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

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

Data and model code can be found at https://osf.io/gnrwz/

Abstract

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

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

A: Alternative CRU Variants of Mixed List Phonological Similarity Effects

In the main text, we endorsed a CRU variant where phonological similarity effects are due to confusions that occur during the output stage, as originally specified in Logan (2018). Here, we explore two other variants of phonological similarity effects. The first assumes such effects arise due to similarity among the item vectors and the second assumes that confusions occur during the encoding stage.

Item Vector Similarity

Common Vector Elements for Each Element in Confusable Items. In CRU, all of the item vectors are orthonormal vectors, where a single element is set to 1 and all other values are set to 0. This makes it such that the vectors are completely dissimilar to each other – the dot product between any item vector and another is always zero. Instead, in CRU, similarity operates at the level of the stored context vectors, which are similar to the extent they contain the same items. For instance, for a list such as $LIST - A - B - C - D - E - F$, the context vectors for D ($LIST - A - B - C$) and E ($LIST - A - B - C - D$) will be similar due to a high proportion of shared item vectors ($LIST - A - B - C$ are both present in their context vectors). The similarity between D and A's context vectors, in contrast, will be much lower, as A's context vector only contains the $LIST$ vector.

We deviated from CRU's similarity scheme by using similar vectors for phonologically confusable letters, specifically the letters B, D, G, P, T, and V, which all share a common rhyme. To construct the item vectors, we constructed a weighted blend of the original orthogonal vectors (which we term r^o) and vectors where the six elements corresponding to the confusable items all have the same value (which we term r^s), namely 1 / $sqrt(6)$, which was chosen to ensure that these vectors are all of length 1. We constructed the item vector

r for each phonologically confusable item i using the following equation:

$$r_i = sr^s + \rho r_i^o \tag{1}$$

where ρ is calculated in the same manner as Equation 2 with the exception that r^s substitutes for c. s is a free parameter that governs the extent to which the vectors are similar to each other - as s approaches 1.0, all of the phonologically similar vectors become identical.

We simulated CRU's predictions for the Page, Madge, Cumming, and Norris (2007) data using the same list structure and number of trials as the original experiment (64 trials for each list type), performing 100 simulations for each trial. The nonconfusable 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), which all used the traditional orthonormal vectors. We set g_{max} and $g_{decrease}$ to .3612 and .8896, respectively, which were the best fitting group-averaged parameters from Logan's (2021) fits to his Experiment 1 data. As in the original model simulations, the threshold of the racing diffusion process θ was set to 200.

Simulations for a range of different β (1.0, .65, .45, and .25) and s values (.95, .8, .6, and .4) can be seen in Figure 1. What is immediately evident from the figure is that none of the combinations of parameter values produce a pattern that even bears a qualitative resemblance to the data from Figure 1 in the main text. Second, similarity effects are most apparent with the higher values of similarity (s =.95 and .8), which are in the top two rows, which we will focus our attention on.

As mentioned previously, when β is higher, there is considerably more emphasis at both encoding and retrieval on the most recently presented item. Sure enough, for the higher values of β (1.0 and .65), one can see that similarity has disastrous consequences for the model. Under these conditions, when an error occurs on the alternating lists, there is little apparent ability for CRU to recover from that error. While previous simulations have demonstrated a strong ability for CRU to recover from errors (Logan, 2018), it is important to consider here that even if the prior recalled items are correct, the high similarity of the confusable items in the context cue employed can prevent such recovery from occurring.

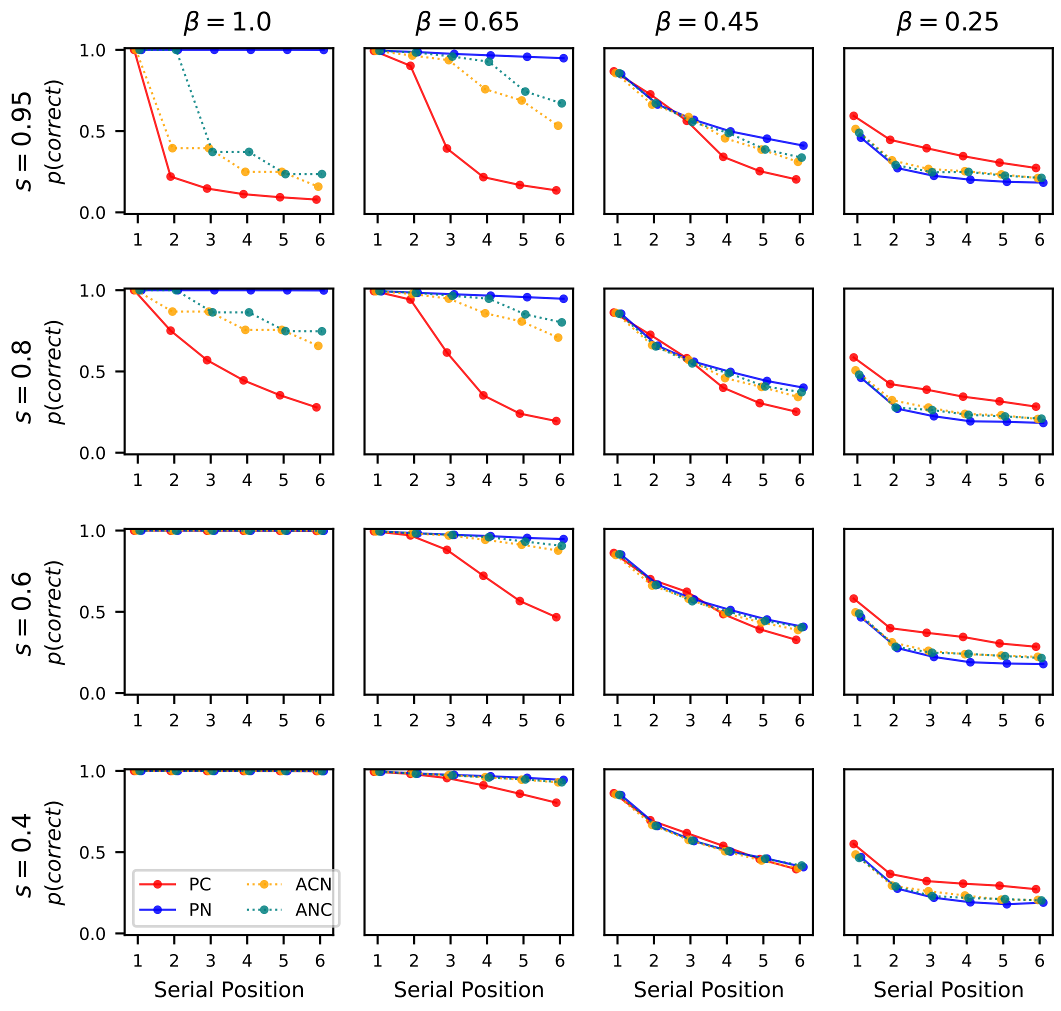

Figure 1. CRU simulations for the Page et al. (2007) paradigm using similar item vectors for phonologically similar items. Notes: PC = pure confusable, PN = pure nonconfusable, ACN = alternating confusable-nonconfusable, ANC = alternating nonconfusable-confusable.

However, an interesting and unexpected result in these circumstances is that the errors in alternating lists are most likely to occur on *nonconfusable* items. To illustrate the consequences of item vector similarity on the resulting context vector similarity, we used Logan (2021)'s method of plotting the similarity between all possible context vectors, where similarity is measured using the dot product. These are depicted in Figure 2 with the four different values of β (1.0, .65, .45, and .25, from top to bottom) for both pure nonconfusable (left column) and mixed lists (right column), where the confusable items are illustrated using dashed lines and were generated using $s = .95$. Context vectors always yield maximum similarity to themselves (1.0), and higher similarity to nearby context vectors than distant ones, but similarity drops off more gradually for lower values of β.

What Figure 2 reveals is that the first confusable item in the mixed list (item 2) yields a very similar similarity gradient to its respective nonconfusable item in the pure nonconfusable list. The most apparent differences in the mixed list occur after the first confusable item. Specifically, the third and fifth items, which are both nonconfusable, have considerably higher similarity to other nearby context vectors. In the limiting case where $\beta = 1.0$, these context vectors are highly similar to each other, but to no other context vectors.

Why would confusable items cause errors on nonconfusable items in CRU? While CRU employs orthonormal vectors for the items, there are counterintuitive consequences to the introduction of similarity between item vectors. Let us consider an alternating nonconfusable-confusable list such as $\text{H}\underline{\text{G}}\text{Q}\underline{\text{T}}\text{J}\underline{\text{V}}$. As mentioned previously, a property of CRU's encoding mechanism is that each context vector does *not* store its own item representation – only the previous items are encoded in it. This means the context vector for the second item G contains $LIST - H$, the context vector for the third item Q contains $LIST - H - G$, and the context vector for the fourth item T contains $LIST - H - G - Q$. Thus, when the item vectors for the rhyming letters G, T, and V are highly similar to each other, it has relatively little influence on the similarity of their context vectors and no influence on the first similar item G, whose context vector does not contain any of the other rhyming letters. Instead, the similar item vectors have a much larger influence on the

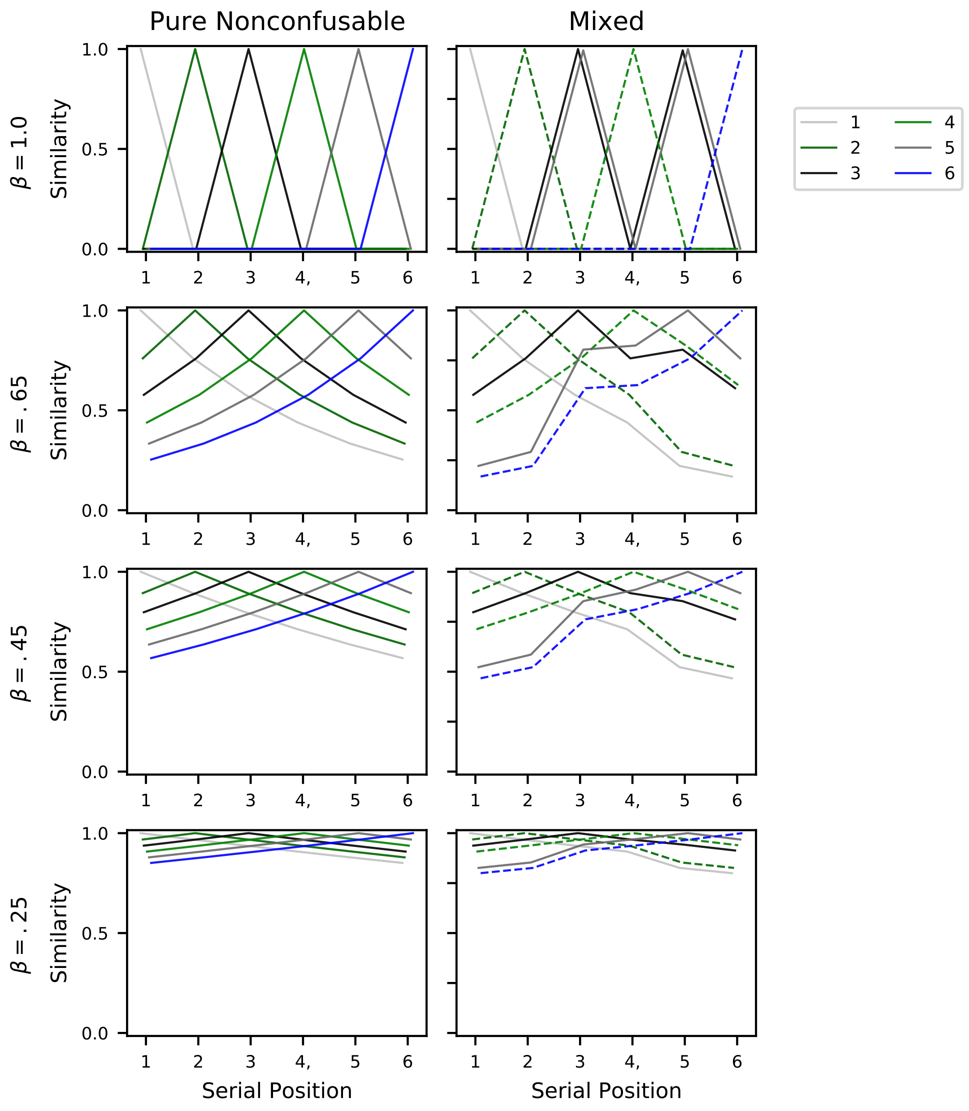

Figure 2. Dot products between all context vectors for pure nonconfusable (left column) and mixed (right column) lists with four different values of β. In the mixed lists, the confusable items were generated with $s = .95$ and are indicated by dashed lines.

nonconfusable items that occur *after* confusable items, because the context vectors for these items contain very strong representations of the confusable items which preceded them, whose item vector representations are all highly similar to each other. This is especially the case when β is high, as the context vectors are dominated by the preceding item in this case.

To illustrate this point, consider when a participant correctly recalls the first nonconfusable letter H. H enters the context vector, which should have high similarity to G but little similarity to other context vectors because H is orthogonal to all the other item vectors, leading to a high probability of correctly recalling G. However, when the item vector for G enters the context, while the context vector becomes highly similar to the (correct) third item Q, it also becomes highly similar to the fifth letter J. This is because the context vector for the fifth letter J strongly represents the item vector for T, which is highly similar to the other confusable letters. Thus, when similarity among the item vectors is introduced into CRU, the model makes the counter-intuitive prediction that high similarity among item vectors is likely to cause errors on the items that *follow* the confusable items. We return to this point later when we simulate the consequences of temporal grouping using group markers with similar vectors, where kindred consequences of vector similarity materialize.

While we had anticipated that the model might fare somewhat better when β is lower, we were surprised to find that under these circumstances the phonological similarity effect decreases and even reverses when $\beta = .25$. This is likely due to the fact that when the item vectors are highly similar, there is a compensation in the ρ normalization, which is sensitive to the similarity between the context and item vectors. As s increases, ρ increases as well, which can result in adjacent context vectors being more dissimilar than when the original orthonormal vectors are employed. This is less consequential when β is higher, as higher values of β result in context vectors that are dissimilar to each other.

Consonant and Vowel Item Vector Structure. A second method of exploring item vector similarity was as follows. We used a scheme where each letter is composed of two components to reflect the combination of a consonant (o) and a vowel (v). For the phonologically confusable letters, the vowel is shared by all of the items and corresponds to the letter "i." For phonologically nonconfusable items, the vowel vector to each of the letters.

The vector for each letter r is a weighted combination of o and v:

$$r_i = \sqrt{(1-s)}o_i + \sqrt{s}v \tag{2}$$

where s is a similarity parameter that governs the contribution of the v vector. As s approaches one, all of the confusable items are dominated by the common v vector. Because the nonconfusable items do not share a vowel, the s parameter merely governs the weighting of two orthogonal components. Consequently, there are no changes in the dot products between their corresponding item or context vectors as s is increased.

CRU simulations were otherwise performed in the same manner as in the previous section. Results depicting the serial position curves can be seen in Figure 3 while the similarities between context vectors can be seen in Figure 4. Results are largely in accordance with the previous method of manipulating similarity. None of the serial position curves in Figure 3 depict a "sawtooth" pattern.

Summary. While these simulations led to some interesting and counter-intuitive results, it is clear that none of the parameterizations across each method of manipulating item vector similarity bear much resemblance to the qualitative pattern of data found across many experiments. While there remain other possible ways of manipulating item vector similarity not pursued in this commentary, it is unclear how any other patterns could circumvent the conceptual problem inherent when vector similarity is approached. In mixed lists, as many as 50% of the retrieved items present in the context vector serve as misleading cues for the next response.

Confusions During the Encoding Stage

In CRU, confusions can occur when each item is encoded into memory. While each item vector is perfectly encoded (no noise is introduced into its features), an item can be mistaken for another item altogether. This has implications for phonological similarity effects because it is possible that phonologically confusable items are mis-identified as other

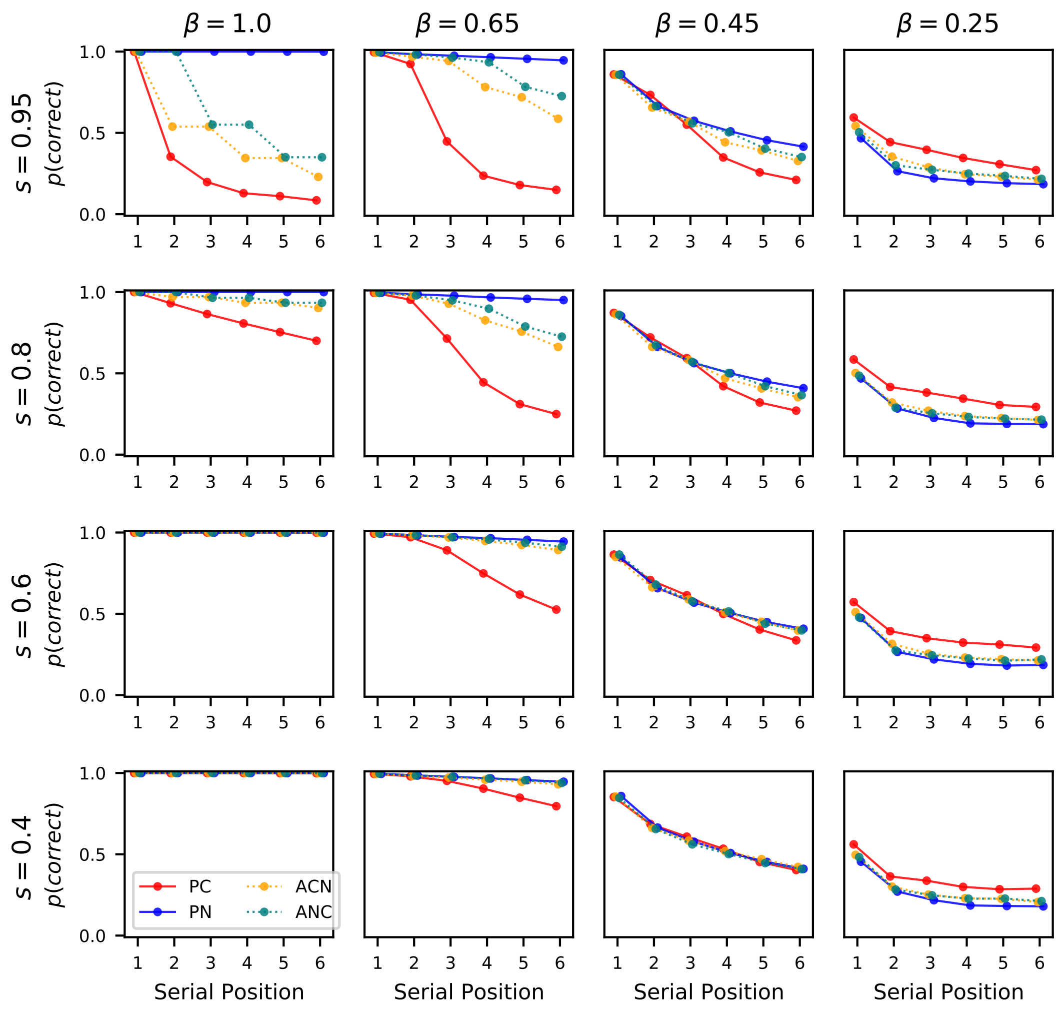

Figure 3. CRU simulations for the Page et al. (2007) paradigm using similar item vectors for phonologically confusable items using the consonant-vowel structure for manipulating item vector similarity. Notes: PC = pure confusable, PN = pure nonconfusable, ACN = alternating confusable-nonconfusable, ANC = alternating nonconfusable-confusable.

confusable items.

As mentioned in the main text, as each letter is input to the model, an identification stage occurs that is implemented as a racing diffusion decision process among all of the available letters. The drift rate v for a given letter i's accumulator after presentation of letter j is:

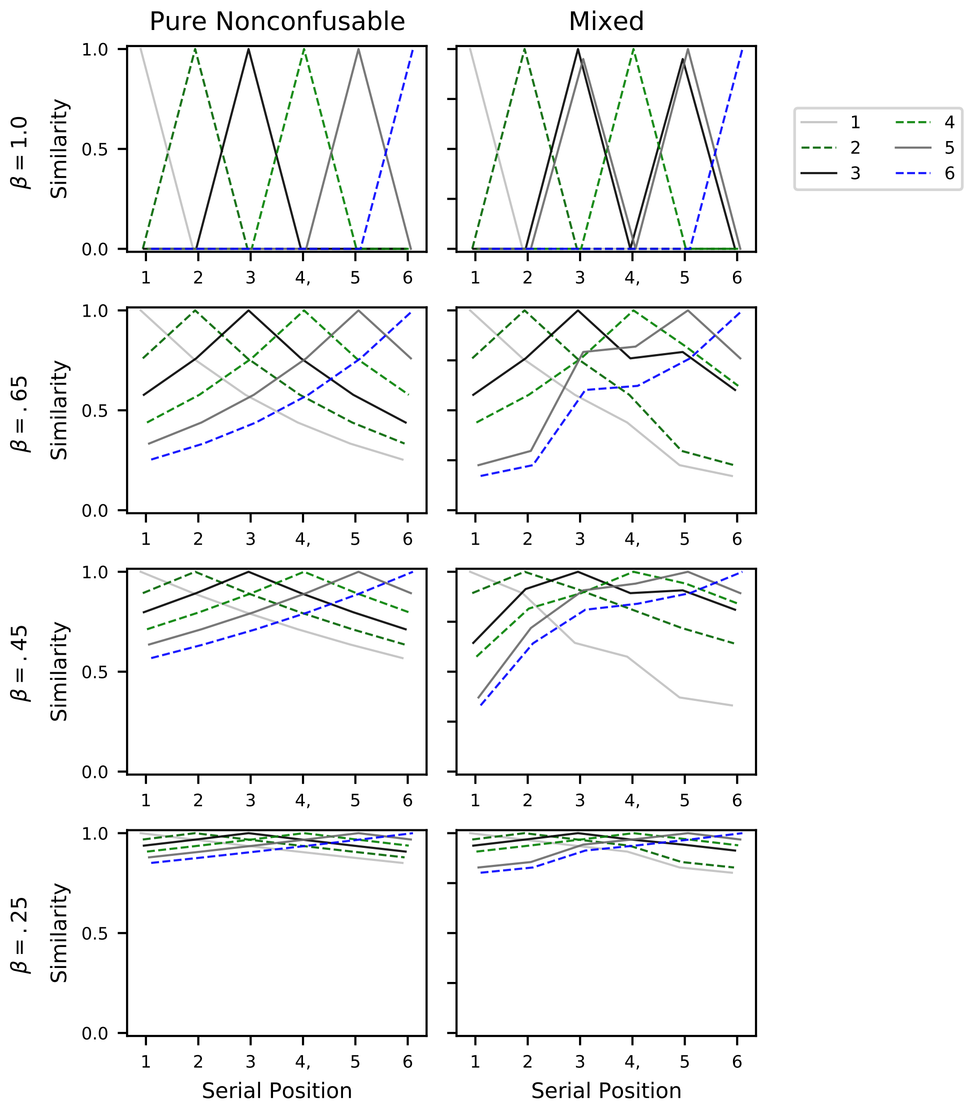

Figure 4. Dot products between all context vectors for pure nonconfusable (left column) and mixed (right column) lists with four different values of β using the consonant-vowel structure to manipualte item similarity. In the mixed lists, the confusable items were generated with $s = .95$ and are indicated by dashed lines.

$$v_i = exp(-gd_{ij}) \tag{3}$$

where d_{ij} is the distance between letters i and j in a multidimensional space. g is a sensitivity parameter – as g increases, the drift rates for letters adjacent to the target letter j are reduced. To construct a multidimensional psychological space, Logan (2021) employed a multidimensional scaling solution based on visual confusions among the letters. In CRU, if a letter is misidentified as another letter from the lexicon, the erroneously perceived letter enters the context, making it unlikely that the correct letter will be recalled during retrieval. Nonetheless, that error does not necessarily contribute to other errors in the list. Consider again the list CKPXGL. If the letter P is erroneously perceived as G, then G's context vector will contain $LIST - C - K$. If the fourth item X is correctly encoded, its context vector will contain $LIST - C - K - G$. Thus, after recall of the list CKG, recall is most likely to be followed by the correct item X, since G will be strongly represented in X's context vector.

We initially performed CRU simulations with the base model and the MDS solution employed by Logan (2021) but found it was not able to produce the phonological similarity effects, which is likely due to the fact that his MDS solution was based on visual confusions of the letters. For this reason, we simulated our own set of distances for confusable and nonconfusable letters to evaluate whether similarity-based confusions during encoding was capable of producing the phonological similarity effects. For each letter pair, we simulated the distance value d from a truncated normal distribution. For pairs of confusable letters, 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 $\sigma = 1.0$.

We simulated CRU using our simulated distance matrix for encoding-based confusions with four different values of β (1.0, .65, .45, and .25) and four different values of g (.1, .3, .5, and 1.0) – these results can be seen in Figure 5. To simplify the predictions, g did not vary across serial positions ($g_{decrease} = 1.0$). Figure 5 reveals that the majority of parameter combinations yield a pattern that is qualitatively similar to the data, namely a sawtooth pattern in mixed lists with peaks corresponding to nonconfusable items and troughs corresponding to confusable items.

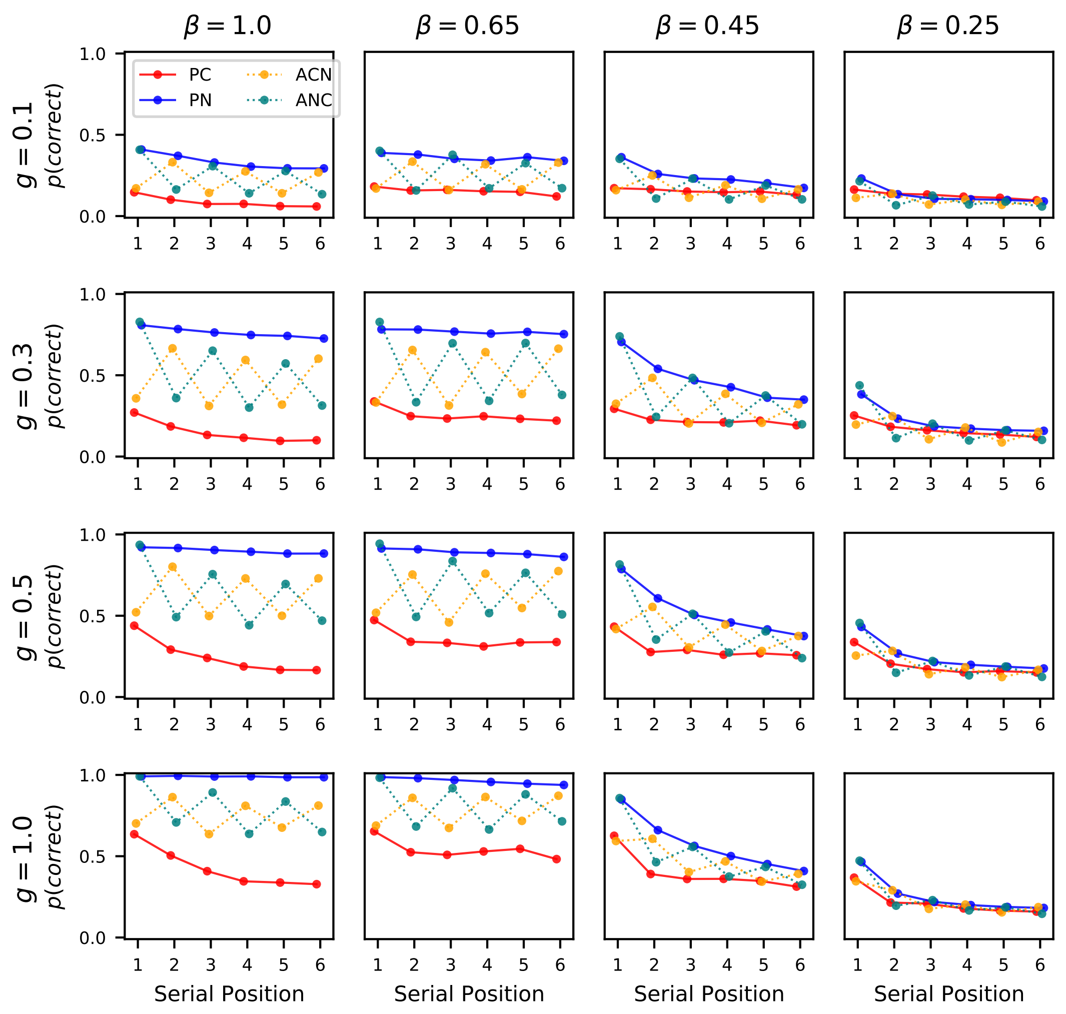

Figure 5. CRU simulations for the Page et al. (2007) paradigm using encoding-based confusions. Notes: PC = pure confusable, PN = pure nonconfusable, ACN = alternating confusable-nonconfusable, ANC = alternating nonconfusable-confusable.

While this demonstration is impressive, there are some deviations from the patterns in previous datasets using this paradigm. First, there are several combinations of parameter values where confusable items in pure lists perform much worse than in mixed lists. While this bears a qualitative correspondence with findings from this paradigm (e.g., Farrell & Lewandowsky, 2003), typically the performance impairment for confusable items in pure lists is not dramatically worse than when presented in mixed lists.

An analysis of the model's errors found that repetition errorsand extralist intrusions were very frequent- these can be seen in Table **??**. Repetition errors during the encoding stage are highly consequential because the same item is associated to two different contexts during list presentation. For instance, for the list CKPXGL, if the letter G is erroneously encoded as a repetition of C, the letter C is present in two different stored context vectors: the first being K, in the second serial position ($LIST - C$), and the second occurrence being the letter L, in the sixth serial positions ($LIST - C - K - P - X - C$). In this case, the erroneous repetition of C will lead to a higher likelihood of transpositions to the end of the list. Thus, repetitions during encoding not only produce an increase in repetition errors during output, but an increase in transpositions as well. Encoding-stage confusions of other items on the list are also more likely in pure confusable lists due to the maximum number of confusable items being studied – this considerably reduces performance in this condition relative to the other list types.

Table 1
Proportions of repetition errors and extralist intrusion errors relative to all responses from a given trial collapsed across list types from the CRU variant with encoding-stage confusions. Extralist intrusion proportions are depicted in italics.

	$\beta = 1.0$	$\beta = 0.65$	$\beta = 0.45$	$\beta = 0.25$
g = 0.1	0.496 *0.075*	0.381 *0.107*	0.384 *0.176*	0.373 *0.256*
g = 0.3	0.314 *0.059*	0.184 *0.082*	0.183 *0.153*	0.178 *0.24*
g = 0.5	0.234 *0.052*	0.108 *0.067*	0.107 *0.138*	0.108 *0.229*
g = 1.0	0.145 *0.036*	0.057 *0.044*	0.056 *0.114*	0.054 *0.213*

In addition to the frequent repetition errors, there were non-negligible frequencies of extralist intrusions for confusable items. These extralist intrusions occurred for a similar reason – phonologically confusable items could often be erroneously encoded as other confusable items that were not presented on the list. While it is possible that the encoding-based repetition errors could be reduced through a form of response suppression during encoding, wherein perception of a given item increases its threshold θ in the racing diffusion process (e.g. Lohnas, Polyn, & Kahana, 2015), it is not clear how the incidence of extralist intrusions could be reduced under parameter regimes that simultaneously give rise to sizeable phonological similarity effects.

B: Simulations of Prior List Intrusions with Both Item Vector and List Context Similarity

In the main text, we explored simulations of a two list paradigm with CRU where either similarity among the list contexts was manipulated or similarity of the item vectors was manipulated. To recap these results, similarity among the list contexts produced a primacy-driven similarity pattern where context vectors from list 2 showed the strongest match to context vectors from the beginning of list 1, because the $LIST$ element is strongest for the beginning-of-list items. Similarity among the item vectors showed a recency-driven similarity pattern, where the context vectors from list 2 showed the strongest match to the context vectors from list 1 that were stored at the end-of-the-list. This occurred because the common item vector component is strongest for the final list items.

While simulations demonstrated a protrusion effect could occur, the parameterizations that caused this pattern suggested the retrieval of the entirety of list 1. This was indicated by high proportions of successive prior list intrusions. The data instead suggest that successive prior list intrusions are quite rare (Osth & Dennis, 2015).

In this section, we explored the consequences of manipulating both the item vector and list context vector similarity simultaneously. To do this, we used separate common vector components for each vector type (m_{item} and $m_{context}$) and weighted the contributions of the unique and common components according to Equations 8 and 9 for list context vectors and item vectors, respectively. The purpose of having separate common components is to reflect the idea that list context elements can be confused with list context elements from other lists, and that different items can be confused with each other, but list contexts are not confused with items and visa versa. Introduction of similarity between list context elements and item vector elements may greatly compromise the model's ability to produce the primacy effect, as the initial $LIST$ cue would show a stronger match to the context vectors where item vectors are strongly represented, such as the final list item.

As before, we manipulated β across a wide range ($\beta = .25$, .45, .65, or 1.0) and additionally manipulated s_{list} ($s_{list} =$, .25, .5, or .75) and s_{item} ($s_{item} = 0$, .25, .5, or .75). Because factorial manipulation of three parameters would produce a rather crowded plot, we instead show these results for separate plots conditioned on each value of s_{list}.

We begin with the analysis of the similarity gradients of the list 2 context cues to the stored context vectors from list 1 and list 2 assuming that the previous list 2 items were perfectly recalled. These can be found in Figures 6, 7, and 8 for s_{list} values of .25, .5, and .75, respectively. We had initially expected that such similarity gradients would show a primacy focus and a recency focus. We were somewhat surprised to find that for values of $\beta < 1.0$, there were some combinations of parameter values where the similarity gradients for output positions 2 and 3 peaked at midlist positions. For instance, Figure 8 (where $s_{list} = .75$) demonstrates similarity gradients for output position 2 and 3 peaking at the same position on list 1.

While these similarity gradients are promising, it is against instructive to evaluate the model via simulation, as the assumption that all items are previously recalled is not a realistic assumption. Simulations of the model can be found in Figure 9, 10, and 11 for $s_{list} = .25$, .5, and .75, respectively. Each of these simulations indicate protrusion effects for some combinations of parameter values, namely when there is a combination of higher values of β and s_{list}.

However, similar to the simulations in the main text when only list context similarity

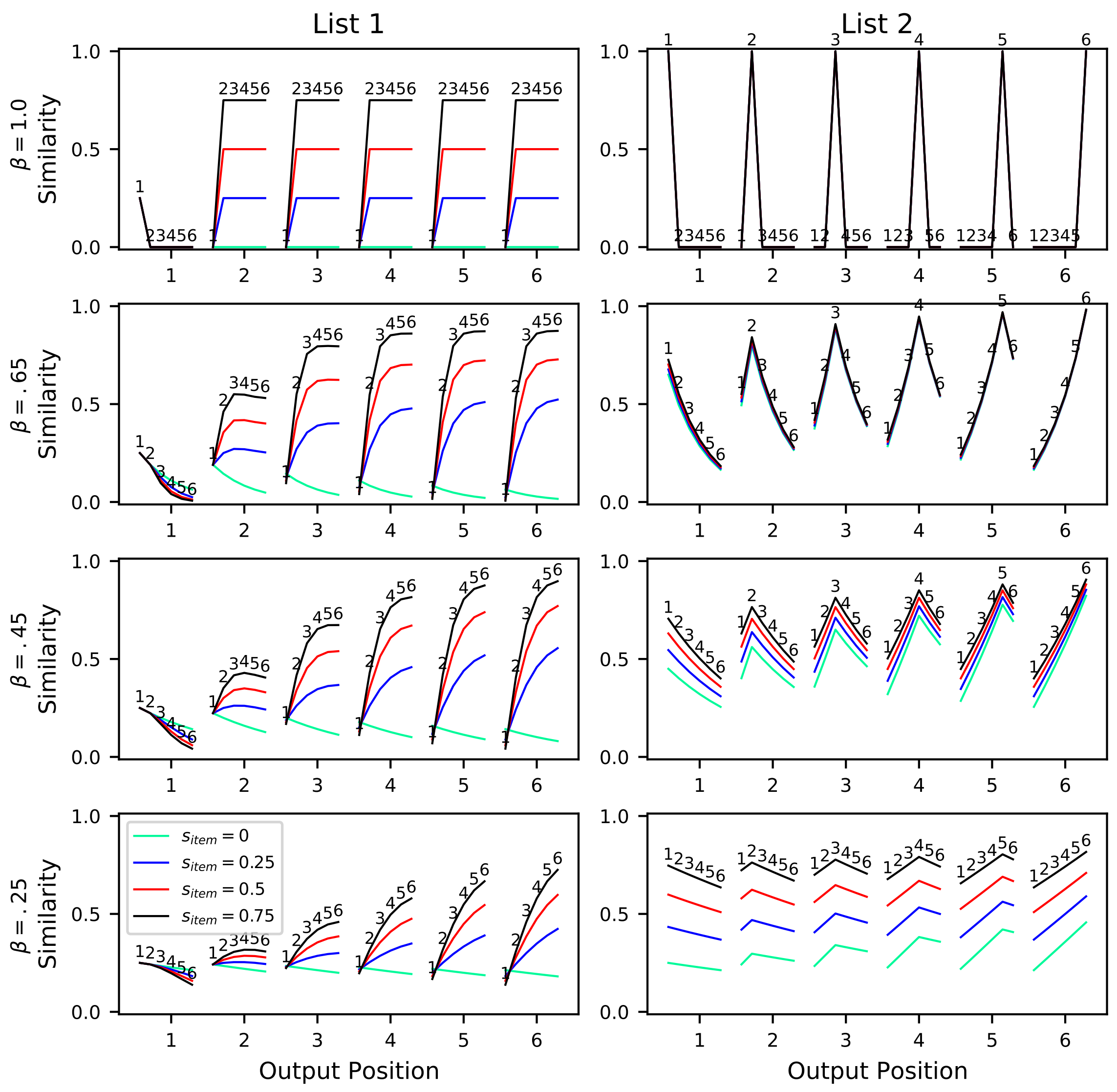

Figure 6. Dot products between the list 2 context cue for each output position (indicated below the x-axis) and the stored list 1 context vectors (left column) and list 2 context vectors (right column) when similarity among list context vectors is manipulated (as indicated by the s_{item} parameter) and $s_{list} = .25$. 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.

was manipulated, each combination of parameter values that shows a protrusion effect is accompanied by a high proportion of successive prior list intrusions, as indicated by the right columns of Figures 9, 10, and 11. These results suggest that it is quite likely that these protrusion effects reflect the retrieval of the entire previous list in order.

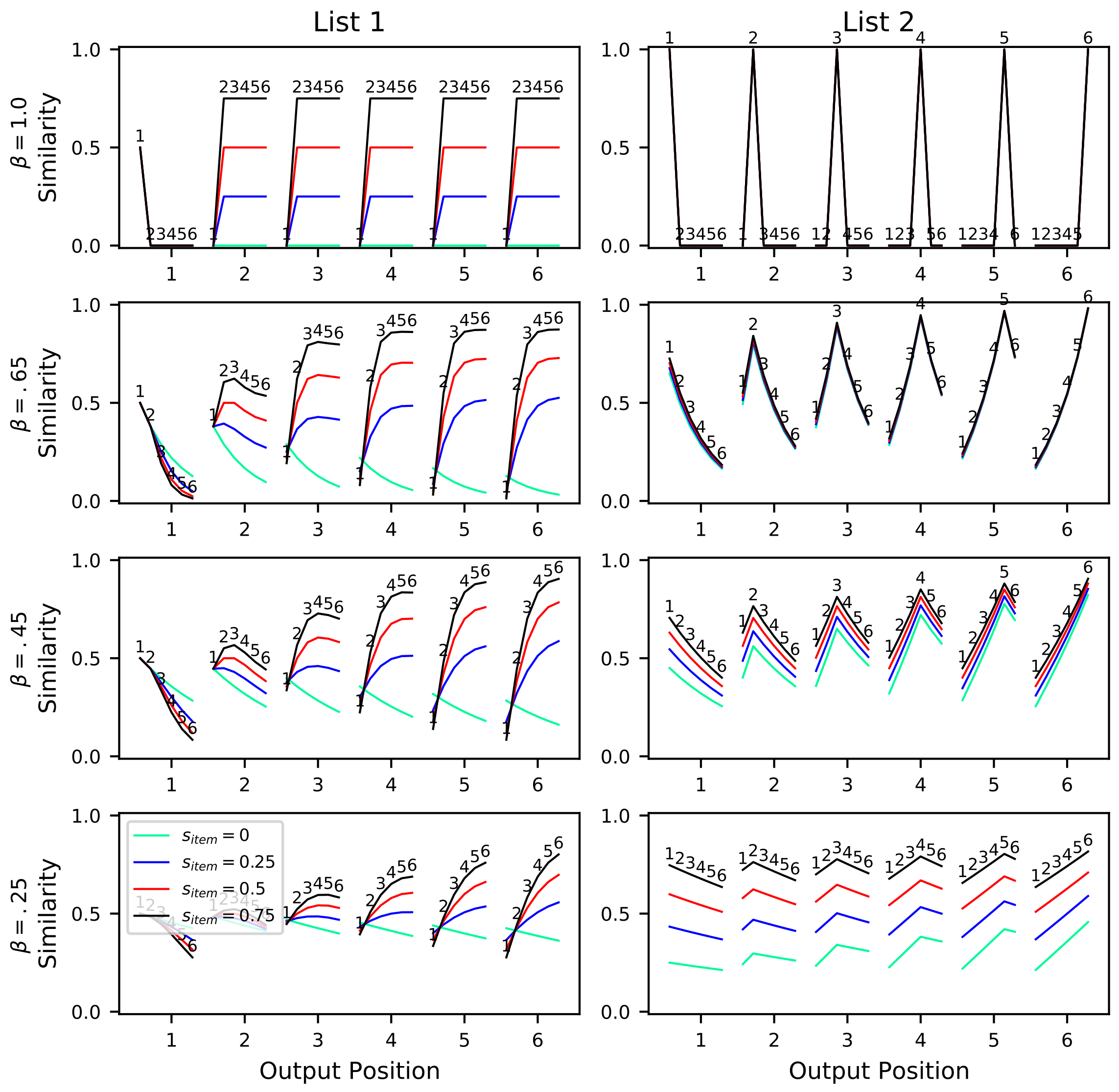

Figure 7. Dot products between the list 2 context cue for each output position (indicated below the x-axis) and the stored list 1 context vectors (left column) and list 2 context vectors (right column) when similarity among list context vectors is manipulated (as indicated by the s_{item} parameter) and $s_{list} = .50$. 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.

C: Grouping Simulations with Different Group Markers

In the main text, we explored simulations of CRU with group markers where each of the elements of the group markers are shared, but are weighted in such a way that markers corresponding to adjacent groups are more similar than markers corresponding to distant

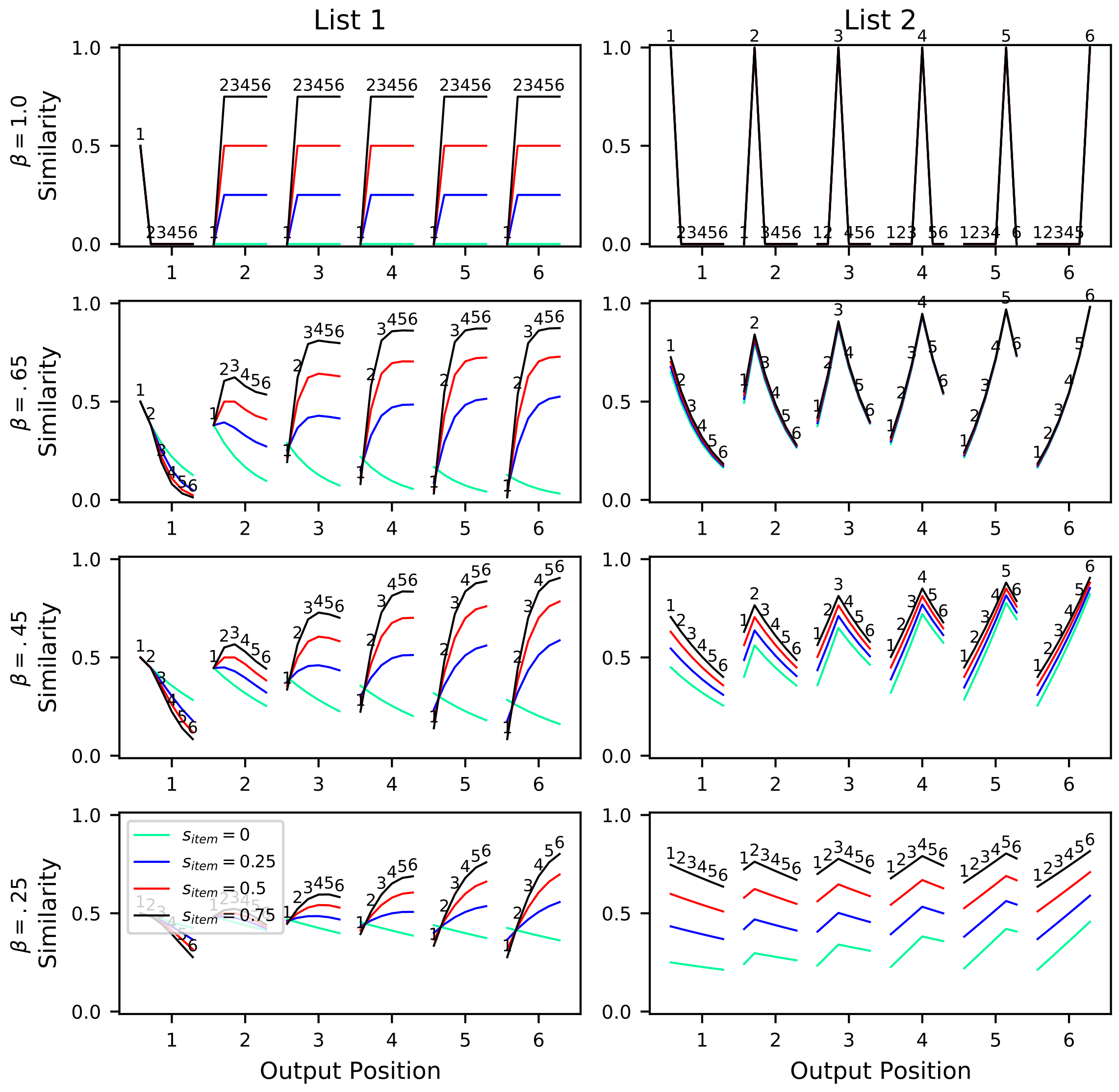

Figure 8. Dot products between the list 2 context cue for each output position (indicated below the x-axis) and the stored list 1 context vectors (left column) and list 2 context vectors (right column) when similarity among list context vectors is manipulated (as indicated by the s_{item} parameter) 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.

groups. 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

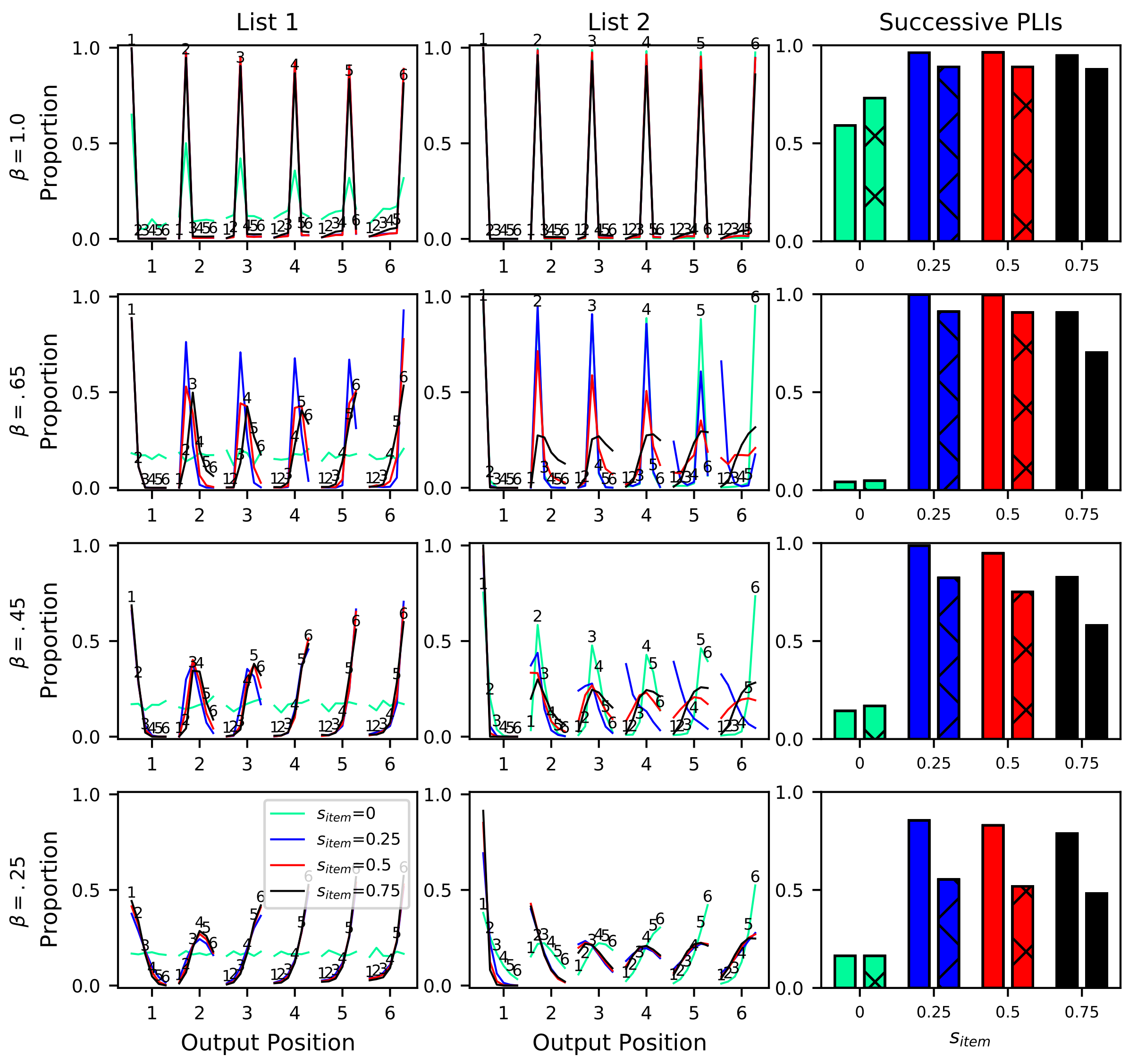

Figure 9. CRU simulations for a two list paradigm with attempted recall of the second list when similarity among the item vectors (s_{item}) is manipulated and $s_{list} = .25$. Depicted for each output position (indicated below the x-axis) are the proportions of recalls from each serial position in 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.

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.

The vectors for the group markers were orthogonal to all other vector representations

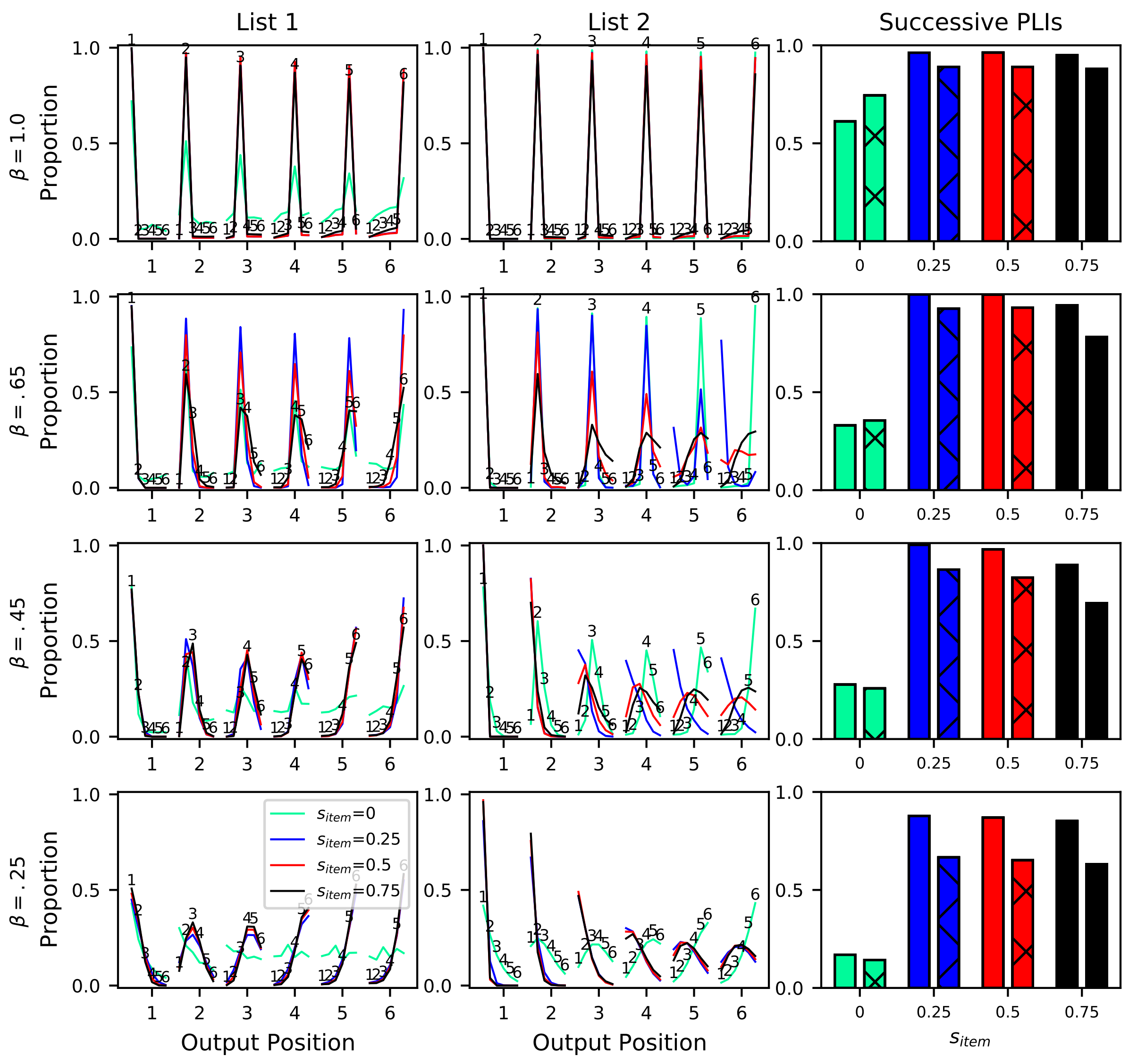

Figure 10. CRU simulations for a two list paradigm with attempted recall of the second list when similarity among the item vectors (s_{item}) is manipulated and s_{list} = .50. Depicted for each output position (indicated below the x-axis) are the proportions of recalls from each serial position in a 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.

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

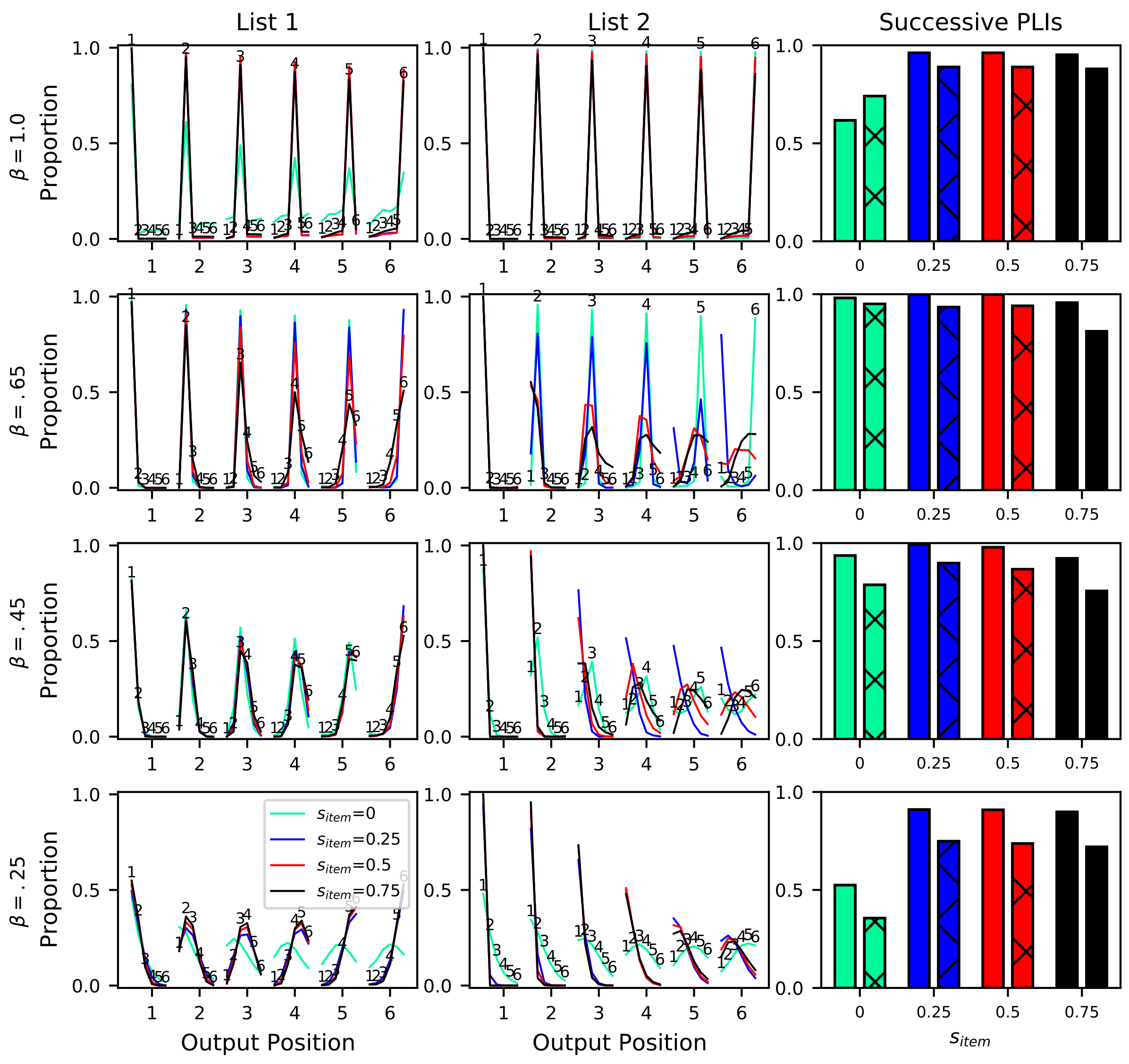

Figure 11. CRU simulations for a two list paradigm with attempted recall of the second list when similarity among the item vectors (s_{item}) is manipulated and $s_{list} = .50$. Depicted for each output position (indicated below the x-axis) are the proportions of recalls from each serial position in 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.

list context.

In the following simulations, we use an alternative scheme that includes both orthogonal and common vector components for the group markers. In these simulations, elements 28-30 correspond to the orthogonal components o_1, o_2, and o_3 corresponding to each group (element 28 is group 1, element 29 is group 2, and element 30 is group 3). Elements 31 and 32 are common vector components (m_1 and m_2) that are shared among each of the group markers, but are weighted differently in order to capture the idea that adjacent group markers are more similar to each other than distant group markers.

In these simulations, we varied two parameters: s_{group} weights the relative contribution of the common and orthogonal vector components in the group markers, while w reflects the relative weight of the two common components. Specifically:

$$g_1 = \sqrt{1 - s_{group}} o_1 + \sqrt{s_{group} w} m_1 + \sqrt{s_{group}(1 - w)} m_2 \tag{4}$$

$$g_2 = \sqrt{1 - s_{group}} o_2 + \sqrt{.5 s_{group}} m_1 + \sqrt{.5 s_{group}} m_2 \tag{5}$$

$$g_3 = \sqrt{1 - s_{group}} o_3 + \sqrt{s_{group}(1 - w)} m_1 + \sqrt{s_{group} w} m_2 \tag{6}$$

As w approaches 1, only m_1 is represented in g_1 while m_2 has zero weight, while in g_3 only m_2 is active and m_1 has zero weight. In g_2, both m_1 and m_2 are equally weighted. Thus, w governs the similarity between g_1 and g_3 – as w approaches 1, both g_1 and g_3 are as similar to each other as g_1 is to g_2 and g_2 is to g_3.

As s_{group} increases, the weight on the orthogonal components decreases. This makes adjacent group markers more similar to each other. Note that this parameter is separate from w because if both s_{group} and w are 1.0, then adjacent group markers are maximally similar to each other, while the dot product between g_1 and g_3 will be zero. If $s_{group} = 1.0$ but $w = .5$, then all of the group markers will be identical to each other. If $s_{group} = 0$, each of the group markers are orthogonal to each other.

Simulations of the model with three different values of s_{group} (.5, .75, and 90) in Figures 12, 13, and 14 which reflect w values of .5, .75, and .90, respectively. The letters and the number of trials correspond to the details of the experimental paradigm of

Hurlstone (2019). One can see that Figures 12 and 13 in particular bear a resemblance to the simulations with group markers in the main text. When both s_{group} and β are higher, there is a tendency for CRU to produce interposition errors, as reflected in an increase in three-apart transpositions relative to ungrouped lists. However, these same combinations of parameters do not demonstrate much of an increase in performance for grouped lists relative to ungrouped lists. Thus, they appear to capture the costs but not the benefits of temporal grouping.

In the main text, we demonstrated that an additional problem with the group markers that we employed was that they only showed elevated similarity for the first members from each group. We performed a similar analysis here and plotted the pairwise similarities between all context vectors in grouped and ungrouped lists, but focused on the cases where group markers have high similarity ($s_{group} = .9$ and $w = .75$).

This plot reveals a very similar pattern as found with the implementation of group markers in the main text. With the highest value of β, there is an elevated similarity between the first members of each group (item 1, 4, and 7). There is no such apparent elevated similarity for the middle or terminal members of each group. The fact that this pattern is restricted to the highest value of β is likely due to the fact that when $\beta > .5$, the group markers dominate the context vectors over the other elements.

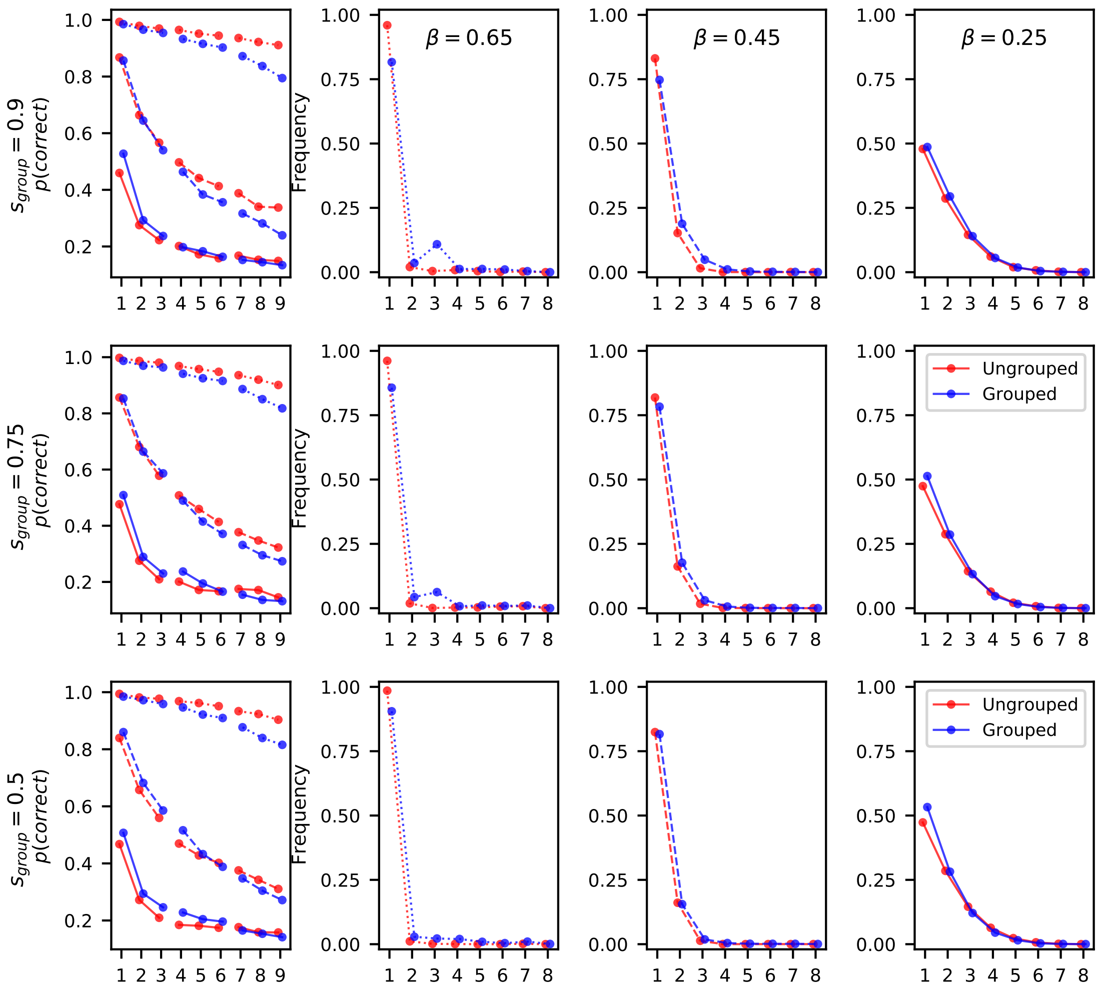

Figure 12. CRU simulations for grouped and ungrouped lists in the Hurlstone (2019) paradigm in which group markers are present in the context vectors and similarity among the group markers is manipulated and $w = .50$. The first column shows the serial position curves ($\beta = .25$: solid lines, $\beta = .45$: dashed lines, $\beta = .65$: dotted lines) while columns 2-4 show transposition gradients (separated for each value of β.)

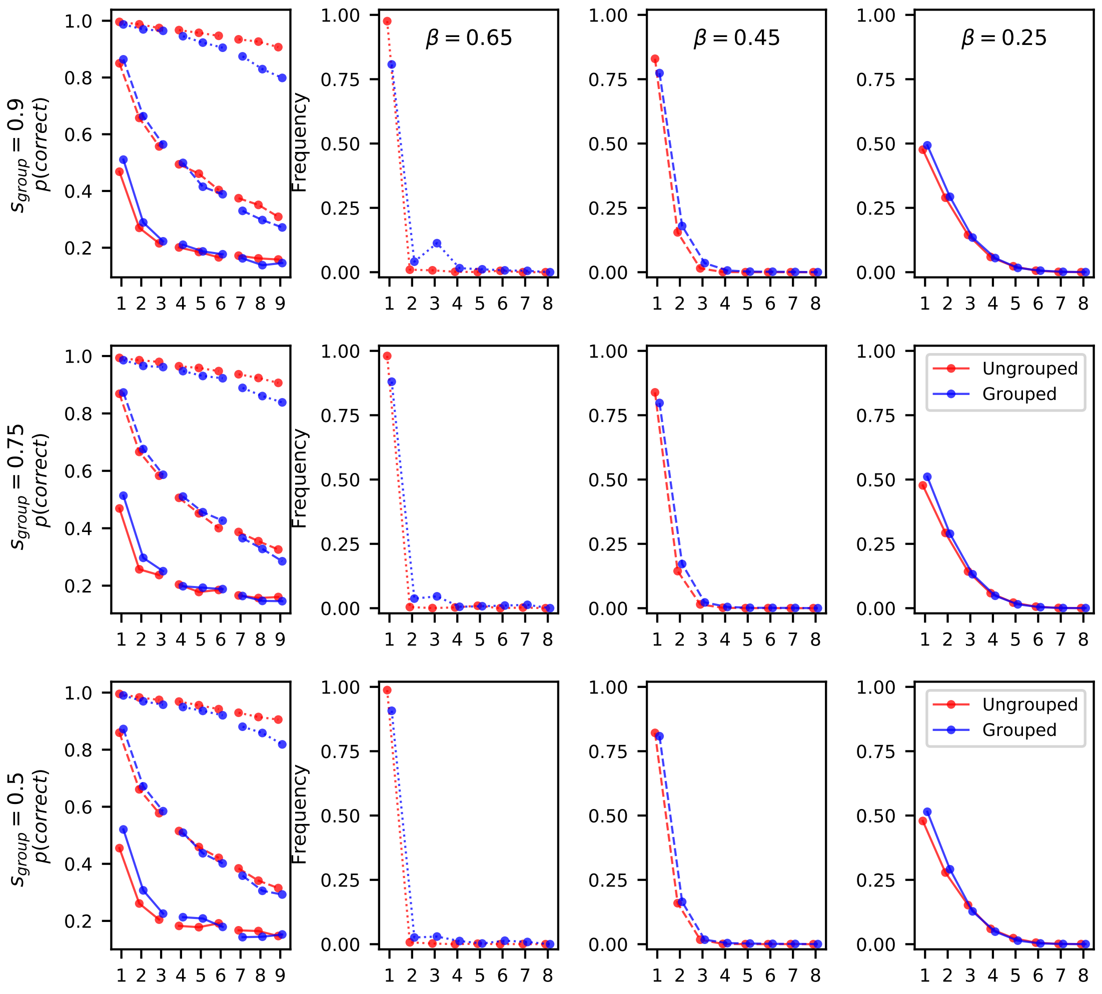

Figure 13. CRU simulations for grouped and ungrouped lists in the Hurlstone (2019) paradigm in which group markers are present in the context vectors and similarity among the group markers is manipulated and $w = .75$. The first column shows the serial position curves ($\beta = .25$: solid lines, $\beta = .45$: dashed lines, $\beta = .65$: dotted lines) while columns 2-4 show transposition gradients (separated for each value of β.)

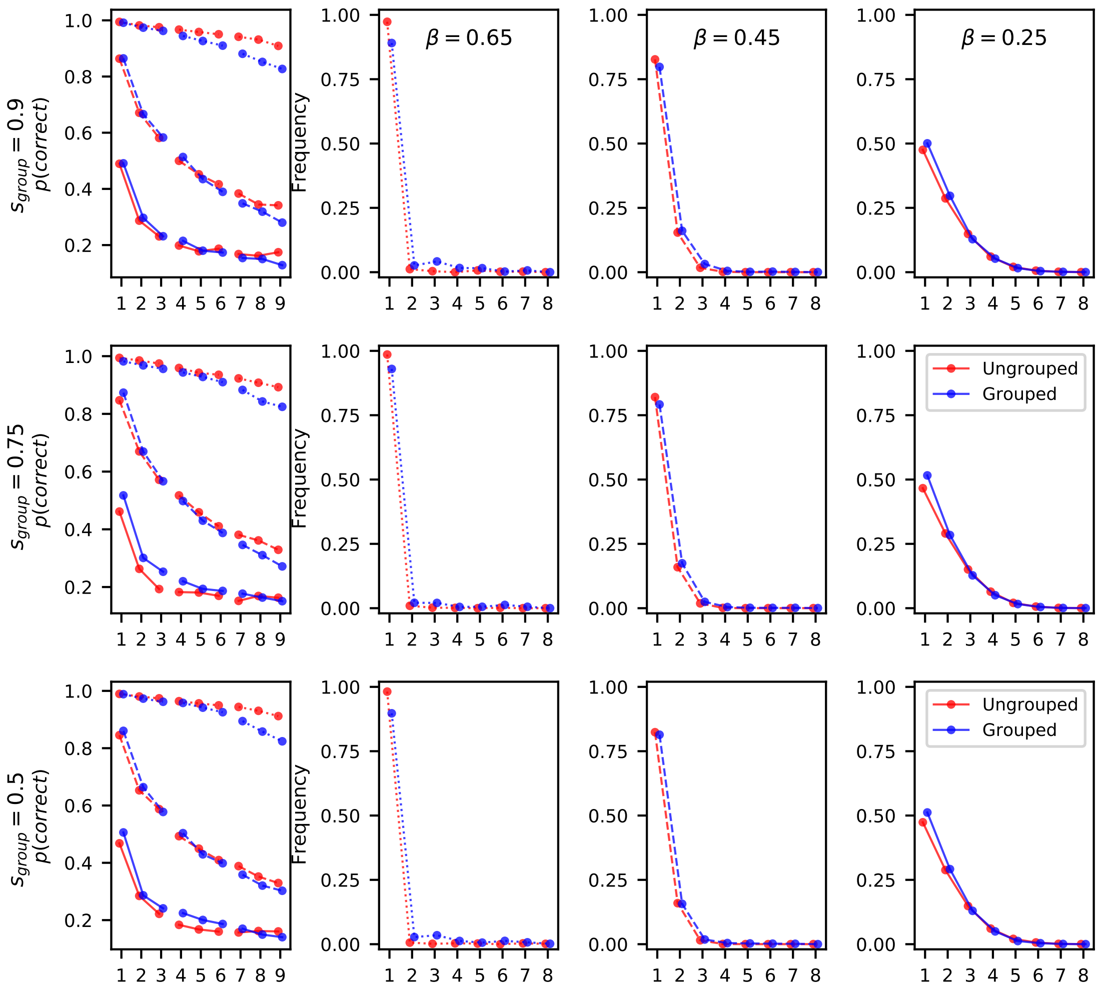

Figure 14. CRU simulations for grouped and ungrouped lists in the Hurlstone (2019) paradigm in which group markers are present in the context vectors and similarity among the group markers is manipulated and $w = .90$. The first column shows the serial position curves ($\beta = .25$: solid lines, $\beta = .45$: dashed lines, $\beta = .65$: dotted lines) while columns 2-4 show transposition gradients (separated for each value of β.)

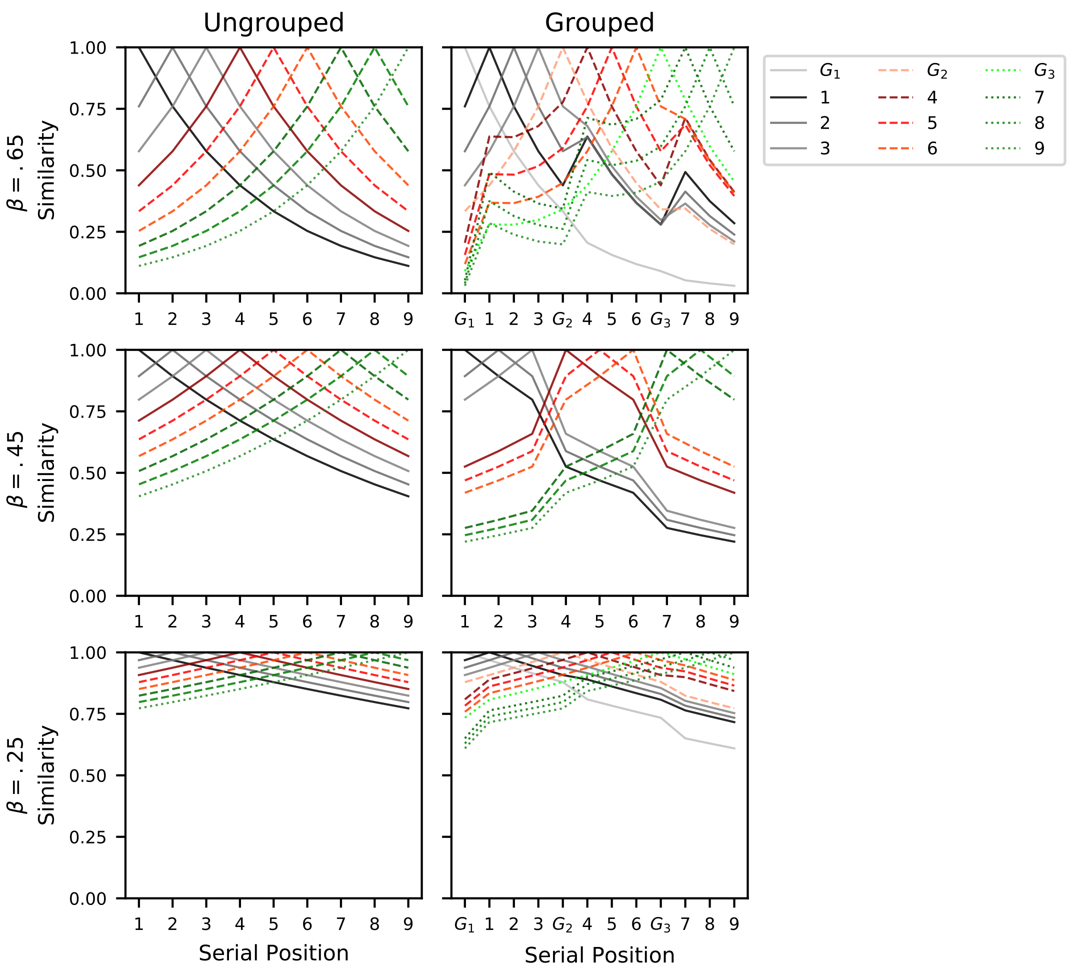

Figure 15. Dot products between all context vectors for ungrouped (left column) and grouped (right column) lists with three different values of β using group markers with high similarity ($s_{group} = .9$ and $w = .75$).

References

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

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.

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.

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

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.