Supplemental Materials for: PEPPR: A Post-Encoding Pre-Production Reinstatement Model of Dual-List Free Recall

M. Karl Healey

Michigan State University

Christopher N. Wahlheim

University of North Carolina at Greensboro

Supplemental Materials for: PEPPR: A Post-Encoding Pre-Production Reinstatement Model of Dual-List Free Recall

Model Details

There are many implementations of the Retrieved Context family of models (e.g., Healey & Kahana, 2016; Howard & Kahana, 2002; Logan & Cox, 2021; Lohnas, Polyn, & Kahana, 2015; Polyn, Norman, & Kahana, 2009; Sederberg, Howard, & Kahana, 2008) that all share the underlying mechanisms of context evolution and reinstatement. Here, we describe the implementation used in the current manuscript. Unless explicitly noted, the Post-Encoding Pre-Production Reinstatement (PEPPR) and Backward-Walk models are identical.

Model Structure

The model simulates an entire session of dual-list free recall that, following the method of the Wahlheim, Ball, and Richmond (2017) experiment, included five blocks of three trials (i.e., 15 total trials), with each block containing one trial from each of the three recall conditions (i.e., recall only from List 1, recall only from List 2, or recall from either List 1 or List 2). The presentation order of conditions was randomized within blocks. On each trial, the model was presented with two 10-item study lists.

In the model, two types of cognitive representations interact: the feature representation (F), in which the currently presented item is activated, and the context representation (C), in which the current state of context is activated. Each of these representational layers is defined as a vector. For both models, the vectors have one element for each item that will be studied during an experimental session: 15 trials each with two 10-item lists for a total of 300 item representations. Both the PEPPR and Backward-Walk models also have additional elements to represent the various events other than item presentations that occur during the experiment and are thought to cause context to drift. Both models have a "experiment start" element to represent the state of context at the beginning of the experiment. Both models also have "between-list disruption" elements to represent the change in context induced by the task switch between the end of List 1 and the beginning of List 2: one unique element for each of the 15 trials. Similarly, both models have "between-trial disruption" elements to represent the context drift associated with the end of one trial and the beginning of the next, again with one unique element for each of the 15 lists. In addition to these shared elements, the PEPPR model has two "List" elements: one to represent the instruction to study/recall List 1 and one to represent the instruction to study/recall List 2. These two List elements are "reused" on each trial, simulating the potential for prior-trial interference. Thus, in total, the feature and context vectors of the Backward-Walk model each have $300 + 1 + 15 + 15 = 331$ elements, and the vectors of the PEPPR model each have $300 + 1 + 15 + 15 + 2 = 333$ elements.

A pair of Hebbian associative matrices connect the two vector representations: features to context (M^{FC}) and context to features (M^{CF}). Each association matrix is a weighted sum of a pre-experimental component (M^{FC}_{pre} and M^{CF}_{pre}) that reflects longstanding semantic relationships and an experimental component (M^{FC}_{exp} and M^{CF}_{exp}) that reflects new learning that occurs during the experiment. Because we were not interested in studying semantic associations, we followed the practice of previous work and initialized M^{FC}_{pre} and M^{CF}_{pre} as identity matrices (i.e., items are associated only with their own context element and vise versa).

Encoding

At the beginning of the simulated experiment, the "experiment-start" feature element is fully activated as is the corresponding context element (i.e., the vectors have a 1 in that position and 0 in all others). The model then proceeds to study List 1 of the first trial. For the PEPPR model, study of List 1 entails first presenting the List 1 feature representation and then presenting the feature representations of the List 1 items one at a time. For the Backward-Walk model, which does not include list representations, study of List 1 simply entails presenting the list 1 items one at a time. During study, presenting any feature representation, i, whether it is a List representation or an item representation, activates its corresponding element on the feature vector producing a vector, $\mathbf{f}_i$, which is has a 1 at position i and 0 at all other positions. $\mathbf{f}_i$ in turn retrieves the context states to which it has previously been associated:

$$\mathbf{c}_i^{\mathbf{IN}} = \frac{M^{FC}\mathbf{f}_i}{||M^{FC}\mathbf{f}_i||}. \tag{1}$$

This retrieved context vector, $\mathbf{c}_i^{\mathbf{IN}}$, which is normalized to have a length of 1, is incorporated into the context representation by adding it to the current context vector $\mathbf{c}_{i-1}$. The context vector is continuously maintained at unit length. Therefore, when a new state of context is added to the existing state, the two vectors, $\mathbf{c}_{i-1}$ and $\mathbf{c}_i^{\mathbf{IN}}$ must be scaled so their sum has a length of one:

$$\mathbf{c}_i = \rho_i \mathbf{c}_{i-1} + \beta \mathbf{c}_i^{\mathbf{IN}}, \tag{2}$$

where β is a model parameter governing how quickly context changes, and ρ_i is chosen such that $||\mathbf{c}_i|| = 1$:

$$\rho_i = \sqrt{1 + \beta^2[(\mathbf{c}_{i-1} \cdot \mathbf{c}_i^{\mathbf{IN}})^2 - 1]} - \beta(\mathbf{c}_{i-1} \cdot \mathbf{c}_i^{\mathbf{IN}}). \tag{3}$$

Because context is always of unit length, it can be thought of as a point on the surface of a hypersphere, with β determining how far along the surface of the sphere it travels with each newly presented item and $\mathbf{c}_i^{\mathbf{IN}}$ determining the direction of travel.

The value of β used when updating context depends on the type of representation being presented. For the PEPPR model, when a list element is presented, a parameter, $\beta_{PEPPR_{encoding}}$, is used. For both models, when a list item is presented a parameter, $\beta_{encoding}$, is used.

Once all List 1 items have been studied, the "between-list disruption" element for the current trial is presented, using a different parameter, $\beta_{between_{lists}}$, to allow for different rates of context change between items versus lists. The model then begins studying List 2. For the PEPPR model, this begins by presenting the List 2 feature representation, using $\beta_{PEPPR_{encoding}}$ to update context. Both models then study the List 2 items one at a time, using $\beta_{encoding}$ when updating context.

During study, as each new list item (and for PEPPR, each List representation) is presented, new experimental associations are formed, both between the item's feature representation and the current state of context (stored in M^{FC}_{exp}) and between the current state of context and the item's feature representation (stored in M^{CF}_{exp}). These associations are formed according to a Hebbian outer-product learning rule:

$$\begin{aligned} \Delta M^{FC}_{exp} &= \mathbf{c}_{i-1}\mathbf{f}_i^{\top}, \\ \Delta M^{CF}_{exp} &= \mathbf{f}_i\mathbf{c}_{i-1}^{\top}\phi_j. \end{aligned} \tag{4}$$

Following Sederberg et al. (2008), ϕ_j simulates increased attention to beginning-of-list items by scaling the magnitude of context-to-feature associations across the list, producing a primacy effect or gradient:

$$\phi_j = \phi_s e^{-\phi_d(j-1)} + 1, \tag{5}$$

where ϕ_s and ϕ_d are model parameters and the index j is the serial position within the current list. Note that because j indexes the serial position within the *current* list (i.e., $j = 1$ for the first item of each list studied across the simulated experiment), the primacy gradient "resets" at the beginning of each list within each trial. For simplicity, the PEPPR version of the model uses the value of $j = 1$ twice per list: once when presenting the List element and again when presenting the actual first item in the list.

Newly formed experimental associations are incorporated with pre-experimental associations. The balance between new and existing associations is controlled by the parameters γ_{FC} and γ_{CF}:

$$\begin{aligned} M^{FC} &= (1-\gamma_{FC})M^{FC}_{pre} + \gamma_{FC}M^{FC}_{exp}, \\ M^{CF} &= (1-\gamma_{CF})M^{CF}_{pre} + \gamma_{CF}M^{CF}_{exp}. \end{aligned} \tag{6}$$

Recall

Pre-Production Reinstatement. For the PEPPR model, recall begins by re-presenting the appropriate List representation. For "recall only from List 1" trials, the List 1 feature representation is represented. For "recall only from List 2" trials, the List 2 representation is represented. For "recall from either List 1 or List 2" trials, we found that the model could fit the summary data in Figure 2 of the main text quite well whether we represented both the List 1 *and* List 2 representation or simply represented just the List 2 representation. Therefore, we choose to present just the List 2 representation to reduce the computational demands of the later simulations of detailed recall dynamics. In all conditions, the presented list element is activated on the feature layer, which, via Equation 1, activates a state of context that we can call $\mathbf{c}^{\mathbf{IN}}_{PEPPR}$. Because the List element had been active when the items of that list were studied, learning has occurred and $\mathbf{c}^{\mathbf{IN}}_{PEPPR}$ will include activation not only of the context representation of the List element but potentially also activation of the context representations of items from the current-trial target list and possibly items from the corresponding lists of prior-trials (i.e., potential intrusions). $\mathbf{c}^{\mathbf{IN}}_{PEPPR}$ is then incorporated with the prior context, $\mathbf{c}_t$ (i.e., the context at the end of the study phase) via Equation 2 using the parameter $\beta_{PEPPR_{reinstatement}}$ to determine the balance between $\mathbf{c}^{\mathbf{IN}}_{PEPPR}$ and $\mathbf{c}_t$. The updated context is then used as the cue for the first retrieval attempt as described below. Higher values of $\beta_{PEPPR_{reinstatement}}$ mean the updated context will provide a stronger cue for items from the target list.

For the Backward-Walk model, the List representation is not re-presented, and context is not updated. Instead, $\mathbf{c}_t$ (i.e., the context at the end of the study phase) is used as the cue for the first retrieval attempt.

Sampling. As described in the main text, the model first samples candidate outputs and then screens them for possible intrusions. In the past, CMR has used a set of competitive leaky accumulators (Healey & Kahana, 2016; Lohnas et al., 2015; Polyn et al., 2009) to simulate the sampling process. Here, we follow Morton and Polyn (2016) and use a simple Luce choice rule to reduce computational demands (see also, Gibson, Healey, & Gondoli, 2019; Mundorf, Lazarus, Uitvlugt, & Healey, 2021). Specifically, the sampling process proceeds as a series of sampling attempts, closely following the implementation used by Mundorf et al. (2021). At each attempt, the model either successfully retrieves an item or fails. After a failure, no further sampling attempts are made. The probability of stopping recall (i.e., failing to sample an item) starts out low for the first recall attempt (i.e., output position) and increases exponentially with each output position:

$$P(stop, j) = \theta_s e^{j\theta_r}, \tag{7}$$

where j is the output position, θ_s is a parameter which determines the scaling of the exponential function, and θ_r is a parameter which controls the rate at which the probability of stopping approaches 1.

If recall does not stop at a particular sampling attempt, the current contextual state is used to cue retrieval via the M^{CF} associations:

$$\mathbf{a} = M^{CF}\mathbf{c}_t, \tag{8}$$

where $\mathbf{c}_t$ is the state of context at the start of the recall period. As described above, the key difference between the PEPPR and Backward-Walk models is whether the appropriate List representation was incorporated into $\mathbf{c}_t$ via Pre-Production Reinstatement.

In either case, the resulting vector, $\mathbf{a}$, gives the degree of support, or activation, for each of the 300 list items in the model's memory. For simplicity, we remove any items from $\mathbf{a}$ that have not yet been presented, as they will have zero activation. We also exclude from $\mathbf{a}$ any items that have already been recalled to avoid the complexity of modeling inhibition of repetitions (but see Lohnas et al., 2015, for how such inhibition can be modeled in CMR2). These exclusions result in a vector $\mathbf{a'}$ of items that are available for recall, which is then used to assign each recall-able item a probability of being sampled according to:

$$P(i) = (1 - P(stop)) \frac{\mathbf{a'}_i^{\tau}}{\sum_k^N \mathbf{a'}_k^{\tau}}, \tag{9}$$

where τ is a parameter that determines how sensitive the model is to differences among items in level of support—when τ is large, the model strongly prefers the item with the highest activation on $\mathbf{a'}$; when it is small, less well-supported items have a greater chance of wining. Note that in computing $P(i)$, each element of $\mathbf{a'}$ is set to a minimum value of 10^{-7} to ensure no item is assigned a zero sampling probability.

Editing Process. Before the sampled item is actually output by the model, it undergoes a post-production editing phase, consistent with the observation that people often report thinking of items that they do not overtly recall during free recall experiments (Keppel, 1968; Wixted & Rohrer, 1994). In the first step of the editing process, the sampled item's representation is reactivated on F, allowing the model to retrieve the contextual state associated with the item via Equation 1. Next, the model compares the context representation retrieved by the candidate item, $\mathbf{c^{IN}}$, with the currently active context representation, $\mathbf{c}_t$, by taking their dot product:

$$u = \mathbf{c^{IN}} \cdot \mathbf{c}_t. \tag{10}$$

The value of u is then used to make a decision about whether the sampled item is from the target list. How this decision is made depends on which list is being targeted and the model (PEPPR versus Backward-Walk). For PEPPR, the context of the targeted list has already been reinstated, so if u is greater than a threshold, u_{min}, the model assumes the sampled item is from the target list and outputs the item. If instead $u < u_{min}$, the model rejects the item and does not output it.

When targeting the most recent list (List 2) or both lists, the Backward-Walk model uses the same rejection rule as described for PEPPR in the previous paragraph. However, when the Backward-Walk model is targeting the non-recent List 1, it uses two thresholds to distinguish between items that are too recent (likely intrusions from List 2) and items that are not recent enough (likely intrusions from prior trials). That is, if $u > u_{max}$, the model decides the item is a List 2 intrusion and rejects it. If $u < u_{min}$, the model decides it is an intrusion from an earlier trial and rejects it. If instead, $u_{min} < u < u_{max}$, the sampled item is accepted. Once a sampled item is accepted, the Backward-Walk model assumes it has accessed the target list and switches back to the standard rejection rule for subsequent sampling attempts.

For all models and conditions, once an item is sampled and the editing decision has been made, context is updated by combining $\mathbf{c}^{\mathbf{IN}}$ and $\mathbf{c}_t$ via Equation 2 with the balance between the two determined by a parameter β_{recall}. That is, even rejected items are allowed to drive context change. This is critical to the Backward-Walk model's ability to iteratively sample progressively less recent items, as the updated context is then used as a cue for the next recall. The cycle of cuing with context, sampling items, editing for intrusions, updating context, and cuing again continues until the model fails to sample an item (Equation 7).

Once the recall period for one trial ends, the "between-trial disruption" element for the current trial is presented using the parameter $\beta_{between_{trials}}$ to update context, and the next trial begins, following the same study/recall equations described above. It is important to note that because the model simulates an entire session of 15 trials, associations formed in prior trials can interfere during recall on the current trial. In particular, for the PEPPR model, as the experiment progresses, more and more prior-trial items become associated with the List 1 and List 2 representations.

Fitting Algorithm

We fit the models by minimizing the root-mean-square deviation (RMSD) difference between the model predictions and data using a differential evolution genetic algorithm (Storn & Price, 1997) as implemented in SciPy (Virtanen et al., 2020). The cross-over probability was set to 0.9 for all generations, and at the beginning of each generation, the differential weight was set to a random value between 0.5 and 1.75. Given that we argued in the main text that the Backward-Walk model was not capable of fitting the data, it was critical that we explored the parameter space of each model extensively enough to find a good-fitting parameter set, if one exists. However, due to the large number of parameters in each model (24 for PEPPR and 23 for Backward-Walk), the spaces are quite large and exploring them exhaustively is computationally prohibitive. Therefore, we designed an algorithm to maximize exploration of the space with the available resources. Specifically, we applied the genetic algorithm iteratively to first provide a very broad search of the parameter space to identify local minima and then to provide a more fined grained search to attempt to find the global minimum. In the first iteration, the algorithm started with an initial generation of 256 parameter sets evenly distributed across the parameter space using the Sobol method. It then was allowed to run further generations of 256 parameter sets for 7 days, with each generation running its sets in parallel on 128 processing cores. For each parameter set, 60 subjects were simulated (i.e., twice the number of actual participants in the Wahlheim et al., 2017, experiment). The best-fitting 128 parameter sets from this initial 7-day run were then used as the starting population for a second 7-day run in which each generation included 128 parameter sets, again run in parallel on 128 processing cores, with 120 subjects being simulated for each parameter set to reduce the error variance in estimating the parameter set's predictions so as to better distinguish among well-fitting parameter sets. We then used the best-fitting parameter set from this final run to produce the simulated data presented in the main text. This entire algorithm was applied 4 times: once for each model to fit the summary data in Figure 2 of the main text and once for each model to fit the recall dynamics data in Figure 6. Consequently, the simulations took 8 weeks to run on 128 processors, which is equivalent to 19.96 years of processing time—approximately 5 years per fit. The algorithm allowed us to find excellent fits to the Figure 2 data for both the PEPPR and Backward-Walk models, as well as an excellent fit to the Figure 6 data for the PEPPR model. However, it was unable to find a good-fitting parameter set for the Backward-Walk model to the Figure 6 data. While it is impossible to rule out the possibility that a good-fitting parameter set exists for the backward-walk model, it seems quite unlikely.

Fitted Parameters

Table 1 provides the best-fitting parameter values used to generate the simulated data reported in the main text. The values here are rounded to 3 decimal places. The actual values with full precision are available in the OSF project for the paper (see the main text for a link). The columns labeled "Figure 2" contain the best-fitting parameter values for each model when fitting the overall recall level data in Figure 2C from the main text. These values were used to generate simulated data for Figures 2A, 2B, 5A, and 5B. The columns labeled "Figure 6" contain the best-fitting parameter values for each model when fitting Figure 6C and F from the main text. These values were used to generate simulated data for the first and second columns of Figures 6 and 7. In all cases, the model was constrained to use a common set of encoding parameters for all conditions. However, retrieval parameters, with the exception of $\beta_{PEPPR_{reinstatement}}$, were allowed to differ among conditions. $\beta_{PEPPR_{reinstatement}}$ was constrained to be the same for all conditions to avoid giving the PEPPR model excessive flexibility. In other words, we assumed that the reinstatement of list context is equally effective regardless of which list is being targeted.

References

Gibson, B. S., Healey, M. K., & Gondoli, D. M. (2019). ADHD reflects impaired externally directed and enhanced internally directed attention in the immediate free-recall task. *Journal of Abnormal Psychology*, *128*(2), 173–183. doi: 10.1037/abn0000393

Healey, M. K., & Kahana, M. J. (2016). A four-component model of age-related memory change. *Psychological Review*, *123*(1), 23–69. doi: 10.1037/rev0000015

Howard, M. W., & Kahana, M. J. (2002). When does semantic similarity help episodic retrieval? *Journal of Memory and Language*, *46*(3), 85–98. doi: 10.1006/jmla.2001.2798

Keppel, G. (1968). Verbal behavior and general behavior theory. In T. R. Dixon & D. L. Horton (Eds.), (pp. 172–213). Prentice-Hall.

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

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

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

Mundorf, A. M. D., Lazarus, L. T. T., Uitvlugt, M. G., & Healey, M. K. (2021). A test of retrieved context theory: Dynamics of recall after incidental encoding. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *47*(8), 1264–1287. doi: 10.1037/xlm0001001

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. doi: 10.1037/a0014420

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), 893–812. doi: 10.1037/a0013396

Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. doi: 10.1023/A:1008202821328

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., . . . SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*(3), 261–272. doi: 10.1038/s41592-019-0686-2

Wahlheim, C. N., Ball, B. H., & Richmond, L. L. (2017). Adult age differences in production and monitoring in dual-list free recall. *Psychology and Aging*, *32*(4), 338–353. doi: 10.1037/pag0000165

Wixted, J. T., & Rohrer, D. (1994). Analyzing the dynamics of free recall: An integrative review of the empirical literature. *Psychonomic Bulletin & Review*, *1*, 89–106. doi: 10.3758/BF03200763

Table 1
The best-fitting parameter values for each model when fitted to the data in Figures 2 and 6.

	Figure 2		Figure 6	
Parameter	Backward-Walk	PEPPR	Backward-Walk	PEPPR
Encoding Parameters Common to All Test List Conditions				
ϕ_s	6.200	44.312	6.492	0.005
ϕ_d	47.066	0.025	35.474	32.366
γ_{fc}	0.735	0.877	0.473	0.939
γ_{cf}	0.077	0.408	0.687	0.985
$\beta_{encoding}$	0.794	0.596	0.834	0.199
$\beta_{between_{lists}}$	0.635	0.234	0.285	0.621
$\beta_{between_{trials}}$	0.692	0.949	0.851	0.997
$\beta_{PEPPR_{encoding}}$	– †	0.432	– †	0.736
$\beta_{PEPPR_{reinstatement}}$	– †	0.947	– †	0.851
Retrieval Parameters for Recall List 1 Condition				
θ_s	0.014	0.003	0.001	0.002
θ_r	0.013	0.377	0.107	0.291
τ	11.375	57.561	91.664	82.095
β_{recall}	0.571	0.665	0.761	0.048
u_{min}	0.881	0.651	0.824	0.894
u_{max}	0.725	– †	0.332	– †
Retrieval Parameters for Recall List 2 Condition				
θ_s	0.027	0.026	0.245	0.331
θ_r	0.145	0.148	0.704	0.416
τ	15.031	77.106	68.465	86.837
β_{recall}	0.859	0.567	0.787	0.323
u_{min}	0.311	0.121	0.297	0.760
Retrieval Parameters for Recall Both List Condition				
θ_s	0.005	0.007	0.233	0.633
θ_r	0.056	0.011	0.358	0.786
τ	0.765	10.253	33.537	29.186
β_{recall}	0.908	0.813	0.841	0.409
u_{min}	0.589	0.618	0.149	0.202

† As discussed in the text, this parameter is used by one model and not the other.