Psychological Review

Serial Order Depends on Item-Dependent and Item-Independent Contexts

Gordon D. Logan and Gregory E. Cox

Online First Publication, March 9, 2023. https://dx.doi.org/10.1037/rev0000422

Psychological Review

ISSN: 0033-295X

https://doi.org/10.1037/rev0000422

THEORETICAL NOTE

Serial Order Depends on Item-Dependent and Item-Independent Contexts

Gordon D. Logan[1] and Gregory E. Cox[2]
[1] Department of Psychology, Vanderbilt University
[2] Department of Psychology, University at Albany, State University of New York

We address four issues in response to Osth and Hurlstone's (2022) commentary on the context retrieval and updating (CRU) theory of serial order (Logan, 2021). First, we clarify the relations between CRU, chains, and associations. We show that CRU is not equivalent to a chaining theory and uses similarity rather than association to retrieve contexts. Second, we fix an error Logan (2021) made in accounting for the tendency to recall ACB instead of ACD in recalling ABCDEF (fill-in vs. in-fill errors, respectively). When implemented correctly, the idea that subjects mix the current context with an initial list cue after the first order error correctly predicts that fill-in errors are more frequent than in-fill errors. Third, we address position-specific prior-list intrusions, suggesting modifications to CRU and introducing a position-coding model based on CRU representations to account for them. We suggest that position-specific prior-list intrusions are evidence for position coding on some proportion of the trials but are not evidence against item coding on other trials. Finally, we address position-specific between-group intrusions in structured lists, agreeing with Osth and Hurlstone that reasonable modifications to CRU cannot account for them. We suggest that such intrusions support position coding on some proportion of the trials but do not rule out CRU-like item-based codes. We conclude by suggesting that item-independent and item-dependent coding are alternative strategies for serial recall and we stress the importance of accounting for immediate performance.

Keywords: serial order, context retrieval, chaining, error ratio, prior-list intrusions

Supplemental materials: https://doi.org/10.1037/rev0000422.supp

Serial order is one of the most fundamental problems in psychology and neuroscience (Lashley, 1951). It challenges our ability to perceive structure in the world, to act coherently in sequential tasks, and to remember the order of our experiences (Logan, 2021). We solve these practical problems routinely in daily life but despite a century and a half of research on serial order (Ebbinghaus, 1885; Ladd & Woodworth, 1911; Nipher, 1878), there is no theoretical consensus on how we solve them. In the last 30 years, research on serial order in memory has focused primarily on serial recall tasks, like the memory span task (for comprehensive reviews, see Hurlstone et al., 2014; Lewandowsky & Farrell, 2008). Early theories based on simple chains of associations between successive items (Lewandowsky & Li, 1994; Lewandowsky & Murdock, 1989; Murdock, 1982, 1993, 1995; Shiffrin & Cook, 1978) were challenged by Henson et al. (1996), who showed that chaining theories cannot recover from errors, respond appropriately to manipulations of phonological similarity, produce transpositions to earlier list positions, or produce position-specific intrusions from previous lists or from different groups in the same list. As a result, theories that assume serial order is based on associations between items and position codes—contexts that are independent of the items—have come to dominate the field (Anderson & Matessa, 1997; Brown et al., 2000, 2007; Burgess & Hitch, 1999; Farrell, 2006; Hartley et al., 2016; Henson, 1998; Lewandowsky & Farrell, 2008; Oberauer et al., 2012).

Recently, Logan and colleagues proposed a context retrieval and updating (CRU) model of serial-order tasks, including serial recall, that does not assume position codes (Logan, 2018, 2021; Logan & Cox, 2021; Logan et al., 2021). Instead, it assumes that serial order is represented by associating items with contexts that are built from fading traces of earlier items, inspired by Howard and Kahana's (2002) temporal context model (TCM) of free recall and its descendants (Lohnas et al., 2015; Polyn et al., 2009; Sederberg et al., 2008). Logan (2021) applied CRU to serial recall, whole-report, and copy-typing tasks, showing that it accounts for several phenomena in these tasks, including list-length effects, serial position curves, transposition gradients, lag conditional recall probabilities, distributions of errors, recovery from errors, and the effects of repeating items in a single list. Logan (2021) found that CRU does not predict

Gordon D. Logan https://orcid.org/0000-0002-8301-7726

This research was funded by National Science Foundation Grant BCS 2147017 to Gordon D. Logan.

Data and code for fits and simulations are available on the Open Science Framework at https://osf.io/3kr5d/. The study was not preregistered; no new data were collected.

Correspondence concerning this article should be addressed to Gordon D. Logan, Department of Psychology, Vanderbilt University, Nashville, TN 37204, United States or Gregory E. Cox, Department of Psychology, University at Albany, State University of New York, 1400 Washington Avenue, Albany, NY 12222, United States. Email: gordon.logan@vanderbilt.edu or gregcox7@gmail.com

the error ratio (transpositions to earlier vs. later positions in the list, e.g., recalling ACB rather than ACD when given list ABCD; Page & Norris, 1998). This misprediction is an important limitation on CRU because virtually all position-coding theories predict the error ratio correctly.

Osth and Hurlstone (2022) provided an extremely valuable and constructive commentary on CRU, evaluating its ability to account for critical results that support position-coding theories: the effects of phonological similarity and position-specific intrusions from prior lists and from different groups in structured lists. They showed that CRU could account for phonological similarity effects with its assumptions that distinguish retrieving an item from selecting a response to report the retrieved item (Logan, 2018). CRU explains phonological similarity effects as confusions in response selection rather than errors in retrieval, much like position-coding theories (Henson, 1998). Osth and Hurlstone tried several ways to model the position specificity of intrusions in CRU but were unable to simulate the correct patterns. They concluded that CRU's failure to predict error ratios and position-specific intrusions were serious problems that warranted further development of the theory.

In this reply to Osth and Hurlstone (2022), we clarify the relations between CRU and item-dependent context models, revisit error ratios, suggest ways in which CRU can be amended or combined with position-coding models to account for position-specific intrusions from prior lists and other parts of structured lists.

CRU, Chains, Associations, and Compound Cues

Osth and Hurlstone (2022) interpreted CRU as a member of a broad class of item-based theories, ranging from classical chains (Ebbinghaus, 1885) to modern implementations (Lewandowsky & Murdock, 1989; Murdock, 1995; Solway et al., 2012), because its contexts are made of fading traces of past items. The broad class contains theories that overcome notorious problems with pairwise chains (Henson et al., 1996; Lashley, 1951) by assuming remote associations and compound cuing. Osth and Hurlstone argued that the elements of CRU's contexts are like remote associations and CRU's retrieval process involves compound cuing, so CRU inherits the interpretations and predictions of the broad class. Here, we show that CRU requires different interpretations, and it need not inherit those predictions. Like all computational models, CRU's predictions depend jointly on its assumptions about memory representations and retrieval processes, and the combination yields predictions that differ from other members of the class.

We contributed to a more general misunderstanding of CRU by drawing analogies between CRU and chaining models and between the elements of CRU vectors and strengths of associations to prior items (Logan, 2018, 2021; Logan & Cox, 2021; Logan et al., 2021). We did that to relate CRU to familiar concepts, but we intended the relations to be treated as analogies and not theoretical equivalences. Osth and Hurlstone (2022) showed that CRU reduces to a pairwise chaining model if $\beta = 1$ (also see Logan & Cox, 2021), but that model cannot produce errors (the only "competitor" in the retrieval process is the next item), so it fails to account for major serial-order phenomena.

To clarify the relations between CRU, chains, associations, and compound cues, and to demonstrate the consequences of mischaracterizing CRU, we compared CRU with the strength-based associative chaining (SBAC) model of serial learning (Solway et al., 2012). We chose SBAC for four reasons: (a) it is the most recent and most successful attempt to model chaining in serial memory, (b) it explicitly assumes remote backward and forward associations, (c) the cue for the next retrieval is the just-retrieved item, in contrast with CRU's "compound cue," and (d) its structure is very similar to CRU's, which facilitates formal comparison. We show that it is not equivalent to CRU, that CRU's elements are not equivalent to remote associations, and that CRU's retrieval cues may be better thought of as configural than compound.

Basic CRU

CRU is a simplification of TCM (Howard & Kahana, 2002) and its descendants (Lohnas et al., 2015; Polyn et al., 2009; Sederberg et al., 2008) that is applied to serial memory tasks (Logan, 2018, 2021; Logan & Cox, 2021; Logan et al., 2021). It assumes items and lists are represented as unit vectors using localist codes. Vectors representing the list have 1 in the element that corresponds to the list and 0 in all other elements. Vector representing the items have 1 in the element that corresponds to the item and 0 in all other elements. Encoding uses these vectors to create a set of stored context vectors that represent the list. It begins with the list vector and proceeds by adding vectors representing the items to the current context vector according to the TCM updating equation:

$$\mathbf{c}_{N+1} = \beta \mathbf{r}_N + \rho \mathbf{c}_N, \tag{1}$$

where $\mathbf{c}_{N+1}$ is the updated current context vector, $\mathbf{r}_N$ is the vector representing the item that was just presented, β is the weight on the presented item, $\mathbf{c}_N$ is the vector representing the context in which the item was presented, and ρ is the weight on that vector. (We represent vectors with lowercase bold italic font and matrices with uppercase bold italic font.) The value of ρ is chosen to normalize the updated context vector $\mathbf{c}_{N+1}$ to unit length. If $\mathbf{r}_N$ and $\mathbf{c}_N$ are orthogonal (as they would be if the new item was not present before in the list), then

$$\rho = \sqrt{1 - \beta^2}. \tag{2}$$

If they are not orthogonal (as they would be if the new item is a repetition of a prior item),

$$\rho = \sqrt{1 + \beta^2(\mathbf{r}_N \cdot \mathbf{c}_N)^2 - 1} - \beta(\mathbf{r}_N \cdot \mathbf{c}_N), \tag{3}$$

where $\mathbf{r}_N \cdot \mathbf{c}_N$ is the dot product between $\mathbf{r}_N$ and $\mathbf{c}_N$. CRU assumes that the current item ($\mathbf{r}_N$) is associated with the context in which it appeared ($\mathbf{c}_N$) and stored, and then the current context vector is updated. A set of stored context vectors generated from Equation 1 with $\beta = .6$ is presented in the top left panel of Figure 1. They show how the context vector for each item is a recency-weighted mixture of the prior items in the list.

Retrieval involves iteratively comparing an evolving current context with the set of stored contexts by calculating the dot products between them to retrieve an item and then updating the current context by adding the retrieved item (and not the retrieved response; Logan, 2018; Osth & Hurlstone, 2022) to it. The current context at retrieval is built with the same updating equation used in encoding (Equation 1). The dot products are drift rates in a racing diffusion decision process (Tillman et al., 2020), which chooses the item to be reported.

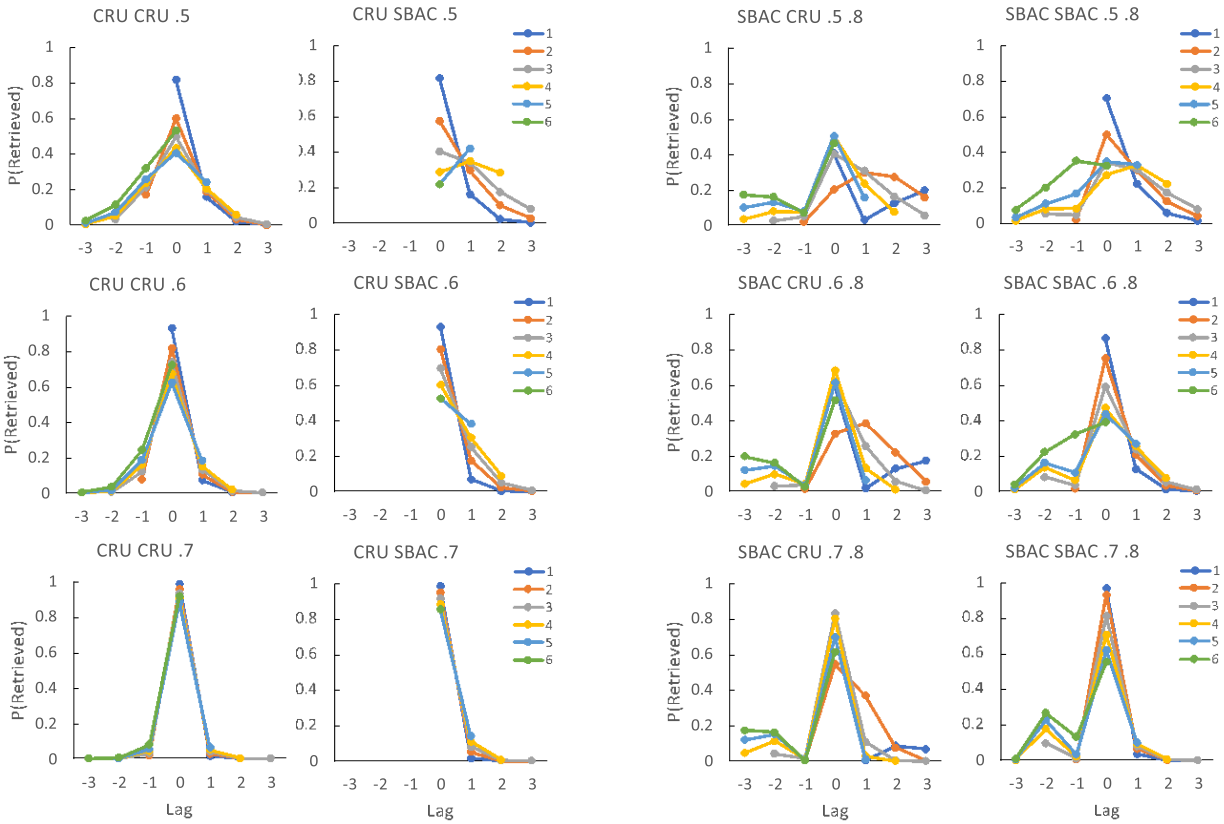
Figure 1
CRU and SBAC Compared

CRU Contexts

	A	B	C	D	E	F
1	1.000	0.800	0.640	0.512	0.410	0.328
2	0	0.600	0.480	0.384	0.307	0.246
3	0	0	0.600	0.480	0.384	0.307
4	0	0	0	0.600	0.480	0.384
5	0	0	0	0	0.600	0.480
6	0	0	0	0	0	0.600

SBAC Contexts

	A	B	C	D	E	F
1	0.600	0.480	0.384	0.307	0.246	0.197
2	0	0.600	0.480	0.384	0.307	0.246
3	0.480	0	0.600	0.480	0.384	0.307
4	0.384	0.480	0	0.600	0.480	0.384
5	0.307	0.384	0.480	0	0.600	0.480
6	0.246	0.307	0.384	0.480	0	0.600



Note. Top left: CRU contexts generated from Equation 1 with β = .6. Top right: SBAC contexts created from CRU contexts with β = .6 and w_b = .8. The rows represent serial positions. The columns represent items associated with serial positions. Second, third, and fourth from top: Probability of retrieval for Serial Positions 1–6 as a function of lag. Each row represents a different value of β (.5, .6, .7). In each row, w_b = .8. CRU–CRU = CRU contexts and CRU retrieval process; CRU–SBAC = CRU contexts and SBAC retrieval process; SBAC–CRU = SBAC contexts and CRU retrieval process; SBAC–SBAC = SBAC contexts and SBAC retrieval process; CRU = context retrieval and updating; SBAC = strength-based associative chaining. See the online article for the color version of this figure.

We interpret Equation 1 as a model of the psychological processes of encoding and retrieval, as if people actually go through the steps of updating the context as they learn and retrieve lists. Coupled with a racing diffusion decision process, CRU predicts the accuracy (Logan, 2018, 2021) and latency (Logan et al., 2021) of individual acts of retrieval. That is a significant strength of the theory that allows it to go beyond summary statistics like serial position curves and transposition matrices to address immediate performance (Kragel et al., 2015; Morton & Polyn, 2016).

Compound Cues and Remote Associations

Osth and Hurlstone (2022) characterize the elements of CRU vectors as recency-weighted associations between the current item and past items, as if the current item is connected to each of the past items with a bond whose strength depends on recency (see their Equation 3). At retrieval, they characterize the current context vector as a compound cue that activates the associations, such that the item with the strongest association is retrieved. While this is a reasonable

interpretation of the content of CRU's representations, it mischaracterizes the processes CRU applies to those representations to drive retrieval. Retrieval in CRU is driven by resonance (Ratcliff, 1978) rather than activation of associations. The current context activates stored contexts in proportion to their similarity, and the item associated with the stored context with the greatest activation is retrieved. This kind of retrieval process is common in theories of categorization, especially exemplar theories (Hintzman, 1986; Nosofsky, 1986). New exemplars must be classified by similarity because they have no prior associations to the category. Retrieval by resonance is part of some theories of recognition (Ratcliff, 1978) and cued recall (Hintzman, 1988; Shiffrin & Steyvers, 1997). It is a central assumption in CRU.

CRU's current context may be viewed as a compound retrieval cue (cf. Ratcliff & McKoon, 1988) but, in the language of Dosher and Rosedale (1997), it is a configural cue that is matched to memory as a holistic unit (Clark, 1995; Murdock, 1995), in contrast with cues whose components combine additively (McNamara, 1992, 1994) or multiplicatively (Humphreys et al., 1989; Raaijmakers & Shiffrin, 1981). The match is calculated with the dot product, which measures the similarity of the pattern of the element values (see Logan, 2021, Figures 3 and 14). The holistic nature of the match arises because CRU's contexts are all of length 1 (as guaranteed by the updating equation). As a result, the elements of CRU's contexts reflect the relative prominence of different items within the context, not the strength of an association. Increasing the prominence of one item necessarily diminishes the prominence of the others. Thus, the multiple items in a CRU context do not combine independently. Instead, they combine interactively in a nonlinear fashion (the nonlinearity arising from normalizing contexts by their length rather than their sum) to form a configural cue. The comparison to SBAC illustrates the importance of appreciating how contexts operate in CRU.

Basic SBAC

The SBAC model represents serial order in terms of associations between items, both adjacent and remote. The strength of the associations decreases with the distance between the items in the list. It assumes forward associations from the current item to items that follow it and backward associations from the current item to previous items in the list. These associations allow SBAC to account for transitions to remote items in both directions. SBAC specifies the computations that determine association strengths but, unlike CRU, it does not interpret those computations as psychological processes.

Solway et al. (2012) configured SBAC to account for serial learning, in which a list is presented several times until subjects can recall it in order without error. The increment in forward association strength between Item i and Item x is given by

$$\Delta F(i - x, i)_T = a_s[1 - F(i - x, i)_{T-1}]e^{-b_s|x-i|}, \quad (4)$$

where F is a matrix representing forward associations, a_x is a learning parameter, and $e^{-b_s|x-i|}$ represents the falloff in association strength with distance. SBAC assumes backward associations are weaker than forward associations by a constant, w_b, so

$$\Delta B(i, i - x)_T = w_b \Delta F(i - x, i)_T: \quad (5)$$

The forward and backward association matrices are added to produce the full association matrix

$$S = F + B: \quad (6)$$

SBAC assumes a primacy gradient such that increments in associative strength decrease exponentially with distance from the start of the list:

$$\mu_{a_s} = c_s e^{-d_s(i-1)} + \min_{a_s}, \quad (7)$$

where μ_{a_s} is the mean learning parameter, c_s is the strengthening parameter, d_s scales the distance between the current item and the beginning of the list, and $\min_{a_s}$ is the minimum value of the learning parameter. CRU also includes a primacy gradient, such that the updating weight (β) decreases as an exponential function of distance from the start of the list (Logan, 2021; Logan et al., 2021). To simplify analysis, we ignored the primacy gradient in SBAC and CRU.

SBAC assumes that the learning rate a_s varies randomly from trial to trial, adding Gaussian noise to its value. This random variability is the source of noise in a SoftMax decision process, which makes SBAC stochastic instead of deterministic. To simplify analysis, we assume a_s is constant over trials and $\min_{a_s} = 0$. Instead of SoftMax, we use CRU's racing diffusion decision process, which adds the necessary noise in the decision process itself (via the diffusion coefficient, set to 1.0) and not in the parameters (drift rates) that drive the decision.

We employed a simpler single-trial version of SBAC to model single-trial serial recall. The forward association matrix is defined by

$$F(i - x, i) = a_s e^{-b_s|x-i|}, \quad (8)$$

and the backward association matrix is defined by

$$B(i, i - x) = w_b F(i - x, i): \quad (9)$$

The top right panel of Figure 1 presents the full matrix ($S = F + B$) with $a_s = .6$, $b_s = .8$, and $w_b = .8$. At retrieval time, SBAC addresses single rows of the matrix one after the other. It begins with a list cue that selects the top row. The cells in the selected row represent association strengths, which are used as drift rates in a racing diffusion model to retrieve the item most strongly associated with the retrieval cue. After an item is retrieved, it serves as a cue to retrieve the next item, and the row corresponding to that item is selected. Retrieval proceeds until the end of the list. SBAC is a chaining model because it uses only the most recently retrieved item as the cue for the next retrieval, which is determined by the strengths of association between recallable items and the most recently retrieved item.

Does CRU Mimic SBAC?

The top left panel of Figure 1 presents a matrix representing the set of stored context vectors CRU would form to represent the same list, with $\beta = .6$ (so $\rho = .8$). CRU addresses columns of the matrix, calculating the dot product between the cue vector and each of the column vectors in the matrix. CRU differs from SBAC in that it uses all the rows in the matrix in the retrieval process while SBAC uses only one. CRU calculates retrieval strength by premultiplying the matrix with the current context vector, whereas SBAC calculates retrieval strength by premultiplying the matrix with a vector containing 1 in the position of the most recently recalled item and 0 elsewhere. From this perspective, CRU could be considered a compound-cuing model, if the SBAC retrieval process was applied to each row and the results were

summed. Each element in the current context multiplies its corresponding row, the products are summed for each column, and the sums are used as drift rates in a racing diffusion decision process. We interpret the mathematics differently: CRU's context vectors are configural cues, and the retrieval process is pattern matching of configurations.

It is tempting to think that CRU and SBAC mimic each other because the elements in the matrices have the same form. In SBAC,

$$F(i - x, i) = a_s e^{-b_s|x-i|}, \quad (10)$$

and in CRU,

$$F(i - x, i) = \beta\rho^{x-i}: \quad (11)$$

The functions have the same form: $b_s = -\log(\rho)$ because $y = e^{\log(y)}$. This equivalence allows us to create SBAC matrices from CRU matrices. Indeed, the SBAC matrix in Figure 1 was made from the CRU matrix in in the same Figure. The elements in the top row of the CRU matrix were multiplied by β. Otherwise, the elements above the main diagonal are identical in the two matrices, and the elements below the main diagonal were made by multiplying the elements above the diagonal by w_b (the weight for backward associations, in this case, .8).

The elements in the CRU matrix are more constrained than the elements in the SBAC matrix because ρ is determined by β (Equations 2 and 3), whereas a_s and b_s are independent. Also, the columns in the CRU matrices are normalized to length = 1, but neither rows nor columns are normalized in SBAC matrices. Consequently, SBAC is more flexible than CRU. We can convert all CRU matrices to SBAC matrices, but we cannot convert all SBAC matrices to CRU matrices. Thus, the memory structures that support retrieval in the two models do not mimic each other. SBAC matrices can be constructed from CRU matrices with appropriate calculations, but those calculations do not represent meaningful psychological processes. Further, the equivalence in form does not imply equivalence in psychological interpretation. The elements in SBAC matrices represent associations between the items, whereas the elements in CRU matrices represent the strength with which items are represented in the context. Thus, CRU does not mimic SBAC despite the equivalence of the elements of their matrices.

To assess mimicry between CRU and SBAC retrieval processes, we used CRU and SBAC retrieval cues to probe the CRU and SBAC matrices in Figure 1. We created matrices for CRU and SBAC and simulated 100,000 trials in which the CRU contexts were run through CRU's retrieval process (CRU–CRU; leftmost bottom panel of Figure 1), the CRU contexts were run through SBAC's retrieval process (CRU–SBAC; second from left), the SBAC associations were run through CRU's retrieval process (SBAC–CRU; third from left), and the SBAC associations were run through SBAC's retrieval process (SBAC–SBAC, rightmost). The result is a 2 × 2 factorial combination of memory representations (CRU and SBAC matrices) and retrieval processes (CRU and SBAC cuing). Details of the simulations are presented in the online Supplemental Material (https://osf.io/3kr5d/).

The bottom panels of Figure 1 show the transposition gradients for each combination of context and retrieval process with β = .5, .6, and .7 and w_b = .8. Retrieving from CRU contexts with CRU's retrieval process produced the familiar symmetrical transposition gradient (Henson, 1998; Logan, 2021). However, retrieving from CRU contexts with SBAC's retrieval process produced a completely asymmetrical transposition gradient. SBAC never made any backward (negative) transitions because there is nothing corresponding to backward associations in the CRU contexts. The CRU retrieval process produces backward transitions because retrieval is driven by similarity, not association, and the similarity of context vectors is symmetrical (see Logan, 2021). The SBAC contexts produced asymmetrical transposition gradients with both CRU and SBAC retrieval processes. (The near-zero values for −1 transitions are discussed in the online Supplemental Material.) The asymmetry suggests that this version of SBAC may not be a viable model of serial recall. Perhaps with the extra assumptions in the complete SBAC model that Solway et al. (2012) implemented would produce symmetrical transitions (see their Figure 4).

Accuracy (Lag 0 in Figure 1) was higher with CRU retrieval cues (.6530) than with SBAC retrieval cues (.6293), indicating the advantage of compound cues over single-item cues noted by Osth and Hurlstone (2022). Accuracy was higher when the retrieval cue and contexts came from the same model (CRU–CRU = .7427; SBAC–SBAC = .5878) than when they came from different models (CRU–SBAC = 6708; SBAC–CRU = .5634). The larger difference with CRU retrieval cues reflects their configural nature: They work better when they match the structure of the memory representation.

Together, the analysis and the simulations show that CRU and SBAC do not mimic each other. The elements of the context and association matrices may have the same basic form, but they are constrained more in CRU than in SBAC and have different structure. There is nothing corresponding to backward associations in CRU's matrices, and the simulations that applied the SBAC retrieval process to CRU representations showed that difference is consequential. The structure of the retrieval processes is different as well. SBAC addresses a single row of the matrices, which corresponds to the item that was just retrieved. CRU addresses columns of the matrices, including both the row that corresponds to the item that was just retrieved and the other rows that represent the context in which it appeared. The context can support retrieval of the correct item even if the previous item was retrieved incorrectly (Logan, 2018, 2021).

Error Ratio Revisited

When subjects omit item N and report item N + 1 instead, they can either "fill in" the missing item, recalling N after N + 1, or they can continue on ("in fill") and recall the item (N + 2) that follows N + 1. Given list ABCDEF and recall of ABD, ABDC ("fill in") is more likely than ABDE ("in fill," see Henson, 1998; Logan, 2021; Osth & Dennis, 2015b; Page & Norris, 1998; Surprenant et al., 2005). The difference in the relative frequency of these errors is often expressed as a ratio of fill-ins to in-fills (called the error ratio), which is typically around 2:1, though there is a large amount of variability in the ratio. The fill-in and in-fill effects distinguish position coding from chaining models. Most position-coding models predict the effect, but simple chaining models with only forward associations cannot predict it. Logan (2021) found that CRU predicts more in-fills than fill-ins, opposite to his own experimental results. This follows mathematically from the structure of the model, which creates a forward asymmetry following the initial omission. Logan suggested that the forward asymmetry could be overcome by adding the initial list cue to the current context after an

omission error, so the retrieval cue is a weighted average of the initial list cue and the current context. The initial list cue creates a primacy gradient (see the top row of the CRU context matrix in Figure 1), and the weight given to it in the model (the λ parameter in Equation 12) controls the strength of the gradient. Simulations showed that adding the initial list cue to the current context could produce more fill-ins than in-fills if the initial list cue is given sufficient weight. However, when the process of adding the initial list cue was implemented in the fitting routine, the fits strongly preferred models in which the weight given to the list cue was effectively zero. Osth and Hurlstone (2022) argued (correctly) that this failure was a serious problem for CRU. Logan (2021) agreed.

In preparing this response, we noticed two errors that Logan (2021) made in implementing the addition of the initial list cue. First and most important, Logan added the initial list cue for every response on every trial. This is inconsistent with the idea that adding the list cue is a response to an omission error. The list cue should only be added when an omission error occurs. Logan's second error was in adding the drift rates for initial and current contexts instead of adding the context vectors themselves. This error is more subtle and is more important psychologically than computationally. The theory assumes that the vectors are combined to form retrieval cues before they generate drift rates. Drift rates represent similarity computations that compare retrieval cues with stored contexts, and those computations logically follow the construction of the retrieval cues.

To address these errors, we refit the data from Logan's (2021) two experiments with the correct procedure. Experiment 1 presented strings of 5, 6, or 7 consonants and required subjects to type them while they remained on the screen, recall them after a 1 s presentation, or report them from a 100 ms display. Experiment 2 presented strings of six letters using the same three tasks (type, recall, report). In half of the displays, the letters were unique. In the other half, one of the letters was repeated. Each experiment tested 24 subjects. To reduce computation time, we only analyzed trials on which fill-in and in-fill errors occurred and we fixed the β and perceptual discriminability parameters to the values obtained in the best-fitting models from the article (encoding decay + serial order decay models). This ensures that the fitted models will still predict the phenomena they predicted in the original article. The details of the fitting procedure are presented in the online Supplemental Material (https://osf.io/3kr5d/).

The fitting routine added the initial context to the current context only after the first omission error in each trial, following the standard procedure for identifying fill-in and in-fill errors. The resulting context, c_{cue}, was

$$c_{cue} = \lambda \times c_1 + \lambda' \times c_N, \tag{12}$$

where c_1 is the initial context, c_N is the current context, λ is a free parameter ranging from 0 to 1, and λ′ is chosen to normalize the cue context vector so its length equals 1. We calculated λ′ using a version of Equation 3. Because the list context c_1 contains 1 in the list position and 0 elsewhere, the dot product $c_1 \cdot c_N = \rho^{N-1}$, and λ′ is

$$\lambda' = \sqrt{1 + \lambda^2 \left(\rho^{2(N-1)} - 1\right) - \lambda \rho^{N-1}}. \tag{13}$$

Each subject was fit individually using the same method as Logan (2021). Table 1 presents the mean values of the λ parameter for type, recall, and report tasks along with the mean numbers of trials that were fit and the minimum log likelihood obtained by the fitting routine. The log likelihood values are not meaningful in themselves without other fits for comparison. We report them for completeness.

We asked whether adding the initial context in the correct way improves CRU's account of fill-in and in-fill errors by counting the number of subjects for whom $\lambda > 0$. In Experiment 1, $\lambda > 0$ for all subjects in all conditions. In Experiment 2, $\lambda > 0$ for all subjects in all conditions except for one who had $\lambda = 0$ in one condition (report) and $\lambda > 0$ for the other conditions (type, recall). We asked whether the fitted λ parameters produce more fill-ins than in-fills by simulating CRU for each subject with their best-fitting parameters (the details are in the online Supplemental Material). The simulated and actual data were scored with the same routine, calculating the rates of fill-in and in-fill responses in the same way. Observed and predicted fill-in and in-fill rates are plotted in Figure 2.

In both experiments, the simulations produced more fill-in responses than in-fill responses, like the data, though the fits were far from perfect. They underestimated fill-ins and overestimated in-fills. They failed to capture the large fill-in rates in the recall condition of Experiment 2. We interpret the fits as proofs of concept, showing that a simple modification of CRU can produce error ratios greater than 1. This suggests that CRU is capable of accounting for fill-in and in-fill effects in fits to the data as well as in simulations. There is room for improvement, however, and obtaining better fits to test the predictions more rigorously is an important goal for future research.

Position-Specific Prior-List Intrusions

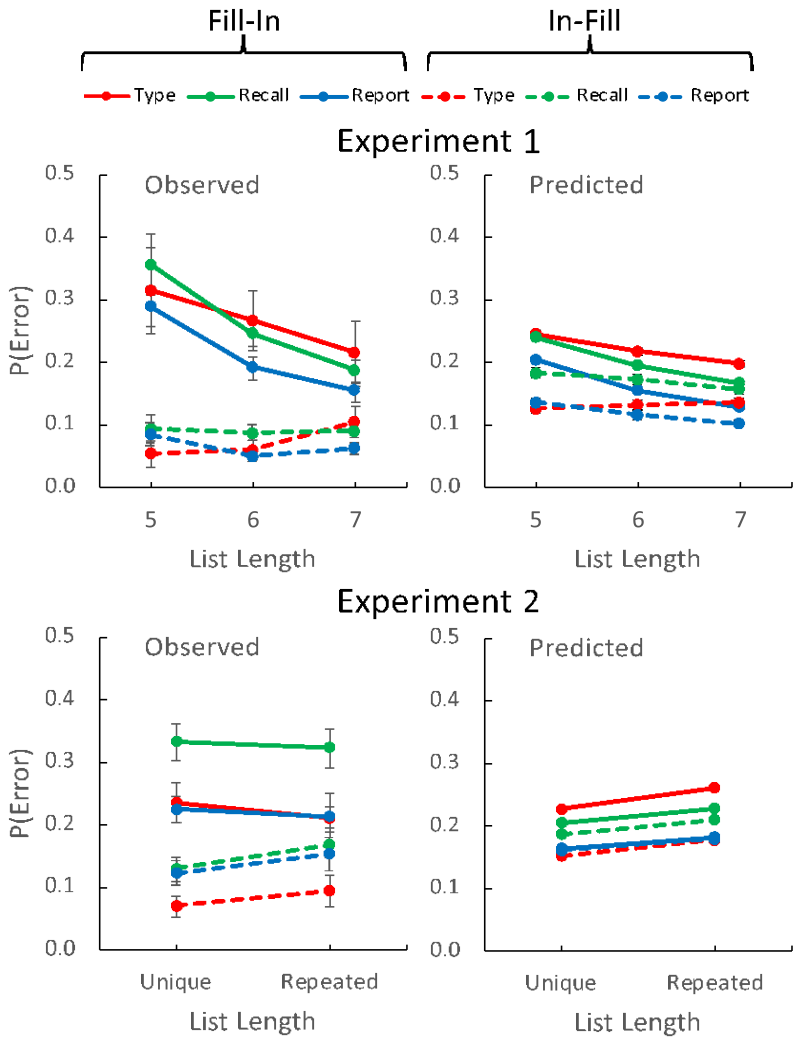
In serial recall, serial learning, and free recall, subjects often recall an item from a prior list. These prior-list intrusions tend to come from the immediately prior list; their frequency decreases exponentially for earlier lists (Kahana et al., 2002; Unsworth, 2008). Prior-list intrusions are often position-specific—recalled in positions at or near their serial position in the prior list (Conrad, 1959; Estes, 1991; Fischer-Baum & McCloskey, 2015; Henson, 1998; Melton & Irwin, 1940; Melton & von Lackum, 1941; Osth & Dennis, 2015a). Position-specific prior-list intrusions are called protrusions (Henson, 1998) and

Table 1
Fits to Fill-In and In-Fill Errors in Logan (2021)

Experiment	Plist type	Plist recall	Plist report	Log likelihood	N Error
1	.2699	.1880	.2216	509.92	49.75
2	.2443	.1644	.1452	556.26	54.88

Note. Estimated plist parameters in fits to Experiments 1 and 2 of Logan (2021), minimum log likelihood, and number of trials (N) on which fill-in and in-fill errors occurred averaged across 24 subjects in each experiment.

Figure 2
Transposition and Omission Errors



Note. Observed and predicted (simulated) rates of fill-in (transposition) and in-fill (omission) errors in copy-typing, serial recall, and whole-report tasks in Logan's (2021) Experiments 1 and 2. See the online article for the color version of this figure.

are interpreted as strong support for position-coding theories (Osth & Hurlstone, 2022): Items in successive lists are associated with the same set of position codes, so prior items may intrude when their position codes are activated to recall the current list. As Osth and Hurlstone (2022) point out, CRU cannot account for protrusions because it only represents the current list. They extended CRU to represent the prior list and varied the similarity of the list cues to produce prior-list intrusions. They found their extension of CRU produced protrusions, but they occurred in long runs as if the simulation switched to the prior list. Runs of prior-list intrusions are rare in real data (Osth & Dennis, 2015a). Osth and Hurlstone argued their findings were strong evidence against this account of protrusions. We agree.

We took two approaches to accounting for protrusions in CRU. First, we adopted an approach to context updating published after the Osth and Hurlstone's (2022) commentary that produced protrusions with a chaining model (Caplan et al., 2022) and applied it to CRU to see if CRU could produce position-specific intrusions without long runs. Second, we explored the idea that item coding and position coding are different strategies that people engage simultaneously or alternatively (Burgess & Hitch, 1992), arguing that protrusions may reflect trials on which people engaged position coding.

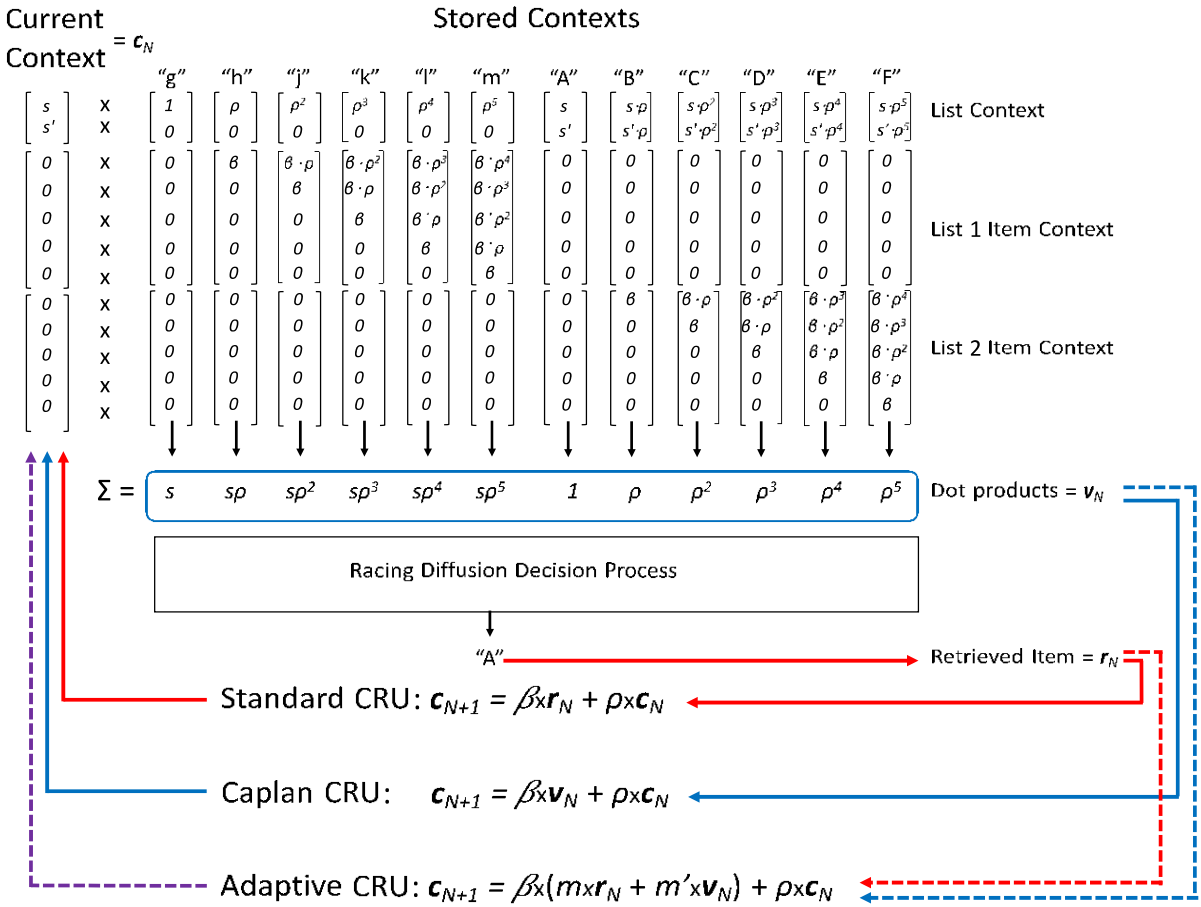
Producing Position Specificity in CRU

Caplan et al. (2022) proposed a chaining model similar to SBAC that explains protrusions without position codes. They added three assumptions to the classic chaining model: (a) Memory is not cleared between lists. Prior lists remain in memory with association strength reduced by forgetting. (b) The first item in all lists is associated with the same "start signal." (c) The retrieval cue for the next recall attempt is not the most recent item but instead is a vector representing the activation of traces from the most recent recall

attempt. The retrieval cue is the evidence that drives the decision that produces a recall response rather than the selected response itself. Imagine a current list ABCDEF and a prior list ghjklm each associated with the same start signal (lowercase represents reduced strength due to forgetting). The start signal activates A + g, which leads to a correct response (A) or a protrusion (g) after redintegration. The Caplan et al. approach uses the retrieval cue vector A + g as the cue for the next retrieval instead of the retrieved item. A + g retrieves B + h. Redintegrating h would produce a protrusion. B + h would then retrieve C + j, and so on, providing an opportunity for a protrusion in each step. The model avoids runs of protrusions because cuing with the evidence vector allows recovery from errors (Lewandowsky & Murdock, 1989), and the current-list items are always stronger than prior-list items. Caplan et al. fit their model to several data sets and found it predicted protrusions well without assuming position codes. Their results are important because they challenge the fundamental belief that only position-coding theories can account for protrusions (Henson et al., 1996).

We adapted the Caplan et al. (2022) version of CRU to see if it could produce protrusions without switching lists. We made three assumptions: (a) Prior lists remain in memory with no forgetting. (b) Items on different lists are associated with start contexts that differ in similarity. We assumed that the start contexts for each list were represented by two vector elements. The context for the start of the first list was [1, 0] and the context for the start of the second list was [s, sqrt($1 - s^2$)] where s represents context similarity (if s = 0, contexts are orthogonal, and if s = 1, they are identical). This decreased similarity makes the prior list less accessible than the current list without forgetting (Osth & Hurlstone, 2022). (c) The retrieval cue for the next recall attempt is the vector of dot products (drift rates) that drove retrieval on the current trial, representing the similarity between the current context cue and the set of stored contexts. Like Caplan et al. (2022), the retrieval cue is the evidence for recall rather than the result of recall, as it was in standard CRU. We evaluated these assumptions with simulations of three mod-els, depicted in Figure 3. Each simulation created stored contexts

Figure 3
Standard, Caplan, and Adaptive Updating



Note. Probing CRU stored contexts from two consecutive lists (ghjklm and ABCDEF) with a current context cue representing the start of the second list. The dot products in the blue box are input to the racing diffusion decision process that chooses a response ("A") represented by the vector r_N. Below the contexts are the context updating equations for the standard CRU model, the Caplan version of CRU, and the adaptive version of CRU. The retrieved item vector r_N is input to the updating equation for standard CRU and adaptive CRU. The dot product vector v_N is input to the updating equation for Caplan CRU and adaptive CRU. The β, ρ, m, and m′ parameters are explained in the text. CRU = context retrieval and updating. See the online article for the color version of this figure.

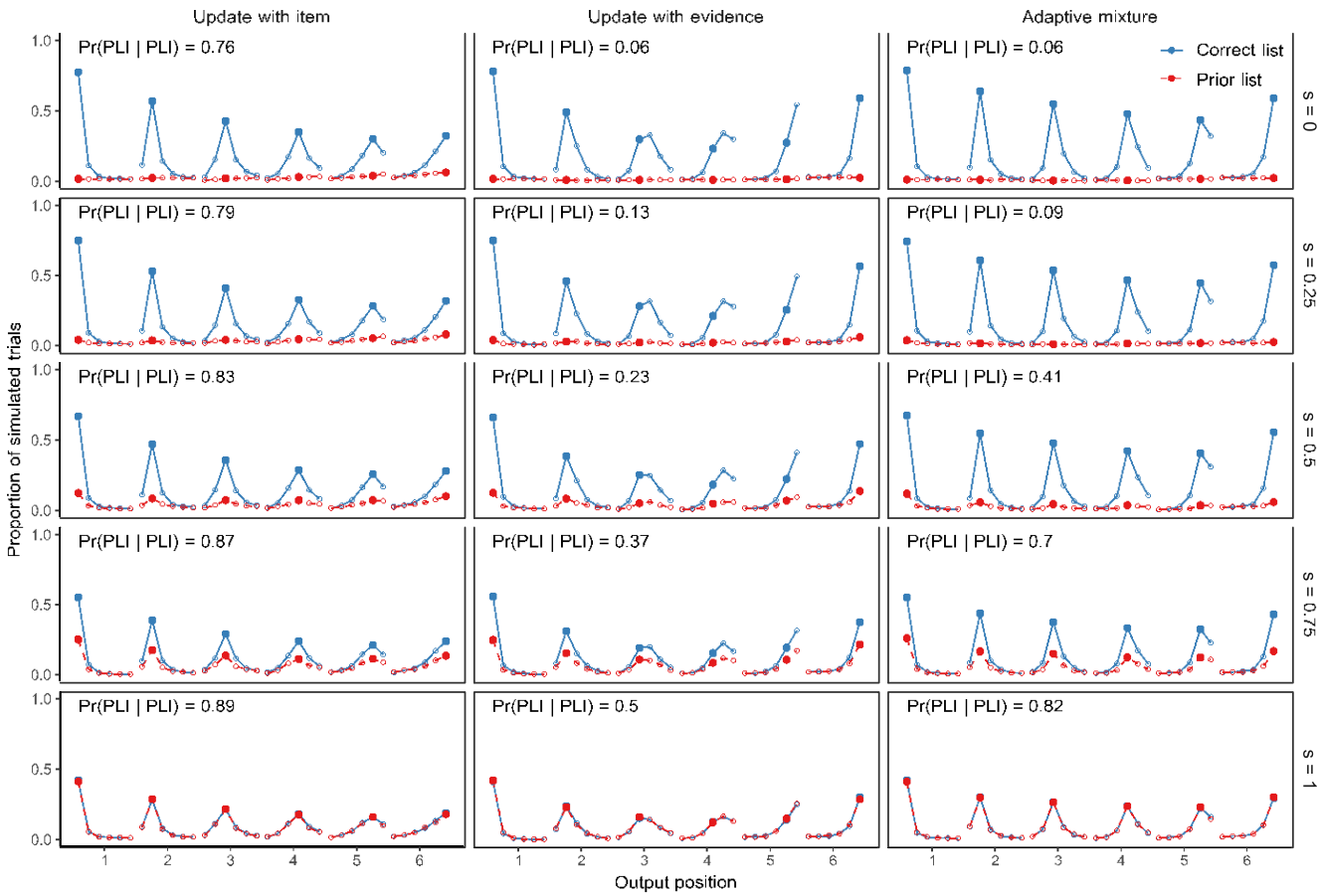
for two lists of six items and modeled recall of the second list by initiating recall with the second list context and terminating recall when CRU had produced six items. We set $\beta = 0.9$ and the racing diffusion threshold $\theta = 7$ for all models. We simulated 5,000 trials for each model. The results of the simulations are plotted in Figure 4. Inset in each panel is the proportion of protrusions conditional on a protrusion on the previous retrieval, Pr(PLI|PLI), where PLI = prior list intrusion. High values indicate runs of protrusions and a tendency to switch lists.

First, we simulated a standard version of CRU supplemented with assumptions (a) to represent prior lists and (b) to make list cues that differ in similarity, following Osth and Hurlstone (2022). Context updating at retrieval used the just-recalled item (Equation 1; Figure 3). When list contexts are sufficiently similar ($s \geq 0.5$), the standard version makes protrusions (Figure 4 left column). The details are in the Supplemental Information (https://osf.io/3kr5d/). But as shown in the insets in Figure 4, protrusions are likely to be followed by more protrusions, regardless of list context similarity. This confirms Osth and Hurlstone's results and is contrary to the data, which do not contain long runs of protrusions. The standard version produces runs because updating with a protrusion increases the similarity between the current context and stored contexts from the prior list, increasing the likelihood of another protrusion.

Second, we simulated a Caplan et al. approach to CRU incorporating all three assumptions. Context updating at retrieval used the vector v_{Norm} of dot products between the cue context and the stored contexts in memory—the drift rates in the racing diffusion decision process—normalized to length 1. This mitigated CRU's tendency to produce runs of protrusions when list contexts were not identical (middle row of Figure 4), but it produced many within-list errors and a tendency to skip to the end of the list, showing that CRU does not perform well with compound retrieval cues other than its current context.

Third, we simulated an adaptive version of CRU that made all three assumptions and supplemented them with a fourth: Context updating at retrieval uses a mixture of the just-retrieved item r_i and the vector of dot products v_{Norm}. The model is adaptive because the mixture gives more weight to r_i if it was probably correct and more

Figure 4
Simulations of Standard CRU, Caplan CRU, and Adaptive CRU



Note. Probability of responding as a function of output position for standard CRU (left panels), Caplan CRU (middle panels), and adaptive CRU. The rows represent different levels of list similarity. Within each panel, the lines represent the probability of responding as a function of input position for each output position. Blue lines represent the current list. Red lines represent prior-list intrusions. The inset in each panel is the probability of a prior-list intrusion given that the previous retrieval was a prior-list intrusion. CRU = context retrieval and updating; PLI = prior list intrusion. See the online article for the color version of this figure.

weight to v_{Norm} if it was probably an error. The model measured the likelihood of an error by retrieving the stored context associated with the retrieved item and comparing it with the current (not yet updated) context by calculating the dot product m_i (this is identical to the ith entry in the unnormalized vector of dot products v). Then it used m_i to construct a vector c_{Mix} that represents the mixture:

$$c_{Mix} = m_i r_i + m_i' v_{Norm}, \quad (14)$$

where

$$m_i' = \sqrt{1 + m_i^2 (r_i \cdot v_{Norm})^2} - 1 - m_i (r_i \cdot v_{Norm}). \quad (15)$$

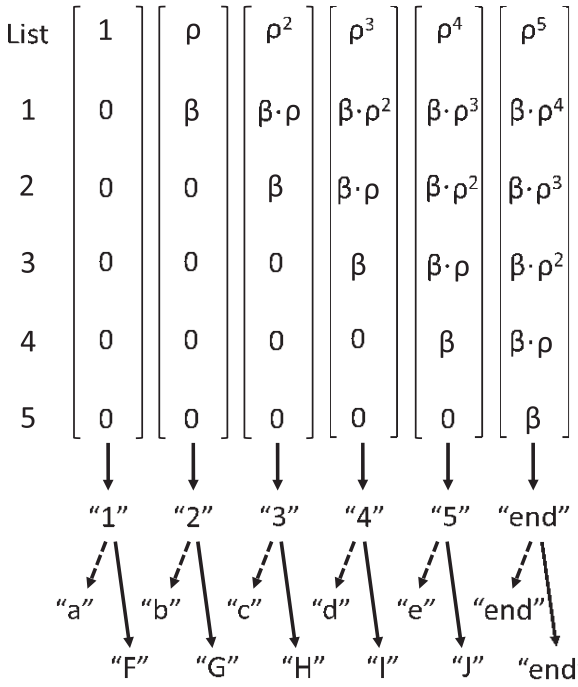
Then, c_{Mix} was used to update the current context. If $m_i = 1$, the just-retrieved item is used in updating (Equation 1), like standard CRU. If $m_i = 0$, the vector of dot products is used in updating, like the Caplan version of CRU. If $0 < m_i < 1$, the vector used in updating is a mixture of the just-retrieved item and the vector of dot products. This adaptive version of CRU produced protrusions without long runs (Figure 4 right column) when list contexts were moderately similar ($s \leq 0.5$). With moderate similarity, protrusions will produce smaller values of m_i and so will have less influence on the context that cues the next retrieval. The results suggest that the adaptive version of CRU may be able to account for protrusions, but much more work is required before the suggestion is conclusive. The Caplan et al. (2022) result is important because it showed conclusively that a chaining model like SBAC can produce protrusions.

Position Coding as a Strategy

Prior-list intrusions are errors produced infrequently by experimental subjects. They are not produced directly by experimental manipulations. Consequently, an observed prior-list intrusion implies that the subject used position coding on the current trial and the immediately preceding trial. It does not imply that position coding was used on other trials on which no prior-list intrusions were produced. Item coding may be used on the other trials. Position coding may be a strategy subjects choose to employ on some trials, but not others (Burgess & Hitch, 1992; Logan & Cox, 2021). Subjects may choose one or the other and alternate between them over the course of the experiment. Or item coding may be the default strategy that is used on every trial and position coding may be used in parallel with it on some proportion of the trials. (Item coding may be obligatory because every trial presents an item in the context of other items; see Logan, 1988.)

We explored these possibilities with simulations, using a standard version of CRU to implement item coding and a position-coding version of CRU (CRUposition) to implement position coding (Logan & Cox, 2021). CRUposition creates a set of generic contexts representing ordinal positions using the CRU context updating equation, associating each context with a generic representation of position (i.e., the third method for generating position codes from CRU in Logan & Cox, 2021; see Figure 5). At encoding, CRUposition steps through the generic contexts and associates each item with the generic position code retrieved from the generic contexts. At retrieval, CRUposition steps through the generic contexts once again, retrieves the position codes, and uses the position codes to retrieve the items associated with them. Instead of updating the current context with a vector representing the item encountered at each position (by having 1 in the vector element for that item and 0 elsewhere), CRUposition updates the current context with a vector that directly encodes the position (by having 1 in the vector element representing that position and 0 elsewhere).

Figure 5
CRU Position-Coding Model

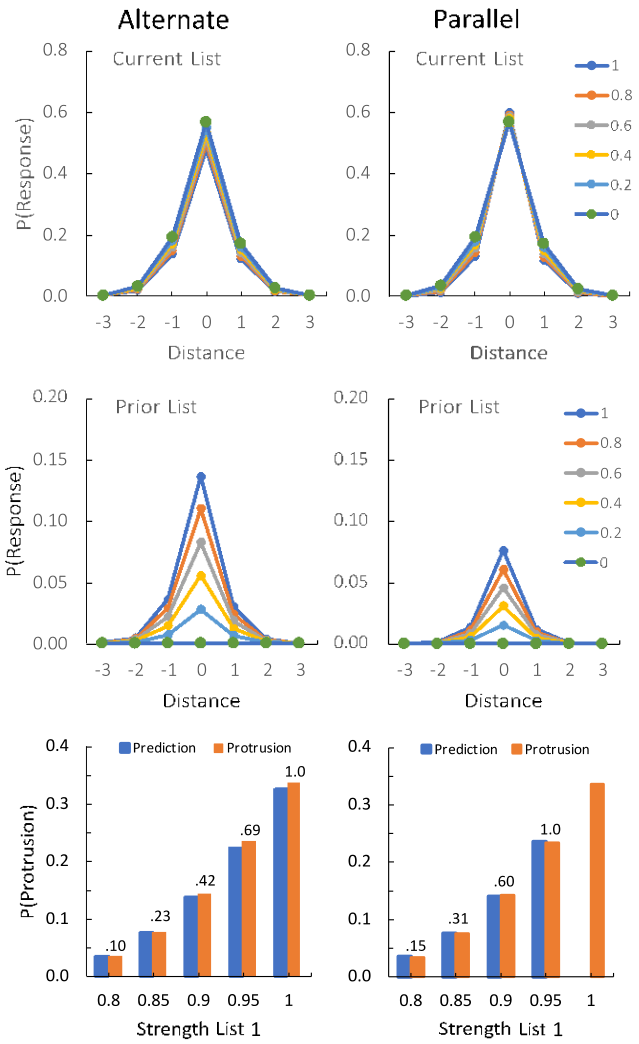


Note. The CRU position-coding model represents generic contexts (top) associated with generic position codes (1–5 and end of list marker). The current list (FGHIJ) and the prior list (abcde) are associated with the generic position codes. The strength of association is represented by the texture of the arrows. Associations to the current list are stronger than associations to the prior list. Solid arrows represent association strength of 1. Dashed arrows represent association strengths between 0 and 1 (represented by the List1strength parameter in the simulations). CRU = context retrieval and updating.

To allow for prior-list intrusions, we assumed that items from the current list and the prior list were associated with the same CRUposition codes with different strengths. The strength of association for the current list was always 1. The strength of association for the prior list varied from 0 to 1 (the strength parameter is List1strength). Consequently, current and prior items compete for retrieval, and the outcome of the competition depends on the relative strengths of the two lists (Figure 5). We also varied the probability that the simulation would engage in position coding (i.e., use CRUposition), either as an alternative to CRU or in parallel with it (the mixture parameter is Pposition). We used the same value of β (.55) for CRU and CRUposition to put them on even ground and set List1strength = .9. When CRU and CRUposition ran in parallel, we simulated retrieval times from each model using CRU's racing diffusion decision process and based the response on the faster of the two. Details of the simulations are reported in the online Supplemental Material (https://osf.io/3kr5d/).

Figure 6 shows the results of the simulations. The top panels show transposition gradients for recall of the current list when CRU and CRUposition alternate (left) and run in parallel (right). The

Figure 6
CRU Simulations of Prior-List Intrusions



Note. Left column: CRU and CRUposition alternate. Right column: CRU and CRUposition run in parallel. Top: probability of recalling items from the current list as a function of lag. β = .55, strength of prior list (List1strength) = .9. The parameter is the probability of using CRUposition (Pposition). Middle: probability of recalling items from the prior list (List 1) as a function of lag with the same parameters as the top panel. Bottom: simulated protrusion errors from CRUposition (orange) as a function of the strength of the prior list (List1-strength; β = .55 and Pposition = 1). Predicted protrusion errors (blue) as a function of Pposition (β = .55 and List1strength = 1). The numbers above the bars indicate the proportion of trials on which CRU position was engaged (Pposition). Pposition and prior list strength trade off. CRU = context retrieval and updating. See the online article for the color version of this figure.

transposition gradients were typical, peaked at lag = 0, and largely unaffected by varying the probability of using position coding (Pposition). The middle panel shows prior-list intrusions. They were clustered around lag = 0, as is typical of real data, and their probability increased as position coding became more likely (as Pposition increased). These results show that CRU combined with CRUposition can produce prior-list intrusions that depend on position.

The strength of the prior list trades off with the probability of using position coding. Prior-list intrusions increase with List1strength and decrease with Pposition. The bottom panels of Figure 6 show this trade-off. The orange bars were generated from CRUposition by itself with β = .55, List1strength varied from .8 to 1.0 (the range that reliably produces prior-list intrusions), and Pposition = 1. The blue bars are predictions generated from CRU and CRUposition combined with β = .55 and List1strength = 1. Pposition was adjusted (by hand) to produce prior-list intrusion rates that matched the rates from CRUposition by itself. The Pposition values required to match the intrusion rates are presented above the bars. It is clear from Figure 6 that List1strength and Pposition trade off. Thus, the observed rate of prior-list intrusions does not imply that position coding is used on every trial.

Other Sources of Prior-List Intrusions

Studies of serial recall have focused on the position specificity of prior-list intrusions without much consideration of other sources of intrusions that are more consistent with item coding (cf. Fischer-Baum & McCloskey, 2015). Prior-list intrusions tend to follow items that repeat from the prior list in serial recall (Fischer-Baum & McCloskey, 2015) and free recall (Kahana et al., 2002; Zaromb et al., 2006). In studies of proactive interference (Wickens, 1970), prior-list intrusions are common when items in the current list are similar semantically to items in the previous list (Loess, 1967). Lohnas et al. (2015) were able to account for both findings with an extension of the context maintenance and retrieval (CMR) model (itself an extension of TCM) to represent prior lists. They attributed the successful predictions to CMR's context representation. CMR is the parent of CRU, so in principle, CRU could be extended in the same way to capture intrusions following repetitions and intrusions from semantically similar lists.

Grouping and Position-Specific Intrusions Between Groups

Memory is often better when lists are divided into groups than when they presented as a single uninterrupted group (Adams, 1915; Hurlstone, 2019; Pollack et al., 1959; Ryan, 1969; Wickelgren, 1964; Wishner et al., 1957). Osth and Hurlstone (2022) showed that CRU can predict the improvement in accuracy in grouped lists. This is important because grouping and organization are important phenomena in memory, and it is encouraging to see that CRU can account for them. Osth and Hurlstone accounted for grouping by increasing β at the beginning of each group such that $\beta_g = \beta + \text{binc}\eth 1 - \beta\text{Þ}$, where β_g is the β value for the first member of each group and $0 \leq \text{binc} \leq 1$ determines the increment (cf. Lohnas et al., 2015). This is another way that a control process can govern context updating adaptively.

We replicated Osth and Hurlstone's (2022) simulations using lists of nine items grouped in sets of three, using β values of .5, .6, and .7 and binc values of .5, .6, .7, .8, and .9, and we compared the results with ungrouped lists with the same β values (binc = 0). The details are in the online Supplemental Material (https://osf.io/3kr5d/). The accuracy advantage of grouping is presented in Table 2. The different values of binc produced very similar advantages, but the advantage was smaller for β = .7 than for the other values of β. The top left plot in

Table 2
Simulated Advantages of Grouping

	CRU binc				
β	.5	.6	.7	.8	.9
.5	.0782	.0767	.0818	.0710	.0645
.6	.0734	.0784	.0690	.0649	.0517
.7	.0283	.0259	.0231	.0205	.0317

Note. Simulated advantages of grouped over ungrouped displays from the Osth and Hurlstone version of CRU (CRU OH). Data are collapsed across serial position. Rows are different β parameters. Columns are different binc parameters. CRU = context retrieval and updating.

Figure 7 shows CRU contexts for β = .5 and binc = .5. Grouped lists are compared with ungrouped lists in the second row of Figure 7. The details of these simulations and the others in this section are presented in the online Supplemental Material. Our simulations show that CRU can produce grouping advantages, but they are proofs of concept rather than accounts of actual data. Fits to data from grouped lists may distinguish between them or suggest other alternatives. Lohnas et al. (2015) and Polyn et al. (2009) showed how CRU's ancestor, CMR, can account for organization and list discrimination in free recall. Perhaps those ideas can be applied profitably to serial recall.

While we are encouraged by CRU's ability to account for the accuracy advantage of grouping, Osth and Hurlstone's (2022) investigation of intrusions between groups is less encouraging. In experiments that manipulate grouping, items from one group often intrude into recall of another group, and the intrusions are position specific: item N of Group x intrudes most often in position N of Group y (Hitch et al., 1996; Hurlstone, 2019; Ryan, 1969). Position-coding models account for these intrusions (Farrell, 2012; Hartley et al., 2016; Henson, 1998). Osth and Hurlstone showed convincingly that their modification of CRU cannot. We replicated their findings with their modification. The transposition gradient with β = .5 and binc = .5, presented in the second row of Figure 7, shows no increase in intrusions for lags of 3 and 6. We accept Osth and Hurlstone's conclusion that position coding is necessary to account for position-specific intrusions. However, as with prior-list intrusions, we question the generality of the position-specific intrusions in grouped lists.

Position-coding theories predict position-specific intrusions by assuming hierarchical position codes with a higher level that represents groups and a lower level that represents positions within groups. CRU's updating equation provides an account of how such position codes can be acquired and then accessed during retrieval (Logan & Cox, 2021). Once acquired, subjects can use hierarchical position codes strategically at encoding, especially when lists are structured. The top right panel of Figure 7 shows hierarchical position codes for a nine-item three-group list derived from CRU with β = .5 for each level.

We simulated retrieval with these position codes using the CRU updating process hierarchically. First, it retrieves a position code for a group, then it retrieves position codes within the chosen group, and then it retrieves the items associated with those position codes. All nine items compete at retrieval, but their contexts are weighted by the similarity of the group contexts, ρ^x, where x = {0, 1, 2} is the lag between the selected group and the competing group. The serial position effect and transposition gradient for the simulation are plotted in the right side of the second row of Figure 7. They show the typical effects: a scalloped serial position curve and spikes in the transposition gradient at Positions 3 and 6 that indicate between-group intrusions.

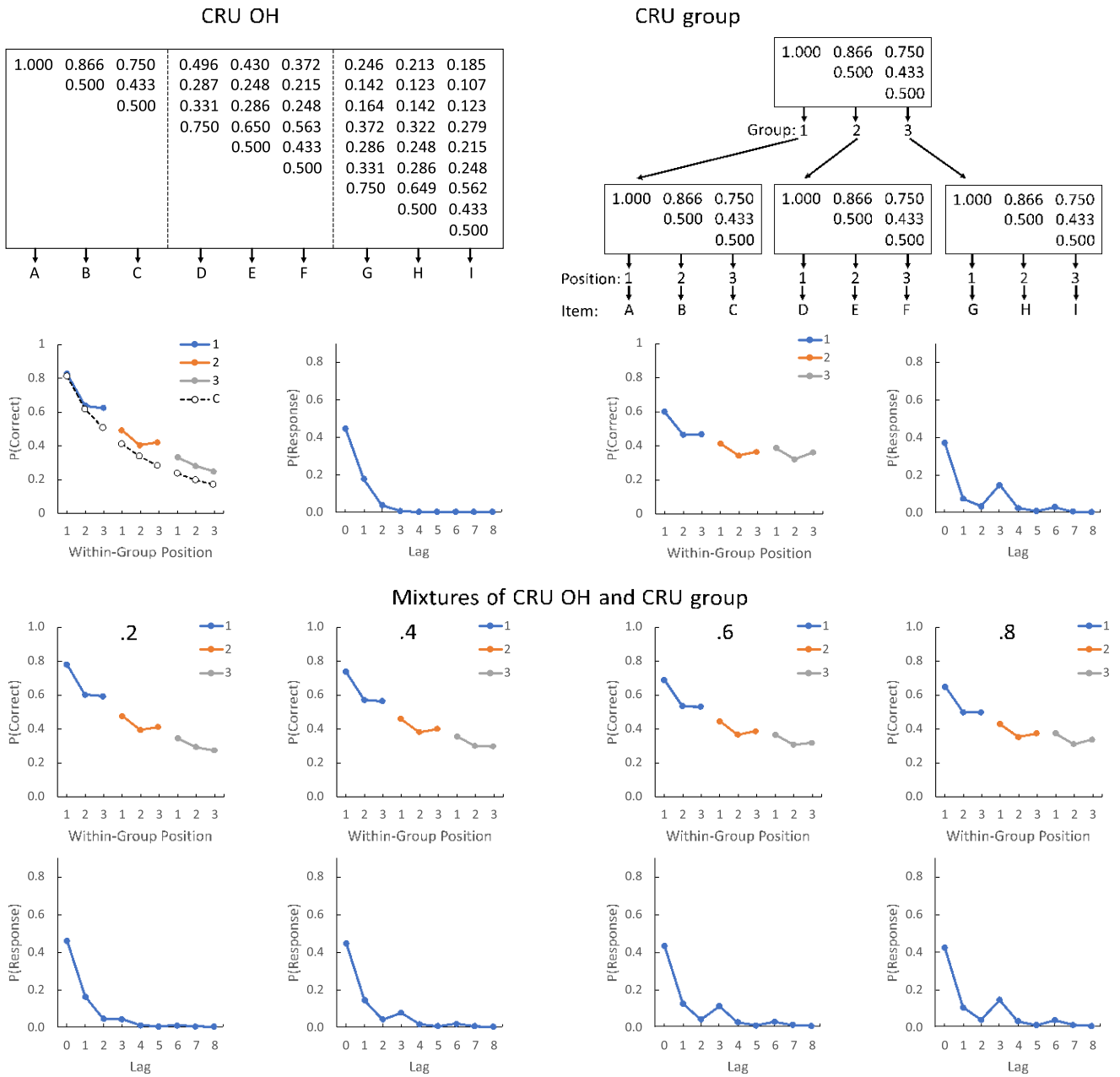
Like prior-list intrusions, between-group intrusions in grouped lists are errors that subjects make occasionally. Between-group intrusions imply that subjects used position coding on trials in which the intrusions were observed but they do not imply that position coding was used on the other trials. Subjects could engage position coding strategically, either alternating between position coding and item coding or running them in parallel. The bottom two panels of Figure 7 plot serial position curves and transposition gradients (respectively) when position coding and item coding alternate. The probability of engaging position coding (Pposition) varied from .2 to .8. The scalloped serial position curve appeared for each value of Pposition and position-specific spikes in the transposition gradient appeared even with Pposition = .2. Thus, the observation of position-specific between-group intrusions indicates that subjects used position coding on some proportion of the trials but it does not rule out item coding.

Discussion

Osth and Hurlstone (2022) asked if CRU's item-dependent context representations could underlie serial order in cognition, focusing on four phenomena that challenge classical chaining theories: phonological confusion effects, error ratios > 1, protrusions, and position-specific between-group intrusions. Our results suggest the answer is "yes." Osth and Hurlstone showed CRU could account for phonological confusion effects. We showed CRU can account for error ratios > 1 (fixing an error in Logan, 2021), that an adaptive version of CRU may be able to account for protrusions, and that both protrusions and position-specific between-group intrusions could be accounted for by a strategic mixture of CRU item coding and (CRU) position coding. More broadly, we showed that CRU's representations are different from chains of associations. Its similarity-based retrieval process produces backward transitions without backward associations and remote transitions without remote associations. On the one hand, our results motivate a renewed interest in item-dependent contexts in models of serial order, counteracting 25 years of neglect since the pivotal Henson et al. (1996) article. On the other hand, our results are proof of principle at best. It remains to be seen whether models implementing these changes will fit the range of serial-order phenomena as well as CRU and other models do.

The successful models required substantial changes in CRU. We had to assume subjects can access the initial list context, the evidence for their decisions, and stored contexts given an item when they detect an error. Many of these assumptions are natural extensions of CRU, having precedents in TCM and its descendants. We had to assume error monitoring to explain the error ratio and protrusions, and Logan (2021) had to assume error monitoring to explain within-list repetition (Ranschburg) effects. Error monitoring is a control processes that involves choices and (internal) actions that take time and make errors (Logan, 2017). In the spirit of CRU, these processes should be modeled and tested on their ability to account

Figure 7
CRU Simulations of Grouping and Between-Group Intrusions



Note. Top left: Osth and Hurlstone's CRU contexts (CRU OH) for a nine-item list grouped in 3 s (1, 2, 3) and a nine-item ungrouped list (C) with β = .5. β increased at group boundaries such that $\beta_{group} = \beta + binc \times (1 - \beta)$. Top right: hierarchical CRU position codes with β = .5. The top contexts retrieve position codes associated with each group, which initiate retrieval within groups based on the second row of contexts. The lower level contexts retrieve positions codes, which retrieve items. Second row: simulated proportions of correct recall as a function of serial position and proportions of responses as a function of transposition lag for the CRU OH model (left) and the CRU group model (right) with β = .5. Third row: Simulated proportions of correct responses as a function of serial position for mixtures of CRU OH and CRU group with β = .5. The proportion of trials using position coding increases from left to right (.2 to .8). Bottom row: simulated transposition gradients for mixtures of CRU OH and CRU group. The proportion of trials using position coding increases from left to right (.2–.8). CRU = context retrieval and updating. See the online article for the color version of this figure.

for effects in immediate performance. We have made no such tests, so our revisions to CRU may be better viewed as directions for future research. We hope our results will inspire more research on the control processes that guide encoding and retrieval. In the 50 years since Atkinson and Shiffrin's (1968) clarion call to study memory structures and control processes, we have learned much more about memory structures than control processes. It is time to redress that imbalance.

An important theme in Osth and Hurlstone's (2022) commentary and our reply is that serial recall may depend on both item-independent and item-dependent contexts. The data support both theories: Position coding uniquely predicts the position specificity of prior-list intrusions and between-group intrusions in structure lists (but see Caplan et al., 2022). Item coding uniquely predicts intrusions of previous list items that follow an item from the current list (Fischer-Baum & McCloskey, 2015; Kahana et al., 2002; Zaromb et al., 2006), intrusions of semantically related items from previous lists (Loess, 1967; Wickens, 1970), the benefits of compound cues (Chance & Kahana, 1997; Kahana & Caplan, 2002; Lohnas & Kahana, 2014), the benefits of sequential constraints on recall of list items (Baddeley, 1964; Botvinick & Bylsma, 2005; Merikle, 1969; Miller, 1958; Miller & Selfridge, 1950), and the learning of spin lists that maintain the order of items while varying absolute position (Kahana et al., 2010; Lindsey & Logan, 2021). Thus, it may be better to accept the validity of both position coding and item coding and ask how they work together than to try to determine which is the One True Theory. The memory system has access to many kinds of information, and it is likely to exploit different kinds in different acts of retrieval, responding strategically and adaptively to task demands (Anderson & Milson, 1989; Shiffrin & Steyvers, 1997). Sometimes position coding may be advantageous, other times item coding may work best. Sometimes they may work best in combination. We hope Osth and Hurlstone's commentary and our reply will inspire future research on control processes and strategies in memory performance within and beyond serial recall.

References

Adams, H. F. (1915). A note on the effect of rhythm on memory. Psychological Review, 22(4), 289–299. https://doi.org/10.1037/h0075190

Anderson, J. R., & Matessa, M. (1997). A production system theory of serial memory. Psychological Review, 104(4), 728–748. https://doi.org/10.1037/0033-295X.104.4.728

Anderson, J. R., & Milson, R. (1989). Human memory: An adaptive perspective. Psychological Review, 96(4), 703–719. https://doi.org/10.1037/0033-295X.96.4.703

Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. Psychology of Learning and Motivation, 2, 89–195. https://doi.org/10.1016/S0079-7421(08)60422-3

Baddeley, A. D. (1964). Immediate memory and the "perception" of letter sequences. The Quarterly Journal of Experimental Psychology, 16(4), 364–367. https://doi.org/10.1080/17470216408416394

Botvinick, M., & Bylsma, L. M. (2005). Regularization in short-term memory for serial order. Journal of Experimental Psychology: Learning, Memory, and Cognition, 31(2), 351–358. https://doi.org/10.1037/0278-7393.31.2.351

Brown, G. D. A., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. Psychological Review, 114(3), 539–576. https://doi.org/10.1037/0033-295X.114.3.539

Brown, G. D. A., 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

Burgess, N., & Hitch, G. J. (1992). Toward a network model of the articulatory loop. Journal of Memory and Language, 31(4), 429–460. https://doi.org/10.1016/0749-596X(92)90022-P

Burgess, N., & Hitch, G. J. (1999). Memory for serial order: A network model of the phonological loop and its timing. Psychological Review, 106(3), 551–581. https://doi.org/10.1037/0033-295X.106.3.551

Caplan, J. B., Ardebili, A. S., & Liu, Y. S. (2022). Chaining models of serial recall can produce positional errors. Journal of Mathematical Psychology, 109, Article 102677. https://doi.org/10.1016/j.jmp.2022.102677

Chance, F. S., & Kahana, M. J. (1997). Testing the role of associative interference and compound cues in sequence memory. In J. Bower (Ed.), Computational neuroscience: Trends in research (pp. 599–603). Plenum Press. https://doi.org/10.1007/978-1-4757-9800-5_93

Clark, S. E. (1995). The generation effect and the modeling of associations in memory. Memory & Cognition, 23(4), 442–455. https://doi.org/10.3758/BF03197245

Conrad, R. (1959). Errors of immediate memory. British Journal of Psychology, 50(4), 349–359. https://doi.org/10.1111/j.2044-8295.1959.tb00714.x

Dosher, B. A., & Rosedale, G. S. (1997). Configural processing in memory retrieval: Multiple cues and ensemble representations. Cognitive Psychology, 33(3), 209–265. https://doi.org/10.1006/cogp.1997.0653

Ebbinghaus, H. (1885). On memory: A contribution to experimental psychology. Teachers College, Columbia University.

Estes, W. K. (1991). On types of item coding and sources of recall in short-term memory. In W. Hockley & S. Lewandowsky (Eds.), Relating theory and data: Essays on human memory in honor of Bennet B. Murdock (pp. 155–176). Erlbaum.

Farrell, S. (2006). Mixed-list phonological similarity effects in delayed serial recall. Journal of Memory and Language, 55(4), 587–600. https://doi.org/10.1016/j.jml.2006.06.002

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

Fischer-Baum, S., & McCloskey, M. (2015). Representation of item position in immediate serial recall: Evidence from intrusion errors. Journal of Experimental Psychology: Learning, Memory, and Cognition, 41(5), 1426–1446. https://doi.org/10.1037/xlm0000102

Hartley, T., Hurlstone, M. J., & Hitch, G. J. (2016). Effects of rhythm on memory for spoken sequences: A model and tests of its stimulus-driven mechanism. Cognitive Psychology, 87, 135–178. https://doi.org/10.1016/j.cogpsych.2016.05.001

Henson, R. N. (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

Henson, R. N. A., Norris, D. G., Page, M. P. A., & Baddeley, A. D. (1996). Unchained memory: Error patterns rule out chaining models of immediate serial recall. Quarterly Journal of Experimental Psychology A: Human Experimental Psychology, 49(1), 80–115. https://doi.org/10.1080/713755612

Hintzman, D. L. (1986). "Schema abstraction" in a multiple-trace memory model. Psychological Review, 93(4), 411–428. https://doi.org/10.1037/0033-295X.93.4.411

Hintzman, D. L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. Psychological Review, 95(4), 528–551. https://doi.org/10.1037/0033-295X.95.4.528

Hitch, G. J., Burgess, N., Towse, J. N., & Culpin, V. (1996). Temporal grouping effects in immediate recall: A working memory analysis. Quarterly Journal of Experimental Psychology A: Human Experimental Psychology, 49(1), 116–140. https://doi.org/10.1080/713755609

Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. Journal of Mathematical Psychology, 46(3), 269–299. https://doi.org/10.1006/jmps.2001.1388

Humphreys, M. S., Bain, J. D., & Pike, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. Psychological Review, 96, 208–233. https://doi.org/10.1037/0033-295X.96.2.208

Hurlstone, M. J. (2019). Functional similarities and differences between the coding of positional information in verbal and spatial short-term order memory. Memory, 27(2), 147–162. https://doi.org/10.1080/09658211.2018.1495235

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., & Caplan, J. B. (2002). Associative asymmetry in probed recall of serial lists. Memory & Cognition, 30(6), 841–849. https://doi.org/10.3758/BF03195770

Kahana, M. J., Howard, M. W., Zaromb, F., & Wingfield, A. (2002). Age dissociates recency and lag recency effects in free recall. Journal of Experimental Psychology: Learning, Memory, and Cognition, 28(3), 530–540. https://doi.org/10.1037/0278-7393.28.3.530

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

Kragel, J. E., Morton, N. W., & Polyn, S. M. (2015). Neural activity in the medial temporal lobe reveals the fidelity of mental time travel. Journal of Neuroscience, 35(7), 2914–2926. https://doi.org/10.1523/JNEUROSCI.3378-14.2015

Ladd, G. T., & Woodworth, R. S. (1911). Elements of physiological psychology: A treatise of the activities and nature of the mind from the physical and experimental point of view. Charles Scribner's Sons.

Lashley, K. S. (1951). The problem of serial order. In L. A. Jeffress (Ed.), Cerebral mechanisms in behavior (pp. 112–136). Wiley.

Lewandowsky, S., & Farrell, S. (2008). Short-term memory: New data and a model. In B. H. Ross (Ed.), The psychology of learning and motivation (Vol. 49, pp. 1–48). Academic Press.

Lewandowsky, S., & Li, S.-C. (1994). Memory for serial order revisited. Psychological Review, 101(3), 539–543. https://doi.org/10.1037/0033-295X.101.3.539

Lewandowsky, S., & Murdock, B. B., Jr. (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. (2021). Previously retrieved items contribute to memory for serial order. Journal of Experimental Psychology: Learning, Memory and Cognition, 47, 1403–1438. https://doi.org/10.1037/xlm0001052

Loess, H. (1967). Short-term memory, word class, and sequence of items. Journal of Experimental Psychology, 74(4), 556–561. https://doi.org/10.1037/h0024787

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. (2017). Taking control of cognition: An instance perspective on acts of control. American Psychologist, 72(9), 875–884. https://doi.org/10.1037/amp0000226

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. Psychological Review, 128(1), 1–44. https://doi.org/10.1037/rev0000253

Logan, G. D., & Cox, G. E. (2021). Serial memory: Putting chains and position codes in context. Psychological Review, 128(6), 1197–1205. https://doi.org/10.1037/rev0000327

Logan, G. D., Cox, G. E., Annis, J., & Lindsey, D. R. B. (2021). The episodic flanker effect: Memory retrieval as attention turned inward. Psychological Review, 128(3), 397–445. https://doi.org/10.1037/rev0000272

Lohnas, L. J., & Kahana, M. J. (2014). Compound cuing in free recall. Journal of Experimental Psychology: Learning, Memory, and Cognition, 40(1), 12–24. https://doi.org/10.1037/a0033698

Lohnas, L. J., Polyn, S. M., & Kahana, M. J. (2015). Expanding the scope of memory search: Modeling intralist and interlist effects in free recall. Psychological Review, 122(2), 337–363. https://doi.org/10.1037/a0039036

McNamara, T. P. (1992). Theories of priming: I. Associative distance and lag. Journal of Experimental Psychology: Learning, Memory, and Cognition, 18(6), 1173–1190. https://doi.org/10.1037/0278-7393.18.6.1173

McNamara, T. P. (1994). Theories of priming: II. Types of primes. Journal of Experimental Psychology: Learning, Memory, and Cognition, 20(3), 507–520. https://doi.org/10.1037/0278-7393.20.3.507

Melton, A. W., & Irwin, J. M. (1940). The influence of degree of interpolated learning on retroactive inhibition and the overt transfer of specific responses. The American Journal of Psychology, 53(2), 173–202. https://doi.org/10.2307/1417415

Melton, A. W., & von Lackum, W. J. (1941). Retroactive and proactive inhibition in retention: Evidence for a two-factor theory of retroactive inhibition. The American Journal of Psychology, 54(2), 157–173. https://doi.org/10.2307/1416789

Merikle, P. M. (1969). Presentation rate and order of approximation to English as determinants of short-term memory. Canadian Journal of Psychology, 23(3), 196–202. https://doi.org/10.1037/h0082806

Miller, G. A. (1958). Free recall of redundant strings of letters. Journal of Experimental Psychology, 56(6), 485–491. https://doi.org/10.1037/h0044933

Miller, G. A., & Selfridge, J. A. (1950). Verbal context and the recall of meaningful material. The American Journal of Psychology, 63(2), 176–185. https://doi.org/10.2307/1418920

Morton, N. W., & Polyn, S. M. (2016). A predictive framework for evaluating models of semantic organization in free recall. Journal of Memory and Language, 86, 119–140. https://doi.org/10.1016/j.jml.2015.10.002

Murdock, B. B. (1982). A theory for the storage and retrieval of item and associative information. Psychological Review, 89(6), 609–626. https://doi.org/10.1037/0033-295X.89.6.609

Murdock, B. B. (1993). TODAM2: A model for the storage and retrieval of item, associative, and serial-order information. Psychological Review, 100(2), 183–203. https://doi.org/10.1037/0033-295X.100.2.183

Murdock, B. B. (1995). Developing TODAM: Three models for serial-order information. Memory & Cognition, 23(5), 631–645. https://doi.org/10.3758/BF03197264

Nipher, F. E. (1878). On the distribution of errors in numbers written from memory. In Transactions of the academy of science of St. Louis (Vol. 3, pp. 201–211).

Nosofsky, R. M. (1986). Attention, similarity, and the identification-categorization relationship. Journal of Experimental Psychology: General, 115(1), 39–57. https://doi.org/10.1037/0096-3445.115.1.39

Oberauer, K., Farrell, S., Jarrold, C., Pasiecznik, K., & Greaves, M. (2012). Interference between maintenance and processing in working memory: The effect of item-distractor similarity in complex span. Journal of Experimental Psychology: Learning, Memory, and Cognition, 38(3), 665–685. https://doi.org/10.1037/a0026337

Osth, A. F., & Dennis, S. (2015a). Prior-list intrusions in serial recall are positional. Journal of Experimental Psychology: Learning, Memory, and Cognition, 41(6), 1893–1901. https://doi.org/10.1037/xlm0000110

Osth, A. F., & Dennis, S. (2015b). 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

Osth, A. F., & Hurlstone, M. J. (2022). Do item-dependent context representations underlie serial order in cognition? Commentary on Logan (2021). Psychological Review. Advance online publication. https://doi.org/10.1037/rev0000352

Page, M. P., & 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

Pollack, I., Johnson, L. B., & Knaff, P. R. (1959). Running memory span. Journal of Experimental Psychology, 57(3), 137–146. https://doi.org/10.1037/h0046137

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

Raaijmakers, J. G. W., & Shiffrin, R. M. (1981). Search of associative memory. Psychological Review, 88(2), 93–134. https://doi.org/10.1037/0033-295X.88.2.93

Ratcliff, R. (1978). A theory of memory retrieval. Psychological Review, 85(2), 59–108. https://doi.org/10.1037/0033-295X.85.2.59

Ratcliff, R., & McKoon, G. (1988). A retrieval theory of priming in memory. Psychological Review, 95(3), 385–408. https://doi.org/10.1037/0033-295X.95.3.385

Ryan, J. (1969). Grouping and short-term memory: Different means and patterns of grouping. The Quarterly Journal of Experimental Psychology, 21(2), 137–147. https://doi.org/10.1080/14640746908400206

Sederberg, P. B., Howard, M. W., & Kahana, M. J. (2008). A context-based theory of recency and contiguity in free recall. Psychological Review, 115(4), 893–912. https://doi.org/10.1037/a0013396

Shiffrin, R. M., & Cook, J. R. (1978). Short-term forgetting of item and order information. Journal of Verbal Learning and Verbal Behavior, 17(2), 189–218. https://doi.org/10.1016/S0022-5371(78)90146-9

Shiffrin, R. M., & Steyvers, M. (1997). A model for recognition memory: REM-retrieving effectively from memory. Psychonomic Bulletin & Review, 4(2), 145–166. https://doi.org/10.3758/BF03209391

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

Surprenant, A. M., Kelley, M. R., Farley, L. A., & Neath, I. (2005). Fill-in and infill errors in order memory. Memory, 13(3–4), 267–273. https://doi.org/10.1080/09658210344000396

Tillman, G., Van Zandt, T., & Logan, G. D. (2020). Sequential sampling models without random between-trial variability: The racing diffusion model of speeded decision making. Psychonomic Bulletin & Review, 27(5), 911–936. https://doi.org/10.3758/s13423-020-01719-6

Unsworth, N. (2008). Exploring the retrieval dynamics of delayed and final free recall: Further evidence for temporal-contextual search. Journal of Memory and Language, 59(2), 223–236. https://doi.org/10.1016/j.jml.2008.04.002

Wickelgren, W. A. (1964). Size of rehearsal group and short-term memory. Journal of Experimental Psychology, 68(4), 413–419. https://doi.org/10.1037/h0043584

Wickens, D. D. (1970). Encoding categories of words: An empirical approach to meaning. Psychological Review, 77(1), 1–15. https://doi.org/10.1037/h0028569

Wishner, J., Shipley, T. E., Jr., & Hurvich, M. S. (1957). The serial-position curve as a function of organization. The American Journal of Psychology, 70(2), 258–262. https://doi.org/10.2307/1419331

Zaromb, F. M., Howard, M. W., Dolan, E. D., Sirotin, Y. B., Tully, M., Wingfield, A., & Kahana, M. J. (2006). Temporal associations and prior-list intrusions in free recall. Journal of Experimental Psychology: Learning, Memory, and Cognition, 32(4), 792–804. https://doi.org/10.1037/0278-7393.32.4.792

Received March 3, 2022
Revision received December 8, 2022
Accepted December 31, 2022 ▪