken	4.065	21	0.343	0.084	<.001	0.364	59.864 (A

Note. DV = dependent variable.

(Appendices continue)

Table B19

Experiment 6: ANOVA and Simple Main Effect Analyses for Each Error Type

DV	F	dfs	MSE	р	3	η_p^2
Hebb Advantage (Spun) vs. Hebb Advantage (Scrambled)						
Omission	0.686	1, 21	3.827	.417	1.000	0.032
Misorder	7.488	1,21	7.360	.012	1.000	0.263
Intrusion	5.270	1, 21	2.830	.032	1.000	0.201
Hebb Advantage (Spun)						
Omission	52.126	1, 21	3.827	<.001	1.000	0.713
Misorder	20.242	1, 21	7.360	<.001	1.000	0.491
Intrusion	113.332	1, 21	2.830	<.001	1.000	0.844
Hebb Advantage (Scrambled)						
Omission	36.582	1, 21	3.827	<.001	1.000	0.635
Misorder	0.395	1, 21	7.360	.528	1.000	0.018
Intrusion	54.731	1, 21	2.830	<.001	1.000	0.723
Learning (Spun Hebb Lists)						
Omission	2.114	7, 147	2.034	.046	0.606	0.091
Misorder	1.360	7, 147	2.830	.226	0.804	0.061
Intrusion	4.647	7,147	2.240	<.001	0.783	0.181
Learning (Control, Spun Hebb List Blocks)						
Omission	1.180	7,147	2.034	.318	0.671	0.053
Misorder	0.270	7,147	2.830	.965	0.827	0.013
Intrusion	1.906	7,147	2.240	.072	0.653	0.083
Learning (Scrambled Hebb Lists)						
Omission	0.973	7,147	2.034	.453	0.543	0.044
Misorder	2.194	7, 147	2.830	.038	0.603	0.095
Intrusion	9.576	7, 147	2.240	<.001	0.851	0.313
Learning (Control, Scrambled Hebb List Blocks)						
Omission	0.412	7, 147	2.034	.894	0.536	0.019
Misorder	0.852	7, 147	2.830	.547	0.814	0.039
Intrusion	2.232	7, 147	2.240	.035	0.826	0.096

Note. DV = dependent variable. Italicized p values are nonsignificant after correcting for violations of sphericity with the Huynh-Feldt epsilon value.

All Experiments Aggregated

Table B20

ANOVA for the Effects of Experiment and Detection of Structure on the Error Rate Advantage for Spun Lists

Effect	F	dfs	MSE	р	η_p^2
Experiment	0.229	5, 130	0.009	.949	0.009
Detection	4.138	1,130	0.009	.044	0.031
Experiment \times Detection	0.479	5, 130	0.009	.791	0.018

Note. The outlier from Experiment 5 is included in these analyses. Removing the outlier only affected the result of the Detection test: F(1, 129) = 3.208, p = .076.

Table B21

- (Dne-S	ampl	e t'i	Tests	on th	ie I	Error 1	Rate A	d	vantage	for	Spun I	Lists, A	4g	gregated	()ver t	he i	Experiments	

Effect	t	df	M _{A-B}	SE_{A-B}	р	d	BF
All	6.055	141	0.047	0.008	<.001	.508	838,150 (A)
No Detection	2.103	65	0.028	0.013	.039	.259	1.058 (A)
Detection	7.578	75	0.063	0.008	<.001	.869	126,384,703 (A)

Note. The outlier from Experiment 5 is included in these analyses. Removing the outlier did not the affect the result of any of the tests. However, it does raise confidence in favor of the alternative hypothesis for the No Detection test: BF = 4.747 (A).

Received February 7, 2020 Revision received March 10, 2021

Accepted April 19, 2021