OPEN-BOOK PRETRAINING FOR LANGUAGE MODELS: SELECTIVE MEMORIZATION WITH DYNAMIC RETRIEVAL

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models use their limited parametric memory to store facts that could instead be retrieved from an external store. We ask whether a model can decide, during pretraining, which facts to memorize and which to leave to a knowledge base (KB). We introduce **OPEN-BOOK PRETRAINING (OBP)**, a pretraining method that makes such decisions dynamically by identifying positions where retrieved context reduces next-token loss, according to the current state of the model. Along with the pretrained language model, we learn a lightweight classifier that decides when to retrieve during inference. Using this approach, we train small language models (<0.5B parameters) from scratch on Wikipedia. Evaluated with retrieval on held-out pages, a 360M OBP model reaches a lower perplexity than a 1.7B model trained on the same data, while retaining competitive perplexity even without retrieval. Relative to standard pretraining plus retrieval-augmented generation, OBP models use retrieved evidence better while memorizing fewer facts in their parameters. Our learned classifier is able to recognize when to retrieve during inference, adapting to different sizes of knowledge base. To our knowledge, OBP is the first framework that dynamically selects between parametric memory and external retrieval during pretraining, based on model- and KB-specific retrieval gains.

1 INTRODUCTION

Modern large language models (LLMs) are pretrained on large amounts of text via next token prediction, before being post-trained and applied to various downstream tasks (Brown et al., 2020; Guo et al., 2025; Yang et al., 2025). Pretraining corpora are usually obtained from web text, which contains a large amount of real-world factual knowledge. As a result, LLMs often memorize world knowledge during pretraining. Storing factual knowledge in model parameters is useful, but which facts should be memorized is debatable.

In real-world use cases, LLMs often combine their parametric knowledge with retrieved knowledge from external sources. Preparing for this real-world setting can be thought of as analogous to studying for an open-book exam. Imagine a student studying for an open-book exam: In the process of studying, they will memorize some facts, while for other facts they will just remember how to quickly look for the information in their books and use it in order to answer exam questions. During the studying process, a student will dynamically adjust what they memorize while also getting better at using information from their books. In addition, depending on various factors—the length of the books they will have at the exam, the time constraints of the exam—they may adapt the extent to which they rely on memorization versus retrieval from their books.

Motivated by this analogy, we propose OPEN-BOOK PRETRAINING (OBP) for language models, which addresses the research question:

> *Can a language model trained from scratch decide which information should be memorized parametrically, and which information should be retrieved from external sources?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: Comparison with related work. **Adaptive Memorization:** parametric memorization is model-adaptive during training. **Dynamic Retrieval:** the retrieval rate during inference can change without retraining. **Annotation Free:** annotation of the training corpus is not required.

	Retrieval in Pretraining	Fact Offloading	Adaptive Memorization	Dynamic Retrieval	Annotation Free
kNN-LM (Khandelwal et al., 2020)	×	×	×	×	✓
RAG (Lewis et al., 2020)	×	×	×	×	✓
RETRO (Borgeaud et al., 2022)	✓	×	×	×	✓
Toolformer (Schick et al., 2023)	×	×	×	×	×
LMLM (Zhao et al., 2026)	✓	✓	×	×	×
OPEN-BOOK PRETRAINING	✓	✓	✓	✓	✓

OBP extends the analogy of studying for an open-book exam into the pretraining of language models. Reducing unnecessary parametric memorization has several practical benefits. One obvious benefit is more efficient use of parameters, leading to smaller models when retrieval is available. Apart from efficiency, keeping facts in an external store naturally supports updating time-sensitive knowledge, keeping private information out of model parameters, and adapting to new domains.

The above considerations have long motivated research on language models that rely more on external data stores and less on parametric memory, including a variety of retrieval-augmentation methods applied at either inference or training time (Lewis et al., 2020; Borgeaud et al., 2022; Shi et al., 2024; Zhao et al., 2026). However, existing work does not yet address all of the desiderata of true open-book study, including adaptive memorization vs. retrieval decisions at training time and dynamic adjustments to retrieval rates depending on conditions and constraints (Tab. 1).

In OPEN-BOOK PRETRAINING, we train a language model to *know when to retrieve* and to *understand what is retrieved* in order to improve next-token prediction, using only the information gain it measures on its training data. Borrowing terminology from Zhao et al. (2026), we call a token *offloaded* when the model is never trained to predict it from the preceding text alone, only with retrieved evidence in the context. An offloaded fact stays in the knowledge base and is not "written" to the weights, while the model is still trained to read the evidence that contains it. A token that is difficult to predict from the prefix may become easy when relevant evidence is retrieved, while another token may already be predictable without retrieval. We measure the effect of retrieval with information gain on next-token prediction, and use this retrieval information gain to determine which tokens should be offloaded. The decision boundary depends on how much retrieval helps the model at the current training step; thus our framework is model-adaptive and can be applied to language models of all sizes and training stages. Our approach is also agnostic to the choice of knowledge base, making it suitable to domain-specific settings, such as legal and medical applications, as well as settings where external knowledge should be private, and supports easy addition, update, or deletion.

Our main contributions are:

- We propose OBP, the first framework that supports dynamic knowledge offloading during training, based on model-adaptive retrieval gains. Unlike previous retrieval-augmented training approaches (Tab. 1), our pipeline requires no annotation of the training corpus, and is agnostic to knowledge base (KB) construction.
- Our evaluation results show that a 360M model trained with OBP can reach lower perplexity than a 1.7B model, and substantially outperforms a retrieval-augmented baseline model in knowledge-intensive question-answering tasks.
- We demonstrate multiple ways of trading off memorization vs. retrieval, by experimenting with different hyperparameter choices and KB sizes.

2 RELATED WORK

A variety of approaches have been developed for augmenting a language model's input with retrieved information. This information can come from any external knowledge source: a corpus (Lewis et al., 2020; Karpukhin et al., 2020), a knowledge base (Zhao et al., 2026), or search engine results (Schick et al., 2023; Jin et al., 2025). Previous work augmenting limited memory of language models can be classified by whether the augmentation happens during training or during inference.

Retrieval-Augmented Generation Retrieval during inference is called retrieval-augmented generation (RAG) (Lewis et al., 2020), where a pretrained language model conditions its generation on retrieved passages (Shi et al., 2024; Karpukhin et al., 2020; Min et al., 2024). These retrievals often occur unconditionally at predefined positions, for example right after a question in question answering tasks (Izacard & Grave, 2021) or at a fixed interval (Khandelwal et al., 2020). RAG can substantially improve language models' performance on answering questions about information that isn't seen during pretraining, and narrows the gap between small language models and large language models (Liška et al., 2022; Kasai et al., 2023; Li et al., 2025).

Retrieval-Augmented Training Beyond augmenting a trained language model at inference time, another line of work incorporates retrieval directly into language-model pre-training (Borgeaud et al., 2022; Guu et al., 2020) or post-training (Schick et al., 2023; Jin et al., 2025). These methods improve language modeling by making external context available during training, but often do not explicitly prevent the same information from being learned parametrically. Retrieval is triggered at fixed positions during training, either by annotating the training corpus with special tokens (Schick et al., 2023; Zhao et al., 2026), or by retrieving at fixed chunk intervals (Borgeaud et al., 2022). Toolformer (Schick et al., 2023) uses a criterion similar to our information gain: It filters candidate API calls by whether providing the API call and its result lowers the weighted cross-entropy loss for a stronger model, and annotates static API call positions based on this measure. REALM (Guu et al., 2020) jointly learns which documents to retrieve with masked-language-model pretraining, retrieving once per sentence. Focusing on limiting parametric memory, LMLM (Zhao et al., 2026) and Co-LMLM (Feldman et al., 2026) first annotate fixed factual spans and retrieval positions in the training data, and then mask those spans from the language-modeling objective so they must be recovered through knowledge base lookups. The data annotation process can be time- and resource-consuming for a large training corpus. Moreover, whether a span is a fact relies on a distilled annotator model, and there's no coverage guarantee. The format of the annotation and knowledge base is constrained: LMLM (Zhao et al., 2026) annotates facts as structured knowledge triplets, while Co-LMLM (Feldman et al., 2026) extends this approach to free-form factual spans and continuous retrieval queries. In contrast, our approach learns where to retrieve on the fly from measured retrieval information gain, without requiring target positions to be annotated in advance, and can in principle be applied to any knowledge base, not necessarily limited to gold factual spans. Our method can also be applied to any general or domain-specific knowledge base, since our language model is trained to understand and filter out irrelevant information in the retrieved documents.

Learning When to Retrieve The retrieval-augmented approaches described above rely on predefined or annotated retrieval positions. Some retrieval-based approaches instead learn when to retrieve, or more generally, when to start an external action (e.g., a tool call) during training or inference. Self-RAG (Asai et al., 2023) learns whether to retrieve on a per-sentence basis, by fine-tuning a pretrained model to generate a RETRIEVE or NO-RETRIEVE reflection token. Search-R1 (Jin et al., 2025) learns when to call a search engine and reason about the results through reinforcement learning over the token trajectories. These methods assume capable pretrained models or task-level supervision. In OBP, in contrast, the retrieval decision is learned dynamically during pretraining.

3 METHOD

OPEN-BOOK PRETRAININGis analogous to letting a language model prepare for an open-book exam. We train a language model from scratch, while allocating which tokens to memorize and which tokens to offload. The decision is based on how predictable the next token is with and without the retrieved context at the current training step, which we measure with retrieval information gain, defined in §3.1. We train a classifier together with the language model to decide when to retrieve during inference. We describe the training objective for the language model in §3.2, and the training objective for the classifier module in §3.3.

3.1 RETRIEVAL INFORMATION GAIN

For token x_t with prefix $x_{<t}$ and retrieved documents d (top-k from the selected knowledge base, prepended as plain text), we define the retrieval information gain δ_t as

$$\delta_t = \log p_\theta(x_t \mid d,\, x_{<t}) - \log p_\theta(x_t \mid x_{<t}) \tag{1}$$

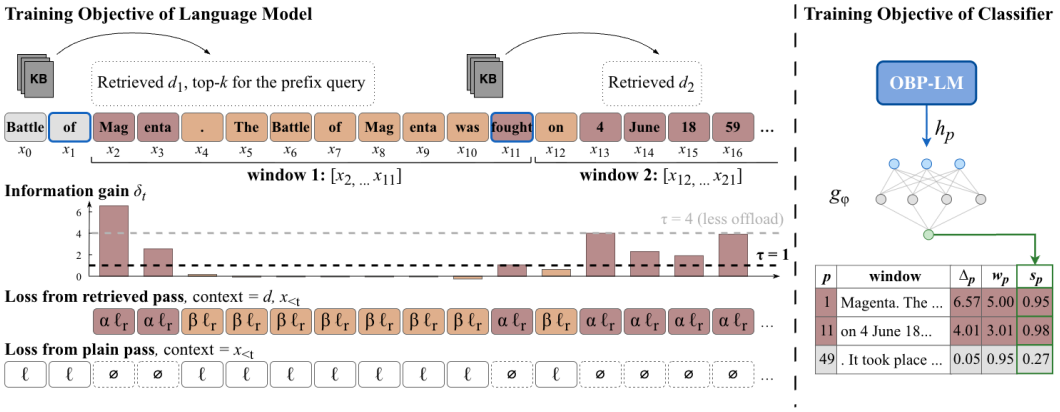


Figure 1: Training objective of OBP shown on the first few tokens of a held-out Wikipedia page, with values measured using the trained `GPT2-124M` model (seed 42). **Left:** As described in §3, we retrieve documents d_p with the prefix as query at each sampled decision position. The bar shows the measured retrieval gain δ_t for each token x_t from adding d_p to the context. Pink tokens are $\mathcal{M}_\mathbf{x}$, orange tokens are $\mathcal{B}_\mathbf{x}$, gray tokens are $\mathcal{P}_\mathbf{x}$. **Right:** Taking the hidden state h_p at each decision position, the classifier g_ϕ is trained to give a score s_p. During inference, the classifier selects position p for retrieval if the predicted s_p is greater than τ_{inf}.

A positive δ_t indicates that retrieval improves the model's prediction of x_t. The score is model-dependent and changes during training. In a typical (non-OBP) language model trained with next-token prediction, as the model becomes better at predicting x_t from the prefix and its parametric memory, the marginal benefit of retrieval should decrease. Conversely, if the model gets better at understanding the retrieved content, the benefit could increase.

3.2 Dynamic Fact Offloading

The computation of our training objective is illustrated in Fig. 1. We use measured retrieval information gain to decide whether for a given token, we should use the retrieval-augmented loss rather than the plain language-modeling objective. Due to compute constraints, instead of making the retrieval decision per token, we sample m non-overlapping spans of length l, and calculate the retrieval information gain for each token inside each span. In our experiments we sample $m = 5$ decision positions by default: Four positions drawn uniformly from template candidates (prefixes ending in a common fact-preceding pattern, listed in Appendix A.3) and one from all word-start positions. At each sampled decision position p in an input sequence $\mathbf{x} = x_0, \dots, x_n$, we use $W_p = x_{p+1}, \dots, x_{p+l}$ to denote the retrieval candidate span. During sampling, we require that spans not overlap with each other. Unless otherwise specified, we use a span length $l = 10$ in our experiments. We use the prefix query $\mathbf{x}_{\leq p} = x_0, \dots, x_p$ to retrieve the top-k documents d_p from the knowledge base, and compute δ_t for each token $x_t \in W_p$.

We denote the decision threshold hyperparameter τ, and we use $\tau = 1$ in our experiments unless otherwise specified. Ablation experiments on τ are in Fig. 5. A token is eligible for retrieval-based training when $\delta_t > \tau$ and it satisfies the following conditions: It belongs to a content word, does not already appear in the prefix $\mathbf{x}_{\leq p}$, and appears in the retrieved documents.[1] Finally, we add tokens that are contiguous with selected tokens into this set. Contiguous tokens include tokens that are in the same word or in a date string. We refer to all of the tokens that have been selected as *offloaded*.

Let $\mathcal{M}_\mathbf{x}$ denote the resulting set of offloaded tokens in a training sequence $\mathbf{x}$. Tokens in $\mathcal{M}_\mathbf{x}$ are masked from the plain language-modeling loss and incur a loss only with retrieved documents in context. Let $\mathcal{B}_\mathbf{x} = \bigcup_p W_p - \mathcal{M}_\mathbf{x}$ denote the remaining tokens inside candidate spans, where the union is over all p that have been chosen for retrieval. We refer to the tokens in $\mathcal{B}_\mathbf{x}$ as *bystander tokens*. Bystander tokens preseve their ordinary plain-context loss and additionally receive

[1] We combine consecutive subword units into words and classify at the word level. A word is a content word if it is not punctuation and not in a fixed function-word list (articles, prepositions, conjunctions, pronouns, auxiliaries); numbers and dates count as content.

a retrieved-context loss. Let $\mathcal{P}_{\mathbf{x}}$ denote all other tokens in the sequence that are not in $\mathcal{M}_{\mathbf{x}}$. Thus, $\mathcal{B}_{\mathbf{x}} \subset \mathcal{P}_{\mathbf{x}}$.

For a token x_t with context $x_{<t}$, we denote the cross-entropy loss without and with retrieval as:

$$\ell(x_t \mid x_{<t}) = -\log p_\theta(x_t \mid x_{<t}), \qquad \ell_{\mathrm{r}} = \ell(x_t \mid d_{p(t)}, x_{<t}) \tag{2}$$

OBP's training objective is:

$$\mathcal{L}_{\mathrm{OBP}} = \frac{\sum_{t\in\mathcal{P}_{\mathbf{x}}} \ell(x_t \mid x_{<t}) + \alpha \sum_{t\in\mathcal{M}_{\mathbf{x}}} \ell(x_t \mid d_{p(t)}, x_{<t}) + \beta \sum_{t\in\mathcal{B}_{\mathbf{x}}} \ell(x_t \mid d_{p(t)}, x_{<t})}{|\mathcal{P}_{\mathbf{x}}| + \alpha|\mathcal{M}_{\mathbf{x}}| + \beta|\mathcal{B}_{\mathbf{x}}|} \tag{3}$$

where $p(t)$ denotes the retrieval position whose window contains token x_t. Retrieved document tokens are used only as context and are masked from the language-modeling loss. The first term applies the standard language-modeling objective to all non-offloaded tokens. The second term applies to offloaded tokens only with external knowledge in context. The third term additionally applies to bystander tokens given the retrieved context, making the model robust to inserted documents. Unless otherwise specified, we use $\alpha = 20$ and $\beta = 1$ in our experiments.

3.3 Learning When to Retrieve

The offloading rule above requires measuring retrieval gain and therefore cannot be applied directly at inference time, where the model doesn't have a gold reference. We instead train a lightweight classifier to predict whether retrieval is likely to be useful from the current hidden state of the model.

Our classifier is a two-layer MLP g_ϕ, which takes as input the hidden state h_p of the language model at each sampled position. Let $\mathcal{M}_p \subseteq W_p$ denote the offloaded tokens of the window at decision position p, and let Δ_p be the largest per-token gain among the window tokens that pass the content, prefix-novelty, and document-support filters, so that $\mathbf{1}[\Delta_p > \tau] = \mathbf{1}[\mathcal{M}_p \neq \emptyset]$. The position-level label and confidence weight are

$$y_p = \mathbf{1}\left[\mathcal{M}_p \neq \emptyset\right], \qquad w_p = \min(|\Delta_p - \tau|,\, 5) \tag{4}$$

with $y_p = 0$ and $w_p = 1$ when no window token qualifies for measurement. The classifier is trained with weighted binary cross-entropy over the measured decision positions of each batch, which we denote $\mathcal{D}$:

$$\mathcal{L}_{\mathrm{classifier}} = \frac{\sum_{p\in\mathcal{D}} w_p[-y_p \log s_p - (1-y_p)\log(1-s_p)]}{\sum_{p\in\mathcal{D}} w_p}, \qquad s_p = \sigma(g_\phi(h_p)) \tag{5}$$

At inference the classifier is applied at the start of every word, and retrieval is triggered when $s_p > \tau_{\mathrm{inf}}$. The training threshold τ controls how much information is offloaded from the model, while the inference threshold τ_{inf} controls retrieval frequency at inference (enabling dynamic retrieval, in the terminology of Table 1).

4 Experiments

We evaluate OBP language models along two dimensions: (1) whether OBP achieves our goal of reducing parametric memorization while being able to use retrieved content; and (2) whether it preserves general language modeling abilities. For memorization, we report perplexity on the annotated fact tokens in our KB evaluated without retrieval in Tab. 10, or close-book accuracy on QA benchmarks (Tab. 3, Tab. 4). For retrieval performance, we report open-book accuracy on QA benchmarks (Tab. 3, Tab. 4). We compare each OBP model with a plain language model baseline that shares its architecture, training data, and schedule and differs only in the training objective.

4.1 Experimental Setup

We test OBP with four architectures from two model families: `GPT2-124M`, `GPT2-355M`, `SmolLM2-135M`, and `SmolLM2-360M` (Radford et al., 2019; Allal et al., 2025). Each model uses the released tokenizer for its model family, extended with one special token, `<|RETRIEVE|>`,

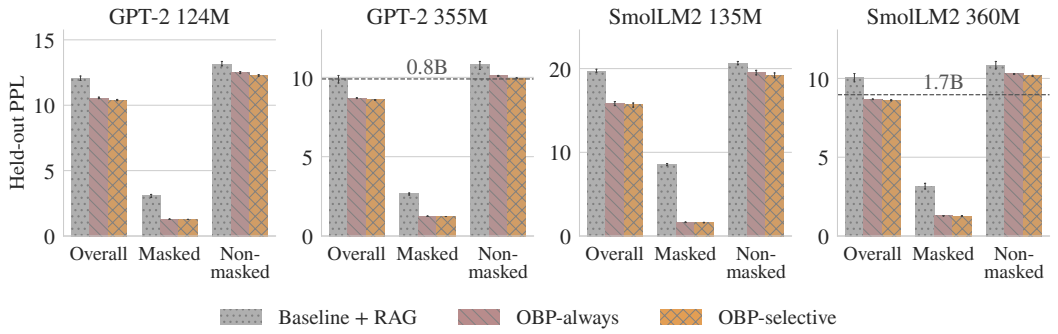


Figure 2: Held-out perplexity with retrieval (i.e., in an open-book setting) from our default knowledge base on the same 1,000 pages as Tab. 2. Each model's "masked" set is selected by whether each token's retrieval gain exceeds the training threshold τ, mimicing training. Thus the "masked" set is model-dependent. The "Overall" group compares the two models on identical tokens. Bars are the mean value over three seeds, with the error bars representing minimum and maximum values. The dashed line in the SmolLM2-360M panel is the plain perplexity of the SmolLM2-1.7B baseline.

that separates retrieved text from the original context. We train all models from scratch on the KILT knowledge source (Petroni et al., 2021), a Wikipedia snapshot from 2019/08/01 with 5.9M articles. We choose KILT because it unifies 11 knowledge-intensive benchmarks and annotates each benchmark entry with supporting sentences in a fixed Wikipedia snapshot, so its benchmarks come with evidence in the corpus we train on and build the default knowledge base from. We hold out every 50$^{\text{th}}$ article as an evaluation split (118K articles, $\sim$80M tokens) and train on the remaining 5.78M articles ($\sim$4.2B tokens). We train each model for one epoch at a context length of 1024 tokens, and repeat each run with three random seeds. Training details and hyperparameters are in Appendix A.2.

4.2 Knowledge Base Annotation

OBP does not restrict how the knowledge base is built. However, a fact can be offloaded only if the knowledge base contains it, so knowledge base size limits how much content can be offloaded. Therefore, we build our default knowledge base to cover the facts in the KILT snapshot as completely as we can: We use `Qwen3-4B-Instruct` to extract atomic facts from each of the 5.9M articles (using the prompt in Appendix B.1), which yields 151.7M facts in total. The knowledge base covers both training and held-out articles. Tab. 9 shows examples of KB entries. The same default KB is used in both open-book training and open-book evaluation. We retrieve with a frozen `e5-base-v2` encoder (Wang et al., 2024)), and store all 151.7M embeddings in a FAISS IVF-PQ index (Douze et al., 2025). At each retrieval position, the query is the page title and the tokens up to that position.

4.3 Perplexity Evaluation

We evaluate token perplexity (PPL) on 1,000 articles sampled uniformly from the held-out split.

Plain Language Modeling Tab. 2 reports held-out perplexity without retrieval. Here the model predicts every token in a "closed-book" setting, including the offloaded tokens, which OBP models are not trained to predict without retrieved context. OBP perplexity stays within 0.6 of the perplexity of the baseline on all four models, confirming that general language modeling abilities are largely preserved even without retrieval.

Language Modeling with Retrieval Fig. 2 reports held-out perplexity with retrieval (i.e., the open-book setting) from the default knowledge base. During this open-book evaluation, every word start is a candidate position for retrieval. When the classifier predicts retrieval, we prepend the top-k retrieved facts and calculate PPL of the following 10-token span, extended by up to three tokens to complete its final word, while the PPL of every token outside the retrieval spans is calculated without them. **Baseline** and **OBP-always** retrieve at every candidate, assigning spans from left to right without overlap. **OBP-selective** retrieves when the classifier score exceeds the inference threshold, $s_p > \tau_{\text{inf}} = 0.5$ by default. For each setting, we report perplexity separately over "Masked" tokens (i.e., the tokens inside a retrieval span which satisfies our offloading criteria §3.2), over all

Table 3: Zero-shot EM scores (%, mean over 3 seeds) on Natural Questions, TriviaQA, and HotpotQA. Bolded values are best performance of our models.

		NQ				TriviaQA				HotpotQA			
Model		None	Top-1	Top-5	Gold	None	Top-1	Top-5	Gold	None	Top-1	Top-5	Gold
GPT2-124M	Baseline	1.2	5.4	9.3	19.4	5.0	11.0	12.7	30.0	5.6	3.6	4.4	10.6
	OBP	1.2	**9.9**	**15.7**	**41.8**	**5.3**	**17.4**	**26.9**	**55.3**	**8.3**	**5.0**	**7.5**	**17.8**
GPT2-355M	Baseline	1.3	5.0	10.1	18.9	**5.7**	9.8	15.2	28.5	6.5	4.4	5.4	12.6
	OBP	**1.4**	**11.4**	**15.7**	**42.8**	5.6	**20.9**	**29.7**	**61.9**	**8.6**	**6.0**	**7.9**	**21.1**
SmolLM2-135M	Baseline	**0.8**	3.5	4.0	9.5	1.9	5.4	6.0	14.8	1.6	1.8	2.2	5.3
	OBP	0.7	**8.8**	**12.4**	**39.1**	**2.4**	**11.1**	**17.3**	**38.5**	**3.0**	**3.2**	**4.9**	**12.4**
SmolLM2-360M	Baseline	**1.3**	6.1	7.7	17.6	3.7	9.8	13.2	26.2	3.4	2.8	4.5	9.9
	OBP	1.2	**11.6**	**15.0**	**41.4**	**3.9**	**19.0**	**26.4**	**54.3**	**4.4**	**5.4**	**7.6**	**18.1**

remaining tokens as "Non-masked", and over all tokens as "Overall". Tab. 5 reports the percentage of masked tokens during training and "Masked" tokens during inference, over corresponding splits.

With the same retrieved documents in context and measured on the same tokens, OBP-always reaches 15% lower overall perplexity than the baseline + RAG for all four models. This results indicates that OBP's training objective improves the base model's ability to extract relevant information and filter out irrelevant noise from retrieved documents. On the "masked" tokens, the perplexities of OBP models are about 3x better than baseline + RAG, and stays at or below baseline + RAG on the remaining tokens. For every OBP model, selective retrieval (the intended use case) matches or slightly improves over always retrieving, while retrieving at 25% fewer positions (Tab. 5). On SmolLM2-360M, both OBP settings reach lower perplexity than plain SmolLM2-1.7B.

Table 2: Perplexity **without retrieval** (mean over three seeds) on 1,000 uniformly sampled held-out Wikipedia pages from KILT. OBP trains with facts masked, so perplexity above the baseline is the expected outcome here.

	Baseline	OBP
GPT2-124M	13.5	14.0
GPT2-355M	11.0	11.4
SmolLM2-135M	21.6	21.5
SmolLM2-360M	11.2	11.6

4.4 QA Benchmarks

Our models are trained on KILT, so open-book evaluation with KILT-grounded QA benchmarks is naturally supported. KILT supports 11 knowledge-intensive language tasks, and we evaluate our models on 3 open-domain QA tasks: Natural Questions (NQ) (Kwiatkowski et al., 2019), HotPotQA (Yang et al., 2018), and TriviaQA (Joshi et al., 2017). We report strict exact match (EM) score with task-level finetuning on the train split (Tab. 11) and a loose EM score using zero-shot prompting (Tab. 3). Both scores are calculated over greedy generation with max 32 tokens. Loose EM means that we consider a generated string correct if a gold alias appears within its first 100 characters for PopQA and within its first five words for T-Rex, following Zhao et al. (2026). The zero-shot EM results on NQ, TriviaQA and HotpotQA are in Tab. 3, and the strict EM results with finetuning are in Tab. 11.

In Tab. 3, **None** corresponds to prompting model with only the question, **Top-1** and **Top-5** add the top retrieved facts, and **Gold** adds the gold provenance sentence. In open-book evaluation, OBP has better performance in every cell, up to 61.9 vs. 28.5 on TriviaQA with `GPT2-355M` given the gold sentence. The Top-1 and Top-5 columns are limited by our retriever choice; the hit rates of the Top-5 retrieved facts are 41.5%, 58.8%, and 21.0% in NQ, TriviaQA, and HotpotQA, respectively. (HotpotQA has the lowest hit rate because it contains questions that need multi-hop evidence from different articles.)

PopQA. Outside of KILT, we also evaluate on PopQA (Mallen et al., 2023). PopQA contains $14,267$ questions from Wikidata, each labeled with the monthly page-view count of its subject entity. Since the supporting evidence of PopQA is not labeled in KILT, we select the gold provenance as the shortest entry on the subject's page that contains a gold alias, which exists for $10,997$ questions. We consider a generation correct when at least one answer alias appears in the first 100 generated characters. **Closed** is the zero-shot accuracy on all $14,267$ questions, with only the question as the

Table 4: Zero-shot EM scores (%, mean over 3 seeds) without and with the knowledge base. Rows with an asterisk (*) and italicized values are the reported results from Feldman et al.. Note that this is not an apples-to-apples comparison: the knowledge bases used by Zhao et al. and Feldman et al. (2026) are specially formatted so that a retrieved entry is the answer, while our less structured KB requires the model to understand and select the answer among distractors on the same subject. Bolded values are best performance of our models.

		PopQA			T-REx		
Model		Closed	Gold	Long-tail	Closed	+KB	Gold
GPT-2 124M	Baseline	6.5	30.0	33.5	36.4	45.3	50.2
	OBP	6.2	37.7	39.4	34.1	52.5	56.9
GPT-2 355M	Baseline	6.9	23.4	27.7	41.3	50.6	55.1
	OBP	6.7	**44.1**	**43.3**	38.7	**57.0**	**60.6**
SmolLM 135M	Baseline	3.3	11.5	16.4	15.9	21.7	26.5
	OBP	3.3	25.2	23.6	15.5	31.6	37.3
SmolLM 360M	Baseline	5.8	25.2	25.6	35.7	44.5	49.3
	OBP	4.8	35.0	33.9	31.9	50.5	56.3
Rel-LMLM-382M* (Feldman et al., 2026)	–	–	–	*26.6*	–	–	*47.4*
Co-LMLM-360M* (Feldman et al., 2026)	–	*13.5*	–	*46.5*	*27.4*	–	*47.4*

prompt. **Gold** includes the gold entry in the prompt for the $10,997$ questions that have answers in our KB and keeps the question-only prompt for the rest, with accuracy averaged over all $14,267$ questions. We separately report the **Long-tail** column following Zhao et al. (2026), restricted to the $1,399$ questions whose subject has fewer than 100 monthly page views: 920 of them have a gold entry in our KB which is prepended to the question; for the rest we use the question-only prompt, and accuracy is averaged over all $1,399$. Our results in Tab. 4 show that OBP can almost double the performance of the baseline model in extracting the correct answer from retrieved evidence, and memorizes less in the closed-book setting.

T-Rex. We use the $11,615$ T-REx (Elsahar et al., 2018) questions in LAMA (Petroni et al., 2019) format from Zhao et al. (2026), and consider a generation correct if the correct answer is in the first 5 words. Our KB has an entry containing the answer (i.e. correct evidence) for $6,210$ questions. **+KB** includes five entries from the subject's page in the prompt, one containing the correct evidence and four distractors, in shuffled order. If the KB does not contain the correct evidence, we include five shuffled distractors. For $2,701$ questions whose subject page has no entries in our KB, we keep the question-only prompt. **Gold** includes only the correct evidence in the prompt. Our GPT2-355M OBP model has the highest accuracy, evaluated in the same setting with a less formatted KB than Feldman et al..

4.5 Ablation study: Memorization vs. Retrieval

Knowledge Base Ablation. We test how the size of the knowledge base affects Open-Book Pretraining. We train GPT2-124M with randomly-selected subsets of our default KB, corresponding to 10, 30, 50, 70 and 90% of the full KB. The subsets are nested so that a larger KB differs from a smaller one only by the entries added, not by a different random sample. Each OBP model variant uses the same KB variant in both training and testing. Our perplexity results on the held-out split are in Fig. 3. At each size of the knowledge base, OBP is better than the baseline at understanding the retrieved information. Furthermore, when the KB is small, retrieval hurts the baseline model, but not OBP. The no-retrieval results show that OBP memorizes less than the baseline, and OBP memorization keeps decreasing as the KB grows larger. Finally, our trained classifier retrieves more frequently when the KB is large, and less frequent when the KB is small.

Information Threshold Ablation. We further test how selecting different values for τ affects the retrieval rate (Fig. 5). τ determines how much a token must gain from evidence to be masked from memorizing, so raising it leaves more facts to be memorized. We experiment with $\tau \in \{0.5, 1, 2, 4\}$ on `SmolLM2-135M`. As τ increases, the classifier chooses to retrieve less during inference, from 33.6 retrievals per sequence at $\tau = 0.5$ to 17.4 at $\tau = 4$.

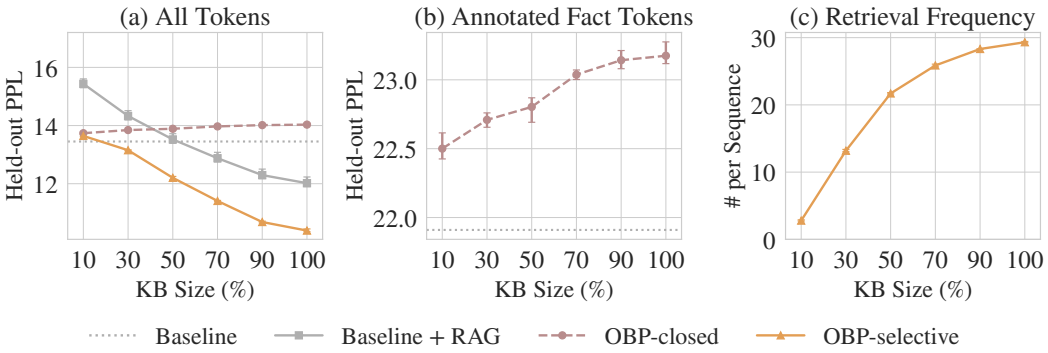


Figure 3: Knowledge base ablation results on held-out perplexity, with `GPT2-124M` model architecture. **(a)** Perplexity on the entire held-out split. **(b)** Perplexity on held-out tokens inside an annotated fact span (annotation details covered in §4.2), evaluated without retrieval. **(c)** Number of times the classifier retrieves per sequence during inference. We report mean values over three seeds, with error bars representing the min and max values.

5 Discussion

The Limits of Memorization vs. Generalization Our main experiment results in §4 show that models learned with OBP are better than standard language models at using retrieved content. What is the minimum amount a model should memorize to reach comparable general performance is still an open question. In the most extreme case, one can exclude facts completely from the model parameters, and learn only linguistic competence. Whether such a model could maintain performance is unclear, as some background knowledge may be necessary to support reasoning and effective use of retrieved evidence. The optimal division between parametric and external knowledge may depend on model capacity and the cost of retrieval. OBP allows us to vary how aggressively information is offloaded during pretraining, and using it to further explore this question would be an interesting direction for future work.

Efficiency Considerations An offloading training step retrieves the top-k results for each of the m sampled spans per sequence, and runs one additional forward pass with the evidence in context. Overall, it is several times slower than a standard language model pretraining step. At inference, selective retrieval reduces retrievals per sequence from 40.6 to 29.4 on GPT2-124M, much less than the one-retrieval-per-token cost of kNN-LM (Khandelwal et al., 2020) but still frequent. Cheaper ways to estimate the information gain during training are another important direction for future work.

Retrieval Information Gain over Training Steps The mean information gain from retrieved evidence increases with training steps (0.23 to 0.39 on SmolLM2-135M in Fig. 4). This implies that during training, OBP models learn to make better use of retrieved evidence. Due to compute constraints, our experiments thus far include training runs of a single epoch. Measuring how the gain evolves in longer training runs could inform how offloading should adapt over training, and may lead to schedules that offload more or less aggressively as training progresses.

6 Conclusion

We ask whether a language model can decide, dynamically during training, which facts to keep in its weights and which to leave to a knowledge base. In Open-Book Pretraining, a token is offloaded when retrieved evidence gives a measured gain above a threshold, and the model learns to predict that token only with the evidence in context. Across four models from two families, OBP models given the same retrieved evidence as a standard model reach lower held-out perplexity (Fig. 2) and higher open-book QA scores (Tab. 3, Tab. 4). These results suggest our training objective can effectively divide knowledge between the parameters and an external store. Our finding is a step towards models whose factual knowledge can be inspected and modified, and we hope future works could use OBP to conduct further analysis on limiting parametric memorization.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing, coding assistance and information searching. We have not used generative AI tools for design, to provide feedback on research methodology or experiments, or to interpret results. The rest of the required disclosure tasks are not applicable to this work. We have reviewed all AI-assisted work. The first drafts of the code and paper are written by the first author, and all the authors polished the final manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022. URL https://arxiv.org/abs/2112.04426.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, and Others. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2025. URL https://arxiv.org/abs/2401.08281.

Hady Elsahar, Pavlos Vougiouklis, Arslen Remaci, Christophe Gravier, Jonathon Hare, Frederique Laforest, and Elena Simperl. T-REx: A large scale alignment of natural language with knowledge base triples. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, and Others (eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), May 2018. URL https://aclanthology.org/L18-1544/.

Yair Feldman, Linxi Zhao, Nathan Godey, Dongyoung Go, Yilun Hua, Kilian Q. Weinberger, Jennifer J. Sun, and Yoav Artzi. Co-lmlm: Continuous-query limited memory language models, 2026. URL https://arxiv.org/abs/2607.07707.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, and Others. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th*

International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, pp. 3929–3938. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/guu20a.html.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 874–880, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74. URL https://aclanthology.org/2021.eacl-main.74/.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550/.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Velocity Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. Realtime QA: What's the answer right now? In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=HfKOIPCvsv.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models, 2020. URL https://arxiv.org/abs/1911.00172.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Yanhong Li, Tianyang Xu, Kenan Tang, Karen Livescu, David McAllester, and Jiawei Zhou. Okbench: Democratizing llm evaluation with fully automated, on-demand, open knowledge benchmarking, 2025. URL https://arxiv.org/abs/2511.08598.

Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models, 2022. URL https://arxiv.org/abs/2205.11388.

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

Sewon Min, Suchin Gururangan, Eric Wallace, Weijia Shi, Hannaneh Hajishirzi, Noah A. Smith, and Luke Zettlemoyer. SILO language models: Isolating legal risk in a nonparametric datastore. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ruk0nyQPec.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250/.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. KILT: a benchmark for knowledge intensive language tasks. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2523–2544, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.200. URL https://aclanthology.org/2021.naacl-main.200/.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.463. URL https://aclanthology.org/2024.naacl-long.463/.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024. URL https://arxiv.org/abs/2212.03533.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, and Others. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question

answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Linxi Zhao, Sofian Zalouk, Christian Belardi, Justin Lovelace, Jin Peng Zhou, Ryan Thomas Noonan, Dongyoung Go, Kilian Q Weinberger, Yoav Artzi, and Jennifer J. Sun. Pre-training limited memory language models with internal and external knowledge. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cvztBvlglK.

A Experimental Setup

A.1 Data Statistics

The two model families use different tokenizers, so token counts differ. The training split is 4.22B tokens under the SmolLM2 tokenizer and 3.89B under the GPT-2 tokenizer, and the held-out split is 86.4M and 79.6M tokens. With batch size 64, one epoch is 64,428 optimizer steps for the SmolLM2 models and 59,327 for the GPT-2 models. Tab. 5 reports how many tokens the offloading rule during training or inference select. During training, we measure the retrieval gain at five sampled windows per sequence, so masked tokens that pass our conditions in §3.2 are 0.4-0.5% of all training tokens. During open-book evaluation, a token counts as masked when it is inside one of that setting's retrieval windows and passes the training-time offloading rules, with the gain measured by the evaluated model on its retrieved entries. In the **always** setting, every token is inside a retrieval window; in the **selective** setting, only spans selected by our trained classifier retrieve. The always and selective columns of Tab. 5 are nearly the same although selective retrieves at about a quarter fewer positions, meaning our trained classifier skips almost only windows in which no token passes the rule.

A.2 Training Hyperparameters

Tab. 6 lists the hyperparameters shared by the baseline and OBP models, and Tab. 7 lists the model architectures. Articles are concatenated and packed into 1,024-token sequences. Position indices restart at each article boundary, so attention never crosses articles and retrieval spans stay inside one article. Each architecture uses the official configuration. The three runs per model differ in weight initialization, data order, and retrieval-position sampling.

Offloading starts after the initial warm-up phase, which is 15,000 plain language-modeling steps. After warm-up, each step samples $m = 5$ spans of $l = 10$ tokens, extended by up to 3 tokens to complete the final word. We retrieve the top-5 entries per span, truncates the retrieved text to 128 tokens, and scores each span once more with the evidence in context, so an offloading step adds one retrieval and one extra forward pass per span. We use $\tau = 1$, $\alpha = 20$, and $\beta = 1$ throughout. These heuristics are selected based on our preliminary experiments. The decision classifier (§3.3) is a two-layer MLP with hidden size 512. Classifier input is the current hidden state of the language model, at each decision position. It uses the same optimizer and schedule with the language model. Retrieval uses the same index and query format in both training and evaluation.

A.3 Templates for Retrieval Position Sampling

Due to compute constraints, we sample 5 windows for retrieval per sequence, and window sampled uniformly yields an offloaded token roughly 20% of the time. To provide more positive signals for our classifier, we selected a few fact-preceding patterns as templates, mostly copulas and prepositions. The 19 templates are *is a, is the, such as, to be, are, and, is, with, from, by, was, on, were, under, during, near, in, of, at*. We sampled 4 positions with the templates and 1 position uniformly from all word-start positions. With enough compute, it's possible to drop the heuristic and sample uniformly at every word start, same as our evaluation setting.

Table 5: Masked-token percentages and retrieval counts on the held-out evaluation pages, mean over three seeds. **Left:** percentage of masked tokens among all tokens, during OBP training and during open-book evaluation of each model and setting. **Right:** retrievals per 1,024-token sequence at evaluation; the always column retrieves at every candidate position while keeping non-overlapping retrieval spans, the selective column retrieves where the OBP classifier score exceeds τ_{inf}.

	Masked Tokens (%)				Retrievals per Seq.	
Model	Training	Baseline	OBP-always	OBP-selective	OBP-always	OBP-selective
GPT2-124M	0.44	6.2	7.4	7.4	40.6	29.3
GPT2-355M	0.42	6.0	7.2	7.1	40.6	28.8
SmolLM2-135M	0.48	5.2	8.5	8.3	42.3	32.5
SmolLM2-360M	0.47	6.2	8.1	7.9	42.3	31.4

Table 6: Training hyperparameters shared by the baseline and OBP.

Hyperparameter	Value
Epochs	1
Sequence length	1,024
Batch size	64
Optimizer	AdamW (β_1=0.9, β_2=0.999, ϵ=10^{-8})
Peak learning rate	3×10^{-4}
lr scheduler	linear warmup over 2,000 steps, cosine decay to 0
Weight decay	0
Gradient clipping	1.0
Precision	bf16 mixed
Seeds	42, 218, 7

A.4 Evaluation Setting

Every entry is embedded with a frozen `e5-base-v2` encoder, with the key as `<title>`t`<prefix>`c, where t is the page title and c is the evidence sentence up to the value, matching the shape of a prefix query. Queries use the same template with the current page title and the current sentence up to the decision position. The 151.7M vectors are stored in a FAISS IVF-PQ index.

Language Modeling The 1,000 evaluation pages are sampled uniformly from the 118K held-out articles, each concatenating to the first 1,024 tokens. Selective retrieval uses $\tau_{\text{inf}} = 0.5$.

QA Prompt format is in Tab. 8. Evidence is truncated to 768 tokens for the GPT-2 family, and to 1,024 tokens for SmolLM2. So QA scores cannot be compared across model families.

B Knowledge Base Construction

B.1 Prompt

We label every paragraph of the KILT Wikipedia snapshot (Petroni et al., 2021) with `Qwen3-4B-Instruct-2507` (Yang et al., 2025), in BF16 precision with greedy decoding and thinking mode disabled. Paragraphs longer than $8,000$ characters are split at sentence boundaries. The model returns a JSON object with a list of facts, each with a `fact` sentence, a `value`, and an `evidence` span. Each request is in the template below:

```
You build a retrieval knowledge base from Wikipedia.

Extract every atomic fact explicitly stated in the supplied paragraph.
Rewrite each fact as one short, standalone, natural declarative sentence
in the style of LAMA, for example: "Marie Curie was born in Warsaw."
```

Table 7: Model architectures. Every model is trained from random initialization with the released configuration and tokenizer.

Model	Params	#Layers	Hidden	#Heads	Vocab Size
GPT2-124M	124M	12	768	12	50,257
GPT2-355M	355M	24	1024	16	50,257
SmolLM2-135M	135M	30	576	9	49,152
SmolLM2-360M	360M	32	960	15	49,152
SmolLM2-1.7B	1.7B	24	2048	32	49,152

Table 8: One PopQA question *What is George Rankin's occupation?* in every retrieval setting. The gold answer is *politician*.

Query	Retrieved Entries	Answer
Closed-book	(no entries)	What is . . .
Gold	*Major General George James Rankin was a politician.*	George James Rankin, **politician** . . .

```
Rules:
- Express exactly one subject-relation-value fact in each sentence.
- State the full, unambiguous subject in each sentence explicitly.
- Never begin a fact with he, she, it, they, this, these, those, the
former, the latter, the following, the work, the film, the book, the
album, the song, or another expression that needs previous context.
- Resolve a source pronoun to the page title or a named entity only when
the reference is unambiguous. If the subject cannot be resolved with
certainty, skip that fact.
- Do not combine a year, nationality, language, type, location,
occupation, or creator in one fact. `X is a 1997 Spanish drama film`
must become separate release-year and type facts.
- Copy names, dates, numbers, places, and other factual values exactly
from the paragraph.
- Put the distinguishing factual value near the end when natural.
- `value` must be the shortest exact substring from the paragraph that
completes the fact.
- `evidence` must be an exact substring from the paragraph, contain
`value`, and support the fact.
- Do not use outside knowledge or infer facts that are not stated.
- Do not emit opinions, speculation, headings, or redundant paraphrases.

Good: "Actrius is a 1997 Catalan language Spanish drama film."
Good: "Animalia was originally published in 1986."
Bad: "It was originally published in 1986."

Return only JSON:
{"facts": [{"fact": "one standalone factual sentence.",
           "value": "exact value", "evidence": "exact support"}]}
Return {"facts": []} when there are no suitable facts.
```

We provide two-shot demonstrations to the model. The first demonstration is about identifying the correct pronoun with the page title, the second is about splitting one sentence into several atomic facts.

```
User:
PAGE TITLE:
Charles Gonnet

PARAGRAPH:
He was born in Laon.
```

```
Assistant:
{"facts": [{"fact": "Charles Gonnet was born in Laon.",
          "value": "Laon",
          "evidence": "He was born in Laon"}]}

User:
PAGE TITLE:
Evanoff Provincial Park

PARAGRAPH:
Evanoff Provincial Park is a provincial park in British Columbia, Canada.

Assistant:
{"facts": [{"fact": "Evanoff Provincial Park is a provincial park.",
          "value": "provincial park",
          "evidence": "Evanoff Provincial Park is a provincial park"},
         {"fact": "Evanoff Provincial Park is located in British
                   Columbia, Canada.",
          "value": "British Columbia, Canada",
          "evidence": "Evanoff Provincial Park is a provincial park in
                      British Columbia, Canada"}]}
```

Finally, we prompt the model to annotate one paragraph, with `<title>` and paragraph `<paragraph>` substituted by the specific instance to annotate.

```
USER:
PAGE TITLE:
<title>

PARAGRAPH:
<paragraph>
```

B.2 Automatic Validation

To prevent hallucination, a returned fact is kept only if all of the following conditions hold:

1. `evidence` is a verbatim substring of the original paragraph.
2. `value` is a verbatim substring of `evidence` and of `fact`.
3. `fact` is a single line of atomic fact that ends with a period and contains no question mark.
4. `fact` does not begin with a context-dependent subject (e.g. *he*, *she*, *it*, *they*, *the former*, *the latter*, *the work*, *the film*, *the book*, etc.).

The `fact` sentence is what gets retrieved as an entry during training and inference. We store the character offsets of `value` and `evidence` in the source page along with it.

Table 9: One paragraph from the training corpus, and the entries annotated from it in the default knowledge base. Each entry is an atomic fact about the entity. The italicized span is the corresponding value each retrieval should look for.

Paragraph: Gregorio Leti (1630–1701) was an Italian historian and satirist from Milan, who sometimes published under the pseudonym Abbe Gualdi, L'abbé Gualdi, or Gualdus known for his works about the Catholic Church, especially the papacy. All of his publications were listed on the "Index Librorum Prohibitorum".
Gregorio Leti was born in *1630*.
Gregorio Leti died in *1701*.
Gregorio Leti was an *Italian historian*.
Gregorio Leti was a *satirist*.
Gregorio Leti was from *Milan*.
All of Gregorio Leti's publications were listed on the *Index Librorum Prohibitorum*.

Table 10: Mean perplexity on annotated fact tokens in the held-out split measured without retrieval. All-token perplexity is in Tab. 2.

Model	Baseline	OBP
GPT2-124M	21.91	**23.17**
GPT2-355M	16.71	**17.69**
SmolLM2-135M	**29.41**	29.38
SmolLM2-360M	13.47	**13.98**

B.3 Example from Knowledge Base

Tab. 9 shows one training paragraph and the six entries annotated from it. The knowledge base has 151.7M entries across the 5.9M articles from KILT knowledge source, about 26 per article. Each entry is one short declarative sentence, stored with the character offsets of its value and evidence spans in the source page.

C More Evaluation Results

C.1 Perplexity on Annotated Fact Tokens

Although our work does not focus on static fact annotation, we denote tokens contained in the annotated fact spans of our default KB as facts in this section. On these fact tokens, we report perplexity without retrieval in Tab. 10. Our results show that OBP models memorize less than the corresponding baseline in three out of four models.

C.2 QA

Loose EM is suitable for zero-shot generation where the model isn't finetuned to match the answer then EOS format. We put additional strict EM scores for the QA benchmarks here. Strict EM follows SQuAD (Rajpurkar et al., 2016): a generation is correct if after lowercasing and removing punctuation and articles, it matches a gold answer alias exactly. Tab. 11 reports our results. For each task we finetune the baseline model on question-answer pairs alone (**"closed"**) to teach them the format, and both the baseline and the OBP model with retrieved sentences or the gold provenance sentence from KILT prepended to the question. Each run has 2 epochs at batch size 16 and learning rate 5×10^{-5}, with loss on answer tokens only. With retrieved or gold evidence in the prompt, OBP matches or improves on the baseline in almost every column, same as the zero-shot loose EM in Tab. 3. Tab. 8 has one PopQA question and generated answer example in every retrieval setting.

C.3 Information Gain Over Training Steps

Fig. 4 reports the retrieval information gain δ on the windows sampled during OBP training, once every 2,000 steps. Measurement starts after the initial warm-up phase. The mean gain rises over

Table 11: Strict EM scores (%) on Natural Questions, TriviaQA, and HotpotQA, mean over three seeds. The KILT validation splits have n=2,837 / 5,355 / 5,600 answerable questions. All models are finetuned on the task's train split: the "Closed" models on question-answer pairs alone, all others with the gold provenance sentence in the prompt (excluded from the loss). "Retrieved" prepends the top-5 retrieved rows (identical for all models; hit@5 = 41.5 / 58.8 / 21.0% on NQ / TriviaQA / HotpotQA), "Gold" prepends the gold provenance sentence. Bold marks the best value per column within each model.

		NQ			TriviaQA			HotpotQA		
Model		Closed	Retrieved	Gold	Closed	Retrieved	Gold	Closed	Retrieved	Gold
GPT2-124M	Closed	**11.2**	10.5	21.6	**11.1**	16.4	26.0	**12.7**	12.5	22.3
	Baseline	0.7	15.2	67.7	4.4	25.2	67.6	10.1	13.0	47.3
	OBP	0.7	**15.4**	**68.5**	4.2	**26.8**	**68.8**	9.5	**13.8**	**48.7**
GPT2-355M	Closed	**14.8**	12.6	26.3	**16.4**	22.1	30.0	**13.3**	13.4	26.1
	Baseline	1.1	16.0	69.4	5.1	28.3	71.4	10.4	13.8	49.6
	OBP	0.6	**16.2**	**69.5**	5.0	**29.1**	**71.8**	10.6	**14.4**	**50.1**
SmolLM2-135M	Closed	**6.3**	3.8	6.4	**3.6**	4.2	5.9	**8.8**	7.3	10.0
	Baseline	0.3	10.9	52.3	1.7	14.9	45.4	7.3	9.5	32.2
	OBP	0.4	**11.7**	**54.6**	2.0	**16.9**	**51.8**	7.1	**10.2**	**35.6**
SmolLM2-360M	Closed	**12.4**	9.4	15.1	**11.0**	12.8	17.6	**11.8**	11.2	17.8
	Baseline	0.7	13.9	59.7	3.9	22.1	63.1	9.7	12.1	42.8
	OBP	0.7	**14.7**	**61.3**	3.6	**24.0**	**64.9**	9.4	**12.1**	**44.5**

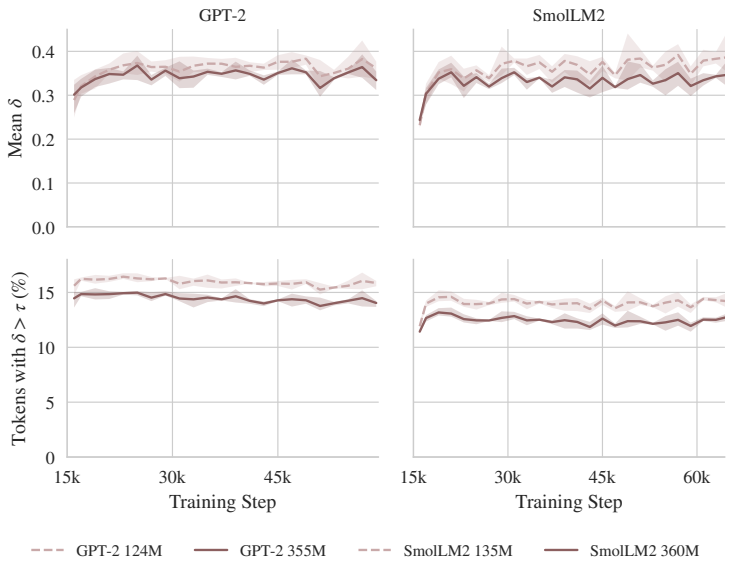


Figure 4: Retrieval information gain δ measured on the retrieval windows sampled during OBP training. Measurement starts at step 15k when offloading begins, and values are averaged over 2k-step training logs. Lines are the mean over three seeds, shaded bands show the min-max range.

training on all four models (0.23 to 0.39 on SmolLM2-135M, 0.29 to 0.36 on GPT2-124M), showing the OBP model learning to use retrieved evidence better rather than memorizing the offloaded facts. The percentage of measured span tokens with $\delta_t > \tau$ stays between 11% and 16% with no consistent trend: the amount of offloadable tokens is stable, and what increases is the gain on them. Due to compute constraints we train models of at most 360M parameters for one epoch. How the gain evolves in longer and larger runs would suggest how offloading should adapt over training, and may support a schedule that offloads more or less aggressively as training progresses.

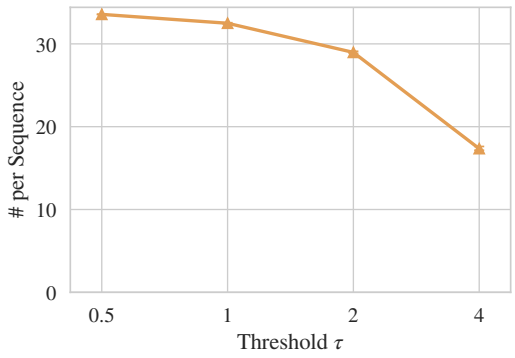


Figure 5: τ ablation results on inference retrieval rates with `SmolLM2-135M` model architecture. We report mean value over three seeds, with error bars representing min-max values.

C.4 τ ABLATION

τ is the threshold parameter on how much a token must gain from evidence before it is masked from the plain objective, so a larger τ reduces retrieval. Fig. 5 shows raising τ from 0.5 to 4 lowers selective retrieval from 33.6 to 17.4 windows per sequence on SmolLM2-135M.