Do item-dependent context representations underlie serial order in cognition? Commentary on Logan (2021)

Adam F. Osth

University of Melbourne

Mark J. Hurlstone

Lancaster University & The University of Western Australia

Address correspondence to:

Adam Osth (E-mail: adamosth@gmail.com)

Author Note

We would like to thank Gordon Logan, Mike Kahana, and two anonymous reviewers for their very helpful comments that improved a previous version of this manuscript. We would also like to thank Philip Smith, Simon Lilburn, and members of the Melbourne Vision Lab for the intriguing discussions that led to the writing of this commentary. Data and model code can be found at https://osf.io/gnrwz/

Abstract

Logan (2021) presented an impressive unification of serial order tasks including whole report, typing, and serial recall in the form of the context retrieval and updating (CRU) model. Despite the wide breadth of the model's coverage, its reliance on encoding and retrieving context representations that consist of the previous items may prevent it from being able to address a number of critical benchmark findings in the serial order literature that have shaped and constrained existing theories. In this commentary, we highlight three major challenges that motivated the development of a rival class of models of serial order, namely positional models. These challenges include the mixed-list phonological similarity effect, the protrusion effect, and interposition errors in temporal grouping. Simulations indicated that CRU can address the mixed list phonological similarity effect if phonological confusions can occur during its output stage, suggesting that the serial position curves from this paradigm do not rule out models that rely on inter-item associations, as has been previously been suggested. The other two challenges are more consequential for the model's representations, and simulations indicated the model was not able to provide a complete account of them. We highlight and discuss how revisions to CRU's representations or retrieval mechanisms can address these phenomena and emphasize that a fruitful direction forward would be to either incorporate positional representations or approximate them with its existing representations.

Keywords: serial order; serial recall; context; chaining; phonological similarity

Do item-dependent context representations underlie serial order in cognition? Commentary on Logan (2021)

Serial order is an important component of cognition. For decades, theorists have observed commonalities in effects and error patterns across different psychological domains, including speech production (Ellis, 1980; Page, Madge, Cumming, & Norris, 2007), typing (Logan, 2018), reading (Hannagan & Grainger, 2012), spelling (Fischer-Baum, McCloskey, & Rapp, 2010), and music performance (Palmer & Pfordresher, 2003; Pfordresher, Palmer, & Jungers, 2007). The commonalities in empirical regularities and error patterns across these different domains have led researchers to hypothesize that each may be served by common representations and retrieval mechanisms of serial order (Fischer-Baum, 2018; Hurlstone, Hitch, & Baddeley, 2014; Hurlstone, in press). However, to date relatively little work has been done to bridge the commonalities across these different domains in a unified model.

Logan (2021) presented what is possibly the most thorough and impressive attempt at unifying serial order tasks to date in his account of whole report, typing, and serial recall procedures using his context retrieval and updating (CRU) model. Each of the tasks are addressed using a common representation and retrieval mechanism. Specifically, the model relies on what we call *item-dependent* context representations, in which items are associated to contexts that are composed of the previous items in the list. Retrieval is determined by the similarity between the current context and the stored contexts in memory. After each item is retrieved, its representation enters into the current context cue, which changes its similarity to the stored context vectors. Specifically, the context vectors belonging to items studied near the just-recalled item will exhibit the highest similarity to the updated context, and will therefore be the most likely to be retrieved. The model is heavily inspired by the temporal context model (TCM: Howard & Kahana, 2002), as CRU even uses the same equations for updating context across item presentations and recalls, although there are a couple of important distinctions from TCM that we highlight below.

The parameters of CRU, which govern the relative weighting of the current item to the preceding context (β) as well as the distinctiveness of letter encoding (g), vary plausibly across tasks to capture the ways in which differences in procedures or task demands may impact the ability to encode or retrieve information.

What is additionally impressive is that the experiments were all collected using the same participants and the model was subsequently fit to the individual responses from the participants. This is an important departure from previous models of serial order, which have a tendency to focus on demonstrations of qualitative phenomena in isolation. While such simulations demonstrate that it is in principle possible for a model to address a phenomenon, it is far more impressive to be able to demonstrate that the model is able to capture such phenomena while being simultaneousy able to explain variations in performance across individuals. CRU's success in capturing variation across individuals at the level of individual responses in each of the aforementioned tasks suggests the model's ability to explain the data is a consequence of its core representations and retrieval mechanisms.

An additional strength of the model is the racing diffusion architecture it employs to decide on which item to encode and retrieve (Tillman, Van Zandt, & Logan, 2020). While latencies were not modeled in the Logan (2021) article, this architecture will allow the model to jointly address patterns of choice and distributions of response latency, producing an integrated model of representation, retrieval, and decision-making (e.g., Cox & Shiffrin, 2017; Fox, Dennis, & Osth, 2020; Nosofsky, Little, Donkin, & Fific, 2011; Osth, Jansson, Dennis, & Heathcote, 2018; Sederberg, Howard, & Kahana, 2008). Racing diffusion processes have recently been found to be successful in accounting for complete RT distributions in free recall (Osth & Farrell, 2019). Extension to latencies will likely be a fruitful endeavor as there have been some important constraints from latencies in serial recall and list reproduction that have yet to be comprehensively addressed in models of serial order (Farrell & Lewandowsky, 2004; Hurlstone & Hitch, 2015, 2018; Thomas,

Milner, & Haberlandt, 2003).

While we believe the field can greatly benefit from both the advances and direction of the Logan (2021) article, the purpose of our commentary is to question the sufficiency of its representations and retrieval mechanisms. In particular, there are a number of consequential challenges from the serial order literature that are constraining for models that principally rely on item-dependent context representations at both encoding and retrieval. Through simuations with CRU, we explore three important challenges, namely the mixed-list phonological similarity effect (Henson, Norris, Page, & Baddeley, 1996), the finding that intrusions from prior lists preserve within-list position (*protrusion errors*: Henson, 1999; Osth & Dennis, 2015a), along with the costs and benefits of temporal grouping manipulations (Henson, 1999). While CRU was able to provide a satisfactory account of the first phenomenon with a revision to its output stage, it was not able to provide a complete account of the other two phenomena. While no model is able to account for every phenomenon, we note that each of these phenomena have been instrumental in motivating a very different class of models, namely models that rely on a specific set of *item-independent* context representations, where the context is composed of elements that are not shared with the items and instead reflect an item's within-list position. Furthermore, these phenomena have been sufficiently influential on the field to be described as "benchmarks" of short-term and working memory in a recent review article published in *Psychological Bulletin* (Oberauer et al., 2018).

Our commentary is organized as follows. We first provide a brief history of item-dependent and item-independent context representations in the serial order literature with a focus on the challenges that arose for theories that are reliant on item-dependent context representations. We follow this with a brief description of the mathematics and parameters of CRU and how these are critical to the model's predictions. We follow this with a comprehensive set of CRU simulations for each phenomenon where theoretically relevant parameters are manipulated across a broad range. We close the commentary with

a synthesis of other findings that are congruent with CRU's representations, and provide recommendations for how both CRU and the field at large can benefit from a broader consideration of how item-dependent context representations can be utilized to represent serial order.

Representations of Serial Order

The question of how the order of a to-be-learned sequence of items is represented in memory is among the oldest questions in research on serial order, dating back to the pioneering studies of Ebbinghaus (1885/1913). The field has generally converged on two common representational schemes, as summarized in a review article by Young (1968): "The serial list, following learning, is a highly organized group of items which are related to one another through a chain of associations, through associations between ordinal positions and items or through something else entirely." The first possibility – associations between items – is most similar to the item-dependent context representations, of which CRU is reliant. While CRU's representations are referred to as "context" representations, its context representations are composed of the previous items, and we will demonstrate later how the mathematics of the model strongly resemble the operations of a model that rely on inter-item associations.

The second representational scheme – associations between items and their position of occurrence – refers to item-independent context representations. Again, while such models describe their representations as "context", this context is not only independent of the other items on the list but corresponds to the within-list position of the items. Throughout the remainder of the article our terminology will focus on the terms "positional representations" or "associations to within-list position," as item-independent context representations do not necessarily correspond to within-list position. In the episodic memory literature, there are several models that employ random context representations that are independent of the items (e.g., Davelaar, Goshen-Gottstein, Ashkenazi, Haarmann, & Usher, 2005; Mensink &

Raaijmakers, 1988; Murdock, 1997; Osth et al., 2018). A critical difference, however, is that while context can vary across item or test presentations in these models, they do not recur across items from different lists that share the same within-list position, and the cues for each retrieval are not reinstated in the same manner as positional models of serial order.

The relative strengths of a given representational scheme are strongly dependent on the form of the representations at encoding and retrieval. In the case of inter-item associations, it is important to note that there are a number of different ways in which associations between items can be used at both encoding and retrieval to support serial order memory, each of which carry important consequences. For instance, the simplest possible method is that associations between adjacent items are formed at encoding, such that a list of items $ABCD$ is encoded as a set of pairwise associations between the items ($A-B$, $B-C$, and $C-D$). At retrieval, the sequence is reconstructed by using each retrieved item as the cue for the next retrieval – after A is retrieved, it is used as a cue and should prompt retrieval of B, which then becomes the cue for C. Such models are often referred to as simple chaining models, because the associations between items resemble links in a chain. Perhaps the best known computational model of this class is the Lewandowsky and Murdock (1989) theory of distributed associative memory (TODAM) model.

Nonetheless, error patterns in serial recall impose strong constraints on simple chaining models. The most noteworthy historical problem is how chaining models recover from errors. If an item is unable to be recalled, how does recall continue from that point? TODAM partially solved the problem of recovery from errors because in the model, a retrieved vector that was unable to produce a valid recall can still be used as a cue for further retrievals, allowing the model to "get back" on the chain after a missing link.

A more serious concern for simple pairwise chaining models is what Henson (1996) referred to as *the locality constraint* - the finding that when an error is made in a given output position, it tends to be an item that was studied near the intended item that was to be output. That is, for a sequence $ABCDE$, when attempting to recall the third item, C is

the most likely response, and errors are more likely to occur from adjacent positions (B or D) and much less likely to come from remote positions (A or E). This regularity in errors is constraining for simple chaining models, which have no obvious mechanism for why, when failing to recall an item, an item studied near that item should be recalled instead. In the case above, when attempting to recall the third item, the cue (B) is only associated to the correct item (C) and has no associations to itself or the other items in the sequence.

The theoretical coverage of inter-item associations can be greatly expanded through the usage of *remote associations* between items at encoding. Consider if instead of merely encoding associations between adjacent items, associations are formed between *all* of the list items, with the strength of the association being proportional to how far apart they are in the sequence. That is, for a list $ABCDE$, $A-C$, $A-D$, and $A-E$ associations are also formed, but are weaker than the adjacent $A-B$ association. Remote associations provide a principled account of the locality constraint – if B is used as a cue, while it has the strongest association to the correct item C, it has a weaker association to its neighbors (B and D) and even weaker associations to the remote items (A and E), making it such that D is more likely to be recalled as an error than E.

In addition, while simple chaining models assume that only the previously retrieved item is the functional cue for the next response, it is possible to instead employ multiple retrieved items as a *compound cue* in a recency-weighted fashion, such that the most recently retrieved item has the greatest emphasis. Usage of multiple retrieved items further bolsters recovery from errors. If $ABCDEF$ is studied and a participant recalls ABE, while the E cue has the greatest weight, the cues corresponding to the previous retrieved items (A and B) can enable retrieval of C. Remote associations can further strengthen this tendency, as the $A-C$ association can further increase the retrieval probability of C.

These two assumptions in tandem – remote associations at encoding and compound cuing at retrieval – can greatly facilitate recovery from errors and coverage of the locality constraint. A prominent example of such a model is the power-set model of Murdock

(1995), which was explicitly built to overcome the limitations of the previous chaining model of Lewandowsky and Murdock (1989). Both assumptions do not have to co-occur– the model of Solway, Murdock, and Kahana (2012), for instance, employs remote associations at encoding, but does not use compound cuing at retrieval. Instead, only the most recently retrieved item is used as a cue. Both models have been demonstrated to be able to capture the locality constraint.

A final relevant consideration for models that rely on inter-item associations concerns what becomes the functional cue for the next response, whether it is the response itself, which we term *response cuing*, or whether it is merely the retrieved content from memory, which we term *memorial cuing*. This distinction has historically not been of interest as the majority of models that employed inter-item associations employed response cuing as the default. However, a number of computational models of serial order have been developed in which output errors can occur independently of memory errors due to errors in the motor or speech production system after memory retrieval has already occurred (e.g., Burgess & Hitch, 1999; Henson, 1998; Page & Norris, 1998). Such an assumption has also been adopted by CRU (Logan, 2018), in which output stage confusions can result in errors, but such erroneous responses do not become the cues for the next retrieval. Instead, it is the retrieved item from memory that enters the context and contributes to the cue for the next response. Memorial cuing is also relevant to recovery from motor errors - if a sequence such as $ABCDE$ is studied, after correctly recalling A the participant might correctly recall the representation of B, but mistakenly output the item as D. However, it is ultimately the representation of B that becomes the cue for the next response, allowing recovery from the error and enabling correct recall of C. As we will demonstrate later in the commentary, this assumption has important consequences for CRU's ability to recover from errors when mixed lists of phonologically similar and dissimilar items are studied.

Historically, the question of what representation underlies serial order remained unresolved for some time despite decades of research. However, a dramatic change occurred

in the 1990's when a number of findings were summarized in Henson's (1996) seminal dissertation that motivated the abandonment of models that rely on inter-item associations and the acceptance of positional models. While models that rely on remote associations can in principle recover from errors, reliance on remote associations still makes it such that after an error, the next response should be located after the erroneously recalled item. While some evidence for this pattern has been found with longer lists using procedures atypical of serial recall (Solway et al., 2012), many studies, including Logan's (2021) own data, have instead uncovered error patterns that are inconsistent with these predictions (e.g., Farrell, Hurlstone, & Lewandowsky, 2013; Henson et al., 1996; Henson, 1999; Osth & Dennis, 2015a, 2015b). This is commonly tested using conditional analyses that focus on what occurs after a participant skips an item during the recall process, such as proceeding from A to C when attempting to recall a list $ABCDEF$. While models that rely on inter-item associations predict that it is more common to continue onward from the erroneous response (producing ACD, referred to as "in-fill"), the data demonstrate that it is much more common to instead proceed backward (producing ACB, referred to as "fill-in"), as if participants are "filling in" the missing response (Farrell et al., 2013; Osth & Dennis, 2015a; Page & Norris, 1998; Surprenant, Kelley, Farley, & Neath, 2005).[1]

An additional phenomenon that was contrary to models that employ inter-item

[1] While Solway et al. (2012) found evidence of an in-fill effect in their re-analyses of several serial recall datasets, these results came from longer lists of items with much higher omission rates. Farrell et al. (2013) responded with a re-analysis of 21 datasets using shorter lists and lower omission rates with the majority of these datasets showing a fill-in pattern. Farrell et al. also conducted a re-analysis of the data from Grenfell-Essam and Ward (2012), who manipulated list length and employed open sets of items, in which the data showed a fill-in effect for lists as short as four items in length but an in-fill effect for list lengths of six or larger. The authors suggested that the inability for participants to indicate omissions as responses may have contributed to this pattern. In response, Osth and Dennis (2015a) conducted several large experiments with six item lists under conditions that reduce omissions (a closed set of items along with a reconstruction of order task), a condition which has high omission rates (an open set of items), and a further experiment with open sets of items where participants can indicate their omission responses. Fill-in was found in conditions with low omission rates (reconstruction of order tasks and closed sets of items). In the open set experiments, an ambiguous result was found (even fill-in and in-fill) when omission rates could not be indicated, while a predominance of fill-in was found when omissions could be indicated. Furthermore, re-analyses of both experiments with open sets found that trials with lower numbers of omissions demonstrated a fill-in pattern.

associations is the finding that with mixed lists of similar and dissimilar items, errors in recalling the similar items produce almost no impairment on the ability to recall the dissimilar items (Henson et al., 1996). When response cuing is employed, like in the TODAM models, the erroneously recalled similar items should serve as misleading cues for further retrievals. While employing remote associations and compound cues at retrieval can potentially mitigate this problem, a more conceptual problem is the fact that in the mixed list half of the items are similar. If compound cues are employed at retrieval, around half of the previously retrieved items should serve as misleading cues for the next response.

While such error patterns are contrary to representations that rely on inter-item associations, other error patterns have been found that strongly suggest that items are associated to the position of their occurrence. The first is the finding that intrusions from prior lists tend to be recalled in the same output position as their within-list position from the previous list (Conrad, 1960; Fischer-Baum & McCloskey, 2015; Henson, 1999; Osth & Dennis, 2015b). The second concerns the finding that when participants study a list of temporally grouped items, when participants recall an item from the incorrect group, it tends to be recalled in the same within-group output position as its position in its original group (Hartley, Hurlstone, & Hitch, 2016; Henson, 1999; Liu & Caplan, 2020; Ng & Mayberry, 2002). Both findings follow naturally from the assumption that items are associated to their position of occurrence and can be captured by positional models (Henson, 1998; Liu & Caplan, 2020), but there isn't an obvious explanation of how such errors could be produced by associations between items.

For these reasons, the majority of current models of serial order have eschewed representations of inter-item associations. Instead, the field has generally converged on item-position associations as the primary representation. In positional models, when an item is retrieved, it is *not* used as the cue for the next retrieval. Instead, the cue corresponding to the next within-list position is employed. Consequently, if a participant is attempting to recall the second item and erroneously recalls the seventh item, the

participant will subsequently employ the cue for the third position, making a correct response the most likely response. Thus, in positional models, errors are far less consequential than in models that are principally reliant on inter-item associations, as an erroneously recalled item does not prevent the correct position cue from being used on the next recall. Such models include the start-end model (SEM: Henson, 1998), ACT-R (Anderson & Matessa, 1997), OSCAR (Brown, Preece, & Hulme, 2000), SIMPLE (Brown, Neath, & Chater, 2007), the Burgess and Hitch (1992, 1999) model, variants of the SOB model (Farrell, 2006; Lewandowsky & Farrell, 2008), the grouping model (Farrell, 2012), and the BUMP model (Hartley et al., 2016). Collectively, such models have been applied to each of the aforementioned phenomena that are challenging to models that rely on inter-item associations. While several of these models have only provided demonstrations of these phenomena in isolation via simulations, some of the models have been able to demonstrate these phenomena after having been fit to either group-averaged data or from individual participants, as we will discuss below.

Despite the field's rejection of inter-item associations at the time, a number of findings have since arisen that demonstrate evidence for inter-item associations. Such findings include repetition advantages for permuted lists that preserve relative positions over repetitions where each of the elements are scrambled (e.g., the spin list advantage Kahana, Mollison, & Addis, 2010; Lindsey & Logan, 2019, 2021), advantages for items being tested in the same relative order as they occur in natural language (Baddeley, Conrad, & Hull, 1965; Botvinick & Bylsma, 2005), and advantages for ordered part-list cues (Basden, Basden, & Stephens, 2002; Serra & Nairne, 2000). For these reasons, we are not suggesting that CRU should abandon item-dependent context representations. Instead, our stance is that CRU should either broaden its representation to include within-list position, or alternatively to consider how its representations can be employed to approximate representations of within-list position. We return to this issue later in the section "Is there evidence for associations between items?" where we provide a more comprehensive review of

these phenomena and our recommendations for both CRU and the field to move forward.

In the next section, we give a brief mathematical description of CRU to illustrate how the model uses inter-item associations. We point out that when the current item dominates the context representation ($\beta = 1.0$), the model resembles a pairwise chaining model, whereas when there is a balance between the current item and the prior context ($\beta < 1.0$), it utilizes remote associations at encoding and compound cuing at retrieval. We follow this section with simulations of CRU with mixed lists of phonologically similar and dissimilar items, intrusions from prior lists, and manipulations of temporal grouping.

CRU and Inter-item Associations

In this section, we give a brief mathematical description of CRU to illustrate its reliance on remote associations and compound cuing, and how the model can mimic pairwise chaining models under some parameterizations. Readers interested in a complete description of CRU should consult the original articles (Logan, 2018, 2021).

As mentioned previously, in CRU, each item is associated to its context of occurrence, which usually consists of the previous items. An exception is the first item in a list, which is bound to a start-of-list context referred to as a $LIST$ vector. All study list items, along with the $LIST$ representation, are represented as an orthonormal vector r.

Each context vector c is composed of a weighted combination of the previous item and the previous context. The context vector for item $N + 1$ is defined as:

$$c_{N+1} = \beta r_N + \rho c_N \tag{1}$$

where the β parameter controls the relative weighting of the current item and the previous context, and ρ is a normalization term to ensure that the length of each context vector is 1:

$$\rho = \sqrt{1 + \beta^2[(r_N \cdot c_N)^2 - 1]} - \beta(r_N \cdot c_N) \tag{2}$$

although $\rho = \sqrt{1 - \beta^2}$ when there are no repetitions and the vectors are orthonormal.

Encoding results in the storage of a context vector associated with each studied item as a separate trace in memory. What is somewhat counterintuitive about the model is that while items are associated to their context vectors, there is not an explicit binding operation between the item vector r_{N+1} and its respective context vector (c_{N+1}). This is one crucial distinction from the temporal context model (Howard & Kahana, 2002), where learning consists of binding the current item vector to its context vector via an outer product operation. As we will demonstrate in simulations, the assumption that an item is not part of its own representation has some important consequences for when item vectors are similar to each other, as the content of an item vector has the largest effect on its successors in the list, but does not affect its own context vector. An additional deviation from the temporal context model is that item repetitions do not produce reinstatement of their prior states of context.

Each context vector is composed of the previous items in the list in addition to the $LIST$ representation. Assuming no repetitions in the list, we can rewrite Equation 1 as:

$$c_{N+1} = \beta r_N + \sum_{i \leq N} (\rho r_i)^{N-i} \tag{3}$$

where it can be seen more clearly that each context vector that is added to memory is composed of the previous item vectors, where each item vector prior to item N is weighted by ρ raised to the power of its recency. As Logan himself notes: "Items are associated with contexts made of previous items, so in effect, items are associated with each other." (Logan, 2021, p.2). This property wherein items are associated to all previous items in a recency-weighted fashion is shared by models that rely on remote associations at encoding. However, one should note that when $\beta = 1$, $\rho = 0$, which simplifies the expression to $c_{N+1} = r_N$. In other words, when $\beta = 1$, each context vector consists of only the previous

item, meaning that only adjacent associations are stored, like in pairwise chaining models.

Context evolution is also used to guide recall. That is, when retrieval begins, the context representation is cleared and initiated with the $LIST$ representation. The similarity between the current context cue c_c and the context vector for a given item i is calculated via the dot product to produce a drift rate v_{mem}:

$$v_{mem,i} = c_c \cdot c_i \tag{4}$$

Drift rates are calculated for all of the letters that were on the list. These drift rates drive a competitive race between each of the list items, which is implemented as a racing diffusion process. Higher drift rates produce more rapid accumulation to the threshold θ, which is conventionally fixed to 200.

Once an item wins the race, its item vector r enters the context, and context evolution proceeds according to Equation 1, and the cycle repeats. In other words, context evolution at retrieval proceeds in the same manner as it does during learning the study list items. Thus, context contains a recency-weighted combination of the recently *retrieved* items as the cue for recall, meaning that the model can employ compound cuing at retrieval. However, similar to its assumptions about encoding, this again depends on the value of the β parameter. When $\beta = 1$, the last retrieved item dominates the context vector. Conventionally (but not necessarily), the same value of β is used at both encoding and retrieval, meaning that when $\beta = 1$, the model can be considered a pairwise chaining model, where adjacent associations are formed at encoding and the cue consists of the last retrieved item. It is for this reason that our simulations below vary the β parameter across a wide range, and we will illustrate how the model can yield very different predictions when $\beta = 1$.

In practice, the fits of CRU to data suggest that $\beta < 1$ (Logan, 2018, 2021), implying

the usage of remote associations at encoding and compound cuing at retrieval. As mentioned previously, such assumptions enable the ability to capture the locality constraint and the ability to recover from errors, both of which were well demonstrated in typing, serial recall, and whole report. Nonetheless, one remaining limitation of the model noted by Logan (2021) was that the model was not able to produce the fill-in effect. Fill-in to in-fill ratios are typically around 2.0 (Page & Norris, 1998), whereas in Logan's data they were much higher, around 3.80. The ratios calculated from CRU were considerably lower, around .5776, indicating the model performed in the opposite manner to how the participants performed and tended to proceed in the forward, rather than backward, direction after an error is made. This prediction is somewhat perplexing because as Logan noted, the similarity between the encoded context vectors is symmetric, meaning that the context of D is equally similar to both B and C. While that would suggest an error ratio of around 1.0, the dynamics of context evolution during retrieval complicate the picture. Using algebraics, Logan demonstrated that after an omission, CRU is actually guided toward later positions: "CRU predicts an asymmetry in favor of later positions following an initial omission." (Logan, 2021, p.17).

One solution Logan attempted was to revise the context representation at retrieval to use a weighted combination of the $LIST$ node and the current context. The $LIST$ node is most strongly associated to the early items on the list, and therefore additional weight on the $LIST$ node in the context vector at retrieval functions to "pull" retrieval in the backward direction after an error. While Logan noted through simulations that this can in principle produce the appropriate fill-in error ratios, unfortunately fits to data demonstrated a trivially low weight of the list node, such that the model still produced error ratios that were in the opposite direction of what was found in the data. For a more complete description of CRU's ability to capture fill-in error ratios, we invite readers to consult the Logan (2021) article. We summarize these results here because we argue that CRU's predicted in-fill pattern is exactly the type of consequence that emerges from

models that rely purely on inter-item associations.

As we mentioned previously, one important deviation that CRU makes from existing models that rely on inter-item associations is its reliance on memorial cuing rather than response cuing. That is, after an item is retrieved, there is an additional decision about which item is to be output, where each item's rate of accumulation in the race is determined by the spatial distance between the retrieved item and all other items on the keyboard. Critically, if an erroneous response is made (typing the letter "g" when "f" was retrieved from memory), the erroneously recalled item is not added to the context vector. Instead, it is the item that was retrieved from memory ("f"). While this component was not present in the (Logan, 2021) article, it was a critical component of the initial CRU model of typing (Logan, 2018). We will demonstrate in the next section that this enables the model to capture the mixed list similarity effect when output-based errors can be phonological in nature instead of reflecting the proximity of the items on the keyboard.

Serial Order Phenomena that Motivated Positional Models

Mixed-lists of Phonologically Confusable Items

Among the most historically constraining patterns of data came from mixed-lists of phonologically confusable items. It has generally been well-established that lists composed of phonologically confusable items, such as rhyming consonants (B, C, P, etc.), are recalled more poorly than lists of nonconfusable items that do not share a rhyme (K, X, L, etc.) due to the frequent order errors among confusable items (Conrad & Hull, 1964). Such an impairment can be accounted for by most theories that posit some confusability or similarity among the item representations for phonologically confusable items. However, a more interesting and constraining test comes from mixed lists in which confusable and non-confusable items are presented in an alternating pattern, such as CKPXGL (where confusable items are underlined).

As mentioned previously, when associations are formed among list items and the

responses associated with retrieved items are used as cues for the next retrieval, if an erroneous item is recalled on a given output position i, there is a higher likelihood that the next output position $i + 1$ will contain an error than if output position i contained a correct response. However, when errors are plotted by output position, such mixed lists show elevated error rates for the confusable items, such as C, P, and G, whereas the error rates for nonconfusable items are not higher than on pure lists of nonconfusable items (Baddeley, 1968; Henson et al., 1996). Figure 1 illustrates this phenomenon with data from Page et al. (2007) in which participants studied and recalled lists composed of purely confusable items (PC), purely nonconfusable items (PN), and alternating lists of confusable and nonconfusable items, with confusable items occurring either in odd (ANC) or even (ACN) serial positions.

Furthermore, investigations have even found that nonconfusable items can *benefit* from the presence of confusable items in mixed lists relative to pure lists of nonconfusable items (Farrell & Lewandowsky, 2003; Farrell, 2006; Lewandowsky & Farrell, 2008) – this pattern is also evident in Figure 1, where the nonconfusable items in mixed lists are often recalled better than their counterparts in pure nonconfusable lists. Furthermore, these same phenomena can also be found in speech production errors (Page et al., 2007). Such findings were highly instrumental in motivating a departure from the reliance on inter-item associations in theories of serial order. Indeed, Henson et al. (1996) went as far as to claim that the mixed-list findings "rule out" chaining models of serial recall. However, these claims critically rest on the assumption that the responses are used as cues in retrieval rather than the retrieved content from memory.

As mentioned previously, CRU departs from other previous models that employ inter-item associations in its treatment of item similarity. In CRU, item vectors are orthonormal and completely dissimilar to each other – it is the similarity of the context vectors that produces interference during the retrieval stage. Confusions between items can occur in two other stages of the model. The first is during an encoding stage, where an

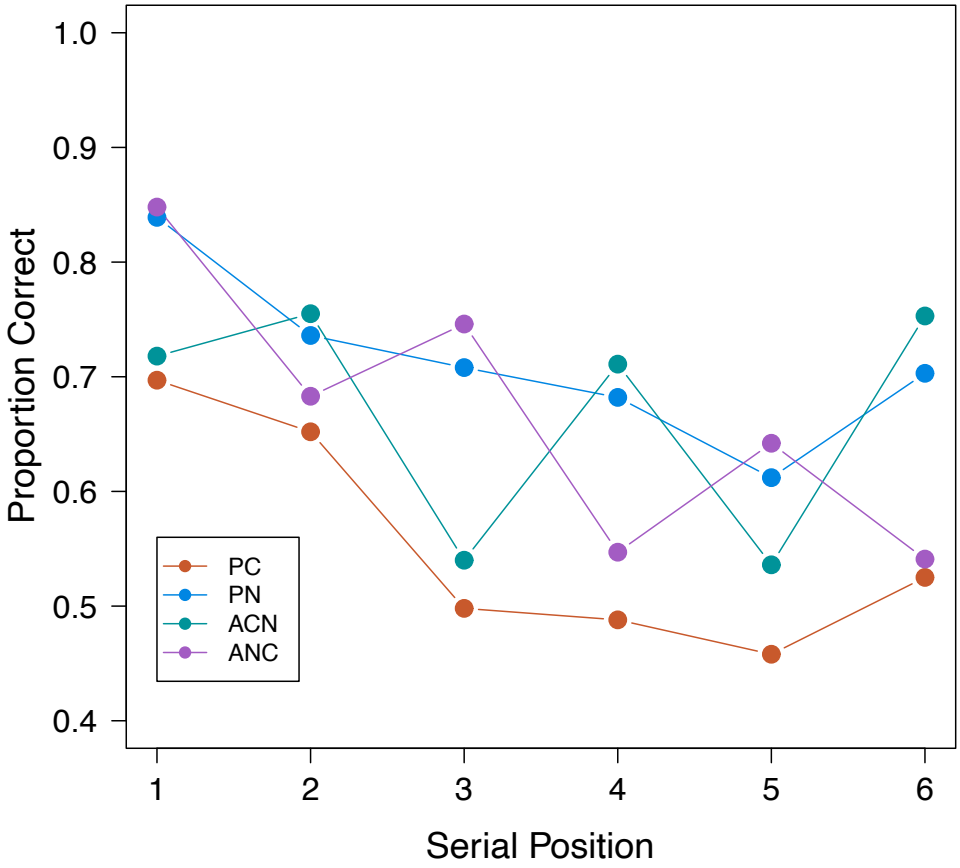

Figure 1. Accuracy serial position curves for verbal serial recall of lists composed of purely confusable consonants (PC), purely nonconfusable consonants (PN), and alternating lists of confusable and nonconfusable consonants, with confusable consonants occurring either in odd (ACN) or even (ANC) serial positions. Data taken from Page et al. (2007).

item can be erroneously perceived as a different item (e.g., reading the letter "e" as "f"). Like in the case of memory retrieval, this process is a competitive race between each of the letters. The strength of each letter i is determined by an exponential transformation of the distance d between it and the presented letter j:

$$v_{encode,i} = exp(-gd_{ij}) \tag{5}$$

where d_{ij} is the distance between the two letters in a multidimensional scaling solution (MDS) based on visual confusions of the letters and g is a distinctiveness parameter.

Increases in g reduce $v_{encode,i}$ for letters that are not the target item, reducing the probability of encoding errors.

The second mechanism for similarity effects is during an output stage that occurs after the memory retrieval stage, where a retrieved item can be output as a different item due to motor errors (Logan, 2018). This operates in the same manner as reading:

$$v_{out,i} = exp(-gd_{ij}) \tag{6}$$

with the crucial distinction being that the distance d_{ij} is derived from the distances between keys on a keyboard, making it such that a letter may be accidentally output as an adjacent letter on a QWERTY keyboard. While this mechanism was not used in the Logan (2021) article, in the General Discussion, Logan notes that this mechanism could be modified to produce confusions based on phonological similarity: "This idea could be generalized to multidimensional representations of response alternatives, like phonological codes for spoken words or letter names." (Logan, 2021, p. 32)

What is critical for predictions of mixed list similarity effects is that output-based confusions do not impact the input to the context layer. For instance, if "r" was retrieved and "t" was erroneously reported, the item vector for the retrieved letter "r" is what enters the context and serves as the functional cue for the next response. In a mixed list of similar and dissimilar items, such a mechanism implies that output-based confusions of phonologically confusable items will not affect context updating, and thus will not disrupt performance on the subsequent retrieval. It is important to note that several models of serial order, including positional models, have adopted a similar approach where confusions of phonologically similar items occur during an output stage and not at memory retrieval (e.g., Brown et al., 2000; Henson, 1998; Page & Norris, 1998).

We adopted a similar approach as Logan (2018) used. However, we had to make an

important departure from that approach in order to address phonological similarity effects. Instead of using distances between letters on a keyboard, we used a simulated distance matrix to make the output-based errors phonological in nature. For each letter pair, we simulated the distance value d from a truncated normal distribution. For pairs of confusable letters (B, D, G, P, T, and V), we used $\mu = .2$ and $\sigma = .2$. For pairs of nonconfusable letters, we used $\mu = 1.80$ and $\sigma = 1.0$. For the distance between confusable and nonconfusable letters, we used $\mu = 3.0$ and $\sigma = 1.0$. These values were chosen to roughly accord with the distances in an MDS solution of a limited pool of confusable and nonconfusable letters performed by Farrell (2006). For the sake of simplicity, we assumed that the g parameter in the output stage was fixed across output positions.

We simulated this CRU variant's predictions for the Page et al. (2007) experiment using the same list structure and number of trials. The CRU variant was simulated with four different values of β (1.0, .65, .45, and .25) crossed with four values of g for the output stage (.1, .3, .5, and 1.0). To simplify the predictions, g did not vary across serial positions ($g_{decrease} = 1.0$). With higher values of g, output-stage confusions are less likely for both confusable and non-confusable items. Additional simulation details can be found in the Appendix.

Results of the CRU simulations with output-based confusions can be seen in Figure 2. The variation in parameter values primarily results in differences in performance – higher values of β and lower values of g produce better performance overall. In terms of the qualitative predictions for the mixed list similarity effect, the model does an impressive job, with virtually all combinations of parameter values showing the qualitative sawtooth pattern that resemble the data in Figure 1. However, an analysis of the model's errors (not depicted in Figure 2) found that repetition errors were high, and there was a much higher incidence of repetition errors in pure-confusable lists relative to mixed lists, with several parameter combinations showing repetition errors that were twice as frequent in pure-confusable lists. These error results are at variance with data showing that repetition

errors are both rare and do not vary across list types (Henson et al., 1996).While the repetition errors caused by output-stage confusions could also be remedied by the introduction of response suppression, this model also yields a high incidence of extralist intrusions. They are particularly common in mixed lists, where as many as 2-3 extralist intrusions occur per list, depending on the parameter values.

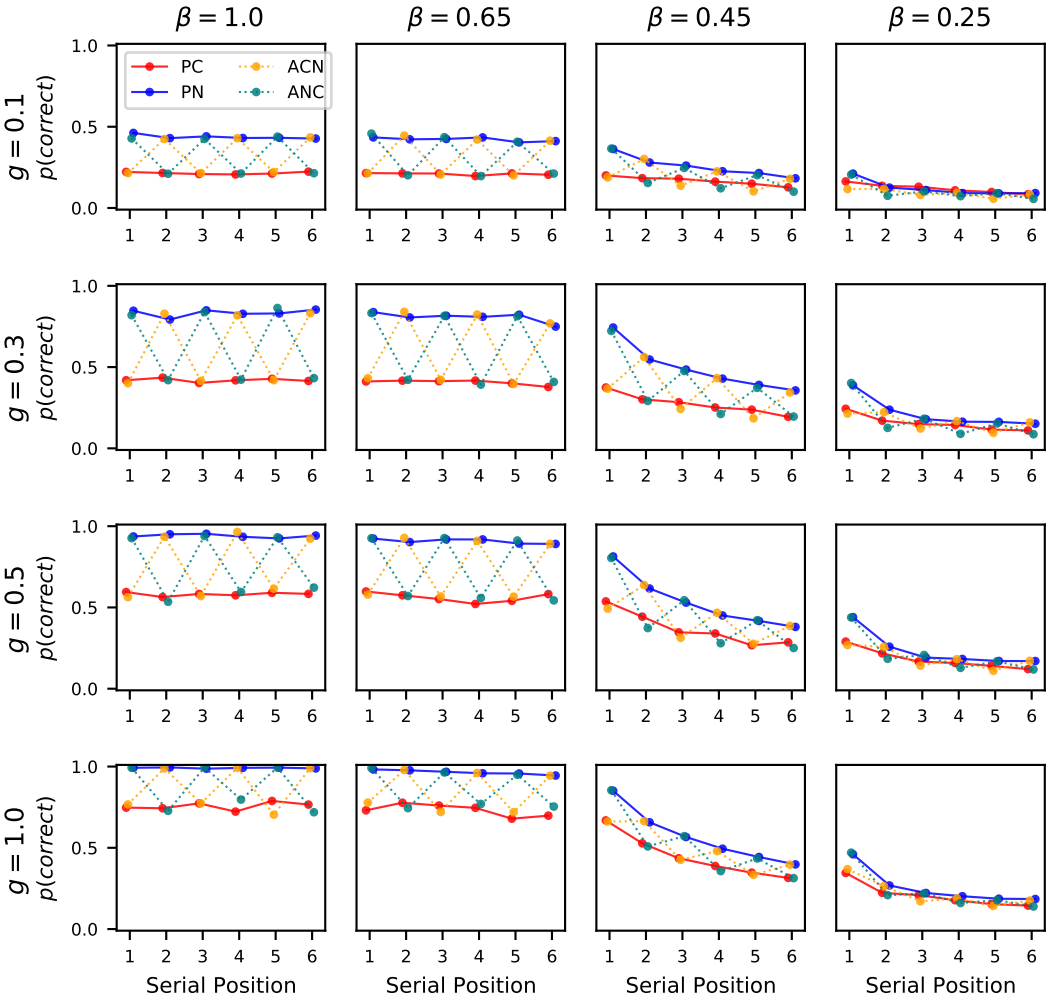

Figure 2. CRU simulations for the Page et al. (2007) paradigm using output-based confusions. Notes: PC = pure confusable, PN = pure nonconfusable, ACN = alternating confusable-nonconfusable, ANC = alternating nonconfusable-confusable.

Why is there such a high incidence of repetition and extralist intrusions, given that such errors are rare in the data? And why are extralist intrusions more common in mixed

lists? The reason why is because confusable items are likely to be confused with other confusable items in the output stage regardless of the contents of memory. This means that when a confusable item such is retrieved from memory, during the output stage it can be confused with another confusable item that was already retrieved or a confusable item that is outside of the study list.

In pure lists of confusable items, six letters are studied out of the seven possible rhyming letters, making it such that there is only one rhyming letter outside the list that can be confused with the study list items, while there is a high probability of confusing a letter with another letter from the experimental set. In mixed lists, in contrast, three of the seven possible rhyming letters are on the study list, providing more opportunities to confuse a rhyming letter with one outside the study list, increasing the proportion of extralist intrusions.

What can be done to improve CRU's predictions for this paradigm? Several models of serial recall similarly rely on output stage confusions to capture similarity effects like we have done here. However, response selection in the output stage is based on a product of the similarity between the context cues and the item cues (e.g., Brown et al., 2000; Burgess & Hitch, 1999; Henson, 1998). In other words, items that are both perceptually similar *and* match the context cues are most likely to be output. Items not presented on the study list do not match the context cues, and thus are unlikely to be output during this stage according to the product rule.

We pursued a similar approach within CRU. In order to use the product rule, we re-used the drift rates from the memory retrieval decision stage (v_{mem}). Specifically, the drift rates for the output stage v_{out} are a weighted product of v_{mem} and drift rates that reflect phonological similarity, which we refer to as v_{phono}, where v_{phono} is calculated according to Equation 6. The final drift rate for a given item i is:

$$v_{out,i} = v_{mem,i}^{w} v_{phono,i}^{(1-w)} \tag{7}$$

where w is a weighting parameter between 0 and 1 that reflects the relative weighting of context similarity and phonological similarity. We pursued a range of different values and generally found the best correspondence with relatively low values of w. One should note that so long as w is greater than zero, v_{out} will be zero for all items where v_{mem} is zero, which reduces v_{out} for all items that were not studied on the list. It also reduces v_{out} to zero for items that were present on the list, but were sufficiently distant from the target item. This is especially the case when β is high, as increases in β reduce the similarity between non-adjacent context vectors. These assumptions place considerable constraint on the phonological confusions that can occur during the output stage. We fixed $w = .05$ in all simulations, as we found this bore the closest resemblance to the patterns found in empirical data. For balanced values of w, such as when $w = .50$, little qualitative correspondence was found between CRU's predictions and the data.

CRU simulations with the output stage that combines contextual and phonological similarity can be seen in Figure 3. In contrast to previous simulations, the patterns of data depend heavily on the value of β. When $\beta = 1$, performance is close to perfect, as in the base version of the model. This is because context vectors only show self-similarity when $\beta = 1.0$ – all other context vectors are completely dissimilar, making their values of $v_{mem} = 0$ and $v = 0$ as a consequence. This eliminates the phonological similarity effect because any confusable items that are more than one position apart from the previously recalled item cannot be produced during the output stage. For the simulations where $\beta = .65$, a potential shortcoming of the model is that performance for confusable items in pure lists is considerably worse than in mixed lists. However, this problem was more evident with lower values of g, suggesting that higher values of this parameter are more

appropriate. Analyses of errors found that this was due to a high incidence of repetition errors. This problem could again be mitigated with the usage of response suppression.

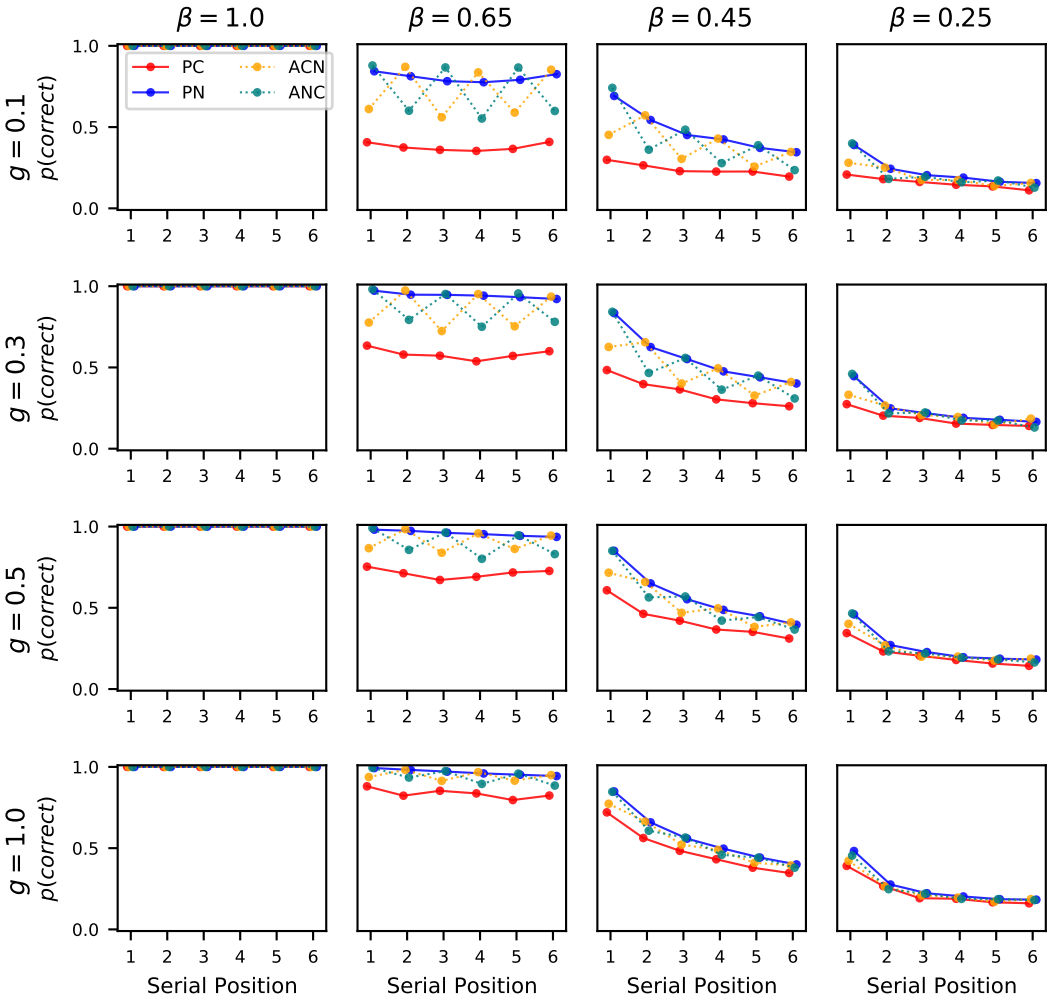

Figure 3. CRU simulations for the Page et al. (2007) paradigm using output-based confusions that combine contextual and phonological similarity. Notes: PC = pure confusable, PN = pure nonconfusable, ACN = alternating confusable-nonconfusable, ANC = alternating nonconfusable-confusable..

When $\beta = .45$, in contrast, a very reasonable correspondence with the qualitative patterns of data is achieved. The model even appears to achieve better performance for both confusable and nonconfusable items in mixed lists than pure lists to a degree that reasonably corresponds with the data. A much weaker correspondence is found when

$\beta = .25$, but memory is very poor in this condition. Nonetheless, what is most encouraging about these simulations is that extralist intrusions were minimal for all parameter combinations, as the product rule ensures that only candidates that bear both non-zero contextual and phonological similarity to the retrieved item can be output.

Other Methods of Accounting for Phonological Similarity Effects. In addition to allowing for phonological output confusions, we also explored two other possible loci of phonological similarity effects, the results of these simulations can be found in Supplementary Materials A. While these methods were not endorsed by Logan (2021), these simulations provide insight into other possible mechanisms, and the limitations of these variants are illustrative and give further weight to the variant where output-stage confusions are responsible for phonological similarity effects.

First, we explored a CRU variant where phonological similarity is between the item vectors in the model and implemented this with a wide range of parameter values. Across the majority of parameterizations, poorer performance was predicted for phonologically similar items, but there was little evidence of a "sawtooth" pattern. Instead, errors in mixed lists continued to result in poorer performance that did not increase afterward.

What was counter-intuitive about these simulations is that errors in mixed lists occurred on the *nonconfusable* items – the opposite of what is found in the data. This is due to the assumption that each item vector is not contained within its own context vector. Because context vectors contain the previously studied item as its strongest element, and the mixed lists in this paradigm use an alternating structure, a confusable item contains its preceding nonconfusable item as its strongest element, whereas a nonconfusable item contains its preceding confusable item as its strongest element. Consequently, when there is similarity between item vectors, the presence of a similar item in the context cue exhibits its highest similarity to nonconfusable items. While these simulations explored only a limited range of model parameters and methods of manipulating item vector similarity, other parameter values and implementations will share this conceptual problem unless

additional assumptions are incorporated into CRU.

Why does the model perform poorly with similar item representations, given that compound cuing and remote associations (which occur when $\beta < 1$) should promote recovery from errors? First, with high values of β, a confusable item dominates the context cue and should mislead memory retrieval. Second, in a mixed list, around half of the preceding retrieved items are likely to be confusable items, making it such that compound cuing will provide relatively little benefit. While these were simulations performed with a somewhat limited range of parameters and means of manipulating item vector similarity, it is unclear how other parameter values or implementations of item vector similarity could overcome this conceptual problem. Such a problem can be compounded in mixed lists where there is only a single nonconfusable item and five confusable items (e.g., Farrell & Lewandowsky, 2003; Farrell, 2006).

The second CRU variant we explored was a model where phonological similarity resulted in confusions at encoding. To implement this model, we used the same simulated distance matrix as our simulations with output-based confusions and implemented it with the same range of parameter values. This variant was similarly able to reproduce the sawtooth serial position curves which appear impressive. However, the model had two shortcomings. First, similar to our first output-stage confusion variant with no contribution of memory-based similarity, the model exhibited a high degree of extralist intrusions and repetition errors. However, with encoding-based confusions, such errors are more consequential. Unlike the output-stage, erroneously perceived items enter the context layer and are learned, which have the potential to causer further errors at recall. Consequently, the model predicted worse performance for both classes of items in mixed lists.

Discussion. In this section, we have pursued a number of different mechanisms for implementing phonological similarity effects in CRU. We found the closest correspondence was when item confusions occur during an output stage. While this can result in a high degree of extralist intrusions or repetition errors, this problem was mitigated by using a

product rule for the output stage, where each item's drift rate is a product of its context similarity and phonological similarity, with context similarity serving to restrict output candidates to items that were studied on the list.

These CRU simulations demonstrate that phonological similarity effects in pure and mixed lists are not nearly as constraining as previously stated. Henson et al. (1996) made the strong claim that such patterns "rule out" models that rely on associations between items. The simulations presented here demonstrate that this is not the case. CRU – which essentially learns associations between an item and all items experienced prior to its presentation – can reproduce the "sawtooth" serial position curves when it relies on confusions during an output stage, if such confusions do not influence the next cue for recall. If the next cue for recall is not changed, then an error during output has no influence on the next retrieval, which enables the model to recover from a motor error.

While the usage of output-based confusions might seem ad hoc given that motor errors were originally specified as confusions among locations on the keyboard (Logan, 2018), extension of the output stage to represent phonological confusions was suggested in the Logan (2021) article. In addition, this mechanism is common to the majority of existing models of serial recall (including positional models), where confusions between phonologically confusable items occur during a second stage where responses are selected for output (e.g., Brown et al., 2000; Burgess & Hitch, 1999; Henson, 1998; Page & Norris, 1998). Indeed, virtually all of these existing models simulated successful qualitative reproductions of the Henson et al. (1996) data. Thus, the criticism that such a mechanism is ad hoc can additionally be levelled against other models that rely on the same mechanism. Furthermore, evidence for phonological confusions in the response stage comes from the fact that the same error patterns can be found in speech production where no memory retrieval is required (Page et al., 2007).

A caveat of all simulation work of this kind is that such simulations demonstrate what a model is capable of with a specific set of parameters. Simulation of a phenomenon

in isolation demonstrates a proof-of-concept that a model can account for a phenomenon under that condition, but does not guarantee that this success will generalize to fits to real data, where the constraints of fitting individual responses and variations across individual participants may "steer" the model into a different parameterization where the phenomenon is no longer predicted. Likewise, introducing a mechanism of this kind may also compromise the model's ability to capture some of the phenomena already demonstrated in the original articles. Thus, it would be fruitful to evaluate the consequences of including phonological confusions during output within CRU for other paradigms, including Logan (2021)'s existing data that includes typing, serial recall, and whole report tasks. Given that output stage confusions can introduce further errors into the model, it is possible that this revision can dramatically change the predictions of the model for the data that the model has already been applied to.

Nonetheless, we are somewhat optimistic about the inclusion of phonological confusions during the output stage given that other models have generally been successful in implementing such a stage even after the models have been fit to data. Models such as SEM, SOB, and the primacy model (Page & Norris, 1998) have been successfully able to capture phonological similarity effects while simultaneously capturing the primacy and recency effects and the shapes of transposition gradients after having been fit to group-level data (Farrell, 2006; Henson, 1998; Lewandowsky & Farrell, 2008). Nonetheless, fits to individual participant data, especially at the level of individual responses as was done by Logan (2021), remains an important direction for future work.

A particularly strong challenge for CRU may be the finding that when only a single nonconfusable item is presented among a set of confusable items, performance is hugely improved for the nonconfusable item relative to both mixed lists comprising 50% confusable and nonconfusable items and pure nonconfusable lists (Farrell & Lewandowsky, 2003; Farrell, 2006), which is essentially a Von Restorff effect (von Restorff, 1933; Hunt, 1995). This advantage for the isolated nonconfusable item is naturally captured by the

SOB model due to its reliance on similarity-sensitive encoding. In SOB, encoding strength is inversely proportional to the similarity between an incoming item and the contents of memory. Because a single nonconfusable item contrasts heavily with the preceding set of confusable items, its encoding strength is much greater than when it is studied amongst other nonconfusable items.

A mechanism such as similarity-sensitive encoding is orthogonal to the issue of representation. With that being said, an architecture such as SOB that relies on positional representations is robust to similarity among the item vectors – a necessary requirement for similarity-sensitive encoding – because errors on confusable items do not influence the cue for the next response. Our simulations demonstrating the highly deleterious effects of item vector similarity on CRU's predictions (which can be found in Supplementary Materials A) show it may not be able to incorporate similarity-sensitive encoding in the same way. While we acknowledge that we have explored CRU with similar item vectors under a limited range of parameters and implementations of item vector similarity, it is not obvious how other parameterizations and implementations could circumvent these problems. In paradigms where only a single nonconfusable item is studied, the context vector that cues the nonconfusable item will potentially be composed of a large number of similar vectors, which can serve as misleading cues.

CRU may be able to account for isolation effects using higher values of the β parameter for nonconfusable items when they are accompanied by confusable items, but ultimately a mechanism is required to explain why this occurs. One possible mechanism might be that items that are more similar to the current context produce less contextual change. Siefke, Smith, and Sederberg (2019) presented a variant of the temporal context model that uses exactly this principle and found it was able to produce isolation effects similar to those of the Farrell and Lewandowsky (2003) paradigm. While CRU could benefit from such a mechanism, it likely incurs the same costs as a similarity-sensitive encoding mechanism. That is, in order for confusable items to be more similar to the

current context than nonconfusable items, there has to be some similarity among the item vectors, which is detrimental to performance. If all item vectors are orthonormal, then all items that are not repetitions exhibit zero similarity to the current context during the encoding stage, predicting no difference between confusable and non-confusable items.

The Protrusion Effect

The effects of phonological similarity in pure and mixed lists were not direct evidence for positional models, as such models had to rely on confusions during an output stage to capture such effects. Indeed, the primacy model of Page and Norris (1998) was able to capture such effects using confusions during the output stage, while principally relying on an *ordinal* representation of serial order. In ordinal models, items are not associated with within-list positions or with other items. Instead, order is represented purely in a declining strength gradient, with the first item being the strongest, the second item being weaker in strength, etc. (e.g., Farrell & Lewandowsky, 2002; Grossberg & Pearson, 2008). We have demonstrated in the previous section, CRU is able to accommodate such effects with a similar mechanism.

More direct evidence for positional representations comes from the correspondence between within-list serial position of prior-list intrusions and output position when recalling the current list. Specifically, participants are most likely to produce an intrusion that matches the within-list serial position of the item they are *attempting* to recall (Conrad, 1960; Fischer-Baum & McCloskey, 2015; Henson, 1999; Osth & Dennis, 2015b). For instance, if a participant is attempting to recall the third item, they are most likely to intrude the third item from the prior list. We depict this pattern in Figure 4 using data from two different list length conditions (5 and 6 items) in Osth and Dennis (2015b). This figure depicts the proportion of intrusions from each serial position in the immediately prior list separately for each output position in the current list. Notably, each output position depicts a gradient that is centered on the same serial position on the prior list, a

pattern which is especially pronounced for the first and final item. This effect was dubbed the protrusion effect because it is as if the items from the prior list protrude downward into their same position in the current list. The most natural explanation of the protrusion effect is in terms of positional representations. If participants are using a positional cue, it will match the items that were studied in similar positions, regardless of the study list they were originally studied in.

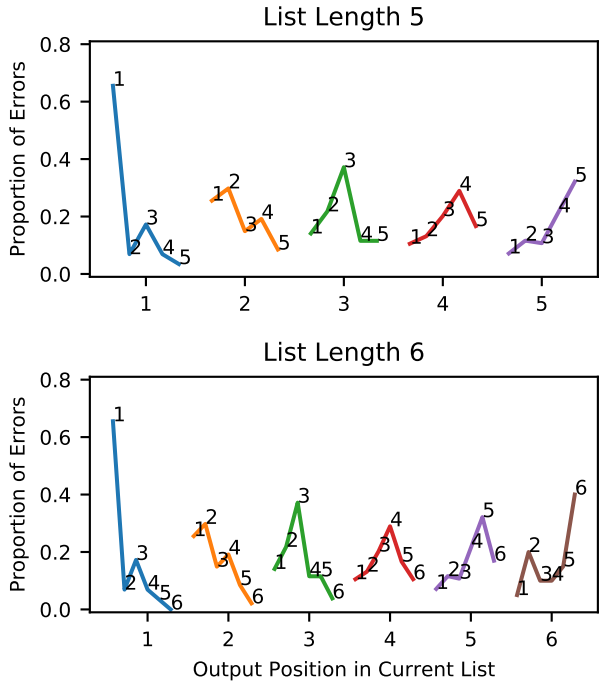

Figure 4. Proportions of prior-list intrusions from each serial position (indicated by the digits) in the immediately prior list plotted as a function of output position from the current list. Data are replotted from Osth and Dennis (2015b) for a dataset with a list length of 5 items (top panel) and 6 items (bottom panel).

One possible explanation for protrusion effects is that there are circumstances that induce participants to associate items to within-list positions instead of other items. In his review of the serial learning literature, Young (1968) provided evidence that the relative reliance on item-position and inter-item associations can depend considerably on a number of factors, including the instructions given to the participants, suggesting that participants

are flexibly adapting their representations to the demands of the task. Many serial recall tasks often employ a small set of items that are repeatedly re-used across trials (e.g., closed sets), and one possibility proposed by Kahana et al. (2010) is that these circumstances induce participants to employ item-position associations to compensate for the massive degree of proactive interference from prior trials.

While it is indeed the case that many demonstrations of the protrusion effect have come from closed sets of items (Henson, 1999; Conrad, 1960; Fischer-Baum & McCloskey, 2015), the experiments of Osth and Dennis (2015b) used a very large set of items (specifically words) such that stimuli were never re-used across trials (e.g., an open set). This procedure was used to specifically test the proposal of Kahana et al. (2010) that item-position associations are employed specifically to compensate for the demands of closed sets. However, as indicated in Figure 4, these experiments also clearly demonstrated a protrusion effect. While these results did not preclude the possibility that the relative reliance of item-position and inter-item associations can vary depending on the size of the experimental set, they do reject the possibility that item-position associations are exclusively employed in experiments utilizing closed sets.

Can CRU produce protrusion effects despite the fact that the model lacks representations of within-list position? While CRU has not been applied to demonstrations of intrusions from prior lists, we performed simulations of a two list paradigm where a study list of six items was studied followed by a second study list. The second study list had a unique $LIST$ representation from the first list. This is undoubtedly an oversimplification, as a "true" reflection of learning would involve either several lists in memory (Lohnas, Polyn, & Kahana, 2015) or all of the lists over the learning episode (Fox et al., 2020). However, it is a useful way to evaluate whether CRU can produce the protrusion pattern.

However, one consequence of these simulations is that the orthonormal vectors within CRU make it such that prior list intrusions almost never occur. This may seem counter-intuitive given that elements of list 1 are present in the list 2 context vectors. For

instance, if list 1 is $ABCDEF$ and list 2 is $GHIJKL$, the context vector for G is $LIST_1 - A - B - C - D - E - F - SPACEBAR - LIST_2$, with the magnitude of the list 1 elements being proportional to the value of β. However, the reason why prior list intrusions will still not occur is that none of the list 2 items are present in the context vectors from the list 1 items.

In order for prior list intrusions to occur due to confusions between the two lists at retrieval, the model requires generalization between the context vectors from the two lists. We implemented this in two ways, namely similarity between the list context vectors ($LIST_1$ and $LIST_2$) and similarity among the item vectors. While it is conventional for CRU to instead employ orthonormal vectors for both types of representations, we found the introduction of vector similarity to be a somewhat desirable reason to capture prior list intrusions as this makes it such that confusions between the two lists are a consequence of the similarity between the stored context vectors from each list. This is consistent with the underlying logic behind CRU, namely that retrieval errors are due to the similarity among the context vectors.

CRU Simulations with Similarity in List Contexts. To manipulate the similarity among the list context vectors, we varied the similarity of the $LIST$ representations in a similar manner as to our simulations of item vector similarity in Supplementary Materials A. Specifically, each $LIST$ vector is a weighted combination of two vectors, a unique orthonormal vector u as well as a common vector m:

$$LIST_i = (\sqrt{1 - s_{list}})u_i + \sqrt{s_{list}}m \tag{8}$$

where s_{list} is a similarity parameter between 0 and 1 that governs the relative weight of the common vector. As s_{list} approaches one, both $LIST$ vectors become identical. Higher values of s_{item} make it such that the $LIST_2$ cue will match any of the context vectors

stored within list 1 to a degree that is proportional to the activation of the $LIST_1$ element.

First, we conducted some analyses on the similarity between the context cues for list 2 in each output position and the stored context vectors for both list 1 and list 2 with a range of parameter values ($s_{list} = 0$, .25, .5, and .75 and $\beta = 1.0$, .65, .45, and .25). That is, for output position 1, we begin with a context vector that is the $LIST_2$ vector and calculate its similarity to all stored context vectors from both lists. For output position 2, we use a context vector that contains $LIST_2 - G$, for the third output position the context vector contains $LIST_2 - G - H$, and so forth.

The similarity gradients for the context vectors for each serial position (indicated by the numbers over the lines) and output position (indicated beneath the x-axis) can be seen in Figure 5 for both list 1 items (left column) and list 2 items (right column). The list 2 columns show the usual expected results for a single list simulation, as the similarity gradients peak for the correct item and are more strongly peaked for higher values of β. Higher similarity among the list contexts (s) does not disrupt this qualitative pattern, although for lower values of β, higher values of s produce lower similarities to the list 2 context vectors.

The similarity gradients for the list 1 context vectors can give some insight into how the model can produce protrusions. First, when $s_{list} = 0$, there is no similarity of the list 2 context cues to any of the stored list 1 context vectors. When $s_{list} > 0$, similarity becomes more evident. However, what is noteworthy is that the shapes of the similarity gradients do not qualitatively change with output position. Instead, they are strongly primacy focused, favoring the first item from list 1 regardless of the output position.

Why do the similarity gradients show a correspondence for the first item, but not for any of the other positions? The answer is because the $LIST$ representation behaves in the same manner as a start-of-list positional representation, which is most active for the first item from each list. The similarity between the two list vectors makes it such that the first output position's context cue ($LIST_2$) matches the first item's stored context vector (which

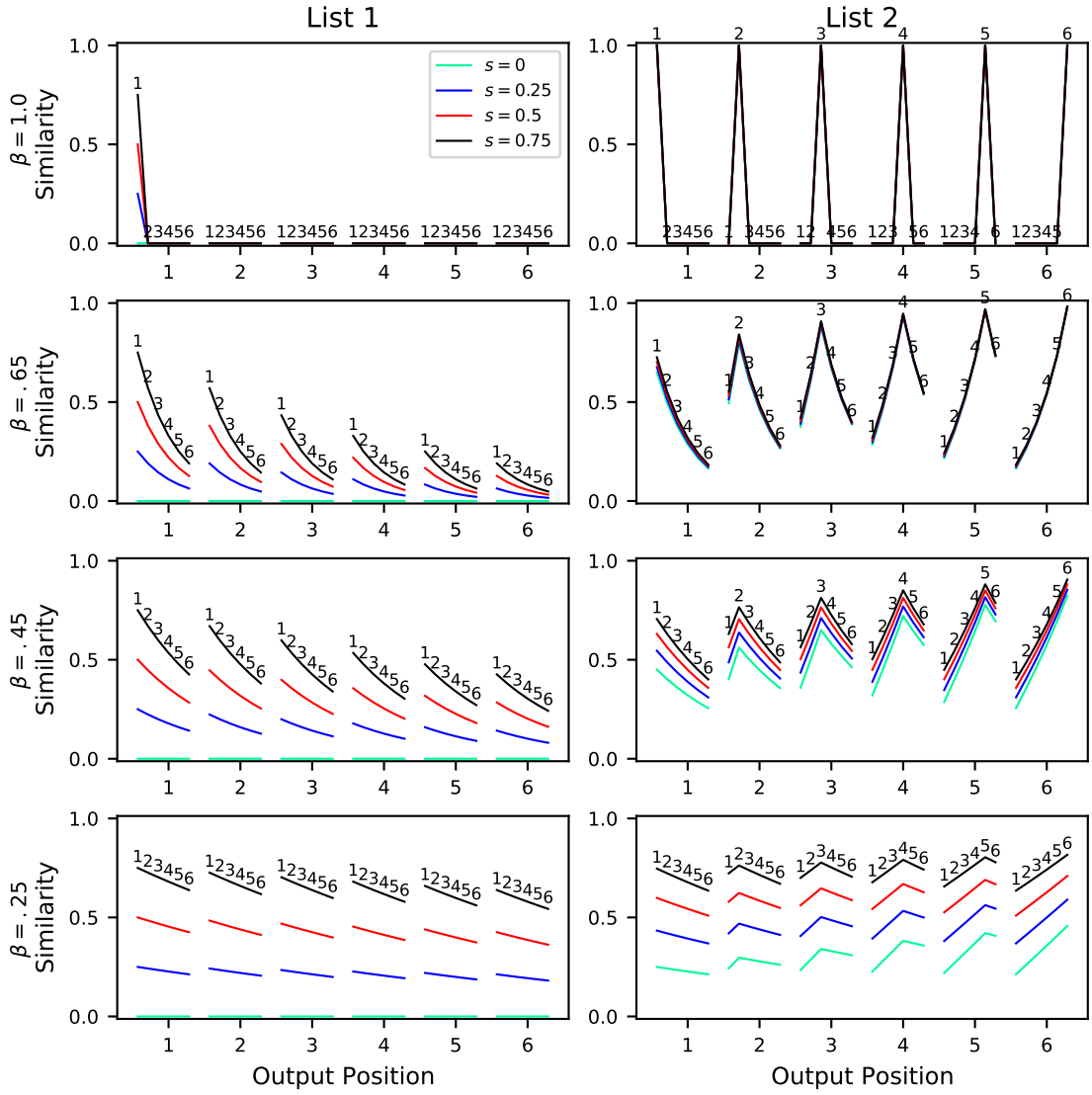

Figure 5. Dot products between the list 2 context cue for each output position (indicated below the x-axis) and the stored list 1 context vectors (left column) and list 2 context vectors (right column) when similarity among list context vectors is manipulated (as indicated by the s_{list} parameter). The serial positions for each context vector are indicated by the numbers above the lines. Note that the context cue for each output position in list 2 assumes the previous items were correctly retrieved.

contains only $LIST_1$). Because $LIST_1$ is most active for the first item in list 1, it shows the strongest match regardless of the output position. Furthermore, there are no components in the model which would allow for in-position matches for the other output positions, as the item vectors, which comprise the remainder of the elements in the context vectors, are

all orthogonal to each other. Other implementations of similarity of the list elements would likely produce the same qualitative patterns in the similarity gradients without additional assumptions. The fact that the $LIST$ elements are most active at the beginning of the list is a property of the context evolution within CRU, not our implementation of list element similarity.

One should note that the similarity gradients in Figure 5 are idealized in that they assume that all of the previous items from list 2 were perfectly recalled. In actuality, there are likely to be errors from the encoding stage and the memory retrieval stage, both of which can change the nature of the context cues employed in each output position. For this reason, we performed simulations of retrieval from list 2 (e.g., recall initiation with $LIST_2$) and plotted the positional uncertainty functions at each output position for list 1 and list 2. The positional uncertainty functions from list 1 are plotted in the same manner as Figure 4, demonstrating for each output position the proportion of recalls from each serial position in the prior list. For each list, we used a random set of six letters. Letters were not reused across each list. For each combination of parameter values, we simulated a total of 250 lists, using 500 simulations for each list to achieve stable predictions. Additional details on these simulations can be found in the Appendix.

Results of the simulations of the two list paradigm can be seen in Figure 6 for list 1 recalls (left column) and list 2 recalls (middle column), where recall is initiated with a $LIST_2$ context. The bar plots in the third column demonstrate the proportions of prior list intrusions (list 1 recalls) that were preceded by (left bar) or followed by (right bar) another recall from the same list. Figure 6 demonstrates the correct expected pattern for list 2 recalls – the positional uncertainty functions are peaked at the correct position for every output position. Nonetheless, the list 1 results depict very different gradients of prior list intrusions than what one would expect from Figure 5. The most striking pattern is for the high values of β (.65 and 1.0), a protrusion effect is observed under some conditions, specifically when $s_{list} = .75$ when $\beta = .65$, or for all values of $s_{list} > 0$ when $\beta = 1.0$. That

is, the most probable response from the prior list is the item that matches the current output position.

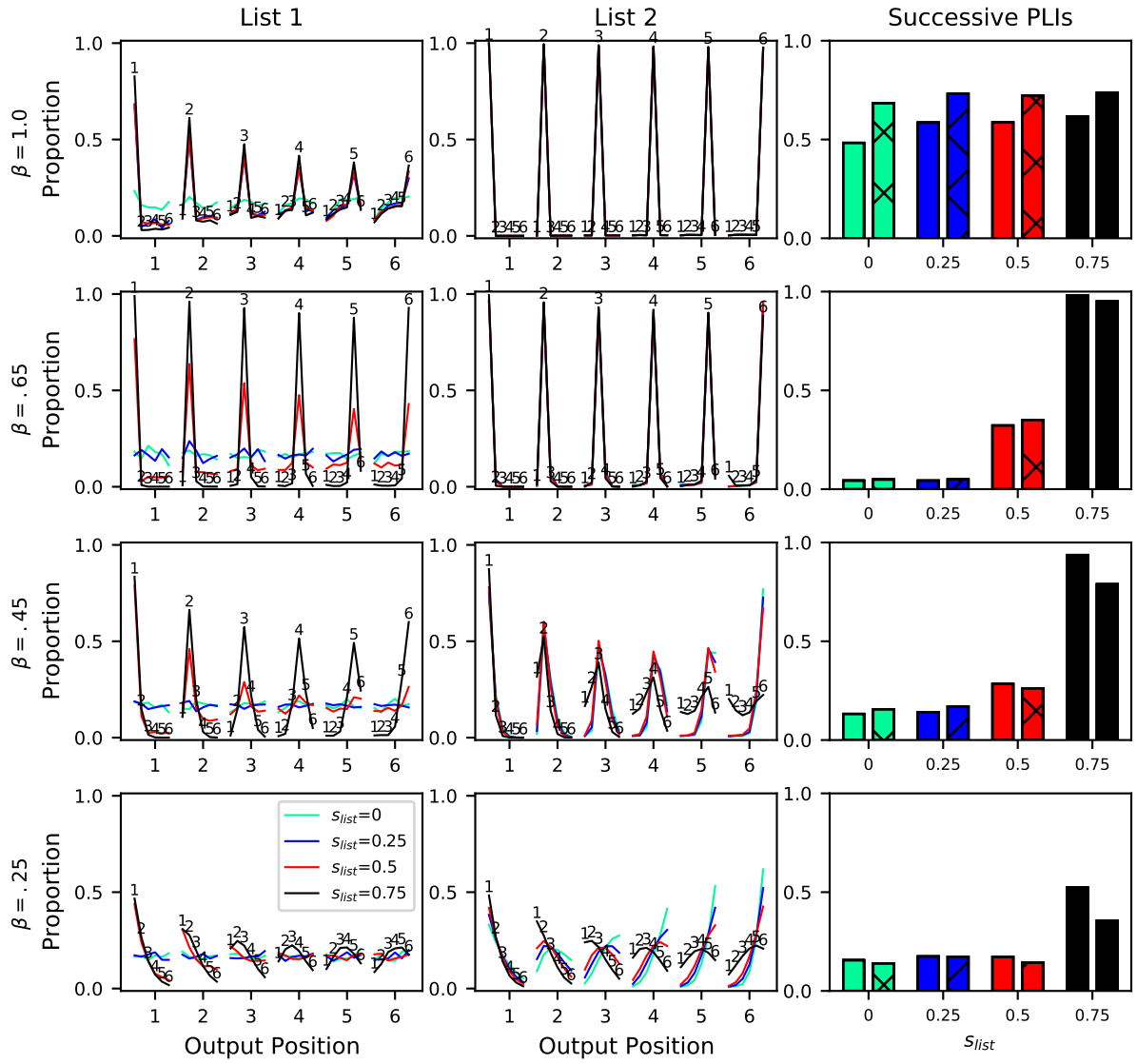

Figure 6. CRU simulations for a two list paradigm with attempted recall of the second list when similarity among the list context vectors (s_{list}) is manipulated. Depicted for each output position (indicated below the x-axis) are the proportions of recalls from each serial position in a given list (list 1 in the left column, list 2 in the middle column). The serial positions of the recalled items are indicated by the numbers above the lines. The right column shows the proportions of successive prior list intrusions (PLIs), with the left bar showing the proportions of intrusions that were preceded by intrusions while the right bar shows the proportions of intrusions that were followed by intrusions.

How can the model produce protrusion effects for each output position if there is only

a representation of the start-of-the-list? It's important to recognize that under these parameterizations the model can retrieve *the entire prior list in order*. To understand how this occurs, consider retrieval of the list 1 items ($ABCDEF$) when both s and β are high. The high value of s can lead to a high probability of initiating retrieval with the first item from list 1 (A), despite the list 2 context cue being employed. A high value of β leads to A dominating the context cue, which will exhibit a strong similarity to the next item from list 1 (B), which will produce a protrusion effect for the second output position. The proportions of successive intrusions in the right column of Figure 6 indicate that this is indeed the case - the parameterizations that produce protrusion effects show very high proportions of cases where list 1 intrusions are preceded by or followed by other intrusions from list 1.

Unfortunately, analyses of prior list intrusions do not suggest that the protrusion effect is due to retrieval of the entire prior list. Osth and Dennis (2015b) explicitly considered this possibility in their analyses and found that prior list intrusions were extremely unlikely to be followed or preceded by prior list intrusions. Specifically, for intrusions from the immediately preceding list that were studied in nonterminal positions, only 6% (list length = 5) and 6.8% (list length = 6) were followed by intrusions from the same preceding list. For intrusions that were studied in positions after the first position, only 2.6% (list length = 5) and 8% (list length = 6) were preceded by intrusions from the same preceding list. Such results suggest that retrieval of the entire preceding list is not a tenable explanation for the protrusion effect.

Several other parameterizations of CRU in Figure 6 do not produce a pattern that resembles the data. In fact, many of the protrusion gradients appear relatively flat, showing no strong tendency for prior list intrusions to be recalled in the same output position as their original list positions. While these analyses and simulations were performed with a restricted range of parameters, it is not obvious how other combinations of parameters could produce a protrusion effect that aligns with the data, especially

considering that the data do not suggest that retrieval of the entire prior list is likely.

CRU Simulations with Similarity Among the Item Vectors. We additionally allowed for similarity among all of the item vectors corresponding to the letters in the same manner as we explored similarity among the list context vectors. Namely, each item vector r was a weighted combination of a unique orthonormal vector u as well as a common vector component m:

$$r_i = (\sqrt{1 - s_{item}})u_i + \sqrt{s_{item}}m \tag{9}$$

As s_{item} approaches one, all of the item vectors become the common vector m.

The similarity gradients for when list 2 context cues are employed in each output position can be seen in Figure 7. Unlike the case when s_{list} was manipulated, changes in s_{item} produce substantial changes to the similarity gradients of the list 2 context vectors (right column of Figure 7). Specifically, increases in s_{item} increase the similarities of the context vectors for the incorrect items. This is because the common component m is present in all context vectors that are not the first item, producing a baseline degree of similarity between context vectors even if they are far apart on the list.

Inspection of the similarity gradients for the context vectors corresponding to the list 1 items (left column of Figure 7) reveals the opposite pattern of the effects of list context similarity. Specifically, when $s_{item} > 0$, there is a recency-focused tendency, where the similarity gradients peak at the final list 2 item for all output positions that are not the first item.

Why does similarity among the item vectors produce a recency pattern? The answer is due to the evolution of context vectors and the gradual decay of the $LIST_1$ context, which makes the common component of the item vectors m most active for the final list item. When the first item from list 2 is studied, $LIST_1$ is maximally active (1.0) and is the

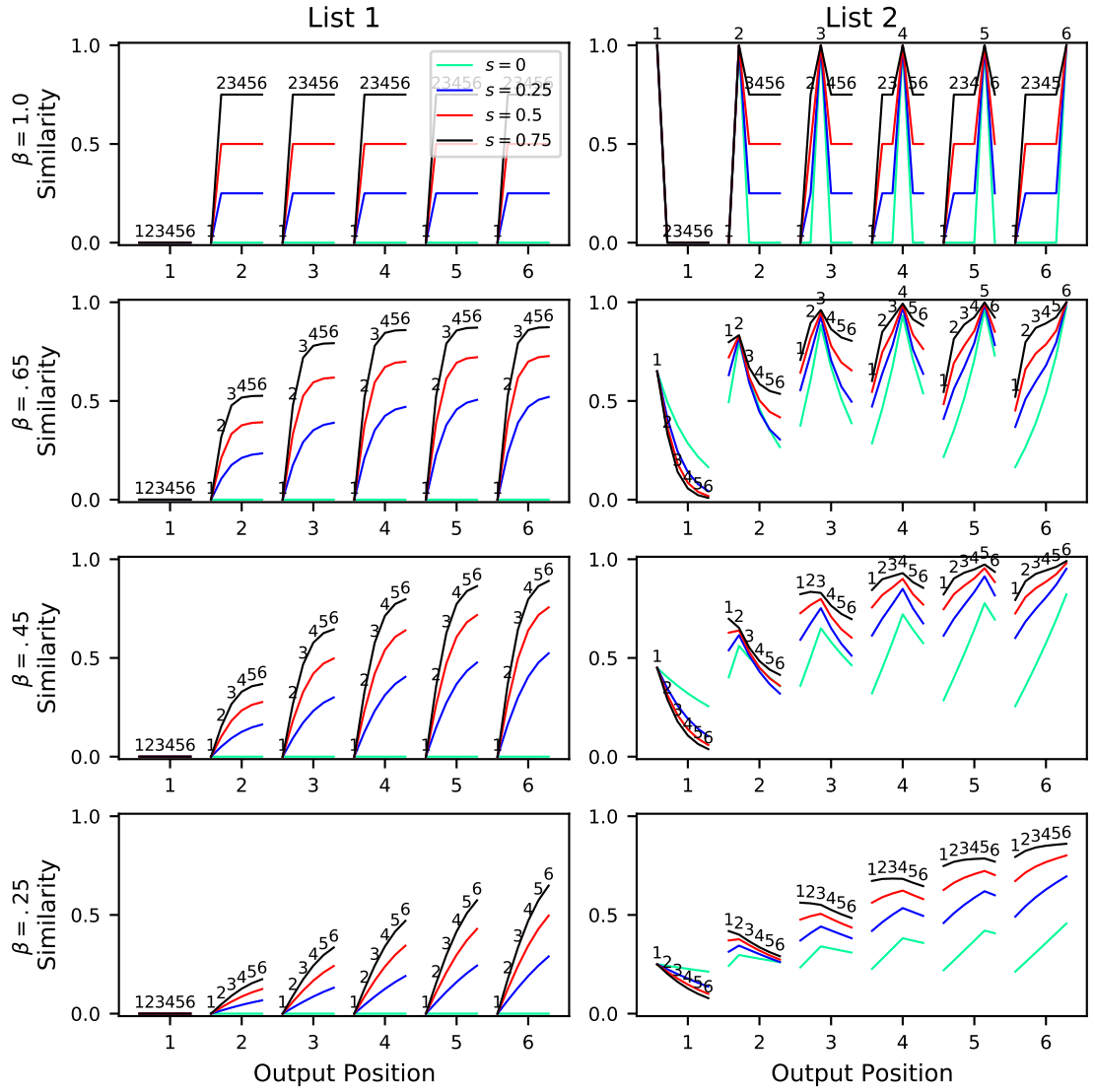

Figure 7. Dot products between the list 2 context cue for each output position (indicated below the x-axis) and the stored list 1 context vectors (left column) and list 2 context vectors (right column) when similarity among item vectors is manipulated (as indicated by the s_{item} parameter). The serial positions for each context vector are indicated by the numbers above the lines. Note that the context cue for each output position in list 2 assumes the previous items were correctly retrieved.

only component in its stored context vector. Since this element has no similarity to any elements in the list 2 context vectors, the first item from list 1 always exhibits zero similarity to the context cues from list 2. However, as each item is added into memory, $LIST_1$ decays while the common component m becomes more active in the context. This

makes it such that the common component has the greatest strength in the context vector for the final item from list 1, producing the highest similarity to context cues from list 2. Thus, in this way the common component approximates an end marker. This is interesting, as a common criticism of end markers in models such as the start-end model (Henson, 1998) is that it is unclear how the model could "know" when the end-of-list is occurring (although see Farrell & Lelièvre, 2009; Henson & Burgess, 1997).

CRU simulations with similarity among the item vectors can be seen in Figure 8. When $s_{item} > 0$ and $\beta \geq .45$, the model is able to produce protrusion effects for the final item. Unfortunately, the model does not appear able to show protrusion effects that generalize to other positions, with the final list 1 item showing the strongest tendency even for output positions 4 and 5. It's also noteworthy that increases in s_{item} come with a cost – recall of list 2 items is compromised for the late list items in several parameterizations. When $\beta \leq .45$ and $s \geq .25$, recall of the final item is compromised, as the positional uncertainty function for output position 6 is no longer peaked on the sixth item. In addition, the right column of Figure 8 indicates that increased item vector similarity also comes at the cost of a very high proportion of successive prior list intrusions.

CRU Simulations with Similarity Among the Item and Context Vectors. The fact that similarity among list context vectors produces a primacy bias in the prior list intrusions while similarity among the item vectors produces a recency bias begs the question – what happens if similarity among both types of vectors is included? After all, the $LIST$ element functions as a start-of-list marker, while the common component among the items can approximate an end marker. One of the most popular positional models is the start-end model of Henson (1998), which constructs a position code using the relative weight of start and end markers. The evolution of context in CRU suggests that it could mimic the start-end model under these conditions – the decay of the $LIST$ element along with the increase in strength of the common item component could approximate the two-dimensional position code in the start-end model, where the relative weights of the

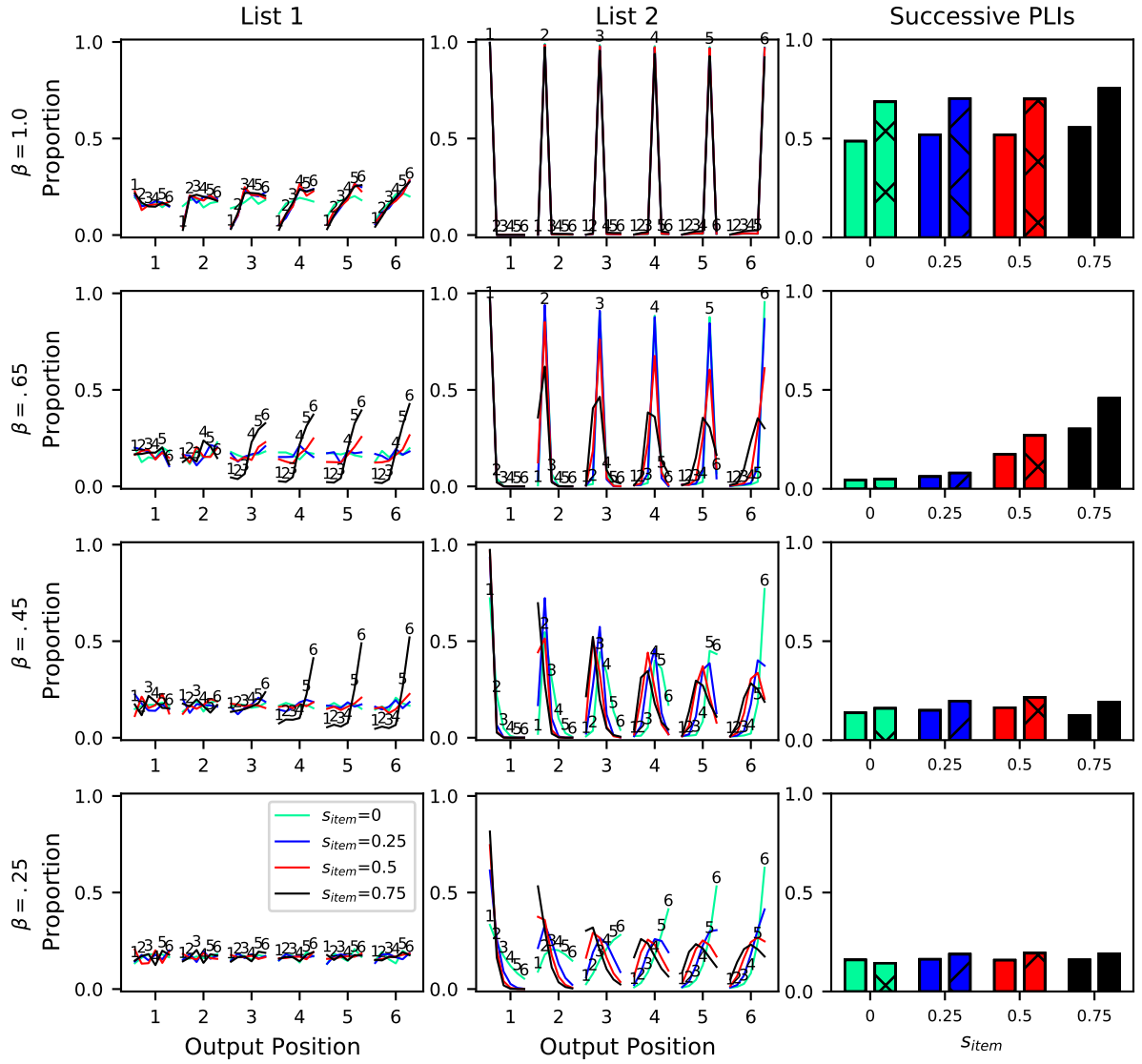

Figure 8. CRU simulations for a two list paradigm with attempted recall of the second list when similarity among the item vectors (s_{item}) is manipulated. Depicted for each output position (indicated below the x-axis) are the proportions of recalls from each serial position in a given list (list 1 in the left column, list 2 in the middle column). The serial positions of the recalled items are indicated by the numbers above the lines. The right column shows the proportions of successive prior list intrusions (PLIs), with the left bar showing the proportions of intrusions that were preceded by intrusions while the right bar shows the proportions of intrusions that were followed by intrusions.

start and end markers give an indication of the relative position of an item within a study list.

In order to simultaneously manipulate similarity among the list elements and item

vectors, we assumed separate common components for each of them to reflect the idea that list elements can be similar to other list elements, and item vectors can be similar to other item vectors, but the two classes do not exhibit any similarity to each other. A complete manipulation of β, s_{list}, and s_{item} results in a large number of figures. For this reason, the complete exploration can be found in Supplementary Materials B. In this section, we would like to highlight our explorations with $s_{list} = .75$, which yielded similarity gradients that most strongly resemble those from positional models.

The similarity gradients are depicted in Figure 9. What was quite interesting was that when $\beta < 1$, the model was able to produce similarity gradients for the list 1 items where the output positions peak on their respective serial positions, similar to what is found for the list 2 items. What was impressive was that this was found not just for the beginning and end items, but even for a portion of the midlist items (output positions 2 and 3). These results suggest that similarity among the list elements and item vectors may be able to approximate a position code within CRU.

How does CRU perform with both types of similarity when the two list paradigm is simulated? Simulation results can be seen in Figure 10. The model can indeed demonstrate the protrusion effect, but shares the limitation of prior simulations, namely that the parameterizations that produce protrusion effects also produce high levels of successive prior list intrusions. Other combinations of parameters found in Supplementary Materials B demonstrate a similar problem.

While the combinations of the two forms of similarity can approximate a position code, a crucial difference from positional models is the assumption that retrieved items are used as cues. When $\beta > .5$, the prior list intrusion has the largest weight in the context vector and can lead to a high probability of the next response being an intrusion from the same list. Thus, CRU differs from positional models because an erroneous retrieval from a prior list can increase the probability of another intrusion from the same list occurring.

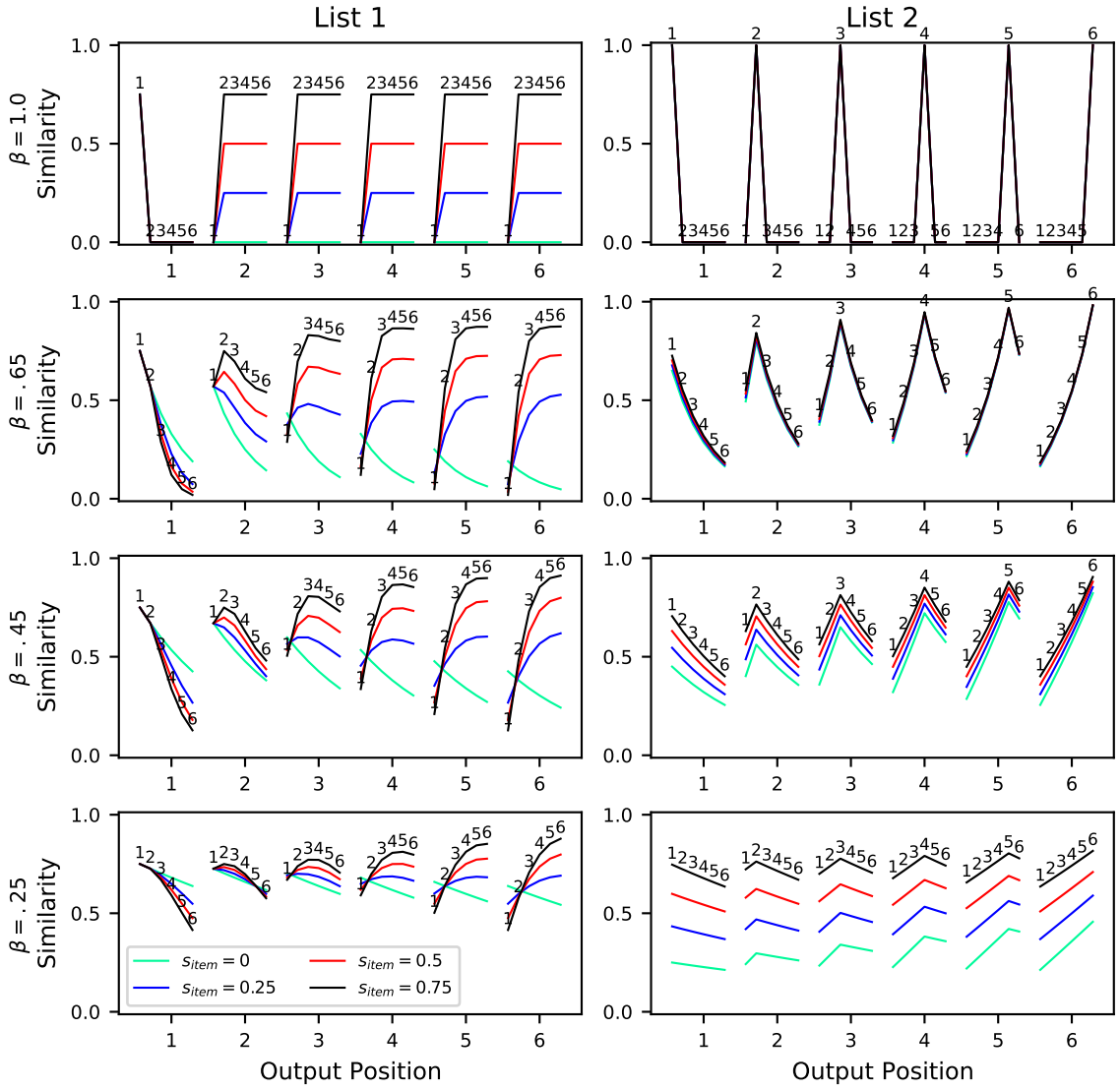

Figure 9. Dot products between the list 2 context cue for each output position (indicated below the x-axis) and the stored list 1 context vectors (left column) and list 2 context vectors (right column) when similarity among item vectors is manipulated (as indicated by the s_{item} parameter) and $s_{list} = .75$. The serial positions for each context vector are indicated by the numbers above the lines. Note that the context cue for each output position in list 2 assumes the previous items were correctly retrieved.

Discussion. In this section, we have explored three different ways to implement prior list intrusions within CRU, namely similarity among the list context vectors, similarity among item vectors, and similarity among both types of vectors. Similarity among the list contexts produces the strongest match between the first item from both lists

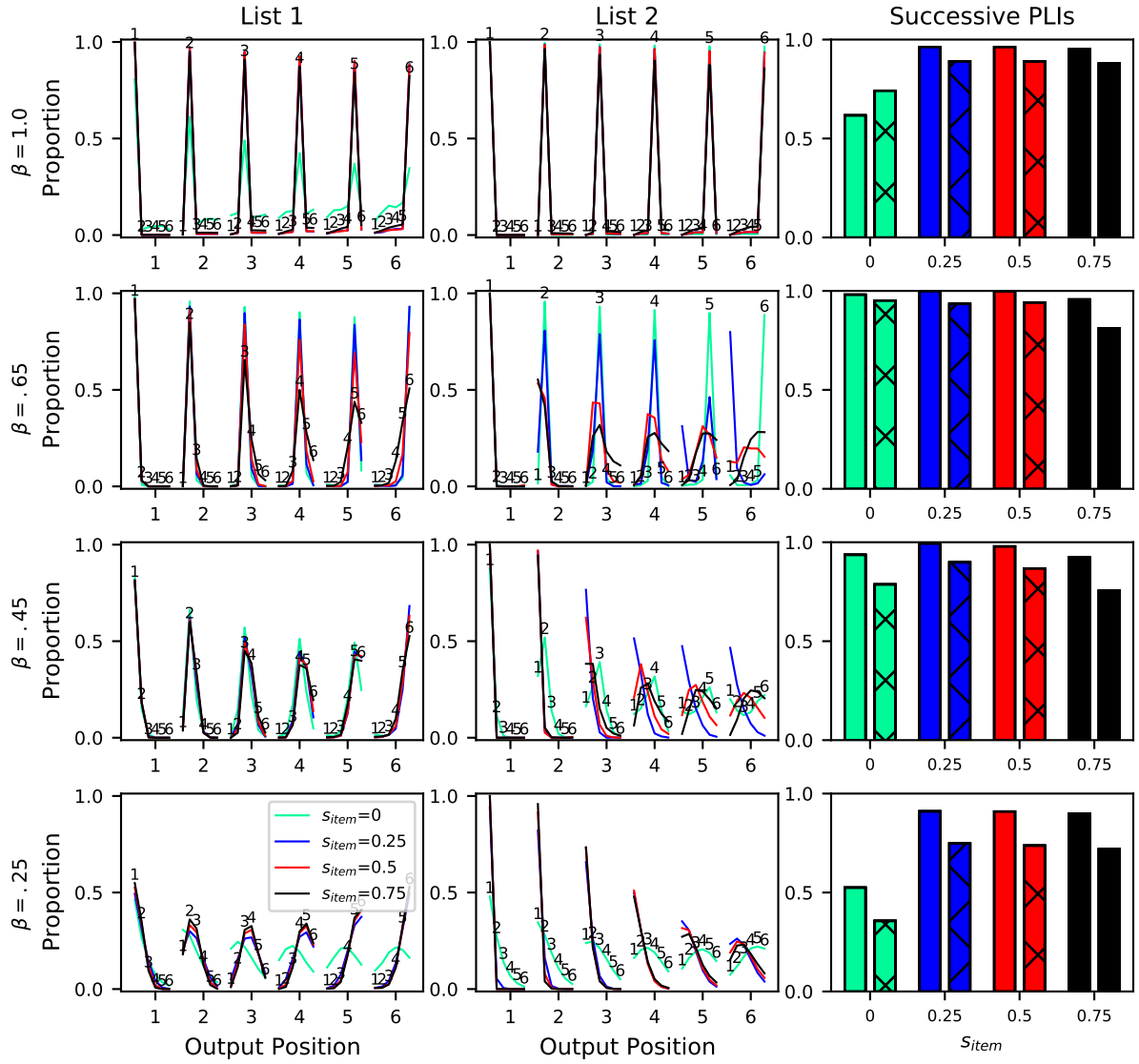

Figure 10. CRU simulations for a two list paradigm with attempted recall of the second list when similarity among the item vectors (s_{item}) is manipulated and $s_{list} = .75$. Depicted for each output position (indicated below the x-axis) are the proportions of recalls from each serial position in a given list (list 1 in the left column, list 2 in the middle column). The serial positions of the recalled items are indicated by the numbers above the lines. The right column shows the proportions of successive prior list intrusions (PLIs), with the left bar showing the proportions of intrusions that were preceded by intrusions while the right bar shows the proportions of intrusions that were followed by intrusions.

(item A in list 1 and item G in list 2). Similarity among the item vectors produces the strongest match between the final item from both lists as the common item vector elements are most active in the context at the end of the list, suggesting that this common

component can approximate an end-of-list marker. Simulations of the model indicated that it can produce protrusion effects, but for the wrong reasons – the parameterizations that demonstrated protrusion effects indicated a very high proportion of successive intrusions from list 1, suggesting that the entirety of list 1 was retrieved in order. The data instead indicate that prior list intrusions are rarely preceded or followed by other intrusions, making retrieval of the entire preceding list an extremely unlikely explanation for the protrusion effect (Osth & Dennis, 2015b).

We would like to explicitly acknowledge, however, that our CRU simulations explored only a limited range of model parameters. This is especially relevant given that similarity among both the item vectors and the list elements in Figure 9 demonstrated similarity gradients that qualitatively accord with the patterns predicted by positional models, where the peak of the similarity gradient at a given output position occurred in the same serial position on list 1. This occurred not just for the first and final item, but for some of the midlist items as well. This is likely due to the fact that the list element is essentially a start-of-list marker, while the common item component approximates an end-of-list marker. This qualitatively corresponds with the start and end markers in the start-end model (Henson, 1998), but avoids the problem of how to associate items to an end marker if the participant does not know the length of the list a priori.

While we cannot rule out the possibility that other combinations of parameter values might produce the protrusion effect, the model would likely be faced with a difficult "balancing act" between the roles of each of the parameters. For instance, similarity among the item vectors was found to be fairly detrimental to recalling the list 2 items due to the increased generalization across many of the context vectors. Another example is that high performance is associated with high values of β, but values of $\beta > .5$ make it such that the most recently retrieved item dominates the context cue at retrieval. If the previous response was a prior list intrusion, the next response is likely to be a prior list intrusion as well. Thus, it may be quite difficult for the model to produce the protrusion effect while

simultaneously keeping such intrusions relatively isolated during recall and capturing performance on the correct list, at least with the current assumptions and architecture of CRU.

Our argument is not that CRU is completely unable to capture this set of constraints, but rather that it would be far less challenging to capture this pattern if it either incorporated positional representations or approximated them to a greater extent. To date, the most successful account of the protrusion effect comes from the usage of item-position associations. Specifically, the start-end model of Henson (1998) was able to account for the protrusion effect in fits to group-level data by assuming that items are associated to both a general list context in addition to a representation of the item's within-list position. The general list context is distinct from CRU's $LIST$ nodes because it is the same for all of the items from a given study list, but does change between different lists. At retrieval, a joint list context and within-list position cue is employed. The predicted frequency of prior list intrusions from the model is rare due to only a minimal overlap between the two list context representations. When prior list intrusions occur, they tend to occur in-position due to the match of the position cues. SEM was able to achieve this after having been fit to group-level data while achieving a number of other benchmarks, including primacy and recency effects and the shapes of transposition gradients. While it would undoubtedly be more persuasive if the model was fit to individual participants and the responses from the individual trials, the protrusion effect follows so naturally from the usage of positional representations that it would be surprising if the model was not able to capture this phenomenon after such an endeavor.

While our introduction of list element and item vector similarity to CRU demonstrate that the model can indeed approximate position codes, a crucial difference is that in CRU retrieved items are used as cues while such an assumption is not made within positional models such as SEM. This makes it such that within CRU, a prior list intrusion is added into the context cue for the next retrieval, which increases the likelihood of a prior list

intrusion on the next recall. One way to further approximate the construction of position cues would be to only use relative activations of the $LIST$ vector and the common component of the item vectors in the context cue. That is, the cue for the beginning of the list would be the $LIST$ element alone, the cue for the final position could be the common item component alone, and the cues for each of the midlist positions could involve relative levels of both activations depending on the cued position in the list. Because retrieved items would not change the nature of the cues, this would prevent prior list intrusions from being accompanied by further prior list intrusions.

In his review of an earlier draft of this manuscript, Gordon Logan pointed out an additional possibility for how CRU could recover from prior list intrusions if there is detection of conflict between the given and intended response (Botvinick, Braver, Barch, Carter, & Cohen, 2001). After the conflict is detected, the appropriate cues for the next response in list 2 could be employed by the model, allowing the model to both recover from errors and produce the protrusion pattern. However, we find that it would be difficult for the model to recover from errors using this mechanism. First, consider if the model were to initiate with two erroneous retrievals from list 1 before the conflict was detected (items A and B). The appropriate context cue for the third item in the second list is $LIST_2 - G - H$. Thus, getting back on track would require an implicit retrieval of the first two items from list 2 to produce the appropriate cue. While generation of such cues is not impossible, it would introduce further challenges to the model, especially at later output positions.

The Costs and Benefits of Temporal Grouping Manipulations

Similar evidence for positional representations can be found in the errors of temporal grouping. Temporal grouping is when extended temporal pauses during list presentation are used to demarcate different groups of items, such as $A.B.C...D.E.F$, where . indicates a temporal pause and ... indicates an extended temporal pause that segments the list into two groups, ABC and DEF. Experiments on temporal grouping reveal that such

manipulations have both benefits and costs in performance. The benefits are enhanced recall of items in grouped lists compared to ungrouped lists, in addition to scalloped serial position curves, with primacy and recency effects found within individual groups as well as the list overall (Farrell & Lelièvre, 2009; Hartley et al., 2016; Ryan, 1969a, 1969b; Ng & Mayberry, 2002). The left panel of Figure 11 depicts this pattern using data from Hurlstone (2019), where in addition to a recency effect for the final item, grouped lists show elevated recall at positions 3 and 6.

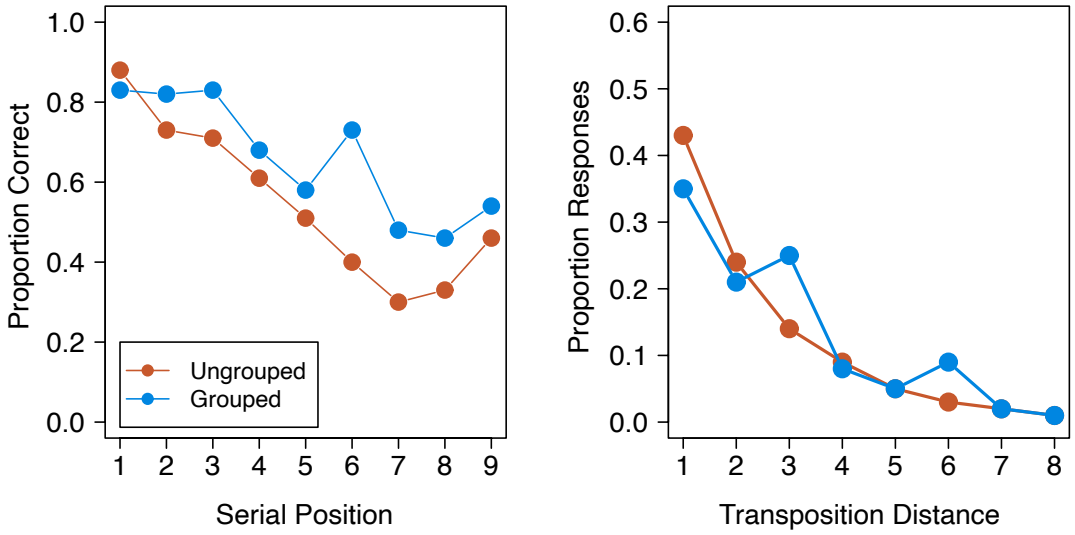

Figure 11. Temporal grouping effects for verbal serial recall of 9-item lists of digits: accuracy serial position curves (left panel) and transposition error gradients (right panel) for ungrouped lists and lists grouped in a 3–3–3 pattern. The peaks in the transposition gradients for grouped lists at distances 3 and 6 correspond to interposition errors. Data taken from Hurlstone (2019).

The costs of temporal grouping are an increase in long-range transpositions between different groups over ungrouped lists. Specifically, there is a tendency for participants to recall an item from the incorrect group that matches the within-group position that is attempting to be recalled. For instance, when attempting to recall the third item from the first group, an erroneous item from group 2 is most likely to be the third item from that group, even though the first item from group 2 is more temporally adjacent to the

previously recalled item. These errors are referred to as *interposition errors* (Hartley et al., 2016; Henson, 1999; Liu & Caplan, 2020; Ng & Mayberry, 2002) and suggest the usage of positional representations. The transposition gradient in the right panel of Figure 11 depicts this pattern, where the peaks in transpositions at distances 3 and 6 from the correct item reveal the interposition errors. Similar errors can be found in speech production, in which phoneme migration errors between syllables often respect within-syllable position without disrupting production of other phonemes in the syllable (Dell, 1986).

The interposition errors bear a strong resemblance to the protrusion effect described in the previous section. In fact, when intrusions from prior lists occur between temporally grouped lists, a similar interposition error can be observed (Ng & Mayberry, 2002). That is, if the intruded item intrudes into a different group than it was presented on the previous list, it is most likely that the intruded item will still share the same within-group serial position that it occupied in the previous list. Just as the protrusion effect is not due to the recall of the entire previous list, interposition errors are not due to the result of entire groups swapping with each other (Lee & Estes, 1981).

In what follows, we pursue two different ways of implementing temporal grouping in CRU to evaluate its ability to capture both the costs and benefits of temporal grouping in Figure 11.

Context Segmentation at the Between-Group Boundaries. The first model we pursued assumes that temporal grouping results in segmenting the contexts before and after the group boundaries, which can be implemented by increasing the β parameter at the group boundaries. This could be due to the fact that the temporal pause is poorly predicted and indicates an upcoming new event, which effectively segments the list into different contexts or episodes (e.g., DuBrow & Davachi, 2013; Polyn, Norman, & Kahana, 2009).

For each item that takes place after a group boundary, we increased the value of β by Δ_β, which is defined as:

$$\Delta_\beta = \beta_{group}(1.0 - \beta) \tag{10}$$

where β_{group} controls the extent to which β increases at the group boundaries. Figure 12 presents simulations of the Hurlstone (2019) paradigm, where the first column shows the serial position curves and columns 2-4 reveal the transposition gradients. These simulations were performed with three different values of β (.65, .45, and .25) crossed with three values of β_{group} (.95, .55, and .2). Additional details of the simulations can be found in the Appendix.

The serial position curves in the first panel reveal that this mechanism is capable of producing the improvement in performance for grouped lists and even can produce within-group recency effects for the first two groups (although not for the final group). This improvement in performance is due to the fact that the increase in β between groups decreases the similarity between the context vectors for the items from different groups. Larger values of β_{group} produce larger decreases in between-group similarity, which is why the effects of temporal grouping are larger for larger values of β_{group}.

While the model does an impressive job at capturing the benefits of temporal grouping, a difficulty with this CRU variant is that it is unable to reproduce the pattern in the transposition gradients seen in Figure 11. Inspection of columns 2-4 in Figure 12 reveals that the benefits to grouped lists come from a "tightening" of the transposition gradient, increasing the relative frequency of one-apart transpositions, but decreasing all others. There is no apparent increase in the frequency of three-apart or six-apart transpositions for any combination of parameter values, which corresponds to transposing an item from another group into the correct within-group position (interposition errors). This is because solely increasing the β parameter under conditions of temporal grouping decreases the similarity between context vectors in different groups, which consequently

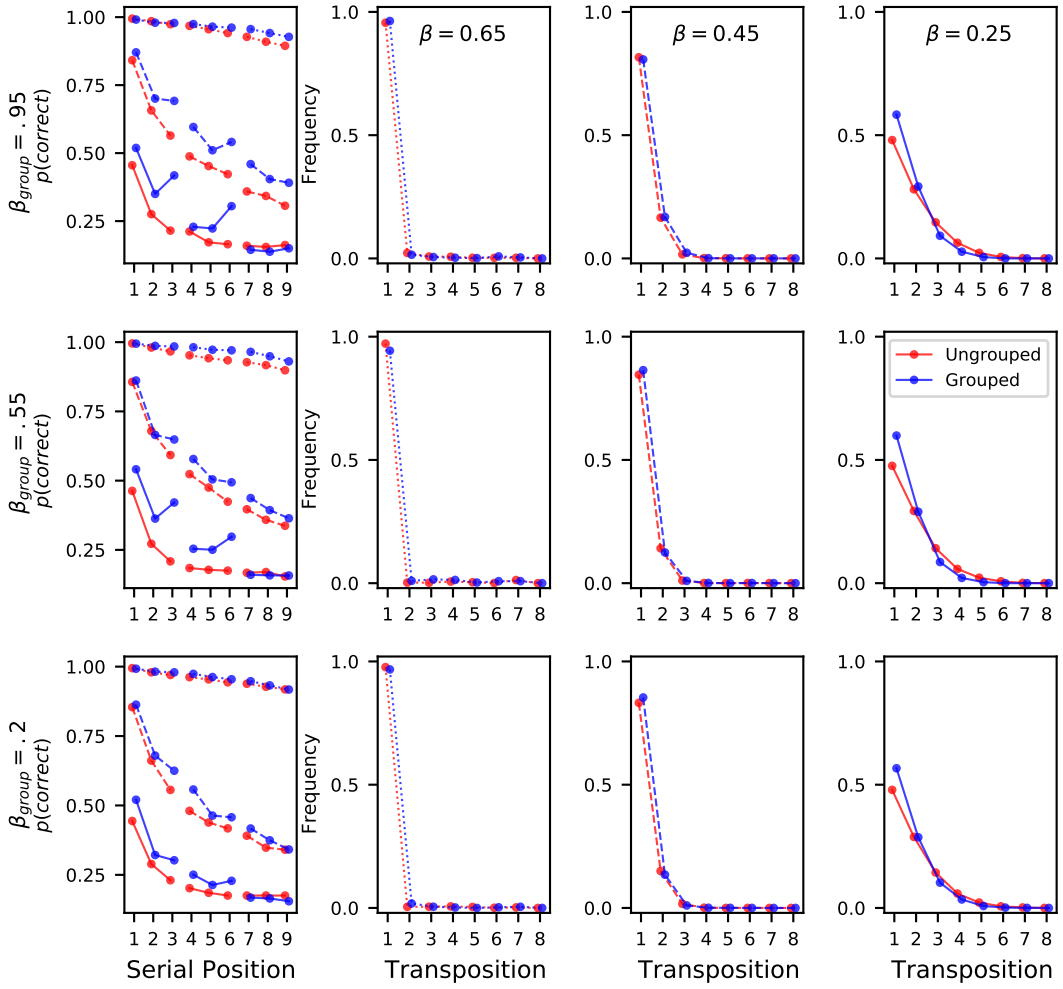

Figure 12. CRU simulations for grouped and ungrouped lists in the Hurlstone (2019) paradigm, where the β_{group} parameter serves to increase the value of β between groups. The first column shows the serial position curve while columns 2-4 show transposition gradients (separated for each value of β.)

decreases all cross-group transpositions.

A clearer illustration of why context segmentation can produce the benefits but not the costs of temporal grouping can be seen from an analysis of the similarities between the context vectors. Figure 13 shows the dot products between all possible context vectors for grouped and ungrouped lists with three different values of β (.25, .45, and .65). For the grouped lists, we used $\beta_{group} = .55$. One can see that the context segmentation serves to considerably decrease the similarities between contexts of different groups, which results in

considerably less competition in grouped lists than ungrouped lists. However, there is not even a single hint of an increase in similarity between items from different groups that share the same within-group position. Context segmentation *on its own* can only serve to push contexts from different groups apart. Thus, while we admit that we have only explored the predictions of this CRU variant with a limited range of model parameters, it is not at all obvious how different parameterizations of this CRU variant could serve to heighten the similarities for items from different groups that share the same within-group position without additional assumptions being introduced into the model.

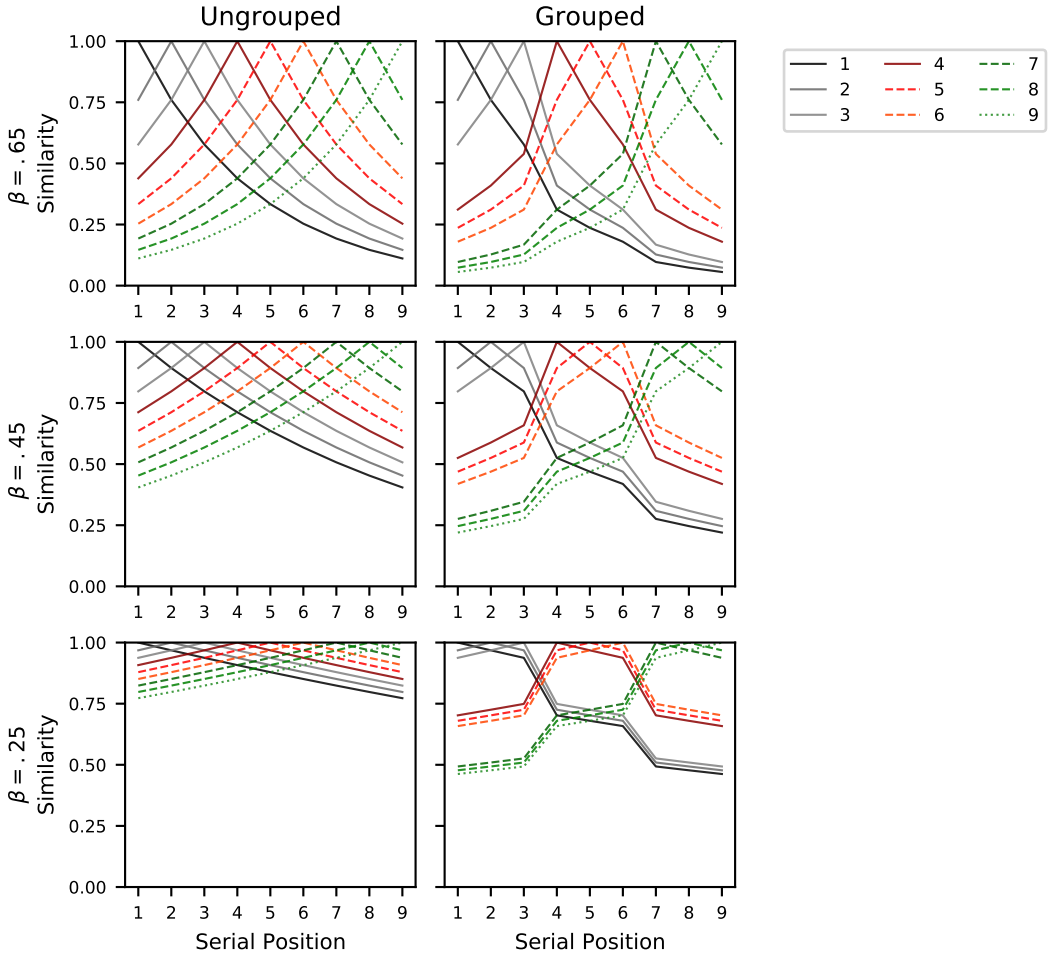

Figure 13. Dot products between all context vectors for ungrouped (left column) and grouped (right column) lists with three different values of β using context segmentation between groups ($\beta_{group} = .55$).

Usage of Group Markers. Positional models tend to account for the effects of temporal grouping by using explicit group markers (e.g., Brown et al., 2000; Farrell, 2012; Henson, 1998). That is, items are associated to their within-group positions, and groups are associated to their within-list positions. Interposition errors occur due to the similarity in within-group position representations from items of different groups, while overall advantages of temporal grouping occur due to the decrease in between-group similarity for items that do not share the same within-group positions.

We attempted a similar approach within CRU. Specifically, in grouped lists we assumed that each group is preceded by a marker that indicates the particular group, such that the list ABCDEFGHI is learned as
$LIST - GROUP_1 - A - B - C - GROUP_2 - D - E - F - GROUP_3 - G - H - I$, where $GROUP_1$, $GROUP_2$, and $GROUP_3$ are treated as item vectors. At retrieval, the group markers can be retrieved, but do not produce responses. Instead, the group markers enter the context representation and can be used to further cue retrievals. A key distinction between this model and previous approaches is that we did not employ within-group position markers.

The vectors for the group markers were orthogonal to all other vector representations from the model, including the vectors corresponding to the letters, spacebar, and list context. In these simulations, elements 1-26 corresponded to the letters, element 27 was the spacebar, elements 28-30 corresponded to the group markers, and element 31 was the list context. However, we did manipulate the similarity of the group marker vectors to each other.

To explore the consequences of similarity, we adopted a similar approach as previous simulations with vector similarity and employed overlapping elements between each group marker. However, we departed from that representation of similarity because with three group markers, it is desirable to have adjacent group markers be more similar to each other than distant group markers, as between-group transpositions are more frequent for

neighboring than distant groups. For each group marker, its own element took a value of 1 (group 1: element 28, group 2: element 29, group 3: element 30). For group 1, elements 29 and 30 represented group lags of 1 and 2. For group 2, elements 28 and 30 represented lags of -1 and 1. For group 3, elements 29 and 30 represented lags of -1 and -2. Element l of each lag was generated from an exponential distribution with decay rate δ:

$$l = exp(-\delta|lag|) \tag{11}$$

Subsequently, each group marker vector was normalized to be of length 1. This formulation made it such that the dot products for adjacent group markers was higher than for distant group markers, and the between-group similarity decreased with increases in the δ parameter.

Simulations of the model with group markers with three different values of β (.65, .45, and .25) and three different similarity values ($\delta = .5$, $\delta = 1.25$, and orthogonal group marker vectors) can be seen in Figure 14. When $\delta = .25$, the dot product between adjacent group markers was .971 while the dot product between distant group markers was .921. When $\delta = 1.25$, the dot product between adjacent group markers was .529 while the dot product between distant group markers was .226.

CRU's predictions with group markers were much less consistent across parameter combinations than in the previous simulations. First, the benefits of temporal grouping were apparent when $\beta = .25$ but not for the higher values of β. Second, the costs of temporal grouping, as reflected in a higher incidence of three-apart (but not six-apart) transpositions, was found when group similarity was high ($\delta = .25$) and when β was .65, but in no other parameter combinations.

Why did group markers hurt performance, and why were interposition errors found with high values of β? To address this question, we adopted Logan (2021)'s approach of

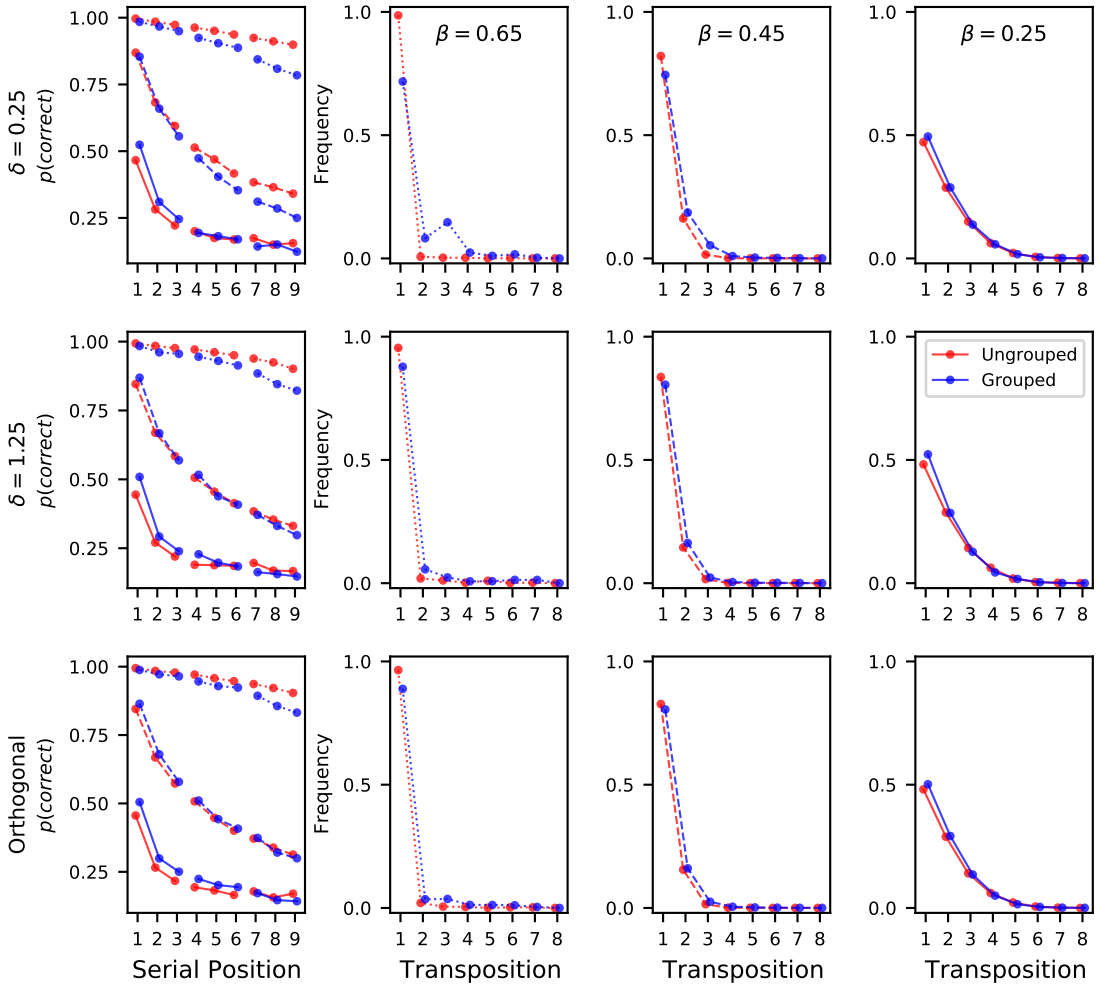

Figure 14. CRU simulations for grouped and ungrouped lists in the Hurlstone (2019) paradigm in which group markers are present in the context vectors. The first column shows the serial position curves ($\beta = .25$: solid lines, $\beta = .45$: dashed lines, $\beta = .65$: dotted lines) while columns 2-4 show transposition gradients (separated for each value of β.)

plotting the pairwise similarities between all context vectors. Figure 15 depicts these similarities for grouped and ungrouped lists with high values of β, but restricted to the case where there is high group similarity ($\delta = .5$). What is initially counterintuitive about the similarities between the context vectors is that each group marker's context vectors are relatively dissimilar to each other, despite the high similarity between their respective item vectors. However, our simulations of item vector similarity in Supplementary Materials A demonstrate that increases in item vector similarity increase the context vector similarity

for items that *follow* the similar items. This is again a consequence of the assumption that an item vector is not present in a given item's own stored context representation. For instance, group 1's context vector just comprises $LIST$, whereas group 2's context vector comprises $LIST - GROUP1 - A - B - C$. While we have manipulated vector similarity in a different fashion than the implementations of phonological similarity, it is ultimately the core assumptions of CRU that produce these consequences.

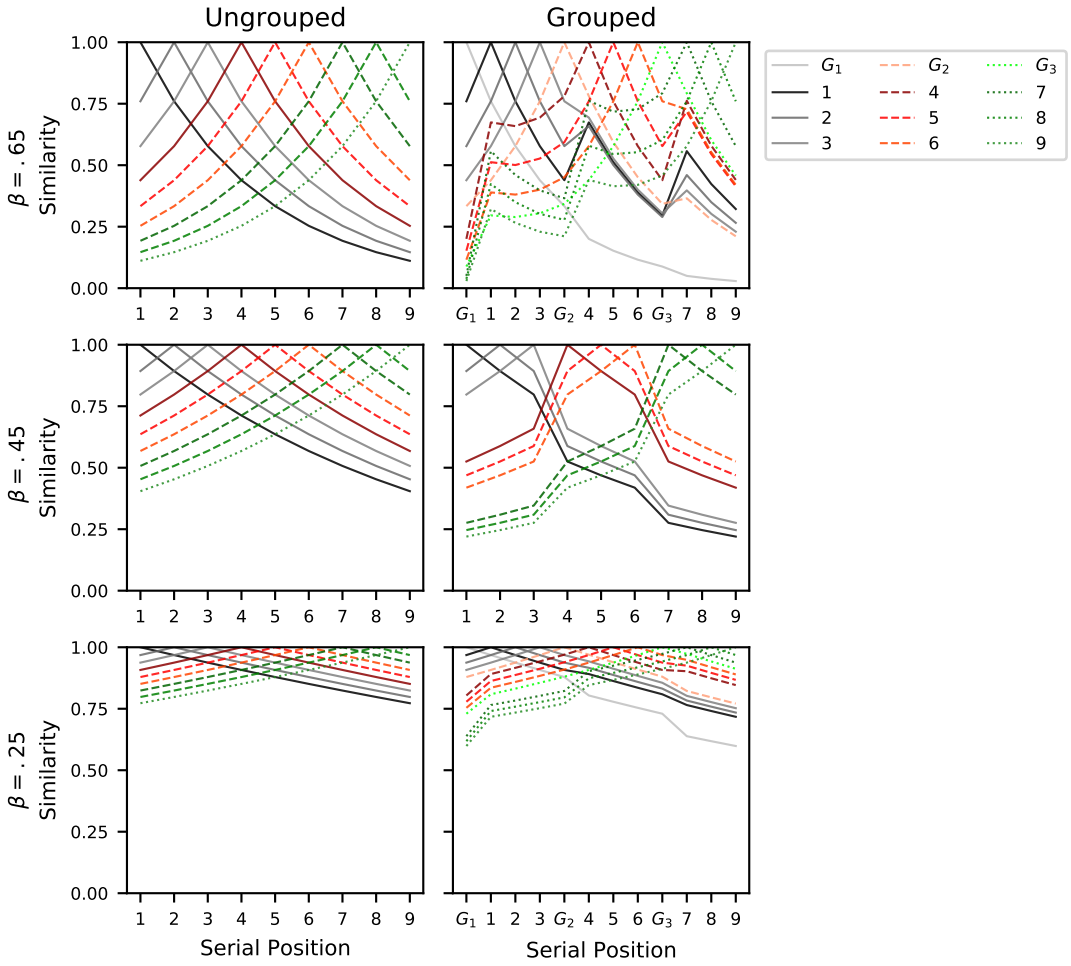

Figure 15. Dot products between all context vectors for ungrouped (left column) and grouped (right column) lists with three different values of β using group markers with high similarity ($\delta = .25$).

Consequently, the high similarity among the group markers does not increase the similarity of their context vectors. However, it does increase the similarity of the context vectors corresponding to the first item within each group, which are all preceded by the

highly similar group markers. This is most evident in Figure 15 when $\beta = .65$ – there is a visibly elevated similarity between the first item's context vector and the similarity of both the fourth and seventh item's context vectors. The reason why this is most evident when $\beta = .65$ is because higher values of β place considerably heavier emphasis on the most recent entry in the context vector, which happens to be the group marker for the first item in each group.

However, the elevated similarity for items that share the first within-group position appears to be restricted to the first member of each group, implying that interposition errors should be the most dominant in the first within-group position. Aside from an investigation by Henson (1999) who found high rates of interposition errors in the final within-group positions, we are not aware of any other analyses that have evaluated whether interposition errors vary by within-group serial position. For this reason, we re-analyzed the data from Hurlstone (2019) and Hartley et al. (2016) – interposition errors for grouped and ungrouped lists can be found in the left and right panels of Figure 16, respectively. These results reveal that interposition errors are not restricted to the first position, nor do they occur most frequently in the first position, but are instead common to virtually all within-group positions, as would be predicted by positional models.

The similarities depicted in Figure 15 also clarify why grouping can hurt memory. When β is high, one can see that the similarity between items in different groups is considerably higher than within grouped lists, even if they do not share the same within-group serial positions. This considerably harms the discriminability between items in different positions.

One concern with the present simulations is that our results may be due to the particular ways in which we implemented similarity among the group markers. In Supplementary Materials C, we explore additional simulations using vectors comprised of weighted orthogonal and common components. These simulations showed very similar patterns to what we depicted here – elevated similarity could be found between the first

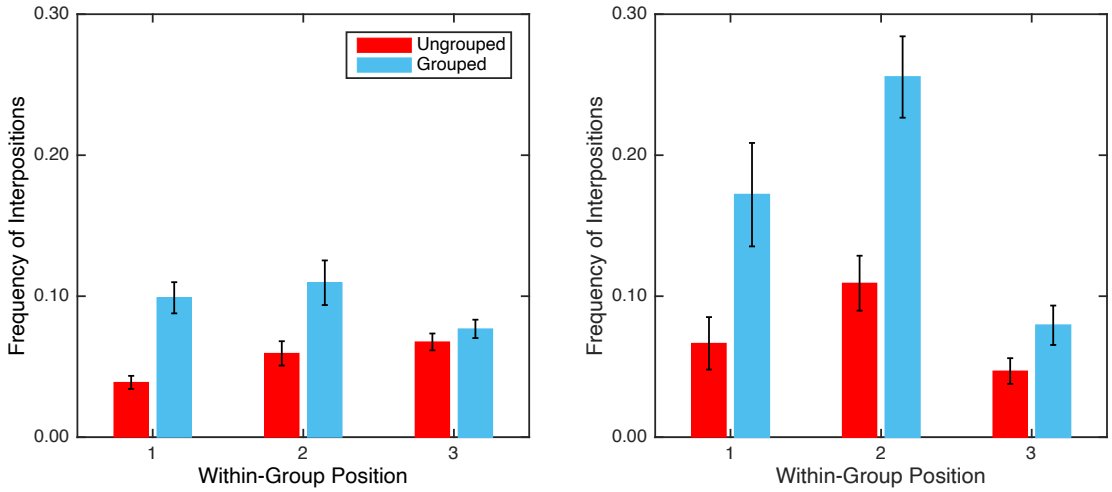

Figure 16. Interposition errors in ungrouped and grouped lists for the data from Hurlstone (2019, left panel) and Experiment 1 of Hartley et al. (2016, right panel). The data for grouped lists in the right panel are taken from the predictable grouping condition of the latter study. Error bars represent the standard error of the mean.

members of each group, but were not evident otherwise. This pattern is likely the product of CRU's architecture, as the similar group markers, which increase the similarity between different groups, are most active for the first presentation of each group. We suspect that additional assumptions would be required to capture the interposition errors for each within-group position and also capture the benefits of temporal grouping.

Discussion. Using two different mechanisms for capturing the effects of temporal grouping, we were unable to simultaneously reproduce the two qualitative phenomena of interest. Increases in the β parameter between groups does an impressive job of improving performance for grouped lists, including the primacy and recency effects within each group, but this mechanism does not capture the interposition pattern because it decreases between-group similarity for all within-group serial positions. Including group markers in the context representations can increase the similarity between members of the first within-group position in some circumstances, but it additionally can hurt performance in grouped lists due to higher overall between-group similarities.

An important caveat is that we have performed these simulations with a limited

range of parameters. However, it is unclear how exactly other parameterizations of the model could address these concerns. For the first CRU variant where β is increased at the group boundaries, there is no obvious mechanism within CRU for producing high similarity between context vectors across group boundaries that share the same within-list position, as increases in β only serve to decrease the similarity between context vectors.

For the CRU variant that uses group markers, it is not obvious how other parameterizations could produce interposition errors at all output positions, as group markers are most strongly associated with the first item from each group and only produce an apparent similarity increase for the first items from remote groups. It is unclear to us how other parameterizations or implementations of the vectors corresponding to the group markers could produce elevated cross-group similarity for the same within-group positions for the second and third positions without additional assumptions.

Our argument here is similar to our argument about the challenges CRU faces with the protrusion effect – that it would be far easier to address the constraints from temporal grouping manipulations if the model incorporated positional representations or approximated them in some fashion. Henson's SEM addresses temporal grouping by assuming hierarchical positional representations, where items are associated to their within-group positions, and groups are associated to their respective list positions. At retrieval, both the group and position cues are used – items are most likely to be retrieved if they match those cues. The benefits of temporal grouping occur because the group cue serves to isolate retrieval to the items located within the group, producing less competition from items from other groups. However, the costs of temporal grouping occur due to the re-usage of within-group position markers across different groups, making it such that long-range transpositions from items in other groups that share the same within-group position become more likely. Ungrouped lists, in contrast, instead use separate within-list position representations for each item on the list.

Positional models such as SEM have even been successful after having been fit to

data from grouped and ungrouped lists. While Henson (1998) fit the model to group-level data, later fits of the model have been successful even after having been applied to individual participant data (Farrell & Lelièvre, 2009; Hurlstone, 2019), producing both the costs (interposition errors) and benefits of temporal grouping (enhanced primacy and recency within groups), while simultaneously being able to address other aspects of the data, including the general primacy and recency effects from ungrouped lists and the shapes of transposition gradients. Likewise, Liu and Caplan (2020) demonstrated successful fits to individual participant data from grouped and ungrouped lists using the SIMPLE model (Brown et al., 2007), which similarly incorporates positional representations to capture the effects of temporal grouping. These fits demonstrated a successful account of both the benefits and costs of temporal grouping in addition to capturing the other trends in the data.

However, one limitation of SEM, and other positional models adopting similar solutions to hierarchical representations (e.g., Brown et al., 2007; Burgess & Hitch, 1999; Farrell, 2012), is they do not specify how the grouping structure is detected and the multilevel positional representations are generated. Instead, hierarchical representations are specified by hand a priori by the modeller. Models such as CRU avoid this problem because the necessary structure (the list items) is specified by the experimental design.

However, it is not the case that all positional models require an ad hoc specification of hierarchical structure. An example of a positional model that has a self-organizing hierarchical structure is the BUMP model (Hartley et al., 2016). In BUMP, hierarchical representations are generated "on-the-fly" by a bottom-up driven timing signal based on a large population of oscillators with frequency tunings spanning the range of presentation rates encountered in a serial recall task. When a grouped list is presented, oscillators with tunings close to the group presentation rate will be recruited that parse the list into groups and track their positions within the list, whereas oscillators with tunings close to the item presentation rate will be recruited that track the positions of items within groups. We note

BUMP bears a family resemblance to the OSCAR model (Brown et al., 2000) mentioned earlier, except OSCAR uses a top-down driven timing signal, rather like an internal clock. The oscillators are therefore ballistic and insensitive to the rhythm and timing of the stimulus input.

Hartley et al. (2016) report experiments showing a bottom-up mechanism is needed to accommodate data showing gross variation in recall performance as a function of different types of grouping patterns. In addition, a major advantage of the BUMP model is that it can accommodate data showing effects of temporal grouping even when the temporal grouping pattern is unpredictable.

Is There Evidence for Associations Between Items?

We have discussed three challenges to CRU's principal reliance on item-dependent context representations. The first hurdle, phonological similarity effects in pure and mixed lists, has been considered to be so challenging for chaining models that they have effectively been "ruled out" by the error patterns (Henson et al., 1996). However, this claim has been overstated, and may only be restricted to models that rely on response cuing, where the responses from the model become the cue for the next response. CRU instead relies on memorial cuing - what is retrieved, and not what is responded with, enters the cue for the next response. Likewise, we found that the "sawtooth" error pattern in the serial position curves in mixed lists could be captured when phonological confusions occur in the output stage. Under these circumstances, a retrieved phonologically confusable item may result in the erroneous output of a different confusable item. Critically, it is the retrieved item that enters the context cue and influences the next response, which prevents the erroneously output item from influencing the next retrieval. These simulations indicate that the "sawtooth" error pattern is not a problem for CRU, and likely wouldn't challenge other models that rely on inter-item associations that utilize a similar output stage. An important contribution of our simulations, therefore, has been to show that the mixed-list

phonological similarity effect offers far less theoretical leverage than was hitherto thought to be the case.

However, the protrusion effect as well as the costs and benefits of temporal grouping are much more challenging for the CRU model as it has been defined in the Logan (2021) article. Both phenomena suggest the existence of positional representations. Our simulations of the protrusion effect indicated that under some parameterizations, the CRU model could capture the protrusion effect, but the high proportion of successive prior list intrusions suggest that the model accommodated this pattern by retrieving the entire prior list. The data instead suggest that is an extremely unlikely explanation for the protrusion effect due to prior list intrusions being unlikely to be followed or preceded by other intrusions from the same list (Osth & Dennis, 2015b).

Our simulations of the effects of temporal grouping found that we were not able to reproduce both the benefits and costs of temporal grouping seen in the data. Specifically, we implemented two variants – one in which temporal grouping has the effect of separating the contexts between the group boundaries, as well as another where explicit group markers are learned. The first variant was successful in capturing the benefits of temporal grouping, but was unsuccessful in capturing the costs. The second mechanism had some success in capturing the interposition errors, but for the first position only and not for the second and third within-group positions. This variant also struggled with its ability to capture the benefits of temporal grouping.

We want to explicitly acknowledge that these explorations were derived from simulations with a limited range of model parameters and implementations of vector similarity. These are much weaker tests of model mechanisms than the fits to individual responses performed in the original Logan (2021) article, which is ultimately the direction the field should be headed in. While we cannot rule out the possibility that other parameters and implementations could produce the correct pattern of results, in several cases it is not obvious how the model could reproduce these phenomena while

simultaneously capturing other benchmarks and qualitative patterns of interest. In several of these cases there are core assumptions of the CRU architecture that would likely cause similar problems for other parameterizations or implementations of vector similarity.

We do not mean to suggest that CRU cannot be revised to address these challenges. Rather, we argue that these challenges point to an *insufficiency* of its complete reliance on item-dependent context representations and the manner in which they are encoded and retrieved within its architecture. Specifically, we argue that overcoming these challenges will be far easier if the model either includes representations of within-list position, or alternatively approximates positional representations using its existing representations. While these solutions may encounter challenges when the model is fit to data, they show a more principled relationship to the problems at hand than the mechanisms within the current architecture.

As mentioned previously, the error patterns we have described have led many theorists to completely eschew associations between items as a viable representation for serial order. Does this imply that there is no positive evidence for associations between items, and that CRU should abandon them entirely? Not exactly.

Strong evidence for associations between items has been found using the *spin-list* paradigm (Kahana et al., 2010; Lindsey & Logan, 2019, 2021). In this paradigm, participants initially study a list of items such as $ABCDEFG$. In the spin-list condition, on subsequent trials participants study a rotated, or "spun" list of the same items, such as $DEFGABC$, whereas in the control-list condition participants study the same list without variation in its starting position. Results show that participants are able to learn both spin and control lists, with the rate of learning of spin lists being only slightly slower than for control lists. Such spin-list learning can be most plausibly attributed to the usage of inter-item associations. This is because the relationships between items are mostly preserved from trial-to-trial in the spin list condition, whereas the item-position associations are completely confounded. Positional models do not possess a natural

explanation of these findings – in order for such models to yield improvements in spin list conditions, they require some mechanism for recognizing the permutation of the list and correctly assigning position representations to the items. Spin-list improvements have also been found in typing tasks (Lindsey & Logan, 2019).

Additional evidence comes from consideration of the sequential history of the studied items. Botvinick and Bylsma (2005) trained participants on an artificial grammar before performing serial recall. Performance was not only better for lists of items that conformed to the structure of the grammar, but participants often biased their recalled responses in favor of the artificial grammar. That is, if participants were trained on a pair of items such as AB, a pair of items on a study list such as AC is likely to be erroneously recalled as AB. These findings resemble other findings in the literature showing performance improvements when lists contain high-frequency bigrams in serial recall (Baddeley et al., 1965) and in typing (Behmer & Crump, 2017). While such findings can be explained by positional models by appealing to processes such as redintegration (e.g., Lewandowsky & Farrell, 2000), such an explanation does not fall naturally from positional representations. Instead, these findings can be most easily explained by assuming that inter-item associations formed from the training influence recall of the current list. Other evidence for inter-item associations comes from the finding that adjacent items on a serial list show improved performance in paired-associate recall relative to distant pairs (Crowder, 1968) as well as the finding that participants show improved performance in reconstruction of order and serial recall when presented with a partial set of cues in their correct order (Basden et al., 2002; Serra & Nairne, 2000).

The fact that there is evidence supporting both inter-item and position-item associations might suggest that a fruitful direction would be to incorporate positional associations into CRU. While positional representations and associations between items are often portrayed in the literature as polar opposites, this is a false dichotomy. There is nothing in CRU's architecture that prevents the incorporation of position markers into its

context representations. Indeed, the original version of the Burgess and Hitch (1992) model contained both item-position and inter-item associations (albeit adjacent and not remote associations), but fits to data suggested such a small reliance on pairwise associations that subsequent versions of the model discarded the inter-item associations entirely (Burgess & Hitch, 1999, 2006).

It would be interesting to see a similar parameter estimation of the relative weighting of position and item associations within CRU, as CRU's item-dependent context representations would likely fare much better than the pairwise associations in the Burgess and Hitch (1992) model due to CRU's formation of remote associations between items at study along with its reliance on compound cues at retrieval. As mentioned previously, pairwise associations often fare quite poorly in models of serial order, as the production of an error is often extremely damaging for retrieval of the rest of the list, while the remote associations in allow the cues prior to the error to influence the next retrieval. In addition, several of Logan (2021)'s methods, including fitting to each response at the individual participant level, are also likely to yield different conclusions than those of Burgess and Hitch (1992). Such a model may also be useful for evaluating Young (1968)'s proposal that the relative weight of item-position and inter-item associations may depend on certain experimental conditions, as fits of such a model may yield different relative weights for different experimental paradigms.

However, a long-standing theoretical challenge for positional models is that it is unclear how positional representations are generated and reinstated at retrieval, which would likewise be shared by a variant of CRU that incorporates position markers. Despite decades of work employing positional models, and some progress on this issue (Brown et al., 2000; Hartley et al., 2016), there still has not been a satisfying solution. An attractive feature of item-dependent context models such as CRU is that the key requirement to explaining remote associations is the maintenance of previously experienced items in order to bind them to the current item, a property which can be attained via recurrent

connections between an item layer and a context layer (Elman, 1990; Howard & Kahana, 2002).

An ideal direction for both CRU and the field may be to consider how associations between items can be used to build or approximate positional representations. Such an approach may be able to account for evidence of both representations without inheriting the theoretical limitation of positional models. In our simulations of the protrusion effect, we found that incorporation of both list element and item vector similarity can partially approximate a position code similar to that of the start-end model. CRU's $LIST$ elements already function as a start-of-list marker. If similarity between item vectors is introduced, the similar elements of the item vectors will be most active in the context layer when the $LIST$ element has decayed, which will tend to be the end-of-the-list.

Thus, similar item elements can approximate end markers without inheriting their limitations. For instance, in the start-end model, participants bind items to an end-of-list marker, and the activation of this marker grows as the list progresses. However, it has never been clear how participants generate such an expectation if they do not know the length of the study list (but see Henson & Burgess, 1997). Within CRU, the activation of the similar item elements grow naturally as a consequence of its context evolution and the normalization of the context vectors as the $LIST$ element declines in strength as the list progresses. Nonetheless, a crucial distinction of CRU from SEM is that in CRU, retrieved items are used as cues, and our simulations of the protrusion effect demonstrated that the model struggled with recovering from prior list intrusions. A more natural analog of position cuing would be to use the relative weights of the $LIST$ element and the common item elements as cues without updating them with retrieved items. These common item elements could simply reflect what is common to each of the experimental items, such as their stimulus type (e.g., letters or words). Future work would likely be required to evaluate such a mechanism.

Other models and architectures have similarly exhibited some success with

approximation of position cues with item representations. Botvinick and Plaut (2006) achieved this using the recurrent neural network model of Elman (1990). The recurrent neural network architecture is very similar to that of the temporal context model (Howard & Kahana, 2002), in that there are connections between an item layer and a context layer that contains the previous items. However, the Botvinick and Plaut model critically differs from many item-dependent context models in that it used a backpropagation learning algorithm instead of Hebbian learning. After extensive training with an experimental set of items, the recurrent model was able to perform serial recall without any learning occurring during the study phase. Instead, it is the maintenance of the activations in the context layer and the learned connections between that layer and the output layer that allow the model to perform serial recall.

Critically, the Botvinick and Plaut model was sensitive to the regularities in the training set, enabling it to perform better on more frequently experienced lists of items, a result which is challenging for purely positional models. At the same time, the model was capable of addressing a number of the same benchmarks that have suggested the existence of positional representations, including the phonological similarity effect and the "sawtooth" error pattern in mixed lists of phonologically confusable and non-confusable items. Analyses of the hidden unit representations uncovered that the model learned conjunctive representations of items in positions, despite the fact that within-list position was never explicitly represented in the network during the training phase. However, a downside of the Botvinick and Plaut (2006) model is the extensive training required. While such training may seem plausible for serial recall of short lists of consonants, it is unlikely to generalize to combinations of novel stimuli, such as random word lists.

Another possibility comes from the model of Dennis (2009). Dennis's model stores forward asymmetric long-range associations between items in a Hebbian outer product matrix. However, unlike many item-dependent context models, which retrieve one item at a time and update the cues for the next retrieval, the entire list of items is retrieved

simultaneously – it is only the output of items which is sequential. Retrieval probabilities for candidate lists are proportional to the difference between an outer product matrix of the candidate and the stored matrix from the learning episode.

Many phenomena emerge "for free" when such a retrieval mechanism is employed. Primacy effects naturally emerge without any reinstatement of the start of the list because the first item is heavily represented in the association matrix, such that any candidate lists for production that lack the first item are unlikely to be produced. List length effects emerge without any competition at retrieval between items because as the length of the study list is increased, there are higher numbers of similar candidate lists for retrieval.

Interestingly, similar to our demonstrations with similar item vectors in our simulations of the protrusion effect, Dennis's model was similarly able to approximate position representations with similarity among items. Specifically, if all items share a common feature, the asymmetric nature of learning will make it such that each item on the list has differing degrees of association to this common feature. That is, in a six item list, just as the first item is associated to all five succeeding items, the first item is likewise associated to the common feature five times. Since the last item is not associated to any other items on the list, it contains no associations to the common feature. Because common features are reused across lists, candidate lists that include intrusions are more likely if they contain protrusions, because the similar strength of association to the common feature makes such a candidate list more similar to the list that was just studied. While this model was not applied to the effects of temporal grouping, it is possible that the same similarity-based mechanism may be able to produce interposition errors.

We illustrate these examples to highlight the fact that item-dependent context representations may be able to account for some of the same phenomena that suggest the existence of positional representations. However, accounting for such phenomena may require rethinking how such representations are employed, either via the learning rule (backpropagation learning) or the retrieval mechanism (simultaneous retrieval of the entire

list instead of single items). There may similarly be other means of re-thinking how item-dependent context representations can be employed to behave in a manner similar to positional associations, and we believe the field could greatly benefit from such considerations.

Concluding Remarks

Logan's CRU model is an impressive and comprehensive account of serial order tasks. It represents an important step forward for the field, as decades of research have uncovered important commonalities between tasks that to date have yet to be unified in a comprehensive framework. The purpose of our commentary and the simulations contained within was to highlight some of the important challenges moving forward for CRU. We acknowledge that no model is able to capture all of the patterns of data in a given task or domain. However, the phenomena we discussed in this commentary, in conjunction with the fill-in effect, have been sufficiently influential to be considered important benchmarks for models of serial order. Furthermore, the phenomena have led theorists of serial order to almost unanimously agree upon the importance of positional representations, which CRU lacks. While we believe that one benchmark finding, namely the mixed-list phonological similarity effect of Henson et al. (1996) is addressable by CRU with modification to the output stage of the model, the other benchmarks will likely require non-trivial revisions to its architecture. Our suggestion is not necessarily that CRU requires positional representations – although this would be one possible route forward – but rather that modification to how its item-dependent context representations are either learned or retrieved may be necessary to account for these challenges. The test for CRU will be to establish whether it can be augmented in such a way as to accommodate these challenges, whilst still retaining its core representations and retrieval mechanisms that are such an attractive feature of the model.

References

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

Baddeley, A. (1968). How does acoustic similarity influence short-term memory. *Quarterly Journal of Experimental Psychology*, *20*.

Baddeley, A., Conrad, R., & Hull, A. (1965). Predictability and immediate memory for consonant sequences. *Quarterly Journal of Experimental Psychology*, *17*.

Basden, B. H., Basden, D. R., & Stephens, J. P. (2002). Part-set cuing of order information in recall tests. *Journal of Memory and Language*, *47*, 517-529.

Behmer, L. P., & Crump, M. J. C. (2017). Spatial knowledge during skilled action sequencing: Hierarchical versus nonhierarchical representations. *Attention, Perception, & Psychophysics*, *79*, 2435-2448.

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological Review*, *108*, 624-652.

Botvinick, M. M., & Bylsma, L. (2005). Regularization in short-term memory for serial order. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *31*.

Botvinick, M. M., & Plaut, D. C. (2006). Short-term memory for serial order: A recurrent neural network model. *Psychological Review*, *113*, 201–233.

Brown, G. D. A., Neath, I., & Chater, N. (2007). A temporal ratio model of memory. *Psychological Review*, *114*, 539–576.

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

Burgess, N., & Hitch, G. (1999). Memory for serial order: A network model of the phonological loop and its timing. *Psychological Review*, *106*, 551-581.

Burgess, N., & Hitch, G. J. (1992). Towards a network model of the articulatory loop. *Journal of Memory and Language*, *31*(4), 429–460.

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*, 627-652.

Conrad, R. (1960). Serial order intrusions in immediate memory. *British Journal of Psychology*, *51*, 45–48.

Conrad, R., & Hull, A. (1964). Information, acoustic confusion and memory span. *British Journal of Psychology*, *55*.

Courrieu, P., Farioli, F., & Grainger, J. (2004). Inverse discrimination time as a perceptual distance for alphabetic characters. *Visual Cognition*, *11*, 901-919.

Cox, G. E., & Shiffrin, R. M. (2017). A dynamic approach to recognition memory. *Psychological Review*, *124*(6), 795–860.

Crowder, R. G. (1968). Intraserial repetition effects in immediate memory. *Journal of Verbal Learning and Verbal Behavior*, *7*, 446–451.

Davelaar, E. J., Goshen-Gottstein, Y., Ashkenazi, A., Haarmann, H., & Usher, M. (2005). The demise of short-term memory revisited: empirical and computational investigations of recency effects. *Psychological Review*, *112*(1), 3–42.

Dell, G. S. (1986). A spreading-activation theory of retrieval in sentence production. *Psychological Review*, *93*, 283-321.

Dennis, S. (2009). Can a chaining model account for serial recall? In L. Carlson, C. Hölscher, & T. Shipley (Eds.), *Proceedings of the XXXI Annual Conference of the Cognitive Science Society* (p. 2813–2818).

DuBrow, S., & Davachi, L. (2013). The influence of context boundaries on memory for the sequential order of events. *Journal of Experimental Psychology: General*, *142*, 1277-1286.

Ebbinghaus, H. (1885/1913). *Memory.* New York: Dover. (H.A. Ruger & C. E. Bussenius Trans. Original work published 1885)

Ellis, A. W. (1980). Errors in speech and short-term memory: The effects of phonemic similarity and syllable position. *Journal of Verbal Learning and Verbal Behavior*, *19*, 624-634.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Farrell, S. (2006). Mixed-list phonological similarity effects in delayed serial recall. *Journal of Memory and Language*, *55*, 587–600.

Farrell, S. (2012). Temporal clustering and sequencing in short-term memory and episodic memory. *Psychological Review*, *119*(2), 223–271.

Farrell, S., Hurlstone, M. J., & Lewandowsky, S. (2013). Sequential dependencies in recall of sequences: Filling in the blanks. *Memory & Cognition*, *41*, 938-952.

Farrell, S., & Lelièvre, A. (2009). End anchoring in short-term order memory. *Journal of Memory and Language*, *60*, 209-227.

Farrell, S., & Lewandowsky, S. (2002). An endogenous distributed model of ordering in serial recall. *Psychonomic Bulletin and Review*, *9*, 59–79.

Farrell, S., & Lewandowsky, S. (2003). Dissimilar items benefit from phonological similarity in serial recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *29*.

Farrell, S., & Lewandowsky, S. (2004). Modelling transposition latencies: Constraints for theories of serial order memory. *Journal of Memory and Language*, *51*, 115-135.

Fischer-Baum, S. (2018). A common representation of serial position in language and memory. In K. D. Federmeier & D. G. Watson (Eds.), *The Psychology of Learning and Motivation* (p. 31-54). San Diego, CA: Academic Press.

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*, 1426–1446.

Fischer-Baum, S., McCloskey, M., & Rapp, B. (2010). Representation of letter position in spelling: Evidence from acquired sygraphia. *Cognition*, *115*, 466–490.

Fox, J., Dennis, S., & Osth, A. F. (2020). Accounting for the build-up of proactive interference across lists in a list length paradigm reveals a dominance of item-noise in recognition memory. *Journal of Memory and Language*, *110*, 104065.

Grenfell-Essam, R., & Ward, G. (2012). Examining the relationship between free recall and immediate seriall recall: The role of list length, strategy use, and test expectancy. *Journal of Memory and Language*, *67*(1), 106–148.

Grossberg, S., & Pearson, L. R. (2008). Laminar cortical dynamics of cognitive and motor working memory, sequence learning and performance: Toward a unified theory of how the cerebral cortex works. *Psychological Review*, *115*(3), 677–732.

Hannagan, T., & Grainger, J. (2012). Protein analysis meets visual word recognition: A case for string kernels in the brain. *Cognitive Science*, *36*, 575-606.

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.

Henson, R. N. A. (1996). *Short-term memory for serial order.* (Unpublished doctoral dissertation. MRC Applied Psychology Unit, University of Cambridge, Cambridge, England.)

Henson, R. N. A. (1998). Short-term memory for serial order: The start-end model. *Cognitive Psychology*, *36*, 73–137.

Henson, R. N. A. (1999). Coding position in short-term memory. *International Journal of Psychology*, *34*, 403–409.

Henson, R. N. A., & Burgess, N. (1997). 4th neural computation and psychology workshop. In J. A. Bullinaria, D. W. Glasspool, & G. Houghton (Eds.), (chap. Representations of serial order). Springer.

Henson, R. N. A., Norris, D., 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*, *49A*, 80–115.

Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, *46*, 268–299.

Hunt, R. R. (1995). The subtlety of distinctiveness: What von restorff really did.

Psychonomic Bulletin & Review, *2*, 105-112.

Hurlstone, M. J. (2019). Functional similarities and differences between the coding of positional information in verbal and spatial short-term order memory. *Memory*, *27*, 147-162.

Hurlstone, M. J. (in press). Serial recall. In M. J. Kahana & A. Wagner (Eds.), *The Oxford Handbook of Human Memory.* Oxford University Press.

Hurlstone, M. J., & Hitch, G. J. (2015). How is the serial order of a spatial sequence represented? Insights from transposition latencies. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *41*, 295-324.

Hurlstone, M. J., & Hitch, G. J. (2018). How is the serial order of a visual sequence represented? Insights from transposition latencies. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *44*, 167-192.

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*, 339-373.

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.

Lee, C. L., & Estes, W. K. (1981). Item and order information in short-term memory: Evidence for multilevel perturbation processes. *Journal of Experimental Psychology*, *7*, 149–169.

Lewandowsky, S., & Farrell, S. (2000). A redintegration account of the effects of speech rate, lexicality, and word frequency in immediate serial recall. *Psychological Research*, *63*, 163-173.

Lewandowsky, S., & Farrell, S. (2008). Phonological similarity in serial recall: Constraints on theories of memory. *Journal of Memory and Language*, *58*, 429-448.

Lewandowsky, S., & Murdock, B. B. (1989). Memory for serial order. *Psychological Review*, *96*, 25–57.

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*, 397-416.

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*.

Liu, Y. S., & Caplan, J. B. (2020). Temporal grouping and direction of serial recall. *Memory & Cognition*.

Logan, G. D. (2018). Automatic control: How experts act without thinking. *Psychological Review*, *125*, 453-485.

Logan, G. D. (2021). Serial order in perception, memory, and action. *Psychological Review*, *128*, 1-44.

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.

Mensink, G. J., & Raaijmakers, J. G. W. (1988). A model for interference and forgetting. *Psychological Review*, *95*(4), 434–455.

Murdock, B. B. (1995). Developing TODAM - 3 Models for Serial-Order Information. *Memory & Cognition*, *23*(5), 631–645.

Murdock, B. B. (1997). Context and mediators in a theory of distributed associative memory (TODAM2). *Psychological Review*, *104*(4), 839–862.

Ng, H. L. H., & Mayberry, M. T. (2002). Grouping in short-term verbal memory: Is position coded temporally? *Quarterly Journal of Experimental Psychology*, *55A*, 391-424.

Nosofsky, R. M., Little, D. R., Donkin, C., & Fific, M. (2011). Short-term memory scanning viewed as exemplar-based categorization. *Psychological Review*, *118*(2), 280–315.

Oberauer, K., Lewandowsky, S., Awh, E., Brown, G. D. A., Conway, A., Cowan, N., ... Ward, G. (2018). Benchmarks for models of short-term and working memory. *Psychological Bulletin*, *144*, 885-958.

Osth, A. F., & Dennis, S. (2015a). The fill-in effect in serial recall can be obscured by omission errors. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *41*, 1447-1455.

Osth, A. F., & Dennis, S. (2015b). Prior-list intrusions in serial recall are positional. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *41*, 1893–1901.

Osth, A. F., & Farrell, S. (2019). Using response time distributions and race models to characterize primacy and recency effects in free recall initiation. *Psychological Review*, *126*, 578–609.

Osth, A. F., Jansson, A., Dennis, S., & Heathcote, A. (2018). Modeling the dynamics of recognition memory testing with an integrated model of retrieval and decision making. *Cognitive Psychology*, *104*, 106–142.

Page, M. P. A., Madge, A., Cumming, N., & Norris, D. G. (2007). Speech errors and the phonological similarity effect in short-term memory: Evidence suggesting a common locus. *Journal of Memory and Language*, *56*, 49-64.

Page, M. P. A., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review*, *105*, 761–781.

Palmer, C., & Pfordresher, P. Q. (2003). Incremental planning in sequence production. *Psychological Review*, *110*, 683-712.

Pfordresher, P. Q., Palmer, C., & Jungers, M. K. (2007). Speed, accuracy, and serial order in sequence production. *Cognitive Science*, *31*, 63-98.

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.

Ryan, J. (1969a). Grouping and short-term memory: Different means and patterns of grouping. *Quarterly Journal of Experimental Psychology*, *21*, 137-147.

Ryan, J. (1969b). Temporal grouping, rehearsal, and short-term memory. *Quarterly Journal of Experimental Psychology*, *21*, 148-155.

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.

Serra, M., & Nairne, J. S. (2000). Part—set cuing of order information: Implications for associative theories of serial order memory. *Memory & Cognition*, *28*, 847-855.

Siefke, B. M., Smith, T. A., & Sederberg, P. B. (2019). A context-change account of temporal distinctiveness. *Memory & Cognition*, *47*, 1158-1172.

Solway, A., Murdock, B. B., & Kahana, M. J. (2012). Positional and temporal clustering in serial order memory. *Memory & Cognition*, *40*, 177-190.

Surprenant, A. M., Kelley, M. R., Farley, L. A., & Neath, I. (2005). Fill-in and infill errors in order memory. *Memory*, *13*(3), 267–273.

Thomas, J. G., Milner, H. R., & Haberlandt, K. F. (2003). Forward and backward recall: Different response time patterns, same retrieval order. *Psychological Science*, *14*, 169–174.

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*.

von Restorff, H. (1933). Über die wirkung von bereichsbildungen im spurenfeld. *Psychologische Forschung*, *18*, 299-342.

Young, R. K. (1968). Verbal behavior and general behavior theory. In T. R. Dixon & D. L. Horton (Eds.), (p. 122-148). Prentice-Hall.

Appendix: Details of CRU Simulations

Our CRU simulations differed depending on the respective phenomenon. When standard assumptions were used about orthonormal vectors, we used 28 element vectors,

where elements 1-26 corresponded to each of the letters of the alphabet, element 27 corresponded to the spacebar, and element 28 corresponded to the $LIST$ representation. Each of these vectors contained a "1" in their respective element and a "0" for all other elements. There were cases where other dimensions were added to reflect common vector components, which we detail below.

Each of our simulations used uppercase letters, as both the experiments of Hurlstone (2019) and Page et al. (2007) used uppercase letters as their experimental stimuli. While the experiments of Osth and Dennis (2015b) used words as stimuli, there to date has not been an extension of CRU to words. We nonetheless used the same encoding process as in the core version of the model to allow for the possibility that words could be encoded as other words. We used the same distance matrix as Logan (2021) to represent confusions between uppercase letters, which was based on a multidimensional scaling solution to a set of response time confusion measures between letters (Courrieu, Farioli, & Grainger, 2004). Similar to Logan (2021), output-based confusions were not employed except for the simulation of phonological similarity effects in the main text, where the details of the construction of the distance matrix can be found. Across all simulations of the encoding process we used $g_{max} = .3612$ and $g_{decrease} = .8896$, which were the best fitting group-averaged parameters for serial recall in Experiment 1 (these can be found in Table 4 of Logan, 2021). An exception to these parameters is the simulation of encoding stage confusions in Supplementary Materials A where the g parameter was explicitly manipulated and a different distance matrix was employed.

Similar to Logan (2021), we fixed the threshold θ of all racing diffusion processes to 200. The β parameter was manipulated across all simulations and these values can be found in their respective sections in the main text. We did not allow β to vary across list positions ($\beta_{decrease} = 1.0$). All model code can be found at https://osf.io/gnrwz/.

Phonological Similarity Effects

As mentioned in the main text, all simulations here used the same stimuli and number of trials as the original Page et al. (2007) experiment - 64 trials for each list type, performing 100 simulations for each trial. The confusable letters were B, C, D, G, P, T, and V. The non-confusable letters were H, J, L, Q, R, Y, and Z (Z is pronounced as "zed" in this study due to the usage of British English). The g parameter for output-stage confusions can be found in the main text. Details of simulation with other methods (item vector similarity and encoding-stage confusions) can be found in Supplementary Materials A.

The Protrusion Effect

As mentioned in the main text, a two list paradigm was simulated with distinct $LIST$ elements for each list ($LIST_1$ and $LIST_2$). Only recall of the second list was simulated. When the second list was learned, the context vector was not "cleared." Instead, the $LIST_2$ marker entered the context and proceeded with evolution (e.g. Lohnas et al., 2015).

When similarity among either the list elements or item vectors was introduced, we employed 30 element vectors. The first 27 elements were used in the same fashion as other simulations. The important differences are that the common component m was element 28, $LIST_1$ was element 29, and $LIST_2$ was element 29. When similarity was introduced to both the list elements and the item vectors, we used 31 element vectors, where the common component m_{item} was element 28, the common list element component m_{list} was element 29, and $LIST_1$ and $LIST_2$ were elements 30 and 31, respectively. The orthogonal components of the item vectors were always their original elements (1-26).

In the simulations of the two list paradigm, we randomly sampled a set of 12 letters and divided these between list 1 and list 2. No items were shared between each of the lists. Because prior list intrusions could be rare with some combinations of parameters, we performed many more simulations than in the other demonstrations. For each combination of parameters, we generated 250 pairs of lists. For each trial, we performed 500 simulations

of the model.

The relevant model parameters that we varied (β, s_{list}, and s_{item}) are detailed in the main text as well as in Supplementary Materials B, where additional simulations are reported where s_{list} and s_{item} were jointly manipulated.

Temporal Grouping

The model simulations incorporated the same letters (F, H, J, L, N, Q, R, S, Y) and experimental parameters (20 trials for each list type) as in the original experiment by Hurlstone (2019). For each list type, 100 simulations were performed. Manipulations of β, the increase in β at group boundaries, as well as the similarity among the group markers is all detailed in the main text.

Our simulations of group markers in the main text changed the nature of the vector representations. In these simulations, we employed 31 element vectors: elements 1-26 corresponded to the letters, element 27 was the spacebar, elements 28-30 corresponded to the group markers, and element 31 was the list context. An additional scheme using orthogonal and common components is described in Supplementary Materials C.