Compression Beyond the Uncompressed: A Two-Stage Training Recipe for Soft Context Compression in RAG

Shuyu Guo
Shandong University
Qingdao, China
guoshuyu225@gmail.com

Shuo Zhang
Bloomberg
London, United Kingdom
szhang611@bloomberg.net

Zhaochun Ren*
Leiden University
Leiden, The Netherlands
z.ren@liacs.leidenuniv.nl

Abstract

Retrieval-Augmented Generation (RAG) enhances language models with external knowledge, but the lengthy retrieved context inflates the input and degrades inference efficiency. Soft context compression encodes each document into a substantially shorter embedding sequence. However, most existing approaches are trained by distilling outputs from uncompressed RAG systems, inherently limiting their performance relative to the original model. To address this limitation, we propose DEX-Comp, a two-stage training recipe: Pure Distillation warm-starts the compression model on the uncompressed RAG's correct responses only, and Hard Exploration then runs reinforcement learning solely on queries the uncompressed RAG fails, forcing the model to explore computation patterns better suited to compressed representations. On five open-domain QA benchmarks at retrieval depths from top-5 to top-30, DEX-Comp compresses retrieved contexts by $16\times$ and accelerates inference by $4\times$–$24\times$, while achieving performance comparable to or exceeding the uncompressed RAG baseline across retrieval depths. Ablations and evaluations across diverse datasets and backbones further confirm the contribution of each stage and the generalization of our approach.

1 Introduction

Retrieval-Augmented Generation (RAG) equips parametric language models with a non-parametric datastore, granting access to up-to-date and long-tail knowledge. By retrieving and prepending relevant documents to the user query, RAG has become a default recipe for knowledge-intensive tasks such as open-domain question answering, fact verification, and domain-specific reasoning [1]. However, this convenience comes at a price. The retrieved context inflates the input length, significantly increasing inference latency [2]. It can also exceed the context window of the underlying LLM.

Context compression is a promising remedy that shortens the retrieved context while preserving the information the LLM needs. Existing methods fall into two families. *Hard compression* operates at the token level, extracting salient spans [3] or summarizing each document into shorter text [4]. *Soft compression* goes one step further by encoding each document into a much shorter sequence of continuous embeddings [5]. Freed from discrete tokens, continuous embeddings carry higher representational capacity and often achieve stronger performance than hard compression in prior work [6]. However, existing soft compression methods are typically trained by distilling responses from an uncompressed RAG model and are therefore implicitly bounded by its behavior [6, 7]. Although

*Corresponding Author.

Preprint.

recent studies suggest that continuous embeddings can encode substantially more information per slot than discrete tokens [8], the empirical gap between soft compression models and their uncompressed counterparts remains non-trivial in practice [9].

We identify that this gap is partly due to the way soft compression models are trained. Distilling the uncompressed model's responses anchors the compression model to a computation pattern shaped by reading discrete tokens, even though its own input is a much shorter, denser embedding sequence (i.e., the model may inherit token-level reasoning behaviors that are suboptimal for compressed representations). Built on this insight, we propose **DEX-Comp** (**D**istillation and **Ex**ploration for soft **Comp**ression), a two-stage training recipe. Stage I, Pure Distillation, warm-starts the compression model by imitating only the uncompressed model's correct responses, in contrast to standard knowledge distillation, which learns from all outputs regardless of correctness, yielding a competent initialization without internalizing its mistakes. Stage II, Hard Exploration, confines reinforcement learning to queries on which the uncompressed model fails, focusing learning on cases where imitation alone is insufficient and forcing the compression model to explore behaviors better suited to operating over compressed representations, potentially surpassing the uncompressed model.

To verify the effectiveness of DEX-Comp, we conduct extensive experiments and analyses on five widely used open-domain QA benchmarks. Prior soft compression work is largely evaluated at shallow retrieval depths (typically top-5), which may not reflect realistic deployment scenarios. We therefore evaluate our method across a wide range of retrieval depths from top-5 to top-30. Despite compressing the input by $16\times$ and accelerating inference by $4\times$–$24\times$, DEX-Comp achieves performance comparable to or exceeding the uncompressed RAG model across retrieval depths and outperforms prior soft compression methods. Ablations and evaluations across diverse datasets and backbone models further confirm the contribution of each stage and the generalization of our approach. This work provides a step toward understanding and improving soft context compression beyond its current limitations.

2 Related Work

Retrieval-Augmented Generation Retrieval-Augmented Generation (RAG) improves language model performance across a wide spectrum of knowledge-intensive tasks [1, 10], including open-domain question answering [11–14], language modeling [15, 16], machine translation [17, 18], and code generation [19, 20]. A broad range of techniques have been studied to push RAG further, such as end-to-end joint training of the language model and the retriever [21], tighter integration with the non-parametric datastore [22, 23], retriever alignment via feedback from the language model [12, 24], and self-reflection mechanisms that interleave retrieval and generation [25, 26]. These efforts mostly target effectiveness, while the efficiency degradation caused by feeding lengthy retrieved context into the language model remains largely underexplored in comparison.

Context Compression Mitigating the efficiency degradation of RAG can be approached by reducing the decoder's effective computation, broadly along two lines: inference-time KV compression [27–30] and pre-inference context compression [31]. The latter directly shortens the input text and can be divided into hard and soft variants based on the form of compression. Hard compression operates on the surface form, either by extracting salient tokens [32, 3, 33, 34] or by summarizing each document into shorter text [35, 36, 4]. Soft compression instead encodes the context into a sequence of continuous embeddings that replace the textual input, and due to the greater flexibility of continuous space, it often achieves stronger performance than hard compression in existing benchmarks [6]. In this work, we focus on query-independent context compression, where documents are encoded offline without access to the query. This setting differs from query-dependent compression methods, which select or compress content conditioned on the query and often aim to improve both accuracy and efficiency [37–45, 35, 46, 47].

Previous works have advanced soft context compression from several perspectives. A number of approaches explore different compression paradigms, including auto-encoding into compressed embeddings [48, 49, 2, 50–55], cross-modal projection of retrieval vectors into the context space [7], and attention-based information aggregation [56, 57]. Another line of work studies training paradigms, spanning a pretraining stage with text reconstruction and continuation objectives [5, 58] and a supervised fine-tuning stage that either fits downstream ground-truth answers [59, 9] or distills the responses of the uncompressed RAG [7, 6]. Several works investigate the interaction between the com-

pressor and decoder, such as joint end-to-end optimization for improved downstream performance [9] and inference-time selection that enables adaptive compression ratios [60]. Some approaches also incorporate hard compression to achieve multi-granularity information compression [39].

While effective, prior work primarily focuses on imitation-based or globally optimized training objectives and does not explicitly study how to leverage the complementary strengths of imitation and exploration in soft compression. Our work addresses this limitation by introducing a two-stage training paradigm that separates reliable imitation from targeted exploration on challenging cases.

3 Method

Problem Formulation. A standard RAG framework consists of a retriever $\mathcal{R}$ and a decoder $\mathcal{T}$. The retriever encodes every document in a corpus $\mathbb{C}$ into a dense vector and stores the embeddings in an offline index. For a given query q, $\mathcal{R}$ encodes q into the same space and returns the top-k documents $\mathbf{D}_q = \{d_1, \ldots, d_k\}$ via a similarity search algorithm such as MIPS [61]. The retrieved documents are concatenated with q and fed to the decoder, which generates the response $r \sim \mathcal{T}(\cdot \mid d_1, \ldots, d_k, q)$. While retrieval grounds $\mathcal{T}$ in up-to-date and long-tail knowledge, processing the full text of k documents quickly inflates the input length, reducing inference efficiency as $|\mathbf{D}_q| \gg |q|$. To alleviate this overhead, an optional compressor $\mathcal{C}_\phi$ can be integrated to reduce each d_i's length from L_i to a more concise m_i, achieving a compression rate of L_i/m_i.

Among compression strategies, soft compression has shown strong empirical performance in prior studies. Each d_i is encoded into a continuous embedding sequence $\mathbf{e}_i = \mathcal{C}_\phi(d_i)$ that replaces the original tokens as input to a compression-aware decoder $\mathcal{G}_\theta$. Such continuous embeddings admit a much higher representational capacity than discrete tokens. In practice, however, soft compression has so far been limited by its training methodology and often does not outperform the uncompressed RAG model. The core difficulty is that the compressed space admits no direct supervision. Neither an oracle embedding $\mathbf{e}_i^\star$ for a given backbone nor oracle decoder parameters $\theta^\star$ that best consume such embeddings are known a priori. This leaves the joint (ϕ, θ) optimization unable to pin down the most effective computation strategy for operating over compressed representations. We propose **DEX-Comp** (**D**istillation and **Ex**ploration for soft **Comp**ression), a two-stage training recipe. We first introduce its architecture (§3.1) and then describe the two complementary stages for training: Pure Distillation (PD, §3.2) and Hard Exploration (HE, §3.3).

3.1 DEX-Comp Architecture

DEX-Comp consists of a compressor $\mathcal{C}_\phi$ and a decoder $\mathcal{G}_\theta$. The compressor adopts an auto-encoding paradigm. For each document d_i, the input text is concatenated with $m_i = \lfloor L_i/\tau \rfloor$ special compression tokens, where L_i is the document length and τ is a fixed compression rate. The full sequence is encoded by $\mathcal{C}_\phi$, and the final-layer hidden states at the compression-token positions form the compressed embedding sequence

$$\mathbf{e}_i = \mathcal{C}_\phi(d_i) = \left(e_i^1, \ldots, e_i^{m_i}\right).$$

The entire compression process is run offline once per document in the corpus $\mathbb{C}$, with all $\mathbf{e}_i$ precomputed and cached for instant access at inference time. The decoder is an autoregressive language model. Given the cached embeddings $\{\mathbf{e}_1, \ldots, \mathbf{e}_k\}$ of the top-k retrieved documents and a query q, $\mathcal{G}_\theta$ generates the response token by token from

$$P_\theta(r_t \mid r_{<t}, q, \mathbf{e}_1, \ldots, \mathbf{e}_k).$$

Training (ϕ, θ) jointly thus raises two questions: (1) how to cold-start the coupled system when neither $\mathcal{C}_\phi$ knows what to write into the soft slots nor $\mathcal{G}_\theta$ knows how to read them; and (2) how to adapt this initialization toward a computation strategy better suited to compressed representations. The two subsections that follow address them in turn.

3.2 Stage I: Pure Distillation

Cold-starting a jointly optimized $(\mathcal{C}_\phi, \mathcal{G}_\theta)$ is non-trivial. Since no oracle embedding is known a priori, we resort to end-to-end distillation from the uncompressed teacher $\mathcal{T}$ as in prior work [7, 6]. In contrast, we distill only the examples that the teacher already answers correctly, and refer to this stage

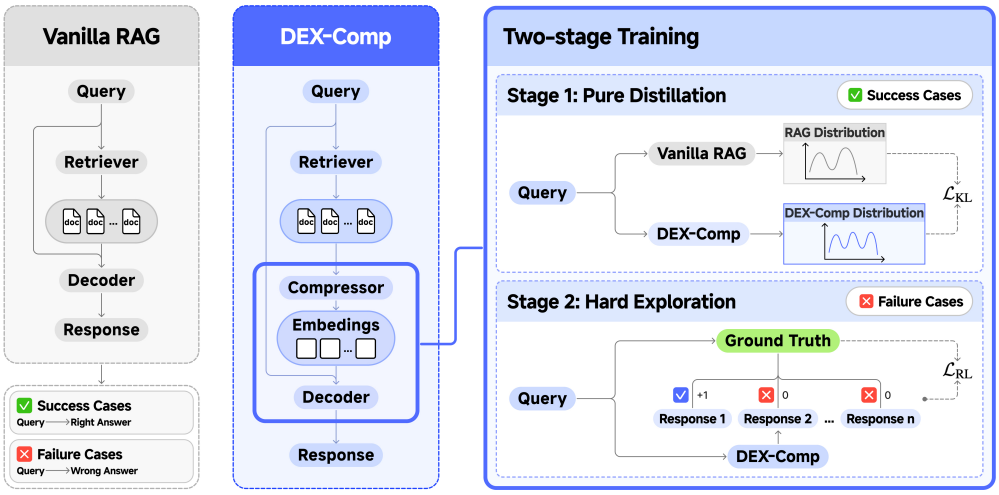

Figure 1: Overview of DEX-Comp. **Left:** a vanilla RAG pipeline feeds the raw retrieved documents to the decoder. **Middle:** DEX-Comp inserts an offline compressor that turns each document into a shorter sequence of soft embeddings, which are consumed by a compression-aware decoder at inference time. **Right:** our two-stage training recipe. Stage I (Pure Distillation) operates on RAG-correct queries and aligns the compressed model's response distribution with the uncompressed RAG model via $\mathcal{L}_{\mathrm{KL}}$. Stage II (Hard Exploration) operates on RAG-failed queries and uses outcome-based RL ($\mathcal{L}_{\mathrm{RL}}$) over groups of rollouts scored against the gold answer, allowing the compressed model to discover behaviors beyond imitation.

as Pure Distillation (PD). Specifically, we partition the training set by whether the teacher's response matches the gold answer:

$$\mathcal{S}^+ = \big\{(q, \mathbf{D}_q) : \mathrm{eval}(\hat{r}_{\mathcal{T}}, r^\star) = 1\big\}, \qquad \mathcal{S}^- = \big\{(q, \mathbf{D}_q) : \mathrm{eval}(\hat{r}_{\mathcal{T}}, r^\star) = 0\big\}, \tag{1}$$

where $\hat{r}_{\mathcal{T}} \sim \mathcal{T}(\cdot \mid \mathbf{D}_q, q)$ is the teacher's response and $\mathrm{eval}(\cdot,\cdot) \in \{0,1\}$ checks whether the prediction matches the gold answer $r^\star$.

PD operates only on the correct slice $\mathcal{S}^+$, while $\mathcal{S}^-$ is reserved for the second stage. For each $(q, \mathbf{D}_q) \in \mathcal{S}^+$, the teacher reads the original retrieved tokens while the student reads their compressed counterparts $\mathbf{e}_i = \mathcal{C}_\phi(d_i)$, and we minimize the Kullback–Leibler divergence between the two resulting response distributions:

$$\mathcal{L}_{\mathrm{PD}}(\phi, \theta) = \mathbb{E}_{(q,\mathbf{D}_q)\sim\mathcal{S}^+}\Big[D_{\mathrm{KL}}\big(P_{\mathcal{T}}(r \mid \mathbf{D}_q, q) \,\big\|\, P_\theta(r \mid \mathbf{e}_1, \ldots, \mathbf{e}_k, q)\big)\Big]. \tag{2}$$

Notably, the gradient flows through both θ and, via $\mathbf{e}_i$, through ϕ.

Rationale. The filtering to $\mathcal{S}^+$ reflects a fundamentally different rationale from prior work. Existing methods take the uncompressed model as an oracle and try to fully clone its behavior over the entire training distribution. In contrast, we treat imitation merely as a fast route to a competent initialization. Cloning is neither necessary nor desirable in our setting, since the compressor and decoder operate on representations that are qualitatively different from original tokens, and their optimal computation strategy does not necessarily coincide with that of the teacher. We therefore expose the student only to the teacher's reliable demonstrations and leave the search for a more suitable computation strategy for compressed inputs to the second stage.

3.3 Stage II: Hard Exploration

A growing body of evidence suggests that soft embeddings can losslessly encode much more information per slot than discrete tokens, yet existing soft compression methods remain far below this theoretical ceiling [8]. Pure imitation cannot fully bridge this gap. The student may be limited by the teacher if trained purely via imitation, and therefore benefits from additional exploration. Building on the competent initialization obtained in Stage I, we make the student explore exclusively on $\mathcal{S}^-$ via reinforcement learning, and refer to this stage as Hard Exploration (HE).

Specifically, we adopt Group Relative Policy Optimization (GRPO) [62]. For each query $q \in \mathcal{S}^-$, we draw a group of G candidate responses from the current student policy $\pi_{\phi,\theta}$:

$$r^{(g)} \sim \pi_{\phi,\theta}\big(\cdot \mid \mathbf{e}_1, \ldots, \mathbf{e}_k, q\big), \qquad \mathbf{e}_i = \mathcal{C}_\phi(d_i), \qquad g = 1, \ldots, G, \tag{3}$$

and assign each rollout a binary outcome reward

$$R^{(g)} = \mathrm{eval}(r^{(g)}, r^\star) \in \{0, 1\}, \tag{4}$$

checking whether it matches the gold answer. Group-relative advantages are then formed by standardizing rewards within each group:

$$A^{(g)} = \frac{R^{(g)} - \mu_R}{\sigma_R + \epsilon}, \qquad \mu_R = \frac{1}{G}\sum_{g=1}^{G} R^{(g)}, \qquad \sigma_R^2 = \frac{1}{G}\sum_{g=1}^{G}\big(R^{(g)} - \mu_R\big)^2. \tag{5}$$

The student is then updated with the clipped policy-gradient surrogate, regularized toward the PD checkpoint π_{ref}:

$$\mathcal{L}_{\text{HE}}(\phi, \theta) \;=\; -\mathbb{E}_{q\sim\mathcal{S}^-}\left[\frac{1}{G}\sum_{g=1}^{G}\frac{1}{|r^{(g)}|}\sum_{t=1}^{|r^{(g)}|}\min\Big(\rho_t^{(g)}A^{(g)}, \mathrm{clip}\big(\rho_t^{(g)}, 1-\varepsilon, 1+\varepsilon\big)A^{(g)}\Big)\right] \tag{6}$$
$$+\beta\, D_{\text{KL}}[\pi_{\phi,\theta} \,\|\, \pi_{\text{ref}}],$$

where

$$\rho_t^{(g)} = \frac{\pi_{\phi,\theta}(r_t^{(g)} \mid r_{<t}^{(g)}, q, \mathbf{e}_{1:k})}{\pi_{\text{old}}(r_t^{(g)} \mid r_{<t}^{(g)}, q, \mathbf{e}_{1:k})}, \tag{7}$$

ε is the clipping range, and β controls the KL regularization.

Confining exploration to $\mathcal{S}^-$ is deliberate. Any gain on this slice is more likely to arise from behaviors not captured by the teacher's predictions. This focused exploration on teacher-failed queries pushes the student beyond imitation, and empirically enables DEX-Comp to match or exceed its uncompressed teacher across standard RAG benchmarks (§5.1).

4 Experimental Setup

4.1 Datasets

We train and evaluate our method on five widely used open-domain QA datasets covering complementary capabilities: (1) Natural Questions [63] and TriviaQA [64] for factoid QA, (2) HotpotQA [65] for multi-hop QA, (3) ASQA [66] for ambiguous long-form QA, and (4) PopQA [67] for long-tail QA. For the retrieval corpus, we use the Wikipedia-KILT corpus split into 128-token chunks following Rau et al. [9].

4.2 Baseline Methods

We compare DEX-Comp against two groups of baselines.

(i) **Uncompressed RAG.** This baseline feeds the full retrieved context into the language model without compression. Note that we omit task-specific training (e.g., RL) for the uncompressed RAG as in prior works due to its unacceptable computational costs on extensive contexts, which our compression method naturally alleviates. Furthermore, we focus on the unprecedented milestone of surpassing standard uncompressed performance, leaving the potential benefits of distilling from a task-tuned model to future work.

(ii) **Query-independent compression methods.** These methods, including DEX-Comp, encode each document offline without access to the query and therefore perform pure information compression rather than query-conditioned selection.

As soft compression generally outperforms hard compression and aligns with our setting, we include one strong hard-compression baseline, LLMLingua-2 [32], alongside four representative soft-compression baselines: xRAG [7], ICAE [5], COCOM [9], and PISCO [6], with PISCO being the strongest prior method. For a fair comparison, all methods use Mistral-7B [68] (or its fine-tuned variant) as the decoder backbone.

Table 1: Results at different retrieval depths (top-5 to top-30). For each retrieval depth, the best result is in bold, and the second-best is underlined. All methods perform inference with Mistral-7B.

Method	Training Top-k	Comp. Rate	NQ		TriviaQA		HotpotQA		ASQA		PopQA		Avg	
			CEM	LLM	CEM	LLM	CEM	LLM	CEM	LLM	CEM	LLM	CEM	LLM
Top-5														
Mistral-7B	-	-	58.65	73.67	90.18	89.04	46.45	55.57	68.99	77.00	58.53	57.36	64.56	70.53
LLMLingua-2	-	4×	49.74	63.59	84.41	83.19	36.43	45.50	59.49	65.30	39.07	39.70	53.83	59.46
xRAG	Top-5	128×	37.36	51.46	78.06	78.54	27.66	35.27	43.25	50.63	28.15	29.12	42.90	49.00
ICAE	Top-5	4×	41.59	56.33	77.78	79.27	28.75	40.48	47.68	60.65	38.28	40.24	46.82	55.39
COCOM	Top-5	16×	30.35	45.05	67.99	71.22	23.25	34.71	37.87	48.31	19.74	22.37	35.84	44.33
PISCO	Top-5	16×	56.61	70.81	88.01	87.06	43.88	51.80	66.24	74.26	55.95	54.78	62.14	67.74
PISCO	Top-30	16×	52.80	67.75	87.00	87.33	40.46	50.73	63.19	72.26	46.54	47.90	58.00	65.19
DEX-Comp(Ours)	Top-30	128×	59.29	73.46	91.32	87.92	44.52	53.82	69.51	77.64	50.80	49.63	63.09	68.49
DEX-Comp(Ours)	Top-30	16×	59.64	74.16	91.69	89.93	46.68	56.84	71.52	78.38	54.95	54.78	64.90	70.82
DEX-Comp(Ours)	Top-5	16×	**62.71**	**76.52**	**92.72**	**89.97**	**51.04**	**57.55**	**75.21**	**79.96**	**60.22**	**58.70**	**68.38**	**72.54**
Top-15														
Mistral-7B	-	-	58.34	73.49	89.65	89.26	45.79	54.91	68.67	78.16	57.24	56.88	63.94	70.54
LLMLingua-2	-	4×	50.30	65.00	84.71	82.70	35.66	44.95	60.02	66.35	38.75	38.18	53.89	59.43
PISCO	Top-5	16×	55.90	70.99	87.13	86.14	41.20	49.98	65.61	74.58	51.42	50.20	60.25	66.38
PISCO	Top-30	16×	52.13	68.14	87.06	87.13	40.50	50.52	64.45	73.21	47.05	48.20	58.24	65.44
DEX-Comp(Ours)	Top-30	16×	61.26	75.93	91.52	89.86	48.00	58.39	71.94	79.32	55.65	55.55	65.67	71.81
DEX-Comp(Ours)	Top-15	16×	**63.27**	**75.61**	**91.84**	**90.38**	**50.36**	**59.30**	**73.84**	**80.27**	**59.87**	**58.84**	**67.83**	**72.88**
Top-30														
Mistral-7B	-	-	57.42	72.75	88.98	88.20	45.45	54.64	67.41	76.79	55.30	55.04	62.91	69.49
LLMLingua-2	-	4×	49.35	63.98	83.42	81.50	34.64	43.89	60.55	67.41	37.74	31.28	53.14	57.61
PISCO	Top-5	16×	50.33	63.34	83.77	82.67	36.88	43.96	60.76	68.57	39.87	38.56	54.32	59.42
PISCO	Top-30	16×	51.99	67.57	86.69	86.59	40.46	50.20	64.24	72.68	46.12	46.89	57.90	64.79
DEX-Comp(Ours)	Top-30	16×	**60.31**	**75.57**	**91.62**	**90.12**	**48.16**	**58.48**	**73.84**	**80.38**	**56.23**	**55.53**	**66.03**	**72.02**

4.3 Evaluation

We evaluate at retrieval depths from top-5 to top-30, since prior compression work has largely focused on shallow settings ($k \leq 5$), which may not reflect realistic deployments. To enable fair comparison at larger retrieval depths, we additionally retrain PISCO at top-30.

We adopt two complementary evaluation metrics: (i) **Containment Exact Match (CEM)**, the standard metric in prior compression work [7, 6], which checks whether the ground-truth answer appears in the model prediction; (ii) **LLM-as-judge**, where a language model determines whether a prediction is correct given the ground truth.

Specifically, we use Gemini 3 Flash[2] as the judge model. We note that Gemini 3 Flash is a proprietary model; however, we provide the full prompt and also report CEM as a fully reproducible metric. The full judging prompt is provided in Appendix A.

4.4 Implementation Details

We implement data preprocessing and document retrieval using the BERGEN library [69], with Splade-v3 [70] as the retriever and DeBERTa-v3 [71] as the reranker. For all methods, we use Mistral-7B-Instruct [68] as the backbone LLM, with a decoding temperature of 0 for deterministic generation.

For DEX-Comp, the compressor and decoder share the same backbone LLM and are implemented as two separate LoRA [72] adapters, substantially reducing training cost. We train DEX-Comp variants at multiple retrieval depths and compression rates to support further analysis.

All experiments are implemented in PyTorch and Transformers and run on 8 RTX PRO 6000 Blackwell GPUs. Hyperparameters for Pure Distillation and Hard Exploration are provided in Appendix B and Appendix C, respectively.

5 Experimental Results

5.1 Main Results

Following prior work, we first evaluate DEX-Comp with training and inference using the same top-k. As shown in Table 1, at $16\times$ compression DEX-Comp surpasses the uncompressed RAG baseline on every dataset and retrieval depth, with statistically significant gains ($p < 0.05$, paired permutation

[2] `https://ai.google.dev/gemini-api/docs/models/gemini-3-flash-preview`

Table 2: Inference efficiency comparison between RAG and DEX-Comp across different top-k retrieval settings. We report Time-To-First-Token (TTFT), computational cost (GFLOPs), and peak GPU memory (MB).

Top-k	TTFT (ms)			GFLOPs			Memory (MB)		
	RAG	DEX-Comp	Speedup	RAG	DEX-Comp	Reduction	RAG	DEX-Comp	Reduction
Top-5	373	84	**4.42**×	82,850	12,423	**6.67**×	16,100	14,242	**1.13**×
Top-15	1,626	131	**12.33**×	245,650	22,654	**10.84**×	26,887	14,457	**1.86**×
Top-30	4,782	201	**23.73**×	518,767	38,129	**13.61**×	59,328	14,766	**4.02**×

test) and average CEM / LLM improvements above two points ($+3.82/+2.01$, $+3.89/+2.34$, $+3.12/+2.53$ at $k=5/15/30$).

To our knowledge, this is the first query-independent soft compression method shown to surpass the uncompressed RAG model across multiple realistic retrieval depths, establishing a new state of the art among comparable compression methods.

Even when using a single model trained at top-30 for all retrieval depths, DEX-Comp continues to outperform the uncompressed RAG model in terms of average score ($+0.34/+0.29$, $+1.73/+1.27$, $+3.12/+2.53$ at $k=5/15/30$), and on all datasets except PopQA at top-5 and top-15, demonstrating strong cross-depth generalization.

5.2 Computational Efficiency

We measure inference efficiency on a single RTX PRO 6000 Blackwell GPU with batch size 8, reporting Time-To-First-Token (TTFT), computational cost (GFLOPs), and peak GPU memory in Table 2. DEX-Comp outperforms the uncompressed RAG model on all efficiency metrics across retrieval depths, and the gains increase with k.

At top-30, DEX-Comp reduces peak GPU memory by $4.02\times$, computational cost by $13.61\times$, and accelerates TTFT by $23.73\times$, demonstrating substantial efficiency improvements under high retrieval settings.

Offline cost. DEX-Comp requires a one-time offline compression pass over the corpus and storage of compressed embeddings. In our setup, corpus compression is performed once and amortized across all queries. The resulting embedding remains fixed unless the corpus is updated, and for dynamic corpora, incremental recompression can be applied only to updated documents.

6 Analysis

6.1 Evaluation Beyond the Overall Score

Beyond overall accuracy, we analyze performance using two RAG metrics [7]. The *Resilience Rate* measures correctness with and without retrieval, reflecting robustness to noisy context. The *Boost Rate* measures cases where retrieval corrects an initially incorrect answer, reflecting effective use of retrieved information. While our evaluation focuses on answer correctness, soft compression introduces opacity in how retrieved information is utilized; future work could incorporate faithfulness or citation-based evaluation.

Figure 2 shows that DEX-Comp surpasses the uncompressed RAG baseline (without task-specific training) on both metrics when training and inference depths match, indicating stronger robustness and information utilization. Gains are larger for Resilience Rate, suggesting that compression acts as an implicit denoising mechanism.

6.2 What makes DEX-Comp effective?

Table 3 shows that both stages are essential. Removing either Pure Distillation or Hard Exploration leads to consistent performance drops. Distilling only teacher-correct examples outperforms using all responses, and RL on harder subsets yields stronger gains. These results indicate that the two stages are complementary and jointly responsible for the improvements.

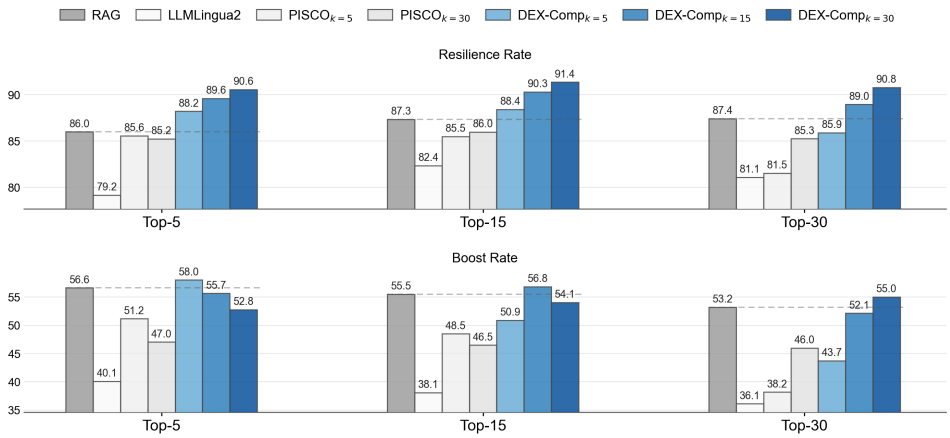

Figure 2: Resilience Rate (top) and Boost Rate (bottom) at retrieval depths $k \in \{5, 15, 30\}$. The subscript k indicates the top-k used during training. Comparison includes DEX-Comp, the uncompressed RAG model, and other compression methods.

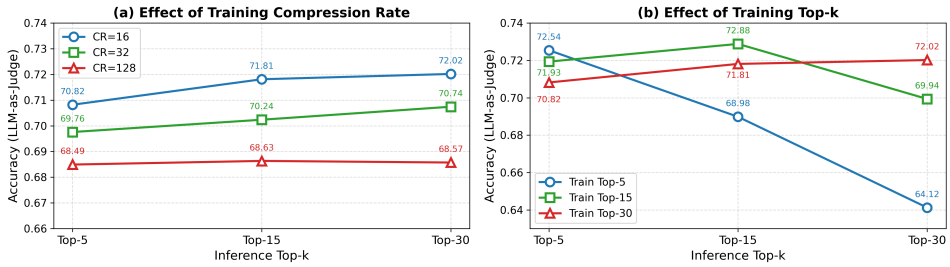

Figure 3: Effect of (a) the compression rate (CR) and (b) the retrieval depth during training. Each panel varies one parameter while keeping the other fixed.

6.3 How to choose training parameters?

Figure 3 studies key training parameters. Accuracy improves as compression rate decreases, consistent with higher representational capacity. CR = 16 achieves the best performance and surpasses the uncompressed RAG baseline, while CR = 32 already matches it, indicating a flexible accuracy–efficiency trade-off. For retrieval depth, performance is highest when training and inference match. However, training at larger k generalizes better across depths, making it preferable for deployment.

6.4 Generalization

DEX-Comp generalizes across both model architectures and data distributions. It surpasses the corresponding uncompressed RAG baseline across backbone families and matches or exceeds it on BioASQ [73], CovidQA [74], and FEVER [75], indicating robustness across domains and tasks. This suggests that the learned compression patterns are not tightly coupled to a specific backbone or dataset.

6.5 How do compression embeddings work?

We analyze learned embeddings using cosine similarity and logit lens projections [76]. Embeddings exhibit spatial specialization, with some focusing on specific document regions and others capturing global information. This suggests that DEX-Comp allocates its limited embedding capacity between local and global representations rather than treating all embeddings uniformly.

Table 3: Ablation on the two-stage recipe. Stage I varies the SFT data (all teacher responses vs. the teacher-correct subset $\mathcal{S}^+$), and Stage II varies the RL data (none vs. correct-only vs. mixed vs. teacher-failed $\mathcal{S}^-$ used by HE). Subscripts report the absolute drop from DEX-Comp.

Method	NQ	TriviaQA	HotpotQA	ASQA	PopQA	Avg		
						Acc	Resilience	Boost
DEX-Comp(Ours)	**75.57**	**90.12**	**58.48**	**80.38**	**55.53**	**72.02**	**90.78**	**55.00**
SFT (all)	67.57$_{\downarrow 8.00}$	86.59$_{\downarrow 3.53}$	50.20$_{\downarrow 8.28}$	72.68$_{\downarrow 7.70}$	46.89$_{\downarrow 8.64}$	64.79$_{\downarrow 7.23}$	85.25$_{\downarrow 5.53}$	45.96$_{\downarrow 9.04}$
PD only	70.96$_{\downarrow 4.61}$	88.74$_{\downarrow 1.38}$	55.32$_{\downarrow 3.16}$	74.79$_{\downarrow 5.59}$	48.95$_{\downarrow 6.58}$	67.75$_{\downarrow 4.27}$	87.63$_{\downarrow 3.15}$	49.56$_{\downarrow 5.44}$
PD + RL (correct)	74.41$_{\downarrow 1.16}$	90.11$_{\downarrow 0.01}$	57.30$_{\downarrow 1.18}$	77.43$_{\downarrow 2.95}$	53.74$_{\downarrow 1.79}$	70.60$_{\downarrow 1.42}$	89.58$_{\downarrow 1.20}$	53.31$_{\downarrow 1.69}$
PD + RL (mixed)	75.15$_{\downarrow 0.42}$	90.10$_{\downarrow 0.02}$	58.13$_{\downarrow 0.35}$	78.90$_{\downarrow 1.48}$	55.07$_{\downarrow 0.46}$	71.47$_{\downarrow 0.55}$	90.13$_{\downarrow 0.65}$	54.49$_{\downarrow 0.51}$

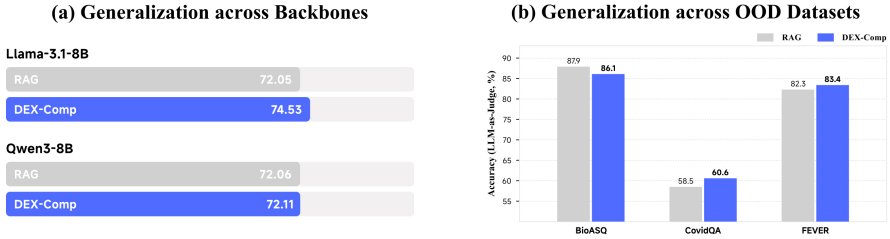

Figure 4: Generalization of DEX-Comp (a) across backbone model families and (b) across out-of-distribution datasets.

7 Limitations

DEX-Comp assumes a query-independent, offline compression setting, which requires precomputing and storing compressed embeddings for the corpus. While this cost is amortized across queries, it may become significant for frequently updated or very large-scale corpora.

Given the inherent difficulty of disentangling the effects of compression from task-specific RL, our current setup does not strictly isolate these gains. A more controlled comparison is left for future work. Finally, our evaluation focuses on QA benchmarks and answer correctness metrics. While we observe strong empirical performance, future work should investigate faithfulness, grounding, and robustness under adversarial or long-form generation settings.

8 Conclusion

We propose DEX-Comp, a two-stage training recipe for soft context compression that combines Pure Distillation with Hard Exploration. To our knowledge, DEX-Comp is the first query-independent soft compression method that surpasses the uncompressed RAG model across multiple retrieval settings.

At top-30 retrieval, DEX-Comp compresses the retrieved context by $16\times$ and accelerates inference by over $20\times$, while achieving performance comparable to or exceeding uncompressed RAG across five benchmarks. Further analyses confirm the contribution of each stage and demonstrate that DEX-Comp generalizes across backbone models and data distributions.

We believe this work provides a step toward understanding and improving soft context compression, and toward narrowing the gap between compressed and uncompressed RAG systems.

References

[1] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997, 2023. doi: 10.48550/ARXIV.2312.10997. URL `https://doi.org/10.48550/arXiv.2312.10997`.

[2] Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of*

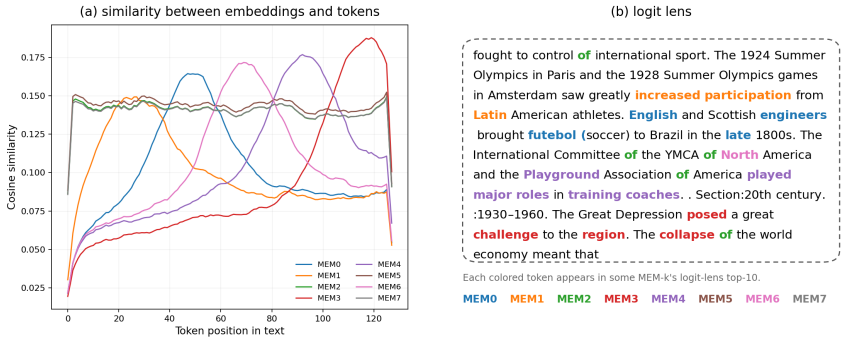

Figure 5: Probing learned compression embeddings. (a) Cosine similarity between embeddings and tokens. (b) Logit lens projections highlighting top tokens associated with each embedding.

the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 3829–3846. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.232. URL https://doi.org/10.18653/v1/2023.emnlp-main.232.

[3] Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.825. URL https://doi.org/10.18653/v1/2023.emnlp-main.825.

[4] Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Ye Qi, and Zhicheng Dou. Hierarchical document refinement for long-context retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 3502–3520. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.176/.

[5] Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=uREj4ZuGJE.

[6] Maxime Louis, Hervé Déjean, and Stéphane Clinchant. PISCO: pretty simple compression for retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 15506–15521. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.800/.

[7] Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. xRAG: Extreme context compression for retrieval-augmented generation with one token. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/c5cf13bfd3762821ef7607e63ee90075-Abstract-Conference.html.

[8] Yuri Kuratov, Mikhail Arkhipov, Aydar Bulatov, and Mikhail Burtsev. Cramming 1568 tokens into a single vector and back again: Exploring the limits of embedding space capacity. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 19323–19339. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.948/.

[9] David Rau, Shuai Wang, Hervé Déjean, and Stéphane Clinchant. Context embeddings for efficient answer generation in RAG. *CoRR*, abs/2407.09252, 2024. doi: 10.48550/ARXIV.2407.09252. URL https://doi.org/10.48550/arXiv.2407.09252.

[10] Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. Retrieval-based language models and applications. In Yun-Nung Vivian Chen, Margot Mieskes, and Siva Reddy, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 41–46. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-TUTORIALS.6. URL https://doi.org/10.18653/v1/2023.acl-tutorials.6.

[11] Patrick S. H. 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 Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

[12] 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 Gómez-Adorno, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 8371–8384. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.NAACL-LONG.463. URL https://doi.org/10.18653/v1/2024.naacl-long.463.

[13] Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. Retrieval-augmented generation with knowledge graphs for customer service question answering. In Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang, editors, *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pages 2905–2909. ACM, 2024. doi: 10.1145/3626772.3661370. URL https://doi.org/10.1145/3626772.3661370.

[14] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6233–6251. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.372. URL https://doi.org/10.18653/v1/2024.findings-acl.372.

[15] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2097–2118. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.132. URL https://doi.org/10.18653/v1/2023.findings-acl.132.

[16] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/ebd82705f44793b6f9ade5a669d0f0bf-Abstract-Conference.html.

[17] Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=7wCBOfJ8hJM.

[18] Xin Cheng, Shen Gao, Lemao Liu, Dongyan Zhao, and Rui Yan. Neural machine translation with contrastive translation memories. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 3591–3601. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.235. URL https://doi.org/10.18653/v1/2022.emnlp-main.235.

[19] Xiangyu Zhang, Yu Zhou, Guang Yang, and Taolue Chen. Syntax-aware retrieval augmented code generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 1291–1302. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.90. URL https://doi.org/10.18653/v1/2023.findings-emnlp.90.

[20] Fengji Zhang, Bei Chen, Yue Zhang, Jacky Keung, Jin Liu, Daoguang Zan, Yi Mao, Jian-Guang Lou, and Weizhu Chen. RepoCoder: Repository-level code completion through iterative retrieval and generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2471–2484. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.151. URL https://doi.org/10.18653/v1/2023.emnlp-main.151.

[21] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR, 2020. URL http://proceedings.mlr.press/v119/guu20a.html.

[22] 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. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR, 2022. URL https://proceedings.mlr.press/v162/borgeaud22a.html.

[23] Xin Cheng, Yankai Lin, Xiuying Chen, Dongyan Zhao, and Rui Yan. Decouple knowledge from parameters for plug-and-play language modeling. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14288–14308, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.901. URL https://aclanthology.org/2023.findings-acl.901/.

[24] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. RA-DIT: retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=22OTbutug9.

[25] Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. Lift yourself up: Retrieval-augmented text generation with self-memory. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December*

10 - 16, 2023, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/887262aeb3eafb01ef0fd0e3a87a8831-Abstract-Conference.html.

[26] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

[27] Jitai Hao, Qiang Huang, Yaowei Wang, Min Zhang, and Jun Yu. DeltaKV: Residual-based KV cache compression via long-range similarity. *CoRR*, abs/2602.08005, 2026. doi: 10.48550/ARXIV.2602.08005. URL https://doi.org/10.48550/arXiv.2602.08005.

[28] Jitai Hao, Yuke Zhu, Tian Wang, Jun Yu, Xin Xin, Bo Zheng, Zhaochun Ren, and Sheng Guo. OmniKV: Dynamic context selection for efficient long-context LLMs. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=ulCAPXYXfa.

[29] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/28ab418242603e0f7323e54185d19bde-Abstract-Conference.html.

[30] Chi-Chih Chang, Wei-Cheng Lin, Chien-Yu Lin, Chong-Yan Chen, Yu-Fang Hu, Pei-Shuo Wang, Ning-Chi Huang, Luis Ceze, Mohamed S. Abdelfattah, and Kai-Chiang Wu. Palu: KV-Cache compression with low-rank projection. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=LWMS4pk2vK.

[31] Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. *CoRR*, abs/2410.12388, 2024. doi: 10.48550/ARXIV.2410.12388. URL https://doi.org/10.48550/arXiv.2410.12388.

[32] Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pages 963–981. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.57. URL https://doi.org/10.18653/v1/2024.findings-acl.57.

[33] Yixiong Fang, Tianran Sun, Yuling Shi, and Xiaodong Gu. AttentionRAG: Attention-guided context pruning in retrieval-augmented generation. *CoRR*, abs/2503.10720, 2025. doi: 10.48550/ARXIV.2503.10720. URL https://doi.org/10.48550/arXiv.2503.10720.

[34] Yucheng Li. Unlocking context constraints of LLMs: Enhancing context efficiency of LLMs with self-information-based content filtering. *CoRR*, abs/2304.12102, 2023. doi: 10.48550/ARXIV.2304.12102. URL https://doi.org/10.48550/arXiv.2304.12102.

[35] Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: improving retrieval-augmented lms with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mlJLVigNHp.

[36] Quancai Liu, Haihui Fan, Jinchao Zhang, Xiangfang Li, Chuanrong Li, and Bo Li. Discomp: A two-stage prompt optimization framework combining task-agnostic and task-aware compression. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA,*

April 29 - May 4, 2025, Findings of ACL, pages 1033–1044. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.58. URL https://doi.org/10.18653/v1/2025.findings-naacl.58.

[37] Yunhao Liu, Zian Jia, Xinyu Gao, Kanjun Xu, and Yun Xiong. Rethinking soft compression in retrieval-augmented generation: A query-conditioned selector perspective. In Hakim Hacid, Yoelle Maarek, Francesco Bonchi, Ido Guy, and Emine Yilmaz, editors, *Proceedings of the ACM Web Conference 2026, WWW 2026, Dubai, United Arab Emirates, originally scheduled for April 13-17, 2026, rescheduled for June 29 - July 3, 2026*, pages 2441–2452. ACM, 2026. doi: 10.1145/3774904.3792700. URL https://doi.org/10.1145/3774904.3792700.

[38] Yi Zhao, Zuchao Li, Hai Zhao, Baoyuan Qi, and Guoming Liu. DAC: A dynamic attention-aware approach for task-agnostic prompt compression. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 19395–19407. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.952/.

[39] Yiqiao Jin, Kartik Sharma, Vineeth Rakesh, Yingtong Dou, Menghai Pan, Mahashweta Das, and Srijan Kumar. SARA: selective and adaptive retrieval-augmented generation with context compression. *CoRR*, abs/2507.05633, 2025. doi: 10.48550/ARXIV.2507.05633. URL https://doi.org/10.48550/arXiv.2507.05633.

[40] Maxime Louis, Thibault Formal, Hervé Déjean, and Stéphane Clinchant. OSCAR: online soft compression and reranking. *CoRR*, abs/2504.07109, 2025. doi: 10.48550/ARXIV.2504.07109. URL https://doi.org/10.48550/arXiv.2504.07109.

[41] Yunlong Zhao, Haoran Wu, and Bo Xu. Leveraging attention to effectively compress prompts for long-context LLMs. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 26048–26056. AAAI Press, 2025. doi: 10.1609/AAAI.V39I24.34800. URL https://doi.org/10.1609/aaai.v39i24.34800.

[42] Barys Liskavets, Maxim Ushakov, Shuvendu Roy, Mark Klibanov, Ali Etemad, and Shane K. Luke. Prompt compression with context-aware sentence encoding for fast and improved LLM inference. In Toby Walsh, Julie Shah, and Zico Kolter, editors, *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 24595–24604. AAAI Press, 2025. doi: 10.1609/AAAI.V39I23.34639. URL https://doi.org/10.1609/aaai.v39i23.34639.

[43] Taeho Hwang, Sukmin Cho, Soyeong Jeong, Hoyun Song, SeungYoon Han, and Jong C. Park. EXIT: context-aware extractive compression for enhancing retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 4895–4924. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.253/.

[44] Nadezhda Chirkova, Thibault Formal, Vassilina Nikoulina, and Stéphane Clinchant. Provence: efficient and robust context pruning for retrieval-augmented generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=TDy5Ih78b4.

[45] Ziqiang Cui, Yunpeng Weng, Xing Tang, Peiyang Liu, Shiwei Li, Bowei He, Jiamin Chen, Xiuqiang He, and Chen Ma. CORE: lossless compression for retrieval-augmented LLMs via reinforcement learning. *CoRR*, abs/2508.19282, 2025. doi: 10.48550/ARXIV.2508.19282. URL https://doi.org/10.48550/arXiv.2508.19282.

[46] Zhiruo Wang, Jun Araki, Zhengbao Jiang, Md. Rizwan Parvez, and Graham Neubig. Learning to filter context for retrieval-augmented generation. *CoRR*, abs/2311.08377, 2023. doi: 10.48550/ARXIV.2311.08377. URL https://doi.org/10.48550/arXiv.2311.08377.

[47] Chanwoong Yoon, Taewhoo Lee, Hyeon Hwang, Minbyul Jeong, and Jaewoo Kang. Compact: Compressing retrieved documents actively for question answering. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 21424–21439. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.1194. URL https://doi.org/10.18653/v1/2024.emnlp-main.1194.

[48] Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/47e288629a6996a17ce50b90a056a0e1-Abstract-Conference.html.

[49] David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5621–5634. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.FINDINGS-EMNLP.412. URL https://doi.org/10.18653/v1/2022.findings-emnlp.412.

[50] Guanghui Qin, Corby Rosset, Ethan C. Chau, Nikhil Rao, and Benjamin Van Durme. Dodo: Dynamic contextual compression for decoder-only LMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9961–9975. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.536. URL https://doi.org/10.18653/v1/2024.acl-long.536.

[51] Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. Long context compression with activation beacon. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=1eQT9OzfNQ.

[52] Jesse Mu, Xiang Li, and Noah D. Goodman. Learning to compress prompts with gist tokens. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract-Conference.html.

[53] Shaoshen Chen, Yangning Li, Zishan Xu, Yongqin Zeng, Shunlong Wu, Xinshuo Hu, Zifei Shan, Xin Su, Jiwei Tang, Yinghui Li, and Hai-Tao Zheng. DAST: context-aware compression in LLMs via dynamic allocation of soft tokens. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, Findings of ACL, pages 20544–20552. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.1055/.

[54] Guojie Liu, Yiqi Wang, Yanfeng Yang, Wenqi Fan, Songlei Jian, Jianfeng Zhang, and Jie Yu. Cognitive chunking for soft prompts: Accelerating compressor learning via block-wise causal masking. *CoRR*, abs/2602.13980, 2026. doi: 10.48550/ARXIV.2602.13980. URL https://doi.org/10.48550/arXiv.2602.13980.

[55] Runsong Zhao, Xin Liu, Xinyu Liu, Pengcheng Huang, Chunyang Xiao, Tong Xiao, and JingBo Zhu. Position ids matter: An enhanced position layout for efficient context compression in large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP*

2025, Suzhou, China, November 4-9, 2025, pages 17715–17734. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-emnlp.962/.

[56] Xin Liu, Runsong Zhao, Pengcheng Huang, Xinyu Liu, Junyi Xiao, Chunyang Xiao, Tong Xiao, Shengxiang Gao, Zhengtao Yu, and Jingbo Zhu. Autoencoding-free context compression for LLMs via contextual semantic anchors. *CoRR*, abs/2510.08907, 2025. doi: 10.48550/ARXIV.2510.08907. URL https://doi.org/10.48550/arXiv.2510.08907.

[57] Yair Feldman and Yoav Artzi. Simple context compression: Mean-pooling and multi-ratio training. *CoRR*, abs/2510.20797, 2025. doi: 10.48550/ARXIV.2510.20797. URL https://doi.org/10.48550/arXiv.2510.20797.

[58] Yuhong Dai, Jianxun Lian, Yitian Huang, Wei Zhang, Mingyang Zhou, Mingqi Wu, Xing Xie, and Hao Liao. Pretraining context compressor for large language models with embedding-based memory. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 28715–28732. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.acl-long.1394/.

[59] Sijun Tan, Xiuyu Li, Shishir G. Patil, Ziyang Wu, Tianjun Zhang, Kurt Keutzer, Joseph Gonzalez, and Raluca A. Popa. LLoCO: Learning long contexts offline. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 17605–17621. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.975. URL https://doi.org/10.18653/v1/2024.emnlp-main.975.

[60] Shuyu Guo, Shuo Zhang, and Zhaochun Ren. Enhancing RAG efficiency with adaptive context compression. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 24061–24076. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-emnlp.1307/.

[61] 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, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

[62] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

[63] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/TACL_A_00276. URL https://doi.org/10.1162/tacl_a_00276.

[64] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

[65] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. 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, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1259. URL https://doi.org/10.18653/v1/d18-1259.

[66] Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. ASQA: factoid questions meet long-form answers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8273–8288. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.566. URL https://doi.org/10.18653/v1/2022.emnlp-main.566.

[67] 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 L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.546. URL https://doi.org/10.18653/v1/2023.acl-long.546.

[68] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

[69] David Rau, Hervé Déjean, Nadezhda Chirkova, Thibault Formal, Shuai Wang, Stéphane Clinchant, and Vassilina Nikoulina. BERGEN: A benchmarking library for retrieval-augmented generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pages 7640–7663. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.449. URL https://doi.org/10.18653/v1/2024.findings-emnlp.449.

[70] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. Splade-v3: New baselines for SPLADE. *CoRR*, abs/2403.06789, 2024. doi: 10.48550/ARXIV.2403.06789. URL https://doi.org/10.48550/arXiv.2403.06789.

[71] Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-Style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=sE7-XhLxHA.

[72] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[73] Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. BioASQ-QA: A manually curated corpus for biomedical question answering. *Scientific data*, 10(1):170, 2023.

[74] Timo Möller, Anthony Reina, Raghavan Jayakumar, and Malte Pietsch. COVID-QA: A question answering dataset for COVID-19. In *Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020*, 2020.

[75] James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*, 2018.

[76] nostalgebraist. interpreting GPT: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens, Aug 2020. LessWrong blog post. Accessed: 2026-05-02.

A LLM-as-Judge Prompt

The following prompt is used to query the judge model for the LLM-as-judge accuracy metric reported in our experiments.

```
You are an evaluation tool. Just answer by {{ Yes }} or {{ No }}.

Given a question, a list of golden answers (any one of them is considered
correct), and an AI-generated answer, judge whether the AI-generated answer is
correct - it is correct if it matches ANY ONE of the golden answers.

Question: {question}
Golden answers: {ground_truth}

Generated answer: {model_answer}
Response: {{
```

B Pure Distillation Hyperparameters

Pure Distillation (§3.2) is performed on the teacher-correct subset $\mathcal{S}^+$, where we determine correctness based on whether the ground truth is contained within the model prediction. The core hyperparameters are listed in Table 4.

Table 4: Hyperparameters for Pure Distillation.

Hyperparameter	**Assignment**
optimizer	AdamW
learning rate	1×10^{-4}
warmup ratio	0.05
weight decay	0.1
gradient clipping	1.0
epochs	1
per-device batch size	2
gradient accumulation steps	1
effective batch size	16
max response length	128
distributed setup	DeepSpeed (8 GPUs)
random seed	42

C Hard Exploration Hyperparameters

Hard Exploration (§3.3) initializes from the PD checkpoint and runs GRPO on the teacher-failed subset $\mathcal{S}^-$. The core hyperparameters are listed in Table 5.

Table 5: Hyperparameters for Hard Exploration.

Hyperparameter	**Assignment**
optimizer	AdamW
learning rate	5×10^{-6}
warmup ratio	0.05
weight decay	0.01
gradient clipping	1.0
gradient checkpointing	True
epochs	2
training micro batch size	8
per-GPU policy batch size	64
gradient accumulation steps	1
GRPO group size G	16
clipping range ε	0.2
KL coefficient β	0.02
reward function	binary CEM
rollout temperature	1
rollout top-p	0.95
max generation length	128
random seed	42