Bridging CRU and CMR in Free Recall: A Factorial Comparison of Retrieved-Context Models

Jordan B Gunn and Sean M Polyn

Department of Psychology, Vanderbilt University

Author Note

Abstract

Retrieved-context theory posits that episodic retrieval is driven by a representation that evolves over time, tying each item to the contextual features present during encoding and later accessing those associations to guide retrieval (Howard & Kahana, 2002). Although the Context Maintenance and Retrieval (CMR) model (Polyn et al., 2009) offers a flexible and well-tested retrieved-context implementation – addressing both free and serial recall (Lohnas, 2024) – it is relatively complex, incorporating mechanisms absent in some other retrieved-context models. In contrast, the Context Retrieval and Updating (CRU) model (Logan, 2018, 2021) provides a simpler, more streamlined specification of context-driven retrieval shown to excel in strictly ordered memory tasks like serial recall. However, it remains unclear whether CRU's leaner architecture extends easily to unconstrained retrieval dynamics in free recall. To investigate the gap between CRU and CMR, we systematically compare them, presenting them side by side and exploring how each can be viewed as a parameterized variant of the same foundational ideas. Using a factorial model selection approach, we selectively incorporate CMR-like features into CRU and compare each hybrid variant to standard CMR on free recall data. We find that selectively incorporating CMR-like features substantially improves CRU's fit to free recall, showing that CRU can be extended to meet many of CMR's capabilities. We highlight how CRU and CMR lie along a continuum of retrieved-context approaches, clarifying how context guides memory search across task domains.

Keywords: episodic memory, free recall, serial recall, retrieved-context theory, computational modeling

Bridging CRU and CMR in Free Recall: A Factorial Comparison of Retrieved-Context Models

Researchers have long recognized that episodic memory does not involve merely storing items but also associating them with the temporal context in which they occur. Early perspectives on temporal context (Estes, 1955; Melton & Martin, 1972; Tulving & Madigan, 1970) proposed that a slowly drifting set of features provides a unique background signal at each point in time, helping the mind keep track of when certain items were experienced. Retrieved-context theory (Howard & Kahana, 1999, 2002) extends these ideas further, suggesting that when an item is recalled, it reactivates the contextual state that was present during its original encoding, thus cueing neighboring items from the same temporal window. Because this context representation reflects a gradually shifting set of features, reactivating a given item's context can bring to mind items from adjacent positions.

The Temporal Context Model (TCM; Howard & Kahana, 2002) crystallized these ideas into a formal model, simulating how a drifting context vector could explain robust patterns in free recall such as the recency effect (in which items from the end of a list are often recalled first) and the temporal contiguity effect (in which items studied near one another in time tend to be recalled successively). Subsequent work showed that TCM could be enriched to capture additional empirical findings, giving rise to a range of models that extend the basic retrieved-context principles to account for a variety of memory phenomena (Howard & Kahana, 2002; Kahana, 2020; Lohnas et al., 2015; Sederberg et al., 2008). Among these extensions, the Context Maintenance and Retrieval (CMR) model (Morton & Polyn, 2016; Polyn et al., 2009) built upon TCM's evolving context representation by adding flexible associative mechanisms, including the incorporation of pre-experimental structures and semantic relationships. The model has been applied to tasks ranging from free recall to recognition memory and has explained a broad array of behavioral and neural data (Healey & Kahana, 2016; Horwath et al., 2023; Kragel et al., 2015; Morton & Polyn, 2016; e.g., Polyn et al., 2009; Sederberg et al., 2011).

Against this background, Gordon Logan has extended the retrieved-context perspective to explain how automatic serial order behavior can emerge in tasks like typing or immediate serial recall (Logan, 2018, 2021). Logan's work on automaticity (Logan, 1988) and hierarchical control underscores that once a top-level goal (for example, "type DOG") is set, a series of lower-level actions can be generated with minimal conscious oversight – an idea that is realized in the Context Retrieval and Updating (CRU) model. CRU assumes that each keystroke or list item is associated with a continuously updated context state and that retrieving one item integrates that item into context, thereby cueing retrieval of the next. After initially demonstrating CRU's potential as an account of automatic typing (Logan, 2018), CRU has been applied to capture a wide range of serial recall phenomena, including transposition gradients, serial position effects, and error patterns (Logan, 2021), with even further work clarifying its contributions against models focused on positional representations of serial order (Logan, 2021; Logan & Cox, 2023; Osth & Hurlstone, 2023). Importantly, CRU highlights item identification mechanisms that handle confusable stimuli (e.g., visually similar letters) and adopts a streamlined approach to context updating. By design, it aspires to offer a general retrieved-context account of sequential performance in domains like typing, musical performance, and language production.

Although CMR was designed primarily for free recall, recent work by Lohnas (2024) has shown that a variant called sCMR can also capture key features of serial recall, hinting at a unified framework spanning both free and serial tasks. That analysis highlights assumptions in CRU that may limit its ability to address free recall, such as its lack of pre-experimental context-to-feature associations and its simplified context updating scheme. Although CRU was proposed as a general model of sequence retrieval, it has so far been tested mainly in serial recall tasks with visually confusable stimuli, leaving open the question of whether its streamlined architecture can address performance in free-recall tasks where participants can recall items in any order. We aim to address this gap by systematically comparing CRU and CMR, exploring how each model can be viewed as a parameterized variant of the same foundational ideas, and clarify whether CRU's simpler approach remains domain-bound or can be extended to capture free recall phenomena traditionally captured by CMR.

In our approach, we analyze CRU and CMR as two ends of a parameterized spectrum with broadly overlapping assumptions. Both models embrace the notion that retrieving an item reactivates its associated context, but they differ in how they handle item confusions, associative structures, recall initiation and termination, and differences in encoding across study positions. We treat these differences as modular components that can be selectively enabled in a factorial manner, creating a family of hybrid variants that systematically isolates the impact of each mechanism. We then fit a broad selection of these variants to empirical data from word-based free recall (Healey & Kahana, 2014), testing how each mechanism contributes to capturing primacy and recency components of the serial position effect, temporal contiguity effects, and patterns of recall termination. By identifying which model-specific assumptions most enhance performance in free recall, we aim to clarify how CRU and CMR can be integrated into a more unified account of episodic memory search, highlighting the shared principles that underlie both models and the unique mechanisms that each brings to the table.

Model Structure

Overall, CRU is a streamlined version of CMR with fewer parameters and a simpler architecture for updating context, though it adds some processing steps absent in CMR. CMR employs dual feature-to-context and context-to-feature linear associative memory matrices to mediate the evolution of context and context-driven retrieval, respectively. By contrast, CRU pairs items directly with context states in an instance-based memory and omits an explicit feature-to-context memory, relying on a simplified context updating scheme suitable for serial recall. However, CRU also includes specialized mechanisms absent from CMR, such as item identification for visually confusable letters (e.g., *p* vs. *q*), to address tasks with high confusion rates.

We outline each model's key features in turn, focusing on how they (i) represent and encode items, (ii) initialize and evolve context, (iii) store item–context associations, and (iv) handle recall—especially stopping rules and transitions. Throughout, we note convergences and divergences, emphasizing that CRU and CMR share a family resemblance in how context guides

Table 1

Parameters and structures specifying CRU and CMR.

Symbol	Description	Applicability across models
C	Temporal context layer	Equivalent in both models
F	Item feature layer	Equivalent in both models
M^{FC}	Item-to-context associations	Omitted (or $\gamma = 0$) in CRU
M^{CF}	Context-to-feature associations	Similar in both models
β_{enc}	Encoding context integration rate	Equivalent in both models
β_{dec}	Encoding integration decay	Omitted (or $\beta_{dec} = 1$) in CMR
β_{rec}	Recall context integration rate	Omitted (or $\beta_{rec} = \beta_{enc}$) in CRU
β_{start}	Start context integration rate	Omitted (or $\beta_{start} = 1$) in CRU
α	Shared support	Omitted (or $\alpha = 0$) in CRU
δ	Pre-experimental context-to-item self-support	Omitted (or $\delta = 0$) in CRU
γ	Item-to-context learning rate	Omitted (or $\gamma = 0$) in CRU
ϕ_s	Primacy scale	Omitted (or $\phi_s = 1$) in CRU
ϕ_d	Primacy decay	Omitted (or $\phi_d = 0$) in CRU
g	Identification sensitivity	Omitted (or $g = \infty$) in CMR
g_{dec}	Identification sensitivity decay	Omitted (or $g = 1$) in CMR
τ	Choice sensitivity	Omitted (or $\tau = 1$) in CRU
θ_r	Stop probability growth	Specific to CMR
θ_s	Stop probability scale	Specific to CMR

retrieval, while still offering distinct approaches to certain subtasks. These differences may at times be interpreted as competing explanations of the same phenomena or as complementary extensions useful in different domains of memory research. In later sections, we examine whether selectively adding certain “missing” CMR-like mechanisms (e.g., dynamic feature-to-context learning or pre-experimental associations) can help CRU capture free recall’s hallmark backward transitions and flexible stopping.

Item Identification in CRU

CRU and CMR each represent items as orthonormal unit vectors. However, CRU can optionally model confusability among similar items via a probabilistic competition step at encoding. When item i is presented, a racing diffusion with drift rates scaled by g determines which representation crosses the threshold first:

$$v_i = \exp\left[-g \cdot d_{ij}\right].$$

Here, v_i is the drift rate for item i, d_{ij} is the distance between the presented item and each candidate j, and g is a sensitivity parameter. Lower g fosters more confusion among similar items, whereas higher g makes identification more accurate.

Because word-based free recall typically involves distinct stimuli that are rarely confused, we disable the item identification mechanism in our CRU free recall fits (i.e., set $g \to \infty$). By contrast, CMR has historically assumed perfect item recognition for tasks like word recall and uses no item identification mechanism.

Context Initialization and Evolution: Equivalent Between Models

Both CMR and CRU represent context as a vector of continuous values. They initialize this vector by setting one dimension to 1.0 and the rest to 0.0. In both models, this initial state represents the start-of-list context and can be reinstated to organize retrieval.

This context vector evolves as items are encoded and retrieved, integrating a new contextual input at each step. At each step i, both models update context as

$$c_i = \rho_i c_{i-1} + \beta\, c_i^{IN},$$

where β controls integration of new input c_i^{IN}, and ρ_i normalizes the vector:

$$\rho_i = \sqrt{1 + \beta^2\left[(c_{i-1} \cdot c_i^{IN})^2 - 1\right]}\; - \;\beta(c_{i-1} \cdot c_i^{IN}).$$

This gradual integration yields a recency-based gradient reflecting the order in which items were presented.

Contextual Input: CRU Omits Feature-To-Context Learning

A key difference is how each model obtains c_i^{IN} from the active item. CRU simply treats the one-hot (orthonormal) item representation as c_i^{IN}. Since each unique item representation is orthogonal to all others, this direct assignment ensures that the context vector is updated in a way that is unique to the item being studied or recalled. By contrast, CMR uses a feature-to-context memory M^{FC} that learns experimental associations between items and context states when items are presented.

Like CRU, CMR's M^{FC} effectively initializes by associating each item representation with a unique context unit such that when an item is presented, the corresponding context unit is activated and integrated into the context vector as contextual input. But unlike CRU, in CMR a learning rate parameter γ configures the relative influence of pre-experimental and experimental associations in this memory. Pre-experimental associations capture these baseline or default item–to-context connections according to:

$$M^{FC}_{pre(ij)} = \begin{cases} 1 - \gamma, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases}$$

Thus, M^{FC}_{pre} resembles a partial identity matrix, ensuring that each item i is initially tied to a corresponding context unit i with weight $1 - \gamma$. Experimental associations are then formed during encoding whenever an item is presented, based on the simultaneous activity of item features F_i and context C_j, where γ works as the learning rate:

$$\Delta M^{FC}_{ij} = \gamma F_i C_j$$

This update rule is a Hebbian outer product, capturing the principle that co-active item features and context states become associated. Accordingly, M^{FC} serves as a matrix of Hebbian associations that stores both the baseline connections (scaled by $1 - \gamma$) and any newly formed links (scaled by γ).

Because $1 - \gamma$ scales the pre-experimental connections while γ scales new learning, retrieving an item leads to reinstatement of the context associated with that item when it was originally presented, retrieving contextual information about preceding items. However, this reinstatement only matters once the item is actually recalled, at which point the stored context in M^{FC} can prime backward transitions in free recall. By contrast, CRU's direct assignment of c_i^{IN} corresponds to CMR with a learning rate $\gamma = 0$. This causes encoding or retrieval of an item to always integrate an orthogonalized context vector unique to that item, eliminating a source of support for backward transitions in sequence recall.

The absence of a feature-to-context memory in CRU (vs CMR) reflects how the temporal contiguity effect – the tendency to transition between items that were presented close together in time (Kahana, 1996, 2020) – manifests differently in serial and free recall tasks. Since participants in serial recall are typically asked to reproduce the studied list in order, the temporal contiguity effect is almost entirely forward-going, with participants transitioning to the next item in the list. But since free recall participants are free to recall items in any order, the temporal contiguity effect is bidirectional, with participants often transitioning to the item that was presented immediately before the retrieved item. A dynamic feature-to-context memory like CMR's may be more effective in capturing this bidirectional effect, while CRU's simpler approach may suffice for serial recall.

Context-to-Item Associations: CRU Omits a Pre-Experimental Structure

CMR uses a context-to-feature matrix M^{CF}, seeded with pre-experimental associations via parameters δ (self-support) and α (shared support):

$$M^{CF}_{pre(ij)} = \begin{cases} \delta, & \text{if } i = j \\ \alpha, & \text{if } i \neq j \end{cases}$$

Here,δ works like γ in M^{FC} to scale the pre-experimental context-to-feature associations compared to the experimental associations acquired during the experiment. By contrast, the parameter α approximates uniform support across all items. When a context unit is activated, any nonzero α provides partial activation to all items, providing a coarse way to capture semantic clustering effects without simulating the semantic identity of individual study items (Polyn et al., 2009). More elaborate versions of CMR can replace this uniform structure with richer semantic representations (Morton & Polyn, 2016).

CRU, by contrast, lacks explicit pre-experimental context-to-item associations, effectively setting $\delta = 0$ and $\alpha = 0$. In CMR, driving up δ while γ is 0 tends to exclusively heighten low-lag forward transitions in free recall, similar to how raising γ (the feature-to-context learning rate) promotes backward transitions. When γ is freed, δ's role is broader, tuning the consistency of short-lag transitions in free recall compared to long-lag, wide-ranging transitions. Because most serial recall tasks emphasize forward order and rarely show backward transitions, CRU omits pre-experimental support in keeping with its streamlined approach.

It is worth noting that TCM's original formulation (Howard & Kahana, 2002) also did not include a pre-experimental context-to-feature matrix, focusing instead on how a drifting context signal promotes both recent and temporally adjacent items during recall. Subsequent extensions – e.g., TCM-A (Sederberg et al., 2008) – introduced additional parameters and retrieval decision rules to address short- vs. long-term recency, whereas CMR (Morton & Polyn, 2016; Polyn et al., 2009) incorporated explicit pre-experimental structures like δ and α to capture semantic clustering and backward transitions more effectively.

During study, CMR updates M^{CF} via a Hebbian rule, whereas CRU encodes item–context pairs as separate instances. Despite these architectural differences, the two approaches can sometimes exhibit functionally similar dynamics (J. R. Anderson & Bower, 1972; Turner, 2019). For example, a linear associator (as in CMR) can approximate an instance-based model (as in CRU) when the input patterns are sufficiently distinct, effectively storing and retrieving each studied episode with minimal interference. We do not explore these equivalences in detail here, but we treat both implementations of context-to-item associations as serving a broadly similar role: binding items to context so that recalling one item can cue related items. Evaluating repeated items or semantic domains where these distinctions become more critical lies outside the scope of this paper, but remains a compelling direction for future research.

Serial Position Effects

Both serial and free recall tasks exhibit primacy and recency effects, where participants exhibit better memory for items from the beginning and end of study lists, respectively (Murdock, 1962). A separate but related observation is that participants often initiate recall with either the first or the last item in a list. CRU and CMR handle these primacy and recency effects differently, reflecting differences in their theoretical emphases and the tasks they are designed to address.

In CMR, two parameters ϕ_s and ϕ_d configure the learning rate of its context-to-feature memory M^{CF} to enforce a primacy gradient, scaling the amount of learning based on the serial position of the studied item according to:

$$\phi_i = \phi_s e^{-\phi_d (i-1)} + 1$$

Here, ϕ_i is the learning rate for the i-th item, ϕ_s is the learning rate at the first serial position, ϕ_d is the decay rate, and i is the serial position of the studied item. The strengths of M^{CF} associations is thus updated according to:

$$\Delta M_{ij}^{CF} = \phi_i C_j F_i$$

By contrast, CRU modulates its sensitivity parameter g and context-integration rate β by serial position. This is done by setting a maximum value for each parameter and a decay rate that scales the maximum value according to the serial position of the studied item such that the integration rate and sensitivity at the i-th serial position are:

$$g_i = g_{max} \cdot g_{dec}^{i-1}$$

$$\beta_i = \beta_{max} \cdot \beta_{dec}^{i-1}$$

Here, g_{max} and β_{max} are the initial values of g and β, and g_{dec} and β_{dec} are the decay rates. The equations for identifying items and updating context are left unchanged, except using the position-specific values of these parameters.

Along with modulating memorability as a function of serial position, the CRU and CMR models also address serial position effects by configuring the state of context when retrieval begins. In CMR, the final state of context at the end of encoding is integrated toward the start-of-list contextual state according to a special integration rate parameter β_{start} before initiating retrieval. When this integration rate is zero, the contextual cue primarily targets items near the end of the study list, and when it is high, the cue targets items near the beginning of the study list. Hence CMR can flexibly shift recall initiation to either early or late list items, whereas CRU always begins from the start-of-list context.

This mechanism is not present in CRU. Instead, retrieval always begins with the start-of-list contextual state, ensuring that initial items in the study list are always most accessible. This is equivalent to fixing $\beta_{start} = 1$ in CMR. The differences in how CRU and CMR account for serial position effects in memory search in part reflect differing emphases in the models. CMR was primarily designed to address free recall, where both primacy and recency effects are prominent (Murdock, 1962), motivating a β_{start} parameter that can be adjusted to target different parts of the study list. By contrast, in serial recall tasks, primacy effects are more salient than recency effects, and the start-of-list contextual state is always the most relevant cue for retrieval. CRU's simpler approach to context initialization and retrieval reflects this emphasis on serial recall, where the goal is to reproduce the studied list in order.

Recall and Recall Termination

In either model, retrieval attempts compare the current context to all stored contexts in the context-to-feature memory, prioritizing items associated with contexts that are most similar to the current context. CRU performs this comparison directly to each context stored in its instance-based memory using a dot product. CMR's context-to-feature memory performs this comparison by passing the current context through its context-to-feature memory M^{CF} to activate item representations. In either model, this context-to-item comparison sets up a probabilistic competition between items for retrieval where items that are more strongly associated with the current context are more likely to be retrieved. Both models also allow the possibility of terminating recall without retrieving any more items, though they differ in how they factor this possibility into the competition between items. In either model, once a retrieval attempt is complete, the item is reported, the context is updated based on the retrieved item, and the process repeats until recall is terminated. The most complete specification of CRU also includes a mechanism where retrieved items can be incorrectly reported due to typos or other errors, but we omit this mechanism here as it adds complexity without consistently improving CRU's performance in prior evaluations (Logan, 2021).

CRU treats the end of the study list as a special "end-of-list" event, assigning it a representation just like any study item and associating it with the final contextual state. Because this representation resides in the same instance-based memory as all other contexts, it competes with studied items during retrieval: if the end-of-list representation wins, recall terminates. As a result, CRU predicts that termination is most likely immediately after recalling the final item—when its associated context is strongly integrated—and least likely after the first item.

By contrast, CMR treats recall termination as a separate process from item retrieval that does not depend on the content of the current context or on which items have been recalled so far. For CMR, the probability of stopping the recall without retrieving any more items, depends on the output position j and is given by:

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

Here, θ_s and θ_r govern the initial stopping probability and the rate at which this probability increases, respectively, which is modeled as an exponential function.

CRU and CMR also differ in how they traditionally model the choice between items during retrieval. CMR applies the Luce choice rule (Luce, 1959) to convert activation strengths into recall probabilities. An additional parameter τ controls the contrast in activation strengths between items by raising the activation of each item to the power of τ. The probability $P(i)$ of recalling a specific item i is thus:

$$P(i) = (1 - P(stop))\frac{A_i^\tau}{\sum_k^N A_k^\tau}$$

In contrast, CRU models the choice between options as race between independent processes where each runner (here, a contextual state stored in memory) is governed by a drift rate parameter v and a threshold θ. This threshold is normally set to a fixed constant, allowing the rate parameter to exclusively determine the winner. The rate parameter for each option is determined by the similarity between current context and the context associated with the item. The finishing time is characterized by the Wald distribution:

$$f(t) = \theta(2\pi t^3)^{-1/2} \cdot exp(-\frac{(\theta - t \cdot v)^2}{2t})$$

And the race between runners is characterized by the following, with the actual probability that the runner i wins being the integral of this function over time:

$$f(t, i) = f_i(t) \prod_{j \neq i}^{N} (1 - f_j(t))$$

The way either model approaches the choice between items during retrieval is not normally treated as core to their theoretical commitments. Instead, the choice of decision rule is typically justified based on computational tractability and simplicity, with the Luce choice rule being more common in CMR and the racing diffusion model being more common in CRU. Given that neither model is committed to a specific decision rule, we will prefer to use matched decision rules in our model comparisons to ensure that differences in model performance are not due to differences in decision rules. Runner drift rates configured for simulation of the racing diffusion process can be treated as item support values in application of the Luce choice rule, and vice versa, to ensure that the models are compared on a level playing field. This leaves evaluation of the relative merits of the racing diffusion and Luce choice rules for modeling memory search within a retrieved-context framework as a topic for future work.

Overall, CMR and CRU differ in how flexibly they address the generation of retrieval sequences. First, CMR includes two separate parameters to model recall termination, θ_s and θ_r, which control the initial tendency to stop and the rate at which this tendency grows over successive recalls. Second, CMR includes a sensitivity parameter τ that controls the contrast in activation strengths between competing items. This parameter can be evaluated independently of the Luce choice rule for its ability to capture recall performance, similarly to the sensitivity parameter g in CRU's item identification mechanism. In addition to these already mentioned differences, CMR uses separate integration rate parameters for encoding and retrieval (β_{enc} and β_{rec}) to control how much the context integrates with each new item during encoding and retrieval.

Factorial Approach

While CRU was originally developed to capture behavioral performance in tasks focused on serial order, many of CRU's structural differences from CMR do not necessarily hinge on serial ordering. We therefore ask which of these differences matter for free recall, where retrieval can proceed in any order and often features backward transitions and flexible stopping. To do so, we adopt a factorial approach in which each key mechanism that CRU omits (or implements differently than CMR) is selectively toggled on or off, creating a series of "hybrid" CRU variants that, in a sense, span the gap between CMR and CRU in a hypothetical space of all possible models. Our evaluation is not comprehensive – we do not explore every possible combination of toggled mechanisms – but instead focuses on the most salient differences between CRU and CMR. In particular, we focus on five key factors that distinguish CRU from CMR in the free recall context: (1) dynamic feature-to-context memory, (2) pre-experimental context-to-feature associations, (3) serial position scaling, (4) recall initiation, and (5) recall termination.

Factors and Variants

In order to permit a fair comparison between CRU and CMR, we equate the two models in several ways. First, some parameters are allowed to vary freely across all model variants (e.g., τ, β_{enc}, and β_{rec}) to avoid complicating the factorial design further. Second, the item identification factor is omitted in free recall fits, as it is most relevant for serial recall tasks with visually confusable stimuli. Finally, we use the same decision rule (the Luce choice rule) for all model variants, ensuring that differences in model performance do not arise from different decision processes. The resulting framework generates model variants that differ along the following dimensions:

1. **Dynamic or Inert Feature-to-Context Memory**. In CMR, a nonzero learning rate γ allows recently presented items to retrieve prior context states, enhancing backward transitions in free recall. By contrast, CRU omits M^{FC} (or sets $\gamma = 0$), instead directly integrating item representations into the context vector with no reinstatement of previously associated contexts. This factor is especially pertinent in free recall, where backward transitions are a key empirical signature. We show that enabling a dynamic M^{FC} in CRU can help capture these transitions and improve its overall fit to free recall data.

2. **Use or Nonuse of a Pre-Experimental Context-to-Feature Memory**. Both α (shared support) and δ (self-support) in CMR reflect pre-existing item associations that can facilitate retrieval. CRU does not include these mechanisms. We consolidate these into a single on/off factor indicating whether such pre-experimental support is active. When on, items have baseline associations that may enhance recall probabilities – particularly relevant in word-based free recall, where semantic or lexical relationships matter. When off, all associations arise solely from the experimental context, mirroring CRU's instance-based memory. By toggling this factor, we show that incorporating pre-experimental support can help CRU capture patterns of temporally distant and adjacent transitions in free recall.

3. **Primacy Gradient**. A third factor involves how each model boosts memory for initial study

items to account for primacy effects. CRU achieves this by modulating contextual integration rate β and item identification sensitivity g by serial position, peaking the strength of these mechanisms at the start of the list and decaying toward the end. CMR instead employs dedicated parameters ϕ_s and ϕ_d that scale the learning rate by position in context-to-feature memory M^{CF} from an initial peak to a final baseline. For free recall, item confusability is negligible ($g \to \infty$), so we focus on assessing the effectiveness of CMR-style primacy gradient in CRU. We show that incorporating CMR's primacy gradient can enhance CRU's ability to capture the primacy effect in free recall without supposing that items can be visually confused between each other during encoding.

4. **Recall Initiation**. CRU always initiates recall from the start-of-list context, whereas CMR can bias the cue toward the beginning or end of the list via β_{start} by integrating the final context state back toward the start. We toggle this factor by either forcing retrieval to begin at the start-of-list state (CRU style) or allowing a flexible mix of start and end (CMR style). In free recall, participants frequently begin recalling from the end of the list. We show that allowing CRU to flexibly initiate retrieval is crucial for its ability to address this pattern and free recall performance more generally.

5. **Recall Termination**. Finally, CRU encodes an "end-of-list" state that competes with items to terminate recall, whereas CMR separately calculates a stopping probability using θ_s and θ_r. In our design, we can adopt either the CRU-style item-based mechanism, the CMR-style exponential stop rule, or, in principle, a hybrid combining both. This factor addresses how each model handles the decision to stop retrieval – especially in free recall, where participants spontaneously cease recall rather than exhaustively listing items in order. We show that CRU's context-based termination mechanism causes the model to collapse when addressing free recall datasets with strong recency effects, and that incorporating a CMR-style stopping probability can help CRU better capture these patterns.

By toggling or switching between these mechanisms – e.g., enabling or disabling feature-to-context learning – we create a spectrum of CRU variants that bridge between CRU's streamlined architecture and CMR's more flexible associative structures. We also include a "baseline CMR" and a "baseline CRU" variant that represents the full CMR and CRU models, respectively, to compare against the hybrid variants.

Dataset

We draw our free recall data from the PEERS dataset (Healey & Kahana, 2014), focusing on a subset well-suited to our model comparisons. **Participants.** A total of 126 individuals (ages 17–30) completed multiple recall trials, each involving a list of 16 unique words. We selected this age range for its relative homogeneity in memory performance and for continuity with prior CMR research. **Stimuli.** Each list consisted of 16 words drawn at random from the Toronto Word Pool (Friendly et al., 1982), which includes common nouns, adjectives, and verbs. Words were chosen to be low in semantic similarity to minimize confusability; thus, items generally had negligible overlap. **Procedure.** Participants studied each list of words, then performed immediate free recall for up to 45 seconds. They repeated this process for multiple lists, although trials featuring an interleaved retrieval task were excluded from our analysis to maintain consistency with standard free recall protocols. In total, this exclusion left 112 trials per participant. **Rationale.** We selected this dataset because (a) it provides a large, diverse sample size (allowing robust parameter estimation), (b) it uses word stimuli, typical for CMR applications, and (c) it exhibits well-documented primacy and recency effects, presenting a challenge for CRU's traditional emphasis on strictly serial tasks.

Likelihood-Based Fitting

We fit each model variant at the individual-participant level via a stepwise likelihood approach (Kragel et al., 2015; Morton & Polyn, 2016; Polyn, 2023). For each trial, the model simulates encoding of the studied items, then scores each observed recall event by the probability of selecting the same item (or stopping) at that point in the retrieval sequence, given the model's evolving context. We sum the log probabilities across trials and maximize likelihood via differential evolution (Storn & Price, 1997). Repeated recalls and extralist intrusions are excluded from the likelihood calculation (consistent with prior CMR evaluations). To compare models, we report mean and bootstrapped 95% confidence intervals for the log-likelihood across participants for each variant. We also examine which variant gives the best fit on a participant-by-participant basis.

Data Simulation

We also simulate each fitted model variant on the same list structure participants experienced to see whether it reproduces hallmark free recall phenomena—serial position curves, probability of first recall (PFR), and lag-conditional response probabilities (lag-CRP). By comparing real and simulated data, we assess whether a model not only matches the trial-by-trial recall sequences but also generates the correct shape of these benchmark effects.

Results

Table 2

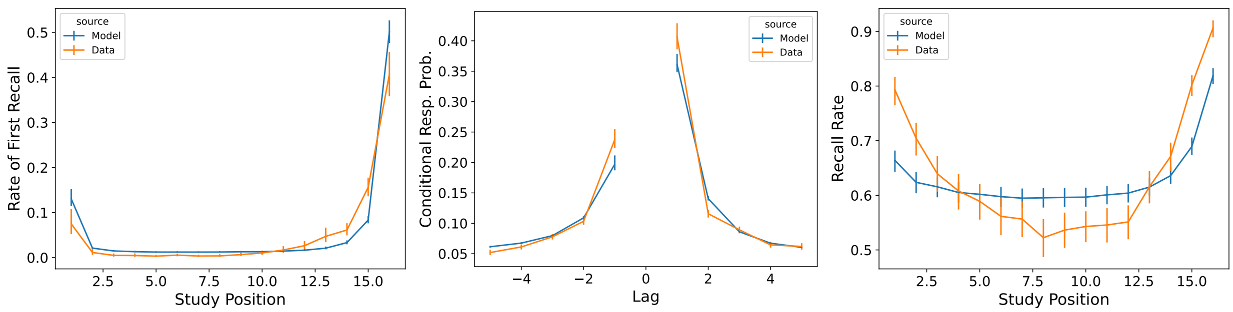
Negative log-likelihood (±95% CI) averaged across participants for selected model variants fit to PEERS data. γ: item-to-context learning rate; α: shared support; δ: self-support; ϕ_s: primacy scale; ϕ_d: primacy decay; β_{start}: start context integration rate.

Model Variant	-LL (±95% CI)
CMR (Free γ, α, δ, ϕ_s, ϕ_d, $\beta start$ and Position-Based Termination)	587.13 +/- 16.84
CRU with Free α, δ, ϕ_s, ϕ_d, $\beta start$	606.05 +/- 16.56
CRU with Free γ, ϕ_s, ϕ_d, $\beta start$	608.00 +/- 17.00
CMR with CRU's Context-Based Termination	627.86 +/- 17.35
CRU with Free ϕ_s, ϕ_d, $\beta start$	645.04 +/- 17.49
CRU with Free $\beta start$	651.35 +/- 17.31
CRU	724.01 +/- 17.76

We begin by examining the overall goodness of fit for each model variant. Standard CMR provides the best fit at the group level (see Table 2) as well as for 100% of individual participants.

Figure 1

Summary statistic fits of the baseline CMR model to PEERS data. ***Left****: probability of recall initiation by serial position.* ***Middle****: conditional response probability as a function of lag.* ***Right****: recall probability by serial position.*



This is no surprise, as CMR was designed to capture free recall phenomena while CRU was developed for serial recall tasks. The value of this approach is in demonstrating how specific model mechanisms improve CRU's fit to the benchmark behavioral phenomena of free recall: serial position effects, recall initiation effects, and temporal organizational effects, as depicted in Figure 1.

A serial position analysis demonstrates both a primacy effect (a memorability advantage for early-list items) and a recency effect (a memorability advantage for late-list items). These memorability advantages are also apparent in the distribution of first recalls. In this dataset, a recency effect is particularly pronounced in the distribution of first recalls; participants usually start recall with the last item from a study list, but sometimes start with the first item. By comparison, the serial position curve shows a more balanced trade-off between primacy and recency effects.

Two mechanisms allow CMR to capture the interplay between primacy and recency effects. First, CMR leverages the primacy learning gradient mechanism modulating learning rates in the context-to-feature memory (ϕ_s, ϕ_d) to enhance the learning of early-list items. Second, CMR uses a Start context integration parameter (β_{start}) to recover the start-of-list context. Both mechanisms influence the initiation of recall and the overall serial position curve, but they have differential impacts on these two stages. The β_{start} parameter is most influential at recall initiation, while the primacy learning gradient influences the primacy effect throughout retrieval. These mechanisms together allow CMR to capture both the distribution of first recalls and the overall serial position curve. CMR exhibits some limitations capturing these phenomena, underpredicting the primacy effect in the serial position curve. Addressing limitations and further exploring these interactions are outside the scope of this paper, but reflect continuing challenges in modeling free recall phenomena.

A focal concern of retrieved-context models is the lag-contiguity effect, the tendency to successively recall items that were presented close together in time (Kahana, 1996, 2020). These models capture this effect by reinstating the temporal context associated with a just-recalled item (step 1) and then using that updated context to cue other items (step 2). In free recall, the lag-contiguity effect is bidirectional but asymmetric, favoring forward transitions over backward ones. To capture all aspects of the lag-contiguity effect, CMR primarily uses its feature-to-context memory to reinstate the context associated with a just-recalled item and then its context-to-feature memory to activate items associated with its newly updated context. The γ parameter influences context reinstatement by weighting the influence of pre-experimental and experimental associations in the feature-to-context memory. CMR's parameters δ and α conversely configure pre-experimental context-to-feature memory to influence the extent to which retrieval of items associated with the current context is focused on nearby or distant neighbors of the last recalled item. Together, these parameters yield the characteristic forward-biased yet bidirectional pattern observed in free recall.

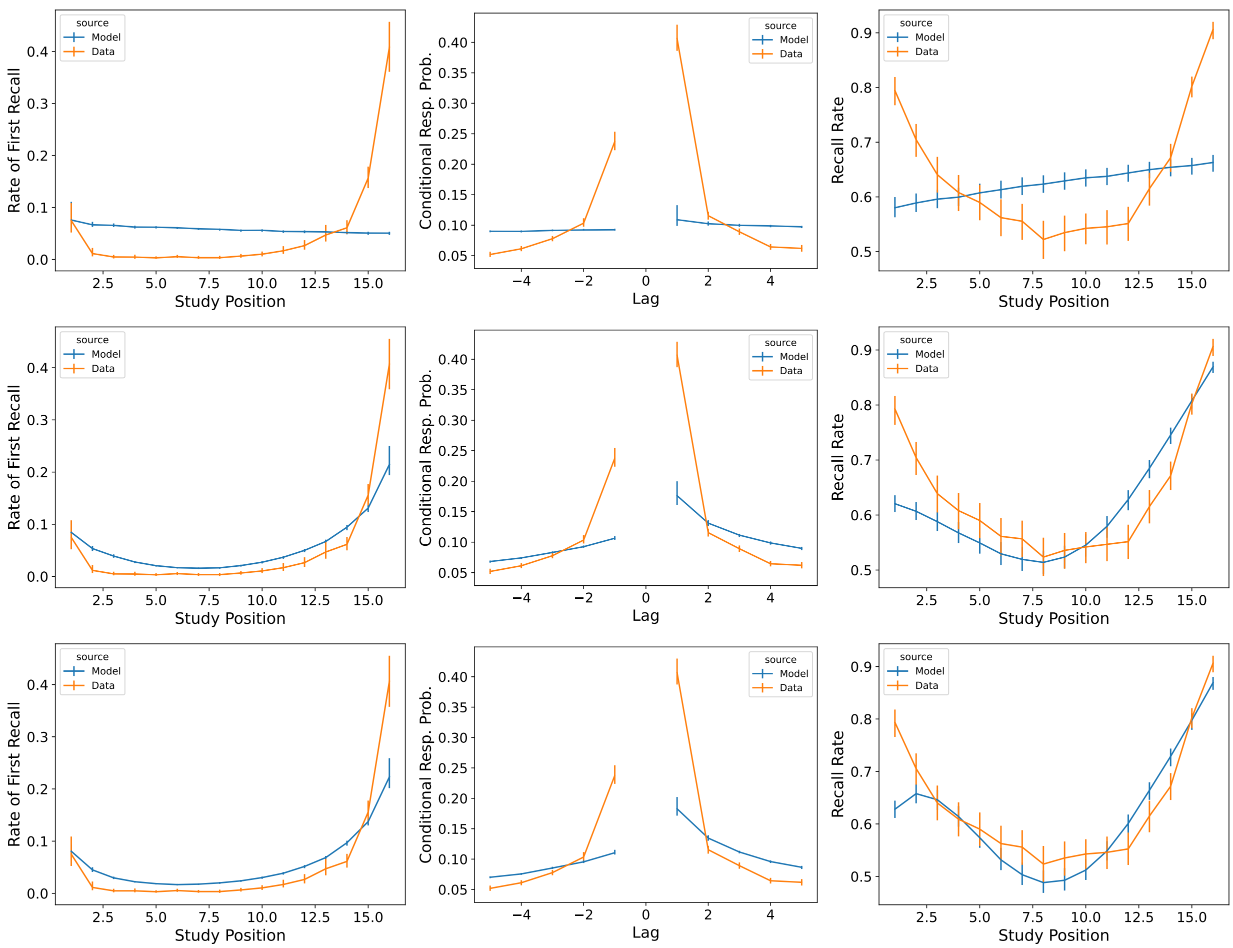
These simulation results establish a baseline, demonstrating the ability of standard CMR to capture key free-recall phenomena. We now turn to CRU and evaluate how its streamlined implementation impacts its ability to capture these phenomena.

Addressing Serial Position Effects in Recall Initiation and Overall

CRU includes most of CMR's core mechanisms. However, CRU's default start-of-list recall initiation mechanism forces it to strongly prioritize the start of the list during recall

Figure 2

*Summary statistic fits of baseline CRU (**Top**), CRU with free start context integration rate β_{start} (**Middle**), and CRU freeing both start context integration rate (β_{start}) and primacy gradient (ϕ_s and ϕ_d) parameters (**Bottom**) to Healey and Kahana (2014). **Left**: probability of recall initiation by serial position. **Middle**: conditional response probability as a function of lag. **Right**: recall probability by serial position.*



initiation, which causes its performance to collapse when fit to free recall data exhibiting a strong recency effect. This failure to capture recall initiation affects the entire response sequence because transitions in free recall depend substantially on prior recalls. Allowing CRU to initiate retrieval with a blend of the end-of-list and start-of-list context according to a flexible start-of-recall parameter like CMR's β_{start} substantially improves its ability to capture these phenomena.

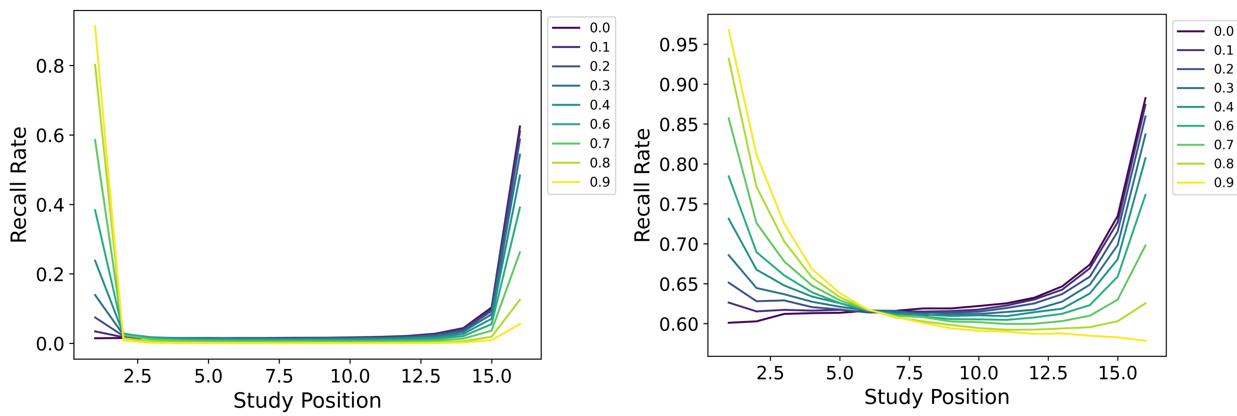
Figure 3 helps interpret the mechanism further by showing the impact of shifting the start context integration rate parameter β_{start} on the probability of starting recall by serial position and the recall probability by serial position for CMR. To perform this simulation, we fit CMR to each individual participant in the subset of the PEERS dataset (Healey & Kahana, 2014) and then shifted the β_{start} parameter from 0 to 1 in increments of 0.1, repeatedly simulating the model on the same list structure it was fit to and generating serial position curves and recall initiation curves. This parameter primarily trades off the strength of the primacy effect in recall initiation against the strength of the recency effect, with higher values of β_{start} leading to stronger primacy effects and lower values leading to stronger recency effects. The serial position curve is sensitive to the value of β_{start} as well since the item retrieval initiates with affects the trajectory of responses throughout the recall sequence.

We illustrate this point by comparing standard CRU to a CRU variant that includes CMR's start context integration rate β_{start} parameter as well as a CRU variant that additionally includes CMR's primacy gradient (ϕ_s and ϕ_d) parameters. Rows 1 and 2 of Figure 2 show simulated benchmark phenomena for CRU and the first variant, respectively. Fixing β_{start} to 1 leads to fits where CRU predicts no consistent serial position or lag-contiguity effects, while allowing β_{start} to vary enables CRU to begin to capture these effects, achieving a U-shaped overall serial position curve, a strong recency effect in recall initiation, and a bidirectional lag-contiguity effect. The model still substantially underestimates the primacy effect in the overall serial position curve, the strength of the recency effect in recall initiation, and the strength of the lag-contiguity effect, but these limitations are less severe than when β_{start} is fixed at 1.0.

Further allowing CRU to include CMR's primacy learning gradient (ϕ_s and ϕ_d

Figure 3

*Simulation of the impact of shifting the start context integration rate parameter β_{start} on the probability of starting recall by serial position (**Left**) and the recall probability by serial position (**Right**) for CMR. Using parameters fit to Healey and Kahana (2014), β_{start} is shifted from 0 to 1 in increments of 0.1, with the color of the lines indicating the value of the parameter.*

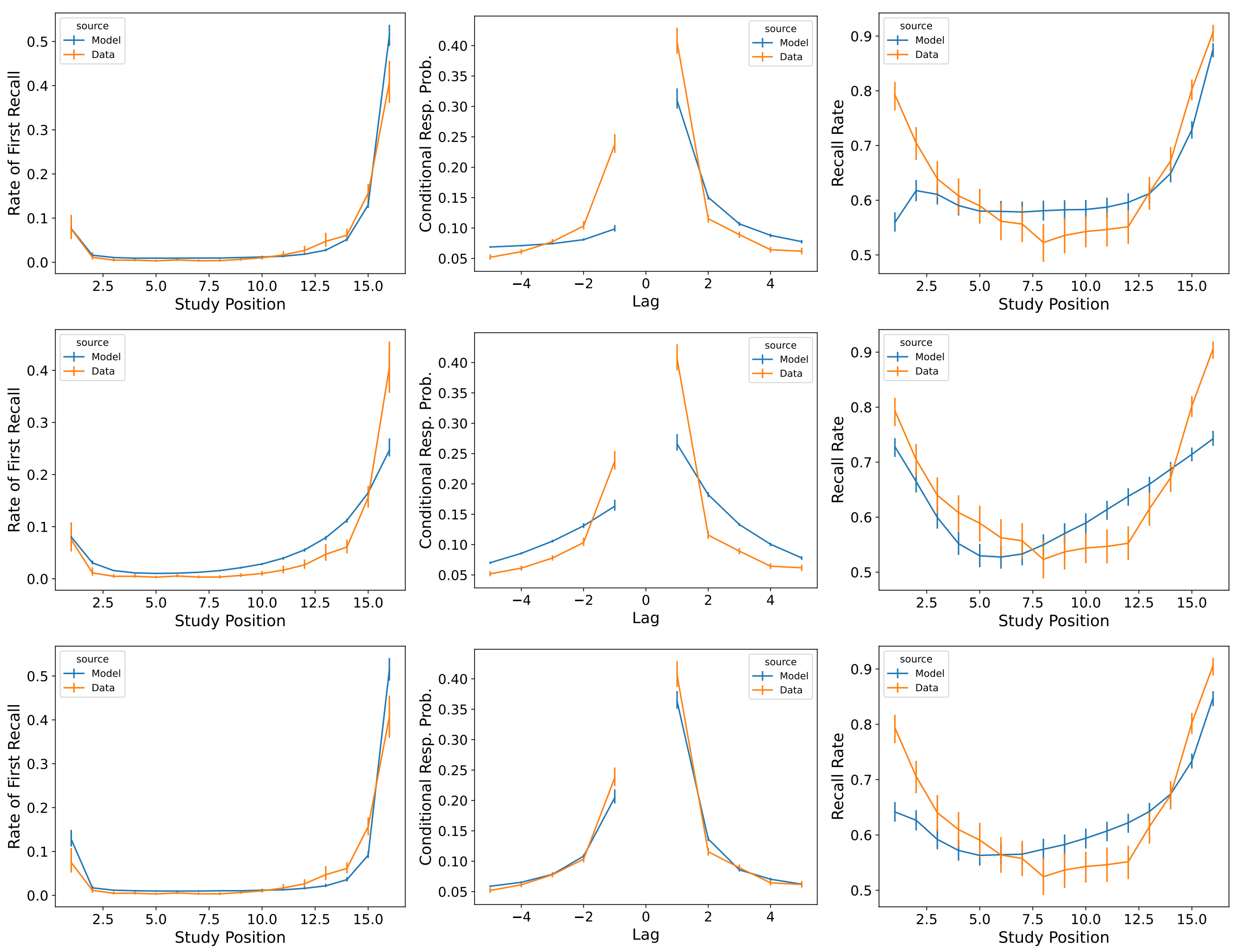


parameters) to modulate context-to-feature memory learning rates to peak at the start of the study list further improves its performance. This variant is better at capturing the strength of the primacy effect in the overall serial position curve Figure 2, but does not as substantially improve the model's ability to capture the recency effect in recall initiation or the lag-contiguity effect.

These differences observed between CRU and CMR in their ability to address serial position effects may be exaggerated by the exclusion of an item identification mechanism in CRU addressing free recall. Logan (2021) accounted for primacy effects in CRU with g_{max} and g_{dec} parameters that allowed the model to modulate the sensitivity of item identification by serial position, but this mechanism does not apply to word free recall data. Here, only CRU's β_{max} and β_{dec} parameters were available to modulate contextual integration rates during encoding as a function of serial position, which may have limited the model's ability to capture the primacy effect in the overall serial position curve.

Figure 4

Summary statistic fits of models to the PEERS dataset (Healey & Kahana, 2014). ***Top****: CRU with free pre-experimental context-to-feature memory (α, δ), primacy gradient (ϕ_s, ϕ_d), and start context integration rate (β_{start}) parameters.* ***Middle****: CRU with free item-to-context learning rate (γ), primacy gradient (ϕ_s, ϕ_d), and start context integration rate (β_{start}) parameters.* ***Bottom****: CRU with free item-to-context learning rate (γ), pre-experimental context-to-feature memory (α, δ), primacy gradient (ϕ_s, ϕ_d), and start context integration rate (β_{start}) parameters – equivalent to CMR.* ***Left****: Probability of starting recall by serial position.* ***Left****: Probability of starting recall by serial position.* ***Middle****: Conditional response probability as a function of lag.* ***Right****: Recall probability by serial position.*



Tuning the Sharpness and Asymmetry of the Lag-Contiguity Effect

In free recall, the lag-contiguity effect is bidirectional and asymmetric, with participants more likely to transition to items that were presented immediately after the just-recalled item, but also sometimes transitioning to items that were presented immediately before the just-recalled item in the original study list. Standard CRU is able to simulate a bidirectional lag-contiguity effect, but it substantially underestimates the strength of the effect compared to the data, predicting around a 20% probability of +1 lag transition while the data shows over a 40% probability of +1 lag transition. Standard CRU shows a and similar discrepancy for a backward -1 lag transition.

CMR controls the shape of the lag-contiguity effect with a set of parameters controlling the relative strength of different associative structures. These parameters include γ, which tunes the strength of learned (experimental) associations between items and context that drive context reinstatement during retrieval, and α and δ, which tune the distribution of context-to-feature associations across items that drive the competition between items during retrieval. CMR's γ parameter provides a dynamic feature-to-context memory by tuning the strength of learned associations between items and context that drive context reinstatement during retrieval. The higher γ is, the more strongly the context associated with a just-recalled item is reinstated during retrieval, helping capture high rates of short lag backward transitions in the lag-contiguity effect. Figure 5 (left) shows the impact of shifting γ on the lag-CRP for CMR, supporting this interpretation. The simulation was configured similarly to the previous section, using parameters fit to Healey and Kahana (2014), and shifting γ from 0 to .9 in increments of 0.1.

CMR's δ and α parameters provide a pre-experimental context-to-feature memory that can influence the lag-contiguity effect by tuning the strength of associations from context representations to item representations. When δ is much higher than α, the lag-contiguity effect is facilitated by favoring neighbors of the just-recalled item in the competition between items during retrieval. Alternatively, setting δ to match α or to zero can flatten the lag-contiguity effect by making all items equally likely to be activated by a context feature associated with a just-recalled item. When γ is configured to 0 – as it effectively is in CRU – tuning δ does not influence rates of backward transitions in the lag-contiguity effect at all, as the context unit pre-experimentally associated with recalled items only helps to activate forward-going neighbors upon retrieval. Figure 5 (right) illustrates the impact of shifting δ on the shape of the lag-CRP for CMR when γ is not set to 0. The simulation was configured similarly to the previous section, using parameters fit to Healey and Kahana (2014), and shifting δ from 0 to 8.9 in increments of 1. When γ is configured to non-zero values, higher values of δ can simultaneously drive both forward and backward transitions in the lag-contiguity effect, reducing the rate of more distant transitions while increasing the rate of closer transitions.

Freeing γ and δ and α in CMR allows the model to flexibly capture the strength and asymmetry of the lag-contiguity effect. Figure 4 (Row 2) illustrates CRU's performance when extended to include CMR's dynamic feature-to-context memory (γ) alongside primacy and recency mechanisms(ϕ_s, ϕ_d, and β_{start}). Simulated summary statistics confirm that this extension provides a more balanced capture of the lag-contiguity effect, but still underestimates its overall strength compared to the data. This produces correspondingly worse performance capturing primacy and recency benchmarks.

Finally, in the third row of Figure 4, extending CRU to include both CMR's dynamic feature-to-context memory (γ) *and* pre-experimental context-to-feature memory (δ and α) alongside established primacy and recency mechanisms provides a more complete capture of all components of the lag-contiguity effect and other benchmarks. However, this model is exactly CMR: CRU with all of CMR's mechanisms enabled. This suggests that standard CRU's streamlined implementation is not sufficient to capture the full range of free recall phenomena, and underlines that all of CMR's mechanisms are useful for capturing free recall data, despite the complexity they introduce.

Position- vs Context-Based Mechanisms for Recall Termination

CRU and CMR's mechanisms for recall termination are fundamentally different. This difference cannot be captured by toggling a parameter from a fixed to a freely adjustable value, but rather can only be swapped between variants. CMR uses exponentially increasing stopping

Figure 5

*Simulation of the impact of shifting CMR's γ (**Left**) and δ (**Right**) parameters on the conditional response probability as a function of lag for CMR. Using parameters fit to Healey and Kahana (2014), the learning rate parameter γ is shifted from 0 to 1 in increments of 0.1, and the item support parameter δ is shifted from 0 to 10 in increments of 1, with the color of the lines indicating the value of the parameter.*

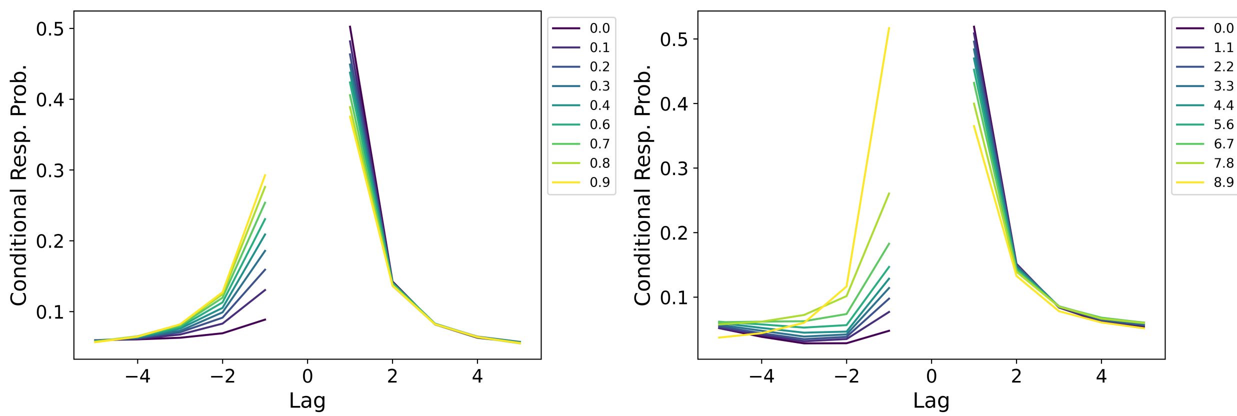
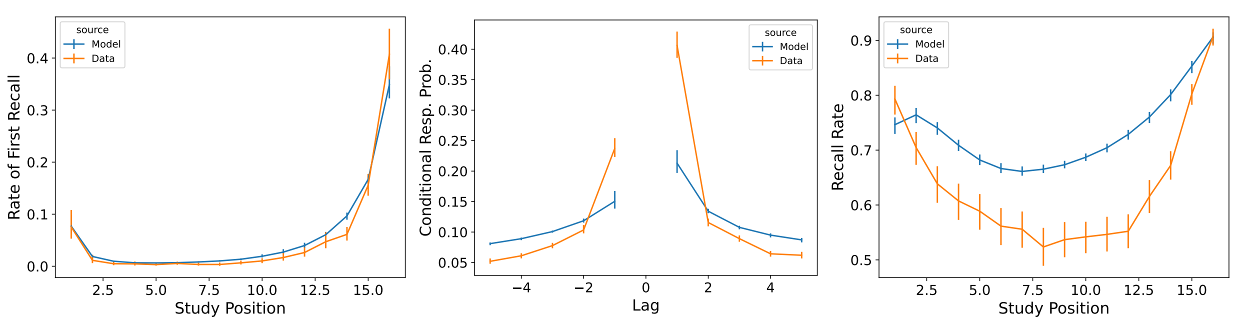


Figure 6

*Summary statistic fits of CMR with CRU's context-based recall termination mechanism to Healey and Kahana (2014). **Left**: probability of starting recall by serial position. **Middle**: conditional response probability as a function of lag. **Right**: recall probability by serial position.*



probabilities θ_s and θ_r to model recall termination; the probability of termination scales only with the number of recalls made so far. By contrast, CRU treats the end of a study sequence as a special item associated in memory with the final state of the study context. This item competes with other items for retrieval at each new recall event, and its activation can terminate recall. In this specification, the probability of termination depends on the state of context at each recall event, and can be influenced by the same mechanisms that influence the probability of recalling other items.

Performance differences between CMR's position-based recall termination mechanism and CRU's context-based recall termination mechanism are substantial. The top row of Figure 6 shows baseline CMR's performance on these benchmarks, while the bottom row shows CMR with CRU's context-based recall termination mechanism. While patterns in response initiation are well-captured, using context-based recall termination mechanism leads CMR to predict overly high recall rates for all study list positions and to fail to capture the sharpness of the lag-contiguity effect. Ideal comparison of these mechanisms would use data with variable study list lengths and focus analyses on the probability of terminating recall as a function of the number of recalls made so far.

The success of CRU's context-based recall termination mechanism depends on how consistently participants terminate recall after recalling the final items from the study list. In most serial recall datasets, participants tend to perform this way, and the mechanism correspondingly predicts that the probability of terminating recall scales with the number of recalls made so far, as context drifts from its start-of-list state to its end-of-list state. By contrast, in free recall datasets where participants exhibit a strong recency effect in recall initiation, the mechanism can predict early termination of recall upon or even before retrieving the last item in the study list. The failure of CMR with CRU's context-based recall termination mechanism to capture benchmark patterns may reflect a discrepancy between this prediction and actual participant behavior in the dataset, and suggests that CMR's position-based recall termination mechanism is more effective for capturing free recall data.

Discussion

The simulation analyses presented here provide a structured comparison of CRU and CMR in the domain of free recall. CRU is a successful model of serial recall performance (Logan, 2018, 2021; Logan & Cox, 2021). As such, comparing the two frameworks gives insight into how specific model mechanisms contribute to behavioral phenomena that differ between the two tasks. At the outset, we noted how CRU's success in strictly ordered memory tasks might obscure its capacity to handle the broader dynamics of free recall, where retrieval can proceed in many directions and often terminates in flexible ways. By systematically enabling or disabling different mechanisms from CMR excluded from CRU, we showed how features like a dynamic feature-to-context memory, pre-experimental context-to-feature associations, serial position memory strength scaling, flexible recall initiation, and different termination rules can critically shape free recall performance. This analysis reveals that CRU can, in fact, capture many hallmark free recall phenomena when progressively endowed with CMR-like machinery – but in so doing, it gradually converges on CMR's complexity.

The factorial approach underscores the specific changes needed for CRU to handle backward transitions and and robust primacy–recency trade-offs. The strongest benefits emerge when we grant CRU the ability to initiate recall flexibly (β_{start}) using a mix of final and initial study context, a feature that CMR uses to balance primacy and recency effects in recall initiation (Kragel et al., 2015; Morton & Polyn, 2016). CMR additionally implements a proposal by Sederberg et al. (2008) that the primacy effect is supported by increased attention to initial items in study lists. Implementing this primacy learning gradient by allowing the strength of context-to-item associations to scale with serial position (ϕ_s, ϕ_d) yields appreciable improvements to CRU when paired with β_{start}, suggesting that mechanisms specific to recall initiation and encoding dynamics are crucial for capturing the serial position curve. This mixed account of primacy and recency effects coheres with patterns of response time distributions in free recall initiation (Osth & Farrell, 2019), suggesting that the balance between these two effects is a key determinant of recall initiation dynamics.

While introducing this flexibility is sufficient to make CRU start to capture key free-recall phenomena, additional mechanisms are needed to fully capture the bidirectional lag-contiguity effect. Adding pre-experimental associations to CRU's context-to-feature memory (α, δ) help it tune the reliability of forward transitions in the lag-contiguity effect. To also capture the strength of backward transitions, CRU needs to enable feature-to-context learning (γ) so that associations necessary for contacting backward neighbors can be more consistently leveraged during retrieval. While addressing backward transitions is crucial for addressing the assymetric lag-contiguity effect in free recall, such mechanisms may also help address performance in serial recall where probed recall of serial lists (Kahana & Caplan, 2002) and dissociations between forward and backward recall (Li & Lewandowsky, 1993) suggest a similar pattern of associative asymmetry in memory search (Howard & Kahana, 2002). While evaluations of CMR for its ability to capture serial recall have embraced a role for a task-specific configuration of γ parameter (Lohnas, 2024), further research is needed to determine how much a dynamic feature-to-context memory is necessary for capturing performance in these tasks.

Finally, CRU's method of modeling recall termination was poorly aligned with the variable stopping patterns often observed in free recall, reinforcing that a purely context-driven account of termination is better suited to tasks where participants explicitly strive for sequential order. On the other hand, CMR's termination is probably overly simplistic, as other research has shown that recall termination probability depends on factors other than output position. Participants are especially likely to terminate free recall after making an error (Miller et al., 2012; Unsworth et al., 2011), when they are less confident in their responses (Unsworth et al., 2011), when they are less motivated to continue (Dougherty & Harbison, 2007), and when a longer amount of time has passed since the last recall (Dougherty & Harbison, 2007). These results together suggest that recall termination probability is a function of the difficulty of continuing recall, not just the number of recalls made so far. Future work may explain these patterns by relating the accessibility in memory of the next item as predicted by a retrieved-context model to the likelihood of continuing or terminating recall.

By showing that CRU and CMR can be seen as points along a continuum of retrieved-context approaches, our results highlight both the flexibility of RCT and the trade-offs involved in simplifying it. CRU's architecture is well-tailored to domains emphasizing forward-ordered retrieval, but omits parameters crucial for capturing the bidirectional and open-ended nature of free recall. Meanwhile, CMR's greater flexibility comes at the cost of added complexity. The results here do not explore potential overfitting in detail, but suggest that each model can be expanded or trimmed depending on task demands. Moreover, CRU's instance-based storage of item–context pairs diverges from CMR's linear associative memory, and the consequences of this difference for model behavior are ambiguous pending further research (J. A. Anderson, 1995; Turner, 2019). It remains unclear which modeling framework better scales to tasks with repeated or highly confusable items. Studying performance across a wider range of list lengths or item sets should clarify the practical boundaries of each approach.

The present analysis sidesteps comparison of CRU and CMR's distinct recall competition mechanisms based on the justification that neither model is committed to a specific mechanism for recall competition (Logan, 2021; Morton & Polyn, 2016; Polyn et al., 2009) and simulation using the Luce choice rule is more computationally efficient. Nonetheless, with response time distributions providing important constraints for accounts of recall initiation (e.g., Osth & Farrell, 2019) and termination (e.g., Dougherty & Harbison, 2007), the racing diffusion model of recall competition favored in demonstrations of CRU (Logan, 2018, 2021) potentially offers a tractable framework for addressing response time distributions as a theoretical constraint within the likelihood-based fitting approach used here (Tillman et al., 2020). To benefit from this approach, future work should clarify assumptions about when recall competitions begin and end across responses in free and serial recall tasks, and how these assumptions can be tested against response time data (Logan, 2021).

In broader terms, these findings advance our understanding of how context-based models can unify disparate memory phenomena. The parallels we draw between CRU and CMR suggest a shared foundation for modeling recall sequences, whether strictly ordered as in serial recall or unconstrained as in free recall. Contemporaneous research indicates that CMR variants can capture core aspects of both free and serial recall (Lohnas, 2024), while CRU's success spans traditional list-learning tasks and broader cognitive-control or motor-sequencing paradigms that rely on forward-chained retrieval (Logan, 2018, 2021). Such breadth underscores the remarkable scope of retrieved-context theory as a unifying explanation across multiple domains, from free recall of word lists to hierarchical production skills. By situating CRU and CMR within a single modeling space, we emphasize how closely related mechanisms can produce seemingly divergent behaviors. This convergence invites researchers to treat free and serial recall findings as complementary constraints on a unified account of memory search, laying groundwork for a more integrated theory of episodic retrieval in both highly directed and more open-ended recall scenarios.

References

Anderson, J. A. (1995). *An introduction to neural networks*. MIT press.

Anderson, J. R., & Bower, G. H. (1972). Recognition and retrieval processes in free recall. *Psychological Review*, *79*(2), 97.

Dougherty, M. R., & Harbison, J. (2007). Motivated to retrieve: How often are you willing to go back to the well when the well is dry? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *33*(6), 1108.

Estes, W. K. (1955). Statistical theory of spontaneous recovery and regression. *Psychological Review*, *62*(3), 145.

Friendly, M., Franklin, P. E., Hoffman, D., & Rubin, D. C. (1982). The toronto word pool: Norms for imagery, concreteness, orthographic variables, and grammatical usage for 1,080 words. *Behavior Research Methods & Instrumentation*, *14*(4), 375–399.

Healey, M. K., & Kahana, M. J. (2014). Is memory search governed by universal principles or idiosyncratic strategies? *Journal of Experimental Psychology: General*, *143*(2), 575.

Healey, M. K., & Kahana, M. J. (2016). A four-component model of age-related memory change. *Psychological Review*, *123*(1), 23.

Horwath, E. A., Rouhani, N., DuBrow, S., & Murty, V. P. (2023). Value restructures the organization of free recall. *Cognition*, *231*, 105315.

Howard, M. W., & Kahana, M. J. (1999). Contextual variability and serial position effects in free recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *25*(4), 923.

Howard, M. W., & Kahana, M. J. (2002). A distributed representation of temporal context. *Journal of Mathematical Psychology*, *46*(3), 269–299.

Kahana, M. J. (1996). Associative retrieval processes in free recall. *Memory & Cognition*, *24*(1), 103–109.

Kahana, M. J. (2020). Computational models of memory search. *Annual Review of Psychology*, *71*(1), 107–138.

Kahana, M. J., & Caplan, J. B. (2002). Associative asymmetry in probed recall of serial lists. *Memory & Cognition*, *30*(6), 841–849.

Kragel, J. E., Morton, N. W., & Polyn, S. M. (2015). Neural activity in the medial temporal lobe reveals the fidelity of mental time travel. *Journal of Neuroscience*, *35*(7), 2914–2926.

Li, S.-C., & Lewandowsky, S. (1993). Intralist distractors and recall direction: Constraints on models of memory for serial order. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *19*(4), 895.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*(4), 492.

Logan, G. D. (2018). Automatic control: How experts act without thinking. *Psychological Review*, *125*(4), 453.

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

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

Logan, G. D., & Cox, G. E. (2023). Serial order depends on item-dependent and item-independent contexts. *Psychological Review*.

Lohnas, L. J. (2024). A retrieved context model of serial recall and free recall. *Computational Brain & Behavior*, 1–35.

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.

Luce, R. D. (1959). *Individual choice behavior* (Vol. 4). Wiley New York.

Melton, A. W., & Martin, E. (1972). *Coding processes in human memory.*

Miller, J. F., Weidemann, C. T., & Kahana, M. J. (2012). Recall termination in free recall. *Memory & Cognition*, *40*, 540–550.

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.

Murdock, B. B. (1962). The serial position effect of free recall. *Journal of Experimental Psychology*, *64*(5), 482.

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*(4), 578.

Osth, A. F., & Hurlstone, M. J. (2023). *Do item-dependent context representations underlie serial order in cognition? Commentary on logan (2021).*

Polyn, S. M. (2023). Assessing neurocognitive hypotheses in a likelihood-based model of the free-recall task. In *An introduction to model-based cognitive neuroscience* (pp. 303–325). Springer.

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.

Sederberg, P. B., Gershman, S. J., Polyn, S. M., & Norman, K. A. (2011). Human memory reconsolidation can be explained using the temporal context model. *Psychonomic Bulletin & Review*, *18*, 455–468.

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.

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.

Tillman, G., Van Zandt, T., & Logan, G. D. (2020). Sequential sampling models without random between-trial variability: The racing diffusion model of speeded decision making. *Psychonomic Bulletin & Review*, *27*(5), 911–936.

Tulving, E., & Madigan, S. A. (1970). Memory and verbal learning. *Annual Review of Psychology*, *21*, 437–484.

Turner, B. M. (2019). Toward a common representational framework for adaptation. *Psychological Review*, *126*(5), 660.

Unsworth, N., Brewer, G. A., & Spillers, G. J. (2011). Factors that influence search termination decisions in free recall: An examination of response type and confidence. *Acta Psychologica*, *138*(1), 19–29.