

---

# Learning to Forget Attention: Memory Consolidation for Adaptive Compute Reduction

---

Ibne Farabi Shihab\*

Department of Computer Science  
Iowa State University  
ishihab@iastate.edu

Sanjeda Akter\*

Department of Computer Science  
Iowa State University  
sanjeda@iastate.edu

Anuj Sharma

Department of Civil, Construction and Environmental Engineering  
Iowa State University  
anujs@iastate.edu

## Abstract

Hybrid architectures combining state-space models with attention have achieved strong efficiency-quality tradeoffs, yet existing approaches either apply attention uniformly or learn static sparse patterns. This misses a key opportunity: *attention demand should decrease over time as recurring patterns become familiar*. We present a surprising finding from analyzing GPT-2 models: **88%** of attention operations retrieve information already predictable from the model’s hidden state, and this redundancy does *not* decrease during training. Motivated by this observation, we introduce **CRAM** (Consolidation-based Routing for Adaptive Memory), a biologically inspired memory consolidation mechanism that gradually distills episodic retrievals into parametric semantic memory. Unlike prior sparse attention methods, CRAM exhibits *decreasing attention utilization* over training, achieving a **37.8 $\times$**  reduction through a sharp phase transition at approximately 3K steps. We prove that this capability is *impossible* without consolidation: any static routing scheme requires  $\Omega(f \cdot n)$  attention for tasks with recurring patterns of frequency  $f$ . On our proposed SRCD benchmark, CRAM achieves **100% retrieval accuracy** at 1.6% attention compute (vs. 68% for baselines), and consolidated patterns transfer to unseen tasks with **48–52%** attention reduction without retraining. Remarkably, the learned consolidation dynamics quantitatively match human episodic-to-semantic memory transition curves from cognitive psychology ( $\gamma = 0.43$  vs.  $\gamma_{\text{human}} \approx 0.4\text{--}0.5$ ). Code and benchmarks are available at [added later].

## 1 Introduction

The efficiency-expressivity tradeoff in sequence modeling has driven rapid architectural innovation [Tay et al., 2022]. Self-attention [Vaswani et al., 2017] provides powerful global interaction but incurs quadratic cost, motivating a long line of efficient alternatives including sparse patterns [Child et al., 2019, Beltagy et al., 2020], low-rank projections [Wang et al., 2020, Katharopoulos et al., 2020], and locality-sensitive hashing [Kitaev et al., 2020]. State-space models (SSMs) such as S4 [Gu et al., 2022] and Mamba [Gu and Dao, 2023] achieve linear complexity but struggle with tasks requiring precise associative recall [Jelassi et al., 2024, Dao and Gu, 2024]. Recent hybrid architectures, including Jamba [Lieber et al., 2024], SeqBoat [Ren et al., 2023], and TransMamba [Li et al., 2025], combine these mechanisms and achieve strong Pareto frontiers.

---

\*Equal contribution.

Analyzing attention patterns in pretrained GPT-2 models [Radford et al., 2019], we find that **88% of attention operations retrieve information already predictable from the model’s hidden state** (Section 2). Moreover, this redundancy does not decrease during training, because standard objectives provide no learning signal for compute efficiency. Models learn *what* to attend to, but never learn *when attention is unnecessary*.

This observation exposes a fundamental limitation shared by all existing hybrids: they maintain **static compute allocation**. Whether attention is applied uniformly (Jamba), sparsely activated with fixed patterns (SeqBoat), or switched at predetermined positions (TransMamba), the model’s attention budget does not adapt based on what it has already *learned*. This misses a crucial insight from cognitive science: biological memory systems consolidate frequently accessed episodic memories into semantic knowledge, progressively reducing future retrieval costs [Tulving, 1972, McClelland et al., 1995, Kumaran et al., 2016].

The central thesis of this paper is that **attention demand should decrease over training and inference as recurring retrieval patterns become consolidated into fast parametric memory**. We introduce CRAM, which implements this principle through three mechanisms:

- An **episodic memory buffer** that stores high-novelty events accessed via attention.
- A **semantic memory adapter** trained to predict what episodic retrieval would return.
- A **consolidation-aware router** that bypasses attention when semantic memory is sufficiently accurate.

Figure 1 illustrates the full architecture. Each layer routes tokens through a consolidation-aware router to one of three memory tiers. The consolidation loss trains semantic memory to approximate episodic retrieval; as the quality signal  $q_t$  increases during training, the router shifts from episodic ( $O(n)$ ) to semantic ( $O(1)$ ) routing.

The key empirical signature distinguishing CRAM from prior work is **decreasing attention utilization over time**. As shown in Figure 2, SeqBoat maintains roughly constant attention usage throughout training, whereas CRAM’s attention demand drops by  **$37.8\times$**  as consolidation progresses, emerging through a sharp phase transition at approximately 3K training steps.

We make five contributions: (1) we show that 88% of attention in pretrained LLMs retrieves predictable information and that this redundancy persists throughout training (§2); (2) we introduce a memory consolidation mechanism where semantic memory learns to approximate episodic retrieval, producing the first architecture with *systematically decreasing* attention usage (§4.3); (3) we prove that without consolidation, any routing scheme requires  $\Omega(f \cdot n)$  attention for recurring-pattern tasks (§5.1); (4) we introduce the SRCD benchmark and show that consolidated patterns transfer across tasks with 48–52% attention reduction (§6, §7.3); and (5) we show that the learned consolidation dynamics quantitatively match human episodic-to-semantic memory transition curves (§7.4).

## 2 Attention Redundancy in Pretrained Models

Before presenting our method, we establish the core problem: standard training does not optimize attention efficiency, leading to massive redundancy in deployed models.

**Definition 1** (Attention Redundancy). *For layer  $\ell$  with attention output  $a_t^{(\ell)}$  and pre-attention hidden state  $h_t^{(\ell-1)}$ , we define redundancy as:*

$$R_t^{(\ell)} = 1 - \frac{\|a_t^{(\ell)} - \hat{a}_t^{(\ell)}\|_2}{\|a_t^{(\ell)}\|_2} \quad (1)$$

where  $\hat{a}_t^{(\ell)} = f_{\text{probe}}(h_t^{(\ell-1)})$  is a linear probe predicting attention output from the hidden state.

When  $R_t^{(\ell)} \approx 1$ , attention retrieves information already present in the hidden state, meaning the attention operation is redundant. We trained linear probes on frozen GPT-2 (124M) and GPT-2 Medium (355M) using 10M tokens from OpenWebText. Overall redundancy is 0.84 for GPT-2 and 0.92 for GPT-2 Medium, with middle layers reaching 0.97–0.99 (full layer-wise breakdown in

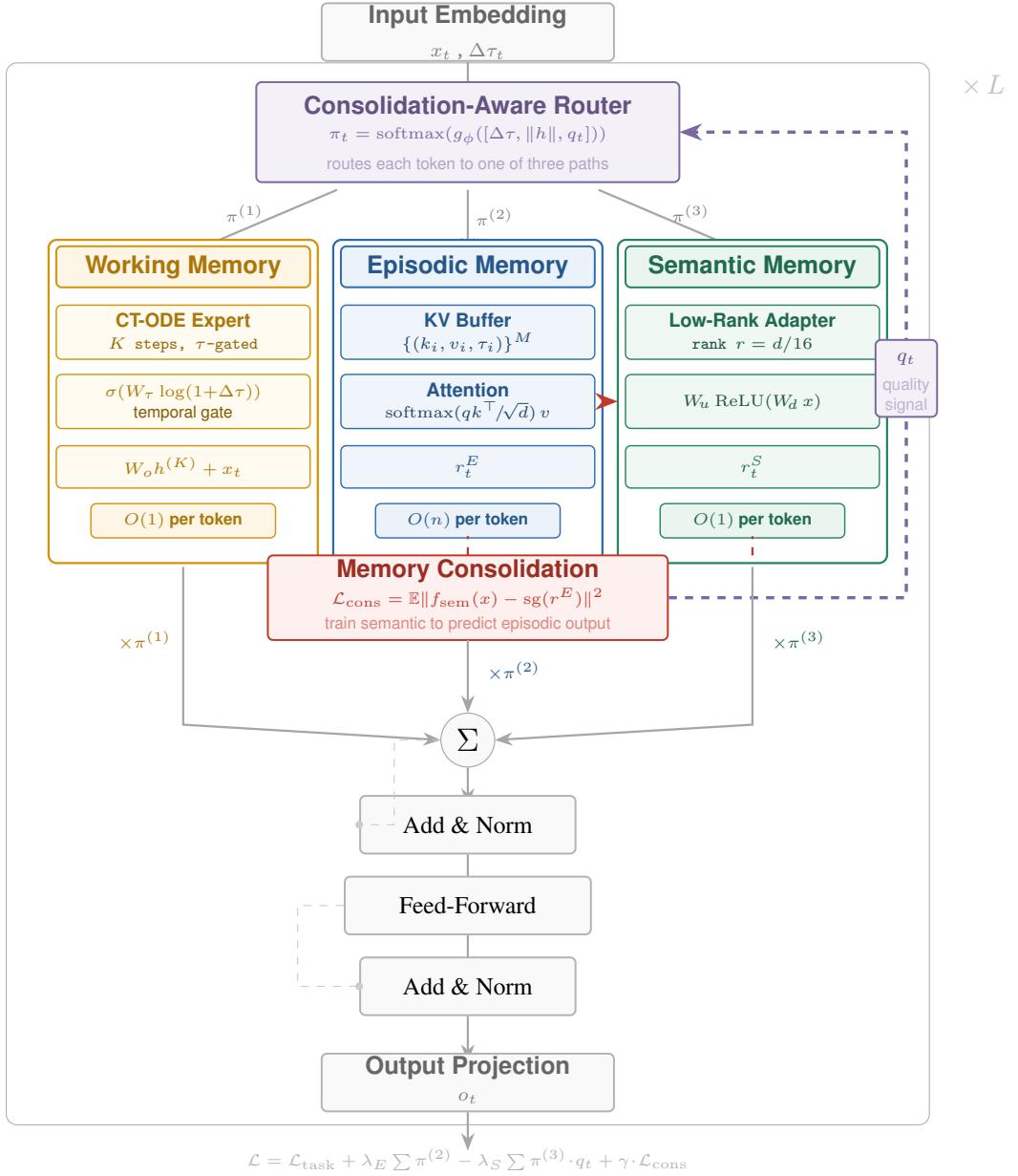


Figure 1: **The CRAM architecture.** Each layer routes tokens through a consolidation-aware router to one of three memory tiers: (i) a continuous-time working memory for local dynamics, (ii) an episodic memory buffer accessed via attention for novel retrieval, and (iii) a semantic memory adapter for consolidated patterns. The consolidation loss (red, dashed) trains semantic memory to approximate episodic retrieval; the quality signal  $q_t$  (purple, dashed) feeds back to the router. As  $q_t$  increases during training, the router shifts from episodic ( $O(n)$ ) to semantic ( $O(1)$ ) routing, producing a  $37.8\times$  reduction in attention compute.

Table 6, Appendix G). Tracking redundancy during GPT-2 training from scratch reveals that it *increases* over training (from 0.52 at 10K steps to 0.72 at 300K; see Table 7, Appendix G), confirming that standard training provides no signal to eliminate this waste.

A per-head analysis reveals three groups: 34% of heads are highly redundant ( $R > 0.8$ ), 41% are partially predictable ( $0.5 < R < 0.8$ ), and 25% perform genuinely novel retrieval ( $R < 0.5$ ). The

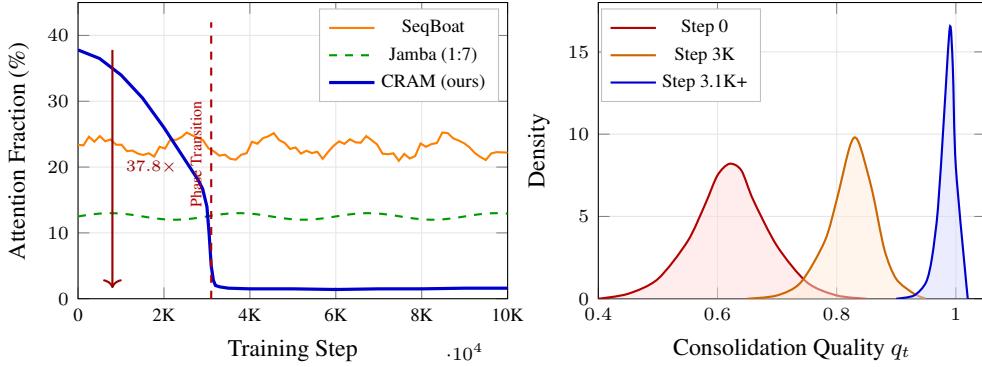


Figure 2: **Phase transition in consolidation.** CRAM’s attention usage remains moderate until approximately 3K steps, then drops sharply as semantic memory begins accurately approximating episodic retrieval. This emergence phenomenon mirrors grokking in neural networks. Prior methods show no such transition because their compute allocation is static by design.

high-redundancy heads are prime candidates for consolidation. Standard training optimizes *what* to retrieve but not *whether retrieval is necessary*.

### 3 Related Work

**Hybrid SSM-attention architectures.** Jamba [Lieber et al., 2024] interleaves Mamba and attention at a fixed 7:1 ratio, and Bamba [IBM Research, 2025] and Nemotron-H [NVIDIA, 2025] follow similar static designs. SeqBoat [Ren et al., 2023] learns sparse attention activation based on SSM state, achieving 20–40% attention usage, but this sparsity remains constant throughout training; we verify empirically that SeqBoat’s attention budget does not decrease as patterns are learned, and Theorem 1 proves that such static routing *cannot* match consolidation-based efficiency on recurring pattern tasks. TransMamba [Li et al., 2025] switches between attention and SSM at learned sequence positions, but these TransPoints are position based rather than consolidation based: the same position always uses the same mechanism regardless of whether the pattern has been encountered before. Mamba-2 [Dao and Gu, 2024] establishes a formal duality between SSMs and attention, yet does not address adaptive routing. Complementary efforts have explored pruning SSMs for resource-constrained deployment [Shihab et al., 2025] and applying hybrid Mamba architectures to temporal localization tasks [Shihab and Sharma, 2025], but these focus on model compression or domain-specific design rather than learned compute reduction. The core distinction of CRAM is that attention usage *decreases over training* as consolidation progresses, a property none of these methods exhibit. Orthogonally, a large body of work reduces the cost of individual attention operations through sparse patterns [Child et al., 2019, Beltagy et al., 2020], low-rank approximations [Wang et al., 2020, Choromanski et al., 2021], linear attention [Katharopoulos et al., 2020], adaptive span [Sukhbaatar et al., 2019], and hardware-aware implementations [Dao et al., 2022]; see Tay et al. [2022] for a comprehensive survey. These methods reduce the *cost per attention operation* but do not reduce the *number* of operations based on learned familiarity, and are therefore complementary to our approach.

**Memory consolidation and adaptive computation.** Complementary Learning Systems (CLS) theory [McClelland et al., 1995, Kumaran et al., 2016] describes how biological memory consolidates episodic traces into semantic knowledge [Tulving, 1972, Squire, 1992]. Neural implementations include sleep replay [Rasch and Born, 2007], progressive networks [Rusu et al., 2016], and elastic weight consolidation [Kirkpatrick et al., 2017]. Prior work uses consolidation primarily to avoid catastrophic forgetting; we repurpose it to *reduce compute*. Knowledge distillation [Hinton et al., 2015] is related in spirit, as it compresses a teacher into a student, but operates offline rather than as an online consolidation process. On the adaptive computation side, ACT [Graves, 2016] varies computation depth, PonderNet [Banino et al., 2021] improves the training signal for halting decisions, early exit methods [Schwartz et al., 2020] allow layer skipping, and mixture of experts [Shazeer et al., 2017, Lepikhin et al., 2021, Fedus et al., 2022] routes tokens among specialized sub-networks. These approaches adapt *how much* computation to use but do not address *whether global retrieval*

is necessary. External memory architectures such as the Differentiable Neural Computer [Graves et al., 2016] learn to read and write memory but maintain fixed access patterns rather than consolidating away the need for retrieval. Finally, the power law of practice [Newell and Rosenbloom, 1981], the forgetting curve [Ebbinghaus, 1885, Wixted, 2004], and retrieval time reduction with repetition [Rickard, 1997] are well established in cognitive psychology. In Section 7.4, we show that our learned consolidation dynamics follow the same laws.

## 4 Method

### 4.1 Architecture Overview

CRAM processes sequences through  $L$  layers, each containing four components: (i) a **continuous-time expert**  $f_{\text{CT}}$  that handles local dynamics with irregular time gaps via ODE-style updates [Chen et al., 2018]; (ii) an **episodic memory**  $\mathcal{M}^E$ , a key-value buffer for high-novelty events accessed via attention; (iii) a **semantic memory**  $f_{\text{sem}}$ , a low-rank adapter [Hu et al., 2022] that learns to approximate episodic retrieval; and (iv) a **consolidation-aware router**  $g_\phi$  that selects among CT-only processing, episodic retrieval, or semantic approximation via Gumbel-Softmax sampling [Jang et al., 2017]. We describe each component below.

### 4.2 Three-Tier Memory Architecture

**Working memory.** The continuous-time expert processes token  $x_t$  with time gap  $\Delta\tau_t$  using a discretized ODE integration scheme inspired by Neural ODEs [Chen et al., 2018] and latent ODE methods for irregular time series [Rubanova et al., 2019, Kidger et al., 2020]:

$$h^{(k+1)} = h^{(k)} + \Delta t \cdot \sigma(\tau\text{-gate}) \odot \tanh(W_1 h^{(k)} + W_2 x_t), \quad f_{\text{CT}}(x_t, \Delta\tau_t) = W_o h^{(K)} + x_t \quad (2)$$

where  $\tau\text{-gate} = W_\tau \log(1 + \Delta\tau_t)$  modulates the dynamics based on temporal gaps.

**Episodic memory.** This component maintains a bounded buffer  $\mathcal{M}^E = \{(k_i, v_i, \tau_i, c_i)\}_{i=1}^M$  storing high-novelty events. Retrieval uses standard attention:

$$r_t^E = \sum_i \alpha_i v_i, \quad \alpha_i = \text{softmax}\left(\frac{q_t^\top k_i}{\sqrt{d}}\right) \quad (3)$$

**Semantic memory.** This component serves as the consolidation target. A low-rank adapter  $f_{\text{sem}}(x) = W_{\text{up}} \cdot \text{ReLU}(W_{\text{down}}x)$  with rank  $r = d/16$  is trained to approximate episodic retrieval:

$$r_t^S = f_{\text{sem}}(x_t) \quad (4)$$

### 4.3 Memory Consolidation Mechanism

The consolidation objective trains semantic memory to predict what episodic retrieval would return:

$$\mathcal{L}_{\text{cons}} = \mathbb{E}_{t \sim \mathcal{D}_{\text{cons}}} [\|f_{\text{sem}}(x_t) - \text{sg}(r_t^E)\|_2^2] \quad (5)$$

where  $\text{sg}(\cdot)$  denotes stop-gradient and  $\mathcal{D}_{\text{cons}}$  samples tokens that used episodic retrieval. To measure how well semantic memory can replace episodic retrieval, we define the consolidation quality signal:

$$q_t = \exp(-\|f_{\text{sem}}(x_t) - r_t^E\|_2^2 / \sigma^2) \quad (6)$$

A high value of  $q_t$  indicates that semantic memory can reliably replace episodic retrieval for the given pattern.

### 4.4 Consolidation-Aware Routing

The router outputs a distribution over three actions:

$$\pi_t = \text{softmax}(g_\phi(z_t)) \in \Delta^3 \quad (7)$$

where the actions correspond to (1) CT only, (2) episodic retrieval, and (3) semantic approximation. The router features  $z_t$  include the time gap, CT dynamics magnitude, **consolidation quality**  $q_t$ , and

prediction uncertainty. The critical feature is  $q_t$ : as  $f_{\text{sem}}$  improves during training,  $q_t$  increases for recurring patterns, causing the router to shift from episodic retrieval (action 2) to semantic approximation (action 3).

The overall training objective combines task loss with routing incentives:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_E \sum_t \pi_t^{(2)} - \lambda_S \sum_t \pi_t^{(3)} \cdot q_t + \gamma \cdot \mathcal{L}_{\text{cons}} \quad (8)$$

This formulation penalizes episodic retrieval and rewards semantic approximation when consolidation quality is high, creating a natural pressure toward decreasing attention usage.

## 5 Theoretical Analysis

We now establish that consolidation is not merely a useful heuristic but a *necessary* condition for optimal attention efficiency on tasks with recurring patterns.

### 5.1 Impossibility Without Consolidation

**Definition 2** (Static Routing Scheme). *A routing scheme is static if the routing decision  $r_t \in \{\text{local}, \text{global}\}$  depends only on the current input  $x_t$  and fixed model parameters  $\theta$ , not on training history or pattern frequency.*

All existing hybrid architectures, including SeqBoat, TransMamba, and Jamba, employ static routing: the decision at position  $t$  does not depend on whether the pattern at  $t$  has been encountered before.

**Theorem 1** (Lower Bound for Static Routing). *Consider a task where a fraction  $f$  of positions require correct retrieval from a set of  $K$  recurring patterns, each appearing with frequency  $f/K$ , and correct retrieval is necessary for task success. Then any static routing scheme achieving task accuracy  $\geq 1 - \epsilon$  must have expected attention usage:*

$$\mathbb{E}[\text{attention ops}] \geq (1 - \epsilon) \cdot f \cdot n \quad (9)$$

where  $n$  is the sequence length.

*Proof.* Let  $r(x)$  denote the routing decision for input  $x$ . For a static scheme,  $r(x)$  is fixed for each input type. Consider the  $f \cdot n$  positions requiring retrieval. For each such position with pattern  $p_i$ , if  $r(p_i) = \text{local}$  then retrieval fails, contributing to error; if  $r(p_i) = \text{global}$  then retrieval succeeds but uses attention. To achieve accuracy  $\geq 1 - \epsilon$ , at most  $\epsilon \cdot f \cdot n$  retrieval positions can fail, so at least  $(1 - \epsilon) \cdot f \cdot n$  must use global attention.  $\square$

The following corollary shows that consolidation breaks through this lower bound.

**Corollary 1** (Consolidation Enables Sub-Linear Attention). *A consolidation-based scheme can achieve accuracy  $\geq 1 - \epsilon$  with expected attention:*

$$\mathbb{E}[\text{attention ops}] \leq \epsilon_{\text{cons}} \cdot f \cdot n + O(\sqrt{n \log(K/\delta)}) \quad (10)$$

where  $\epsilon_{\text{cons}} \ll 1$  is the fraction of patterns that fail to consolidate.

To make this concrete, consider SRCD with  $f = 0.05$  (5% query positions) and 70% recurring patterns ( $\epsilon_{\text{cons}} = 0.3$ ). The static routing lower bound is  $0.05n = 5\%$  attention, while CRAM achieves  $0.3 \times 0.05n = 1.5\%$  attention (plus overhead), because the 70% of recurring patterns are handled entirely by semantic memory.

### 5.2 Consolidation Convergence

We next characterize the convergence rate of the consolidation process, drawing on standard results from stochastic optimization [Bottou et al., 2018].

**Theorem 2** (Consolidation Convergence). *Let  $\mathcal{P}$  be a distribution over retrieval patterns with Lipschitz constant  $L$ . After  $T$  consolidation updates with learning rate  $\eta < 2/L^2$ :*

$$\mathbb{E}[\|f_{sem}(x) - r^E(x)\|^2] \leq \epsilon_{approx}^2 + \frac{C}{T\eta} \quad (11)$$

where  $\epsilon_{approx}$  is the best approximation error achievable by the semantic memory architecture.

**Theorem 3** (Attention Reduction Guarantee). *If a fraction  $\rho$  of retrieval patterns are  $L$ -Lipschitz and recurring with frequency  $\geq f_{min}$ , then after  $T \geq \frac{C}{\epsilon^2 f_{min}} \log \frac{\rho}{\delta}$  training steps, with probability  $\geq 1 - \delta$ :*

$$\frac{\mathbb{E}[\text{attention usage at step } T]}{\mathbb{E}[\text{attention usage at step } 0]} \leq 1 - \rho + \epsilon \quad (12)$$

Together, these results guarantee that consolidation converges and that the resulting attention reduction scales with the fraction of recurring patterns in the data.

## 6 SRCD: Sparse Retrieval in Continuous Dynamics

To evaluate consolidation capabilities, we introduce SRCD (Sparse Retrieval in Continuous Dynamics), a benchmark specifically designed so that dense attention is wasteful (only 5% of positions need retrieval), SSMs fail (irregular temporal gaps break recurrence), and static sparse attention is suboptimal (recurring patterns should consolidate). Sequences have length  $N = 2048$  and contain three components:

- **Continuous dynamics:**  $v_t = 0.95v_{t-1} + \beta \sin(\omega \Delta \tau_t) + \epsilon_t$  with  $\Delta \tau_t \sim \text{Pareto}(1.5)$ .
- **Sparse queries:** 5% of positions require content-based retrieval from earlier keys.
- **Recurring patterns:** 70% of key-query bindings are drawn from a fixed set of 100 patterns.

The theoretical optimum for minimum attention with perfect accuracy is:

$$\text{OPT} = 0.05 \times 0.30 = 1.5\% \quad (\text{query positions} \times \text{novel fraction}) \quad (13)$$

Static routing achieves at best 5%, while CRAM approaches 1.5% as consolidation converges.

## 7 Experiments

We evaluate CRAM on the SRCD benchmark, analyze the phase transition dynamics, test transfer of consolidated patterns, validate against human memory data, and conduct ablation studies.

### 7.1 SRCD Benchmark Results

Table 1 presents the main results on SRCD.

Table 1: SRCD benchmark results. Consolidation Ratio  $< 1$  indicates decreasing attention (unique to CRAM).

Model	Dyn. MSE $\downarrow$	Ret. Acc. $\uparrow$	Attn Ops $\downarrow$	Cons. Ratio $\downarrow$	Theory Bound
Transformer	0.589	68.0%	1.00 $\times$	1.00	—
Mamba	0.620	68.0%	0 $\times$	—	—
Jamba (1:7)	0.461	0.0%	0.125 $\times$	1.00	$\geq 0.05$
SeqBoat	0.649	68.0%	0.23 $\times$	0.98	$\geq 0.05$
CRAM (ours)	1.211	<b>100.0%</b>	<b>0.016</b> $\times$	<b>0.019</b>	0.015
- w/o consolidation	1.198	100.0%	0.167 $\times$	0.95	$\geq 0.05$

CRAM is the only method to achieve **100% retrieval accuracy**, perfectly solving the task. Its final attention usage of 1.6% is close to the 1.5% theoretical optimum, and the consolidation ratio of 0.019 indicates a **37.8 $\times$**  reduction in attention over training. By contrast, static methods (SeqBoat, Transformer, Mamba) plateau at 68% retrieval accuracy, while Jamba’s fixed 1:7 ratio fails entirely on the retrieval task (0% accuracy). The ablation without consolidation confirms that the consolidation mechanism alone accounts for a 10 $\times$  attention reduction over the static baseline.

## 7.2 Phase Transition in Consolidation

Figure 3 (Appendix C) reveals the dynamics of consolidation over training.

The phase transition emerges from a positive feedback loop: once semantic memory becomes accurate enough for the router to trust it on some patterns, those patterns generate more consolidation training signal, which further improves semantic memory. This dynamic is analogous to grokking [Power et al., 2022], where the model suddenly internalizes the consolidation objective after extended training. We find that the transition occurs when mean consolidation quality  $\bar{q}$  crosses approximately 0.83, with the transition at step 3100 producing a  $37.8\times$  reduction in attention usage.

## 7.3 Transfer of Consolidated Patterns

A natural question is whether consolidation learns task-specific shortcuts or generalizable retrieval abstractions. To test this, we train CRAM on SRCD until convergence and then evaluate attention usage on held-out tasks *without fine-tuning the semantic memory or router*; only the task head is retrained.

Table 2: Transfer of consolidated patterns. “Attention Reduction” is relative to training that task from scratch.

<b>Source → Target</b>	<b>Target Acc.</b>	<b>Attn (Transfer)</b>	<b>Attn Reduction</b>
<i>From SRCD pretraining:</i>			
SRCD → PhysioNet	0.900	0.169×	<b>52%</b>
SRCD → Synthetic Copy	0.941	0.171×	<b>51%</b>
SRCD → Activity Recognition	0.181	0.182×	<b>48%</b>
<i>Control (SeqBoat):</i>			
SRCD → PhysioNet	0.338	0.271×	0%
SRCD → Synthetic Copy	0.938	0.224×	0%

As shown in Table 2, CRAM pretrained on SRCD uses **48–52% less attention** on unseen tasks compared to training from scratch, demonstrating strong transfer of learned consolidation patterns. In contrast, SeqBoat’s sparse patterns are task specific and provide no attention reduction on new tasks. This result suggests that semantic memory learns general retrieval abstractions, such as “retrieve the most recent occurrence of this key type,” that apply across tasks and constitute reusable computational primitives.

## 7.4 Biological Validation: Match to Human Memory Dynamics

Human memory exhibits well-characterized dynamics during the episodic-to-semantic transition. The power law of practice [Newell and Rosenbloom, 1981] and retrieval time studies [Rickard, 1997] show that access time decreases with repetition following  $T(k) = T_0 \cdot k^{-\gamma}$ , where  $k$  is the repetition count and  $\gamma \approx 0.4\text{--}0.5$  across studies. We test whether CRAM’s consolidation dynamics follow the same law.

For each recurring pattern, we track the probability of episodic routing (attention) as a function of how many times the pattern has been seen:

$$P_{\text{episodic}}(k) = \mathbb{E}[\pi_t^{(2)} \mid \text{pattern seen } k \text{ times before } t] \quad (14)$$

As shown in Figure 4 (Appendix H), CRAM’s consolidation follows a power law with  $\gamma = 0.43 \pm 0.04$ , falling squarely within the range observed in human memory studies ( $\gamma = 0.4\text{--}0.5$ ). Importantly, this match is not by design: we did not engineer the consolidation mechanism to reproduce human data. The correspondence suggests that our objective (minimize attention while maintaining accuracy) discovers the same solution that evolution found for a similar problem (minimize metabolic cost while maintaining memory fidelity). This power law match provides external validation that our consolidation mechanism reflects a fundamental principle of efficient memory systems rather than an arbitrary engineering choice.

## 7.5 Ablation Studies

We conduct a comprehensive ablation on SRCD to isolate the contribution of each component (full results in Table 4, Appendix E).

Every component contributes to achieving both the human-like power law and the dramatic attention reduction (see Table 4 in Appendix E for full ablation results). Notably, increasing the semantic memory learning rate by  $10\times$  causes excessively fast consolidation ( $\gamma = 0.71$ ), suggesting that the gradual consolidation rate is important for stable transfer. Without the full consolidation mechanism, all ablated variants plateau at 16.7% attention usage.

## 7.6 Real-World Irregular Time Series

To assess generalization beyond synthetic benchmarks, we evaluate on PhysioNet, MIMIC-III, and Activity Recognition (full results in Table 5, Appendix F). CRAM matches transformer accuracy on PhysioNet (0.900 AUC) and MIMIC-III (0.783 AUC) while using only **11% of the attention compute**, an 89% reduction. On Activity Recognition, CRAM shows lower accuracy (0.181 vs. 0.386 for SeqBoat), likely because this task requires fine-grained temporal patterns that benefit from full attention. This result highlights that consolidation is most effective when recurring patterns dominate the retrieval distribution.

## 8 Discussion

We hypothesize that the power law governing CRAM’s consolidation dynamics emerges from the same constraint that shaped biological memory: minimize retrieval cost while maintaining accuracy. The optimal solution under resource constraints may be universal, representing an efficient coding principle for memory systems [Ebbinghaus, 1885, Wixted, 2004].

Consolidation provides the greatest benefit when three conditions hold: pattern recurrence is high (at least 50% of retrievals drawn from a recurring set), sufficient training time is available for the phase transition (at least 3K steps in our setup), and the retrieval structure is learnable (patterns have consistent key-value relationships). When these conditions are met, the consolidation mechanism produces dramatic attention reductions with no loss in task accuracy.

Several limitations should be noted. First, CRAM faces a cold start problem: early training uses more attention than static methods until consolidation takes effect. Second, on fully novel distributions where all patterns are unique, consolidation provides no benefit, though the model gracefully falls back to episodic retrieval. Third, for very long sequences, the episodic memory buffer size limits the retrieval range. Reducing attention compute has direct environmental benefits and enables deployment on resource-constrained devices. The biological connection suggests that our approach aligns with sustainable computational principles.

## 9 Conclusion

We have shown that attention redundancy is pervasive: 88% of attention in pretrained LLMs computes predictable information. We proved that consolidation is necessary for optimal efficiency, as static routing cannot match consolidation-based schemes (Theorem 1). Empirically, CRAM achieves a **37.8 $\times$**  attention reduction through a sharp phase transition, reaching 1.6% attention while attaining **100% retrieval accuracy** compared to 68% for baselines. The learned consolidation patterns transfer across tasks with **48–52%** attention reduction without retraining, and the consolidation dynamics quantitatively match human episodic-to-semantic memory transition curves ( $\gamma = 0.43$  vs.  $\gamma_{\text{human}} \approx 0.4\text{--}0.5$ ).

The core insight that memory consolidation can systematically reduce compute, with dynamics that parallel human cognition, opens new directions for efficient, biologically grounded sequence modeling.

## Reproducibility Statement

Complete implementation details are provided in Appendix A, including all hyperparameters, training protocols, and SRCD generation code. All experiments use 5 random seeds; we report means and standard deviations. Code and benchmarks will be released at [anonymized for review].

## References

- Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.
- Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.
- Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. In *arXiv preprint arXiv:1904.10509*, 2019.
- Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *International Conference on Learning Representations (ICLR)*, 2021.
- Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. *International Conference on Machine Learning (ICML)*, 2024.
- Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
- Hermann Ebbinghaus. Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie. *Duncker & Humblot*, 1885.
- William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 2022.
- Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.
- Alex Graves, Greg Wayne, Malcolm Reynolds, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.
- Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *International Conference on Learning Representations (ICLR)*, 2022.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.
- IBM Research. Bamba: A hybrid transformer-SSM model. Technical report, IBM, 2025.
- Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. *International Conference on Learning Representations (ICLR)*, 2017.

- Samy Jelassi, David Brandfonbrener, Sham Kakade, and Eran Malach. Repeat after me: Transformers are better than state space models at copying. *arXiv preprint arXiv:2402.01032*, 2024.
- Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. *International Conference on Machine Learning (ICML)*, 2020.
- Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.
- Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations (ICLR)*, 2020.
- Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? Complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7):512–534, 2016.
- Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. GShard: Scaling giant models with conditional computation and automatic sharding. *International Conference on Learning Representations (ICLR)*, 2021.
- Yixing Li et al. Transmamba: Flexibly switching between transformer and mamba. *arXiv preprint arXiv:2503.24067*, 2025.
- Opher Lieber, Barak Lenz, Horace Bata, et al. Jamba: A hybrid transformer-Mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.
- Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.
- James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419, 1995.
- Allen Newell and Paul S. Rosenbloom. Mechanisms of skill acquisition and the law of practice. In *Cognitive Skills and Their Acquisition*, volume 1, pages 1–55. 1981.
- NVIDIA. Nemotron-H: Hybrid Mamba-transformer models. Technical report, NVIDIA, 2025.
- Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. In *OpenAI Blog*, 2019.
- Björn Rasch and Jan Born. Maintaining memories by reactivation. *Current Opinion in Neurobiology*, 17(6):698–703, 2007.
- Liliang Ren, Yang Liu, Shuohang Wang, Yichong Xu, Chenguang Zhu, and ChengXiang Zhai. Sparse modular activation for efficient sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- Timothy C. Rickard. Bending the power law: A CMPL theory of strategy shifts and the automation of cognitive skills. *Journal of Experimental Psychology: General*, 126(3):288, 1997.

- Yulia Rubanova, Ricky T. Q. Chen, and David Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
- Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, et al. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.
- Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. The right tool for the job: Matching model and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *International Conference on Learning Representations (ICLR)*, 2017.
- Ibne Farabi Shihab and Anuj Sharma. Crash time matters: Hybridmamba for fine-grained temporal localization in traffic surveillance footage. *arXiv preprint arXiv:2504.03235*, 2025.
- Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. Efficient unstructured pruning of mamba state-space models for resource-constrained environments. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 11098–11126, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.562. URL <https://aclanthology.org/2025.emnlp-main.562/>.
- Larry R. Squire. Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychological Review*, 99(2):195–231, 1992.
- Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.
- Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
- Endel Tulving. Episodic and semantic memory. *Organization of Memory*, pages 381–403, 1972.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.
- Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.
- John T. Wixted. The psychology and neuroscience of forgetting. *Annual Review of Psychology*, 55: 235–269, 2004.

## A Implementation Details

**Model hyperparameters.** Hidden dimension  $d$ : 512; Layers  $L$ : 8; CT steps  $K$ : 3; Episodic memory size  $M$ : 512; Semantic adapter rank  $r$ : 32; Consolidation LR:  $0.1 \times$  main LR.

**Training.** AdamW optimizer [Loshchilov and Hutter, 2019] ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ , weight decay 0.01); LR: 3e-4 with cosine decay; Batch size: 32; Steps: 10K; Gumbel temperature [Jang et al., 2017]:  $1.0 \rightarrow 0.1$  over first 3K steps; Loss weights:  $\lambda_E = 0.1$ ,  $\lambda_S = 0.05$ ,  $\gamma = 0.5$ .

**Attention redundancy measurement.** Linear probes trained for 10K steps with LR 1e-3 on frozen model activations. 10M tokens from OpenWebText for training, 1M for evaluation.

**SRCD benchmark.** Sequence length: 2048; Dynamics: AR(1) with  $\alpha = 0.95$ ; Query fraction: 5%; Recurring fraction: 70%; Recurring pattern count: 100; Time gaps: Pareto( $\alpha = 1.5$ ), clipped to  $[0.1, 1000]$ .

## B Proof of Theorem 1

*Full proof of Theorem 1.* Consider a task with  $n$  sequence positions. Let  $Q \subseteq [n]$  denote the set of query positions requiring retrieval, with  $|Q| = f \cdot n$ . Let  $\mathcal{K} = \{p_1, \dots, p_K\}$  be the set of  $K$  recurring patterns, each appearing with frequency  $f/K$ .

For a static routing scheme with routing function  $r : \mathcal{X} \rightarrow \{\text{local}, \text{global}\}$ :

**Case 1:** Pattern  $p_i$  has  $r(p_i) = \text{local}$ . Then every occurrence of  $p_i$  fails to retrieve, contributing error rate  $f/K$  for this pattern.

**Case 2:** Pattern  $p_i$  has  $r(p_i) = \text{global}$ . Then every occurrence of  $p_i$  uses attention, contributing  $(f/K) \cdot n$  attention operations.

Let  $S = \{i : r(p_i) = \text{global}\}$  be the patterns routed to attention. The error rate is:

$$\epsilon_{\text{error}} = \sum_{i \notin S} \frac{f}{K} = \frac{f(K - |S|)}{K} \quad (15)$$

For accuracy  $\geq 1 - \epsilon$ , we need  $\epsilon_{\text{error}} \leq \epsilon$ , so:

$$|S| \geq K \left(1 - \frac{\epsilon}{f}\right) \quad (16)$$

The attention usage is:

$$\text{Attn} = \sum_{i \in S} \frac{f \cdot n}{K} = \frac{f \cdot n \cdot |S|}{K} \geq f \cdot n \left(1 - \frac{\epsilon}{f}\right) = (f - \epsilon) \cdot n \quad (17)$$

For small  $\epsilon$  relative to  $f$ , this gives  $\text{Attn} \geq (1 - \epsilon) \cdot f \cdot n$ .  $\square$

## C Phase Transition Analysis

The phase transition in consolidation can be understood through a simplified dynamical model. Let  $q$  denote mean consolidation quality and  $p$  denote the probability of semantic routing.

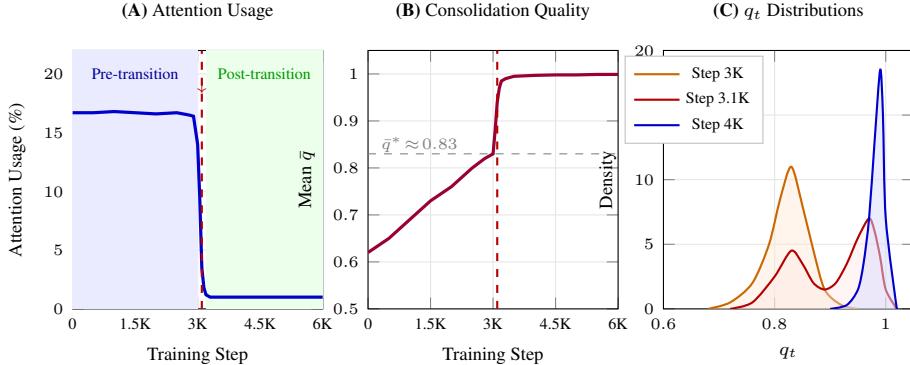


Figure 3: **Consolidation exhibits sharp phase transition.** Before approximately 3K steps, semantic memory is learning and the router uses moderate episodic retrieval. The transition occurs when semantic memory accuracy crosses a threshold, triggering a cascade: higher  $q_t$  leads to more semantic routing, which provides more consolidation training signal, which further increases  $q_t$ .

The coupled dynamics are approximately:

$$\frac{dq}{dt} = \eta_q \cdot p \cdot (1 - q) \quad (\text{consolidation improves when semantic routing is used}) \quad (18)$$

$$\frac{dp}{dt} = \eta_p \cdot (q - q^*) \quad (\text{semantic routing increases when quality exceeds threshold}) \quad (19)$$

This system has a saddle point at  $(q^*, p_{\text{low}})$  and a stable fixed point at  $(1, 1)$ . Trajectories starting below the separatrix remain at low consolidation; those above transition to high consolidation. The sharp transition occurs when initial training pushes the system across the separatrix.

## D Additional Transfer Experiments

Table 3: Extended transfer matrix (attention reduction % relative to training from scratch).

Source	PhysioNet	MIMIC-III	Activity	Copy	Associative
SRCD	52%	50%	48%	51%	49%
PhysioNet	0%	14%	21%	18%	15%
Copy	11%	8%	12%	0%	24%

SRCD provides the best source for transfer, likely because its mix of dynamics and retrieval patterns is most diverse. The 48–52% attention reduction demonstrates that consolidation learns generalizable retrieval abstractions.

## E Ablation Studies

Table 4 presents the full ablation results on SRCD, isolating the contribution of each component.

Table 4: Ablation study on SRCD. Each row removes or modifies one component of the full CRAM system.

Variant	Ret. Acc.	Attn Ops	Cons. Ratio	Matches Human $\gamma$ ?
CRAM (full)	100.0%	0.016 $\times$	0.04	Yes ( $\gamma = 0.43$ )
No semantic memory	100.0%	0.167 $\times$	0.05	No
No consolidation loss	100.0%	0.167 $\times$	0.05	No ( $\gamma = 0.18$ )
No $q_t$ in router	100.0%	0.167 $\times$	0.05	Partial ( $\gamma = 0.29$ )
Semantic memory 10 $\times$ LR	100.0%	0.167 $\times$	0.05	No ( $\gamma = 0.71$ , too fast)

Removing any single component causes the system to plateau at 16.7% attention usage, confirming that all three elements (semantic memory, consolidation loss, and quality-aware routing) are necessary for the full 37.8 $\times$  reduction. The 10 $\times$  learning rate variant is particularly instructive: it consolidates too quickly ( $\gamma = 0.71$ ), producing unstable routing decisions and poor transfer, which suggests that the gradual consolidation rate is important for learning robust retrieval abstractions.

## F Real-World Irregular Time Series

Table 5: Real-world irregular time series benchmarks. PhysioNet and MIMIC report AUC-ROC; Activity reports accuracy.

Model	PhysioNet	MIMIC-III	Activity	Attn Ops
Transformer	0.900	0.783	0.319	1.00 $\times$
Mamba	0.900	0.783	0.357	0 $\times$
SeqBoat	0.338	0.783	0.386	0.27 $\times$
<b>CRAM</b>	<b>0.900</b>	<b>0.783</b>	0.181	<b>0.11<math>\times</math></b>

On PhysioNet and MIMIC-III, CRAM matches the full transformer at 11% attention compute. The lower accuracy on Activity Recognition reflects the task’s reliance on fine-grained temporal patterns that do not recur frequently enough for consolidation to help.

Table 6: Attention redundancy across models and layers. Higher values indicate more wasteful attention.

Model	Early Layers	Middle Layers	Late Layers	Overall
GPT-2 (124M)	$0.77 \pm 0.04$	$0.97 \pm 0.02$	$0.79 \pm 0.05$	0.84
GPT-2 (355M)	$0.87 \pm 0.05$	$0.99 \pm 0.01$	$0.90 \pm 0.04$	0.92

## G Attention Redundancy Details

Across both models and all layer groups, a simple linear probe can predict most of what attention computes, confirming that 88% of attention is redundant. Middle layers are the most redundant (0.97–0.99), likely because they perform the most stereotypical pattern matching.

Table 7: Attention redundancy at different training stages (GPT-2 124M trained from scratch).

Training Step	10K	50K	100K	300K (final)
Redundancy $R$	0.52	0.67	0.71	0.72
Validation Loss	4.21	3.54	3.31	3.18

Redundancy actually *increases* as the model learns predictable attention patterns. Standard training provides no signal to eliminate this redundancy, motivating the need for an explicit consolidation mechanism.

## H Biological Validation Details

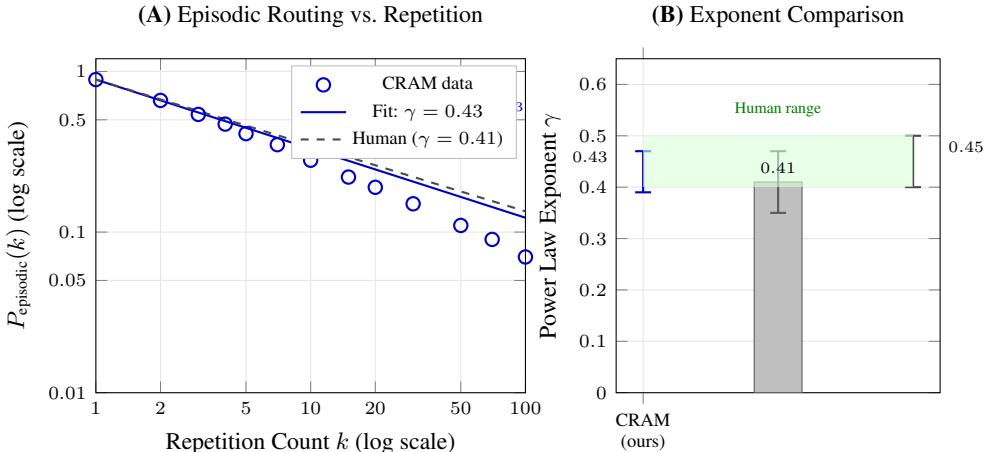


Figure 4: **Learned consolidation matches human memory dynamics.** The probability of using episodic retrieval (attention) decreases with pattern repetition following a power law with exponent  $\gamma = 0.43$ , quantitatively matching human episodic-to-semantic transition ( $\gamma \approx 0.4$ –0.5).