

# 000 SEMI-PARAMETRIC LANGUAGE MODEL WITH SELEC- 001 002 TIVE MEMORY 003 004

005 **Anonymous authors**

006 Paper under double-blind review

## 007 008 ABSTRACT 009

011 Pretrained on trillions of tokens, LLMs are known for their ability to store a  
012 large amount of factual knowledge in their parametric memory. However, re-  
013 calling facts from this memory is known to be unreliable, particularly for long-  
014 tail knowledge—obscure facts infrequently mentioned in training data. Although  
015 retrieval-augmented generation (RAG) is the standard solution, it introduces over-  
016 heads such as increased inference costs due to longer input contexts; preprocess-  
017 ing and indexing extensive document collections also adds additional engineering  
018 complexity. In this work, we propose a novel approach to improve the factuality  
019 of LLMs on long-tail knowledge. We begin by identifying *atomic facts*—short  
020 statements detailing relationships of some entities—that are not present in a pre-  
021 trained LLM’s parametric memory. These facts are then stored in an external,  
022 non-parametric memory. Subsequently, the model undergoes continuous pretrain-  
023 ing, enabling it to learn *when* to consult this external memory at inference time.  
024 Compared with existing approaches, our method uses a compact external mem-  
025 ory that selectively stores only the atomic facts not known to the LLM, resulting  
026 in minimal additional inference-time costs in terms of both time and space. Fur-  
027 thermore, our method outperforms fully trained models of comparable size on  
028 knowledge-intensive benchmarks by more than 10% on some benchmarks and  
029 achieves competitive results against larger models.

## 030 031 1 INTRODUCTION

032 Large language models (LLMs) have demonstrated an outstanding ability to learn a substantial  
033 amount of world language from their training corpus, storing this knowledge in their parameters  
034 and excelling in a wide range of applications. However, despite these advanced capabilities, LLMs  
035 frequently encounter the problem of hallucination, particularly when dealing with long-tail knowl-  
036 edge that is less represented in their training data (Mallen et al., 2023; Asai et al., 2023; Kandpal  
037 et al., 2023; Wei et al., 2024).

038 Recent work has explored integrating external memory into language models to improve factuality  
039 and reduce memorization at inference time (Li et al., 2025; Mallen et al., 2023; Li et al., 2024;  
040 He et al., 2023). However, a key limitation of this training-free approach is the need for a com-  
041 plex proposal procedure to decide between generating or retrieving from memory. Another line of  
042 work augments language model training with retrieval-based memory (Guu et al., 2020; Borgeaud  
043 et al., 2021). This type of methods force all retrieved knowledge into model parameters, which adds  
044 significant overhead during training and inference. Memory<sup>3</sup> (Yang et al., 2024) integrates exter-  
045 nalized knowledge into attention layers by compressing key value representations. More recently,  
046 Large Memory Language Model (LMLM) (Zhao et al., 2025) introduced a pretraining recipe that  
047 stores factual knowledge both in model weights and in an external database. While effective, LMLM  
048 does not differentiate long-tail knowledge from sufficiently learned knowledge, which can lead to  
049 unnecessary retrievals and even degrade downstream performance.

050 A central challenge still remains: when and how to decouple long-tail knowledge during pre-training  
051 to improve factuality without sacrificing other capabilities. Offloading knowledge too early may  
052 limit the model’s ability to form long-range associations between knowledge entities, while offload-  
053 ing too late may fail to sufficiently remove long-tail knowledge from the model parameters.

We address this with continual pretraining and selective memory, which uses partially trained models to detect knowledge gaps and adaptively externalize them. Our method begins by employing model-based or frequency-based methods to score long-tail knowledge within the training corpus. We preprocess the corpus into interleaved sequences of standard text and extracted knowledge segments, where the long-tail knowledge is stored in an external non-parametric memory. During continued pretraining, we implement an adaptive masking strategy: when the cross-entropy loss for a knowledge segment exceeds a predefined threshold, indicating potential memorization difficulty or long-tail knowledge, we mask these segments from the next-token prediction objective. To enable effective retrieval from this external memory during inference, we also finetune a lightweight query adapter that learns to generate appropriate query representations and maintains the separation between parametric knowledge (encoded in model weights) and non-parametric knowledge (stored in external memory).

Our method demonstrates superior performance on memory-intensive benchmarks compared to fully trained models of equivalent size, and even achieves competitive results compared to larger models. Compared to full offloading, this approach achieves a favorable trade-off: it preserves performance on general and reasoning-heavy benchmarks while shows clear improvements on long-tail knowledge tasks.

## 2 SEMI-PARAMETRIC LANGUAGE MODELS (SPLM)

### 2.1 IDENTIFY LONG-TAIL KNOWLEDGE

Common knowledge—frequent facts and patterns—are best captured in the model’s parametric weights, allowing for fast, generalizable inference. Offloading these to external memory can force the model to rely on retrieval for even simple knowledge, increasing inference time and reducing generalizability. On the other hand, long-tail knowledge is difficult to internalize, requires repeated exposures, and offers little generalization benefit, yet it consumes disproportionate model capacity. Following Zhao et al. (2025), we pre-processed the training corpus into interleaved regular text and memory segments using a finetuned extraction model. Each memory segment contains a query–answer pair wrapped in special tokens: `m_start` [query text] `m_retrieve` [answer text] `m_end`, where `m_start` marks the beginning of a memory segment, and the `m_retrieve` token indicates the start of the answer which should be looked up from external memory. For example, the sentence “Sugeno received the IEEE Frank Rosenblatt Award in 2010 for his contributions to the field of fuzzy systems.” is preprocessed as follows:

Sugeno received the `m_start` Michio Sugeno Award Received `m_retrieve` IEEE Frank Rosenblatt Award `m_end` IEEE Frank Rosenblatt Award in `m_start` Michio Sugeno Year of Award `m_retrieve` 2010 `m_end` 2010 for his contributions to the field of fuzzy systems.

In addition, we score each memory segments in terms of how rare or hard the knowledge segments. We experiment with a few variants for proxy of long tail knowledge, including model based (loss of answer tokens) and model agnostic (frequency of entities). We leave the discussion on this part to Section 4.2.

### 2.2 TRAINING WITH SELECTIVE MEMORY

To prevent the model from inefficiently memorizing these rare or difficult knowledge pieces, we selectively delegate them to an external memory. In practice, this is achieved by masking out the answer tokens for hard knowledge segments during continued pre-training, so the model is trained to intermix free text generation with knowledge retrieval calls. This selective masking ensures that the model’s parametric capacity is focused on common knowledge, while long-tail knowledge is handled via retrieval.

**Selective Memory** Offloading all knowledge to memory can bloat the retrieval store with trivial facts, slow down inference, and make the system brittle when retrieval fails. Prior work has shown that pretrained models’ loss is a reliable proxy for knowledge gaps Feng et al. (2024), where persistent high loss usually indicates unreliable memorization. For a given memory segment  $s_i$ , let  $A_i$  denote the set of token positions corresponding to its answer text. We compute

108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120

Sugeno received the IEEE Frank Rosenblatt | start Michio Sugeno  
Award received | retrieve IEEE Frank Rosenblatt end in  
start Michio Sugeno Year of award | retrieve 2010 end  
2010 for his contributions to the field of | start Frank  
Rosenblatt Award Field | retrieve fuzzy systems end  
fuzzy systems.

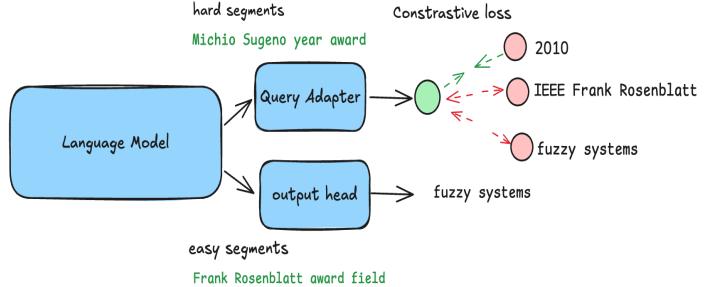


Figure 1: Training with selective memory and query adapter. Easy memory segments are trained directly with the standard next-token prediction loss, while hard segments are routed through a lightweight query adapter. The query adapter is optimized with a contrastive loss: embeddings for query span is pulled closer to the correct answer embedding (positive) and pushed away from incorrect answers (negatives) drawn from other segments within the same document.

the average negative log-likelihood of the answer tokens  $\ell_i$  as a proxy for the difficulty of the memory segment. We define hard memory segments as those with average loss above a threshold  $S_H = \{s_i : \ell_i > \tau\}$ . If the model finds a particular memory segment consistently difficult (loss beyond  $\tau$ ), we mark that segment as a candidate for externalization. In practice, we experiment with either a fixed threshold (only mask answer span when the loss is above a fixed threshold) or an adaptive threshold (mask answer span when the loss is in the top  $p\%$  of all the memory segments seen so far in training). For a memory segment of length  $T$ , we introduce a binary mask  $M = [m_1, m_2, \dots, m_T]$  where  $m_t = 1$  if  $s_i \in \{S_H\}$  and  $x_t$  is between [m\_start]. We apply this mask to the language modeling objective, so that the model is trained only on unmasked tokens.  $\mathcal{L}_{\text{masked}} = -\frac{1}{\sum_{t=1}^T m_t} \sum_{t=1}^T m_t \cdot \log P(x_t | x_{<t}; \theta)$ . This effectively removes the loss contributions from the answer spans of hard memory segments, discouraging the model from attempting to memorize them and instead encouraging reliance on external memory for factual recall. This adaptive masking mechanism enables the model to allocate its parametric capacity to common knowledge and compositional reasoning, while selectively offloading long-tail or hard-to-learn knowledge to a non-parametric memory.

**Query adapter** While the above mechanism teaches the model when to retrieve, we also need to train how to retrieve the correct information from the external memory. To enable efficient and robust retrieval at inference time, we introduce a lightweight, contextualized MLP query adapter that projects both query context and answer text into a shared embedding space where vectors of relevant pairs of query and answer are close to each other. The adapter is trained to map the high-dimensional contextual representation of a query (the pooled embeddings over the query span) to a lower-dimensional dense query embedding that can be used to retrieve relevant answers. The final hidden representation at the position of the [m\_retrieve] token is passed through the adapter to obtain a dense query embedding  $z_{q_i}$ . Similarly, for each memory segment’s answer, we obtain a dense answer embedding by applying a pooling function over the LLM’s final hidden states for the answer text to obtain  $z_{a_i}$ . We train the adapter using a contrastive InfoNCE loss: For a batch of  $N$  memory segments (i.e.,  $N$  query–answer pairs  $(q_i, a_i)$ ), we treat each  $(q_i, a_i)$  as a positive pair and all other  $N - 1$  answers in the batch as negatives for  $q_i$ :

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp(\text{sim}(z_{q_i}, z_{a_i})/\tau)}{\sum_{j=1}^N \exp(\text{sim}(z_{q_i}, z_{a_j})/\tau)}, \quad (1)$$

where  $\text{sim}(u, v) = \frac{u^\top v}{\|u\| \|v\|}$  denotes the cosine similarity between vectors and  $\tau > 0$  is a temperature hyperparameter controlling the sharpness of the distribution. We leverage in-batch negatives during training, since each batch typically contains multiple memory segments extracted from the same document or entity, naturally serving as hard negative examples. The full training process is

162 described in Figure-1. Finally, the adapter loss is added on top of the training loss through:  
 163

$$\mathcal{L} = \mathcal{L}_{\text{masked}} + \alpha_{\text{adapter}} \mathcal{L}_{\text{InfoNCE}} \quad (2)$$

166 **Inference** At inference time, if the model outputs the special `m_retrieve` token, this signals that  
 167 a memory retrieval is needed for the current query. At that point, we take the model’s current hidden  
 168 state over the query span and feed it through the adapter to get a contextualized query embedding  
 169  $z_q$ . This embedding is then used to perform a nearest-neighbor search in the external memory store  
 170 for the most relevant stored answer.

## 172 3 EXPERIMENTS

### 173 3.1 EXPERIMENT SETUP

176 **Training details** We pretrained a baseline LLAMA-3-1B model for 150k steps on the DCLM  
 177 corpus (220B tokens). We continued pretraining on the Wikipedia subset of the Dolmino corpus  
 178 (3.7B tokens) for an additional 20k steps, using a learning rate of  $4 \times 10^{-4}$ , batch size of 4, and  
 179 sequence length of 4,096. For the query adapter, we use a two-layer bottleneck MLP where both the  
 180 hidden dim and output dim is set 384. The loss weight for the adapter ( $\alpha_{\text{adapter}}$ ) is set to be 1.

181 **Inference details** For generation, we compare using an off-the-shelf sentence embedding  
 182 model (all-MiniLM-L6-v2 (Reimers & Gurevych, 2019; Wang et al., 2020)) to embed queries  
 183 against using a trained query adapter. For the query adapter, we build the retrieval store for each  
 184 memory segment in the corpus, where answer embedding generated by the adapter is used as re-  
 185 trieval key, and the answer tokens are used as retrieval value. For the sentence embedding model,  
 186 we use a retrieval threshold of 0.6, and for the query adapter we apply a higher threshold of 0.7.  
 187 To support efficient approximate nearest-neighbor search, we employ a FAISS IVF index (John-  
 188 son et al., 2019), which clusters vectors into 16,384 centroids, and we set `nprobe`= 64 during  
 189 retrieval to balance recall and efficiency. For all benchmarks, we use answer level recall (ALR) as  
 190 the evaluation metrics.

### 191 3.2 EVALUATIONS

194 We evaluate our models on a set of benchmarks designed to test general or long-tail factual knowl-  
 195 edge in language models. For general-purpose QA, we adopt widely used benchmarks that serve as  
 196 guardrails for assessing open-domain question answering: TRIVIAQA (TQA) (Joshi et al., 2017),  
 197 NATURALQUESTION (NQ) (Kwiatkowski et al., 2019), ENTITYQ (Sciavolino et al., 2021) and  
 198 HOTPOTQA (HQA) (Yang et al., 2018). These datasets emphasize broad coverage of knowledge  
 199 and multi-hop reasoning, making them representative tests of general QA ability.

200 To specifically target long-tail factual knowledge, we use a suite of benchmarks including  
 201 POPQA (Mallen et al., 2023), HEAD-TO-TAIL (Sun et al., 2024), and SIMPLEQA (Wei et al., 2024).  
 202 POPQA is curated to cover questions on widely recognized entities and facts, while also incorpor-  
 203 ating a substantial proportion of less common, long-tail items. HEAD-TO-TAIL explicitly measures  
 204 the popularity of entities: it consists of 18K question–answer pairs categorized into *head* (frequent),  
 205 *torso* (moderately frequent), and *tail* (rare) entities. SIMPLEQA is a fact-seeking benchmark of  
 206 short, unambiguous questions designed to challenge SOTA models.

### 207 3.3 BASELINES

209 **Vanilla training** As a baseline, we continued pretraining the model on the original Wikipedia sub-  
 210 set of Dolmino for the same number of steps, without introducing any external memory or selective  
 211 masking. This setup measures the effect of additional domain-adapted pretraining alone, isolating  
 212 gains that come purely from further exposure to the corpus.

214 **Full Memory** Following the setup in the LMLM framework (Zhao et al., 2025), we construct a  
 215 memory-only baseline in which all factual tokens are offloaded to external memory. In this setting,  
 the model never learns to parametrize knowledge internally; instead, retrieval is performed at every

216 `m_retrieve` call using a separate embedding model and fuzzy string matching against the memory  
 217 index.  
 218

219 **3.4 MAIN RESULTS**  
 220

221 <b>Model</b>	222 <b>TQA</b>	223 <b>NQ</b>	224 <b>HotpotQA</b>	225 <b>EntityQ</b>	226 <b>PopQA</b>	227 <b>Head-to-Tail</b> All	228 <b>SimpleQA</b>
229 LLaMA3 8B	62.92	32.05	26.16	31.16	22.65	14.77	10.74
229 LLaMA3 1B	40.94	18.48	21.01	14.12	14.30	7.78	6.22
229 LLaMA3 1B - 150k	31.72	17.37	18.10	17.34	14.54	7.51	5.30
229 +Vanilla training	23.72	9.39	13.45	12.43	11.13	6.26	4.49
229 +Memory	24.64	14.71	13.06	23.95	23.04	10.29	10.09
229 +SM (75%)	26.79	16.04	15.72	25.84	21.81	11.37	10.58
229 +SPLM	<b>28.65</b>	<b>16.24</b>	<b>16.86</b>	<b>26.38</b>	<b>25.37</b>	11.3	10.28
							<b>11.35</b>

231 Table 1: Answer-level recall (ALR) on open-domain QA benchmarks. The baselines include  
 232 LLaMA3 8B, LLaMA3.2 1B, and a partially trained LLaMA3 1B (150k steps). For selective mem-  
 233 ory (SM), we report the result with an adaptive threshold of 0.75 on quantile of average answer  
 234 token losses (25% of hard segments are included in memory).

235 In Table 1 we show the results on the answer-level recall (ALR) of a LLaMA-3 1B trained under  
 236 different paradigms: continual pretraining on regular corpus (vanilla training), continual pretraining  
 237 with fully offloaded memory (Memory), continual pretraining with selective memory (SM), and our  
 238 full approach with selective memory combined with a query adapter (SPLM).  
 239

240 **Continual pretraining with memory outperforms fully trained models on knowledge intensive**  
 241 **tasks** Compared to a fully trained Llama 1B model, training with memory demonstrates super-  
 242 ior performance particularly on benchmarks that emphasize long-tail or entity-centric knowledge,  
 243 such as EntityQ (17.34% for fully trained model vs), PopQA (14.54% for fully trained model vs  
 244 23.04% with memory), and head-to-tail. It also achieves comparable results compared to a much  
 245 bigger model (Llama 8B), indicating that externalizing long-tail knowledge can enhance factual  
 246 recall without increasing model size.  
 247

248 **Selective memory outperforms full knowledge offloading** We also observe clearly improve-  
 249 ments when using selective memory compared to fully offloading memory. While offloading all  
 250 memory improves recall on certain long-tail benchmarks such as PopQA, Head-to-tail and Sim-  
 251 pleQA, it leads to degradation on general QA datasets (TQA, NQ) or those require multi-hop rea-  
 252 soning (HotpotQA). In contrast, selective memory achieves a better balance: it maintains com-  
 253 petitive performance on general QA datasets while further improves the performance on long-tail  
 254 benchmarks. This shows that adaptively masking only the hardest segments allows the model to pre-  
 255 serve its parametric capacity for common knowledge and reasoning, while only relying on external  
 256 memory for rare knowledge. Finally, incorporating a lightweight query adapter on top of selective  
 257 memory (SPLM) recovers much of the degradation observed on general QA benchmarks while fur-  
 258 ther improving long-tail factual recall. We observe the degradation on certain benchmarks is a result  
 259 of the model generating imperfect and unnecessary retrieval as shown in the sample genera-  
 260 tions in Appendix C. Selective memory combined with query adapter mitigates this issue by learning to  
 261 produce contextualized, discriminative, and richer query embeddings that are naturally aligned with  
 the model’s internal representation.  
 262

263 **4 ABLATIONS**  
 264

265 **4.1 WHEN SHOULD WE DECOUPLE FACTUAL KNOWLEDGE?**  
 266

267 **LMs are inefficient in memorizing factual knowledge in parametric memory** In Figure 2 we  
 268 plot the answer-level recall (ALR) of a Llama3 1B and Llama 8B model trained on DCLM corpus as  
 269 training progresses. We observe that some benchmarks exhibit clear improvements with additional  
 training, while others remain stagnant. For example, NQ and HotpotQA continue to benefit from

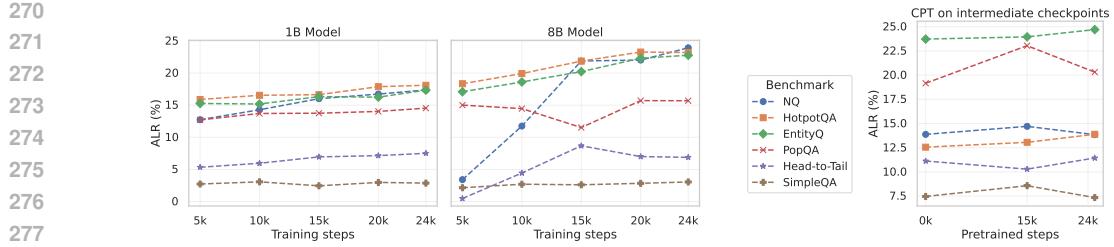


Figure 2: Left: Answer-level recall (ALR) across training steps when the model is trained on regular corpus. Right: Continual pretraining (CPT) on memory corpus starting from an pretrained model at intermediate checkpoints.

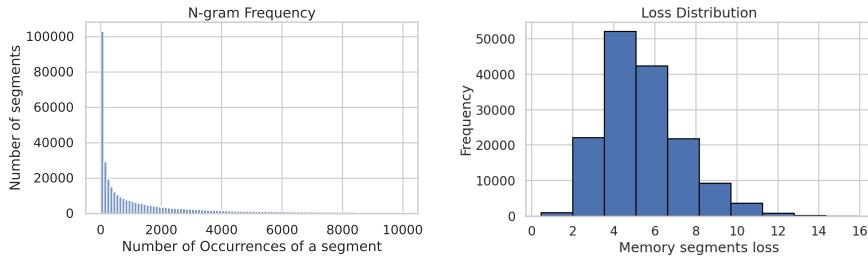


Figure 3: Model based and model agnostic scoring for long-tail knowledge.

longer training, showing steady gains up to 240k steps. By contrast, simpleQA and popQA plateau quickly, indicating that continued training on the same corpus does little to improve performance on factual benchmarks. This is consistent with previous observation Chang et al. (2024) that factual knowledge are prone to forgetting and pretraining on more data shows no significant improvement.

**Pretraining vs. Continual Pretraining** Decoupling long-tail facts to external memory can be done during pre-training or during continuous pre-training time. However, it's hard to separate "factual memory" from "reasoning" in a clean way. Many reasoning capabilities are scaffolded and interleaved with factual text. For example, solving multi-step questions in math often relies on domain-specific facts or theorems embedded in the training data. If large spans of such text are masked, the model may be under-trained on certain compositional patterns, hurting downstream reasoning. In figure 2, we plot the performance of the final model when starting from an intermediate model trained on regular corpus for  $x$  steps and then continual pretrain on memory corpus for 20k steps. We observed that the final performances when training from an intermediate checkpoints are better than training from memory corpus from scratch.

#### 4.2 DIFFERENT PROXIES FOR LONG-TAIL KNOWLEDGE

In this section, we compare model-based proxies and model-agnostic proxies for detecting long-tail knowledge in a corpus. Model-agnostic approaches estimate rarity directly from corpus statistics. One effective tool is *Infinigram* (Liu et al., 2024), which computes exact  $n$ -gram frequency distributions in a given corpus. We count the frequency if either the objective or the subjective entity in the memory segment appear in the corpus. While Infinigram provides an exact measure of  $n$ -gram frequencies, its distribution is extremely skewed (Figure 3), making it challenging to draw a clear boundary between head and long-tail knowledge. Alternatively, we also experimented with model based scoring, where we prompt the Llama3-70B model (see Appendix for prompt details) to assign a popularity score to each knowledge segment on a scale from 1 to 10, with higher scores corresponding to more frequently occurring knowledge and lower scores marking long-tail or rare facts. We observe the same trend: the model assesses most of the knowledge segments as long-tail knowledge.

324 On the other hand, model based proxies such as training loss provides a natural signal of difficulty  
 325 and has been widely used for curriculum learning. Figure 3 shows the distribution of memory  
 326 segment losses at the start of continual pretraining: most losses concentrate around moderate values,  
 327 while the long tail represents knowledge the model struggle to internalize.

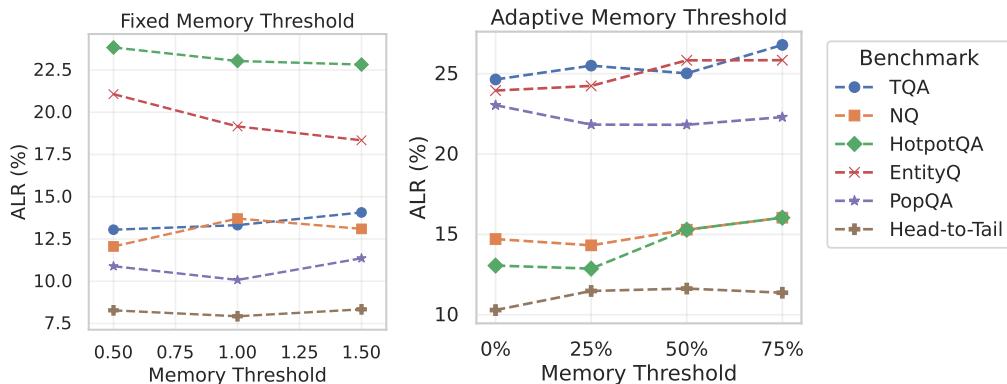
328 As shown in 3, knowledge frequency tend to display Zipf law, where most memory segments are rare  
 329 and low frequency. In contrast, the distribution under loss-based scoring is closer to normal, with  
 330 many segments falling into a moderate difficulty range rather than being pushed to the extremes.  
 331 This discrepancy suggests that, despite the frequency of appearing in the corpus, the model exhibits  
 332 heterogeneous difficulty in internalizing difficult factual knowledge. This motivates us to use model-  
 333 based proxies at continual pretraining time to distinguish hard knowledge that the model struggles  
 334 with.

335

## 336 4.3 DIFFERENT MEMORY THRESHOLD

337

338 **Fixed vs adaptive threshold** In the simplest variant, we apply a fixed threshold  $\tau$ , masking any  
 339 answer segment with average loss above  $\tau$ . This provides a straightforward knob on how aggres-  
 340 sively long-tail knowledge is offloaded: the higher  $\tau$  is, the fewer segments are delegated to external  
 341 memory. Alternatively, we experiment with an adaptive threshold, where at each step we compute  
 342 the distribution of average losses across all answer segments seen so far and mask those falling in  
 343 the top  $\tau\%$  (e.g., 75th percentile). The adaptive strategy dynamically adjusts to training progress  
 344 and corpus difficulty: early in training, more segments are masked, while later only the most chal-  
 345 lenging segments are delegated to memory. We report answer-level recall (ALR) under different  
 346 fixed thresholds shown in Figure 4. For fixed threshold, as the memory threshold increases, ALR  
 347 generally decreases for the entity-centric and longtail benchmarks such as EntityQ and PopQA, but  
 348 increases for the common knowledge QA datsets such as NQ and HotpotQA. This aligns with our  
 349 assumption that aggressive memory offloading (lower threshold) could hurt performances on general  
 350 benchmarks. On general QA benchmarks such as TQA, NQ, and HotpotQA, we observe a positive  
 351 trend as the adaptive memory threshold increases, whereas performance on long-tail benchmarks is  
 352 comparatively stable. This pattern aligns with our hypothesis: by delegating only the hardest factual  
 353 segments to external memory, the model preserves parametric capacity for common knowledge and  
 354 improves generalizability, while maintaining strong recall on rare, entity-centric queries.



367 Figure 4: Answer level recall (ALR) for different memory threshold. The higher the threshold is, the  
 368 fewer segments are delegated to external memory. Left: Fixed memory threshold. Right: Adaptive  
 369 memory threshold.

370

371 **Trade-off between accuracy and latency** Long-tail datasets such as POPQA and SIMPLEQA  
 372 trigger retrieval far more frequently than other benchmarks, indicating that the model is more reliant  
 373 on external memory for long tail questions. While one might expect retrieval frequency to increase  
 374 linearly with more higher thresholds, this is not observed in practice. The reason is that segment  
 375 losses are highly skewed and clustered as shown in previous histogram, so adjusting thresholds  
 376 often has non-linear effects: small changes may have little impact or may trigger sudden jumps in  
 377 retrieval.

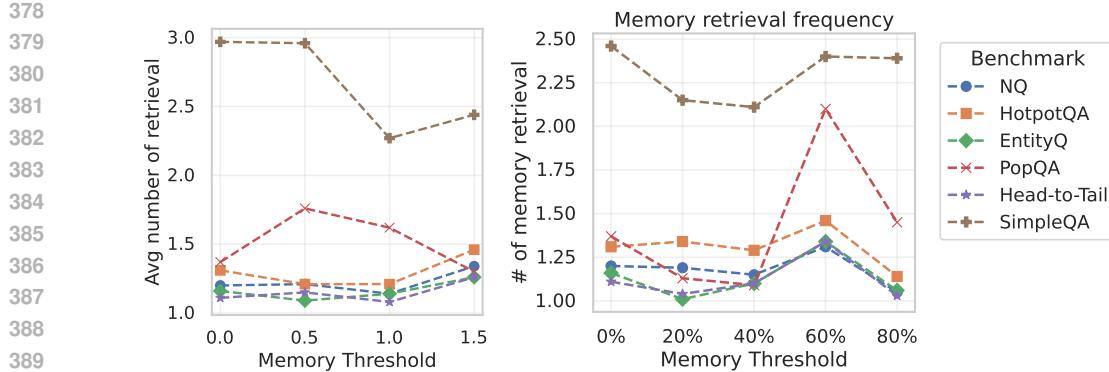


Figure 5: Average number of retrieval calls for different memory threshold, left is fixed threshold, and right is adaptive threshold. The higher the threshold is, the fewer segments are delegated to external memory.

## 5 RELATED WORK

**Non-parametric language modeling** One recent line of research has explored integrating non-parametric memory with language models generation to reduce memorization and improve factuality. The earliest work includes kNN-LM, which interpolates language model generation and retrieved nearest neighbors at inference time to adjust output distribution (Khandelwal et al., 2019). More recent inference-time only approaches include Chunk-Distilled LM (Li et al., 2025), which speculates multi-token chunks using a retrieval datastore to accelerate generation, as well as REST (He et al., 2023) and NEST (Li et al., 2024), which combine speculative decoding with retrieval to improve efficiency and factuality.

**Retrieval-augmented pretraining** Several recent approaches have explored incorporating retrieval directly into the pretraining process. RETRO (Borgeaud et al., 2021) demonstrated that integrating retrieved context during pretraining improves generalization and reduces undesired memorization by allowing smaller models to match the performance of much larger purely parametric LMs. More recently and most relevant to us, Zhao et al. (Zhao et al., 2025) introduced Large Memory Language Models (LMLM), which explicitly separates factual knowledge storage from model weights by offloading specific factual details to an external database during pre-training. Our approach shares a similar motivation, but instead of we leverage signals from a partially trained model to adaptively decide what knowledge to offload.

## 6 CONCLUSION

We introduced semi-parametric language models with selective memory (SPLM), a framework that improves factuality on long-tail knowledge without sacrificing general capabilities. By masking high-loss segments and delegating them to external memory, SPLM preserves parametric capacity for common knowledge and reasoning while offloading rare facts. A lightweight query adapter enables contextualized retrieval aligned with the model’s representations. Experiments demonstrate that SPLM achieves a favorable trade-off, outperforming memory-only baselines and rivaling larger models. This highlights selective memory as a promising direction for scaling factuality in LLMs efficiently.

There are several interesting directions for future research. We primarily targeted factual recall on short QA benchmarks, but the framework could naturally extend to other domains such as mathematical theorems or code snippets that occur in training data. An interesting question for future work is whether decoupling knowledge in this way could also improve performance on reasoning-intensive benchmarks. By externalizing rare factual content, the model could dedicate more of its parametric capacity and compute to learning reasoning strategies rather than memorizing infrequent facts. Another interesting direction would be to explore joint training of the adapter and the language

432 model, which may further improve performance by enabling integration of external knowledge dur-  
 433 ing inference.  
 434

435 **REFERENCES**  
 436

437 Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Retrieval-based language models and  
 438 applications. In Yun-Nung (Vivian) Chen, Margot Margot, and Siva Reddy (eds.), *Proceedings*  
 439 *of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tuto-*  
 440 *rial Abstracts)*, pp. 41–46, Toronto, Canada, July 2023. Association for Computational Linguis-  
 441 *tics.* doi: 10.18653/v1/2023.acl-tutorials.6. URL <https://aclanthology.org/2023.acl-tutorials.6/>.  
 442

443 Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Mil-  
 444 lican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego  
 445 de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, T. W. Hennigan, Saffron Huang,  
 446 Lorenzo Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irv-  
 447 ing, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and L. Sifre.  
 448 Improving language models by retrieving from trillions of tokens. In *International Conference*  
 449 *on Machine Learning*, 2021. URL <https://api.semanticscholar.org/CorpusID:244954723>.  
 450

451 Hoyeon Chang, Jinho Park, Seonghyeon Ye, Sohee Yang, Youngkyung Seo, Du-Seong Chang,  
 452 and Minjoon Seo. How do large language models acquire factual knowledge during pre-  
 453 training? *ArXiv*, abs/2406.11813, 2024. URL <https://api.semanticscholar.org/CorpusID:270559235>.  
 454

455 Shangbin Feng, Weijia Shi, Yike Wang, Wenzuan Ding, Vidhisha Balachandran, and Yulia  
 456 Tsvetkov. Don’t hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collab-  
 457 oration. *ArXiv*, abs/2402.00367, 2024. URL <https://api.semanticscholar.org/CorpusID:267365203>.  
 458

459 Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-  
 460 augmented language model pre-training. *ArXiv*, abs/2002.08909, 2020. URL <https://api.semanticscholar.org/CorpusID:211204736>.  
 461

462 Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. Rest: Retrieval-based speculative  
 463 decoding. *ArXiv*, abs/2311.08252, 2023. URL <https://api.semanticscholar.org/CorpusID:265157884>.  
 464

465 Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. In *IEEE*  
 466 *Transactions on Big Data*, 2019.  
 467

468 Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly  
 469 supervised challenge dataset for reading comprehension. *ArXiv*, abs/1705.03551, 2017. URL  
 470 <https://api.semanticscholar.org/CorpusID:26501419>.  
 471

472 Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language  
 473 models struggle to learn long-tail knowledge. In *Proceedings of the 40th International Conference*  
 474 *on Machine Learning*, ICML’23. JMLR.org, 2023.  
 475

476 Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization  
 477 through memorization: Nearest neighbor language models. *ArXiv*, abs/1911.00172, 2019. URL  
 478 <https://api.semanticscholar.org/CorpusID:207870430>.  
 479

480 Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris  
 481 Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion  
 482 Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le,  
 483 and Slav Petrov. Natural questions: A benchmark for question answering research. *Trans-*  
 484 *actions of the Association for Computational Linguistics*, 7:453–466, 2019. URL <https://api.semanticscholar.org/CorpusID:86611921>.  
 485

486 Minghan Li, Xilun Chen, Ari Holtzman, Beidi Chen, Jimmy Lin, Wen tau Yih, and Xi Victoria Lin.  
 487 Nearest neighbor speculative decoding for llm generation and attribution. *ArXiv*, abs/2405.19325,  
 488 2024. URL <https://api.semanticscholar.org/CorpusID:270095057>.

489 Yanhong Li, Karen Livescu, and Jiawei Zhou. Chunk-distilled language modeling. In *The Thirteenth*  
 490 *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=nrv0W0cyg>.

491 Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram:  
 492 Scaling unbounded n-gram language models to a trillion tokens. In *First Conference on Language*  
 493 *Modeling*, 2024. URL <https://openreview.net/forum?id=u2vAyMeLMm>.

494 Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi.  
 495 When not to trust language models: Investigating effectiveness of parametric and non-parametric  
 496 memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of*  
 497 *the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*  
 498 *Papers)*, pp. 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguis-  
 499 *tics*. doi: 10.18653/v1/2023.acl-long.546. URL <https://aclanthology.org/2023.acl-long.546>.

500 Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-  
 501 networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language*  
 502 *Processing (EMNLP)*, 2019.

503 Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. Simple entity-centric ques-  
 504 tions challenge dense retrievers. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and  
 505 Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural*  
 506 *Language Processing*, pp. 6138–6148, Online and Punta Cana, Dominican Republic, November  
 507 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.496. URL  
 508 <https://aclanthology.org/2021.emnlp-main.496>.

509 Kai Sun, Yifan Xu, Hanwen Zha, Yue Liu, and Xin Luna Dong. Head-to-tail: How knowl-  
 510 edgeable are large language models (LLMs)? A.K.A. will LLMs replace knowledge graphs?  
 511 In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Confer-*  
 512 *ence of the North American Chapter of the Association for Computational Linguistics: Human*  
 513 *Language Technologies (Volume 1: Long Papers)*, pp. 311–325, Mexico City, Mexico, June  
 514 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.18. URL  
 515 <https://aclanthology.org/2024.naacl-long.18>.

516 Wenhui Wang, Hangbo Bao, Li Dong, and Furu Wei. Minilm: Deep self-attention distillation for  
 517 task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Pro-*  
 518 *cessing Systems (NeurIPS)*, 2020.

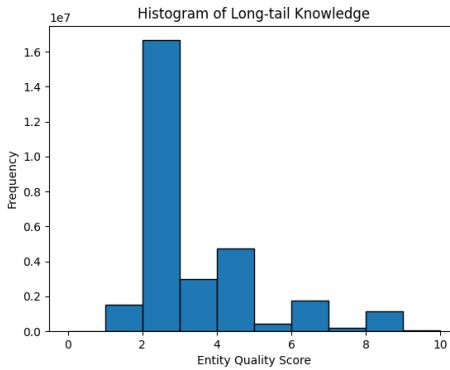
519 Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese,  
 520 John Schulman, and William Fedus. Measuring short-form factuality in large language models.  
 521 *arXiv preprint arXiv:2411.04368*, 2024. URL <https://arxiv.org/html/2411.04368>.

522 Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo  
 523 Wang, Zeyun Tang, Shichao Song, Chenyang Xi, Yu Yu, Kai Chen, Feiyu Xiong, Linpeng Tang,  
 524 and E Weinan. Memory3: Language modeling with explicit memory. *ArXiv*, abs/2407.01178,  
 525 2024. URL <https://api.semanticscholar.org/CorpusID:270870116>.

526 Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov,  
 527 and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question  
 528 answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL  
 529 <https://api.semanticscholar.org/CorpusID:52822214>.

530 Linxi Zhao, Sofian Zalouk, Christian K. Belardi, Justin Lovelace, Jin Peng Zhou, Kilian Q.  
 531 Weinberger, Yoav Artzi, and Jennifer J. Sun. Pre-training large memory language models  
 532 with internal and external knowledge. *ArXiv*, abs/2505.15962, 2025. URL <https://api.semanticscholar.org/CorpusID:278789132>.

540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552



553  
554 Figure 6: Histogram of long tail scoring using Llama3-70B. 1 indicates low popularity and 10  
555 indicates high popularity.

556  
557 **A APPENDIX**  
558

559 **A ADDITIONAL IMPLEMENTATION DETAILS**  
560

561 **B MODEL BASED SCORING**  
562

563 **B.1 PROMPT USED FOR LLAMA3 70B**  
564

565 You are a capable model that can determine, based on the atomic factual knowledge query,  
566 whether the atomic factual knowledge belongs to a long-tailed (niche or less commonly refer-  
567 enced) domain. To assess the entities popularity, use the following rules: 1) The number of times  
568 the atomic factual knowledge potentially can be referenced or cited by other documents on internet.  
569 2) The number of times the atomic factual knowledge potentially can be linked to or duplicated  
570 on internet. Please rate the atomic factual knowledge with a score between 1 and 10, where 1  
571 indicates low popularity (long-tailed knowledge) and 10 indicates high popularity (widely refer-  
572 enced knowledge). Return the score in JSON format, for example: {"quality\_score": predicted  
573 score } text:

574 Interestingly, as shown in Figure 6, the model rates most extracted factual knowledge as long-tail,  
575 which aligns closely with the skewed  $n$ -gram frequency distribution estimated by Infinigram (Fig-  
576 ure 3). In contrast, the distribution under loss-based scoring is closer to normal, with many segments  
577 falling into a moderate difficulty range rather than being pushed to the extremes. This discrepancy  
578 suggests that, despite the frequency of appearing in the corpus, the model exhibits heterogeneous  
579 difficulty in internalizing difficult factual knowledge.

581 **C SAMPLE GENERATIONS FROM SIMPLEQA**  
582

583 **Q1.** When was the exact day, month, and year the fpt university was founded?  
584

585 **Gold Answer:** september 8, 2006

586 **Full Memory:** The university was founded in m\_start University of California,  
587 Berkeley Founded In m\_retrieve 1868 m\_end 1868.

588 **Selective memory:** m\_start FPT University Founded On m\_retrieve septem-  
589 ber 8, 2006 m\_end september 8, 2006.

590  
591 **Q2.** Q: What was the word of the decade (2010–2019) according to the american  
592 dialect society?  
593

**Gold Answer:** They.

594                   **Full Memory:** `m_start` Q: What was the word of the decade (2010–2019)  
 595                   Answer `m_retrieve` web `m_end` web

596                   **Selective memory:** The word of the decade was `m_start` Word of the Decade  
 597                   2010-2019 `m_retrieve` they `m_end` they, according to the american dialect  
 598                   society.  
 599

600                   **Q3.** Q: What was the first name of the irish-born computer programmer kathleen  
 601                   rita mcnulty mauchly antonelli's father?  
 602

603                   **Gold Answer:** james mccullough

604                   **Selective memory:** He was called `m_start` Kathleen Rita McNulty FatherName  
 605                   `m_retrieve` troy and claudette mcghee `m_end` troy and claudette mcghee.

606                   **SPLM (ours):** She was called `m_start` Kathleen Rita Mcculty Mauchly  
 607                   FatherName `m_retrieve` james mccullough `m_end` james mccullough

608                   **Q4.** Q: Who was the 9th chief minister of jammu and kashmir?  
 609

610                   **Gold Answer:** mehbooba mufti

611                   **Selective memory:** `m_start` 9th Chief Minister of Jammu and Kashmir Name  
 612                   `m_retrieve` mehbooba mufti `m_end` mehbooba mufti

613                   **SPLM (ours):** Bakhsh Singh, who was the 9th chief minister of jammu and  
 614                   kashmir from `m_start` Bakhsh Singh Term Start Date `m_retrieve` 9 april 1964  
 615                   `m_end` 9 april 1964 to `m_start` Bakhsh Singh Term End Date `m_retrieve` 10  
 616                   july 1970 `m_end` 10 july 1970.

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647