Dual-Purpose Context Compression: Aligning Generation and Retrieval Spaces for RAG

Anonymous ACL submission

Abstract

Retrieval-augmented generation (RAG) reduces LLM hallucinations but requires heavy infrastructure, making it unsuitable for constrained deployments. Soft context compression pre-computes documents into compact, on-device neural tokens (MEM-tokens). We investigate whether these compressed representations can support both generation and passage retrieval on a single GPU without external servers. Among three evaluated methods, OSCAR excels at generation, though its raw embeddings yield near-random retrieval. A lightweight MLP projector trained via knowledge distillation maps OSCAR into a retrieval-compatible space for multi-hop QA, whereas contrastive training fails due to domain shift. Our results establish the feasibility of server-free, GPU-resident RAG using compressed representations.

1 Introduction

Large language models (LLMs) hallucinate on entities rare in their pre-training corpora (Mallen et al., 2023). Retrieval-augmented generation (RAG) (Lewis et al., 2021) mitigates this but requires persistent vector indexes, heavy encoders (300M–7B parameters), and network-accessible servers—making it impractical for mobile devices, edge nodes, or closed environments where external servers are forbidden. Fine-tuning suffers from catastrophic forgetting (McCloskey and Cohen, 1989; Luo et al., 2025) and cannot keep up with evolving knowledge.

Context compression offers a middle ground: a compression model distills each document into a small number of dense neural tokens ("MEM-tokens"), pre-computed offline and stored on a single GPU. Prior work on soft compression (Chevalier et al., 2023; Ge et al., 2024) has focused on generation quality; it remains unknown whether the same compressed representations can also serve as a retrieval index.

We investigate whether MEM-tokens from three recent methods (xRAG, PISCO, OSCAR) can support both high-quality generation and passage retrieval on a single GPU without external infrastructure. Raw OSCAR MEM-tokens (8 per passage) perform near-randomly on retrieval benchmarks, but a lightweight MLP projector trained via knowledge distillation from SFR-Embedding-Mistral enables effective retrieval on multi-hop QA (HotpotQA (Yang et al., 2018a)). In contrast, contrastive training on MS-MARCO (Bajaj et al., 2016) fails to transfer to entity-centric benchmarks (PopQA (Mallen et al., 2023), HotpotQA) due to domain shift (Thakur et al., 2021). Our results establish the feasibility of GPU-resident, server-free RAG using compressed document representations.

We address the following research questions:

RQ1 Can passage retrieval be supported by off-the-shelf soft compression, e.g. OSCAR embeddings, under any reasonable pooling strategy?

RQ2 Is it feasible for a projector, trained on a SOTA frozen soft compression embeddings, to bridge the retrieval gap between compression space and a SOTA retrieval embedding space?

RQ3 Which training strategy (contrastive learning or knowledge distillation) demonstrates superior generalization across heterogeneous retrieval benchmarks, including entity-centric question answering and open-domain passage ranking?

2 Related Work

Retrieval-augmented generation (RAG) (Lewis et al., 2021) grounds LLM outputs in external documents. Dense retrieval methods such as DPR (Karpukhin et al., 2020), ANCE (Xiong et al., 2020), E5 (Wang et al., 2024), and SFR-Embedding (Rui Meng, 2024) have pushed performance well beyond sparse methods like BM25 (Robertson and Zaragoza, 2009). SPLADE

(Formal et al., 2021) offers a hybrid sparse-dense approach, and BEIR (Thakur et al., 2021) provides a benchmark for zero-shot retrieval evaluation.

Context compression aims to reduce inference cost by representing long documents as compact vectors (soft tokens or MEM-tokens). ICAE (Ge et al., 2024) and AutoCompressors (Chevalier et al., 2023) train autoencoders within an LLM's latent space; LLMLingua (Jiang et al., 2023b) prunes low-importance tokens instead. xRAG (Cheng et al., 2024) compresses a document into a single retrieval embedding token projected into the LLM's token space – although we later show this is insufficient for factual reconstruction. PISCO (Louis et al., 2025) uses sequence-level knowledge distillation, and OSCAR (Louis et al., 2026), the focus of this work, adds online reranking to dynamically allocate MEM-tokens. Both demonstrate that 8-32 MEM-tokens preserve generation quality. However, no prior work has investigated whether the same compressed representations can also serve as a retrieval index, enabling dual-purpose generation and retrieval on a single GPU without external servers.

3 Methodology

Compression Method Selection. We applied three selection criteria to identify compression methods suitable for resource-constrained deployment: query-independent operation (offline indexing), vector output (compatible with cosine similarity), and empirical functionality (basic QA capability via paraphrase reconstruction). This yielded three candidate methods—xRAG, PISCO, and OSCAR—each producing query-independent vector representations. The paraphrase reconstruction diagnostic revealed that xRAG's single-token bottleneck achieves near-zero exact match, confirming catastrophic information loss. OSCAR (8 MEM-tokens) and PISCO both pass the reconstruction test, with OSCAR showing stronger entity-level fidelity and therefore selected as the primary focus for retrieval alignment. See the Appendix A for details.

Architecture Overview. The end-to-end system proposed in this work operates as follows. In an offline pre-computation phase, each document in the target corpus is processed by the OSCAR compressor (naver/oscar-qwen2-7B), yielding 8 MEM-token vectors per document. These vectors are stored in a GPU-resident vector store. A lightweight MLP projector, trained via knowledge distillation from SFR-Embedding-Mistral, transforms the flattened 8×3,584-dimensional OSCAR representation into a 4,096-dimensional retrieval-compatible embedding. At inference time, the user query is encoded by SFR-Embedding-Mistral (a fast forward pass over a short query), and cosine similarity is computed against all stored document projections. The top-k retrieved document embeddings are then passed, together with the query, to the OSCAR-conditioned language model generator, by which the final answer is produced. The key architectural novelty lies in the asymmetric encoding: queries are encoded directly by SFR (lightweight, fast, suitable for real-time operation), whereas documents are encoded by the OSCAR+MLP pipeline offline (computationally heavier, but pre-computed over many queries). The MLP projector is trained to align the document embedding space of OSCAR with that of SFR, thereby ensuring that cosine similarities between the two are meaningful. Proposed architecture could be represented as on Figure **??**.

Problem Statement. The target system could be formalized as follows. Let $D = \{d_1, d_2, ..., d_N\}$ be a corpus of N documents, where each document d_i is a passage of text drawn from a knowledge source such as Wikipedia. Let q be a user query issued at inference time. A compression model C maps each document d_i to a set of k MEM-token embeddings:

$$C(d_i) \rightarrow e_i \in \mathbb{R}^{k \times h}, \tag{1}$$

where $k = 8$ is the fixed number of MEM-tokens and h is the hidden dimensionality of the compressor model ($h = 3,584$ for the Qwen2-7B backbone used in OSCAR). The compression is designed to be performed offline, without knowledge of q — a property that could be termed query-independence.

The system must fulfil two simultaneous objectives:

1. **Retrieval objective**: Given query q, rank documents in D by relevance to q using the precomputed representations $\{e_i\}$. Formally, define a query encoder $Q : q \rightarrow \mathbb{R}^d$ and a document encoder $E : \mathbb{R}^{k \times h} \rightarrow \mathbb{R}^d$ such that $rank(d_i|q)$ is determined by $\cos(Q(q), E(e_i))$.

2. **Generation objective**: Given the top-ranked document $d^\star$ with representation $e^\star$, generate

the answer: $a = LLM(e^\star, q)$. Here $e^\star$ is consumed by the LLM as a soft prefix, replacing the raw document text in the context window.

The alignment problem can be solved precise by examining the geometry of the two spaces. In an ideal retrieval embedding space, documents relevant to a given query are clustered near the query vector in cosine distance, while irrelevant documents remain distant. This geometry is enforced through contrastive training on query–document relevance pairs (e.g., MS-MARCO). An ideal generation conditioning space, by contrast, is characterized by the ability of the LLM to recover the information content of the original document from the compressed representation under cross-attention; equivalently, the information content can be recovered by the LLM from that compressed representation. Such geometry is imposed by a reconstruction or distillation loss computed against the LLM's internal representations when processing the full document. A priori, there is no reason to expect these two geometries to be compatible. Concrete evidence is provided by the retrieval results presented later: raw OSCAR MEM-token embeddings, under all pooling strategies, exhibit retrieval performance at approximately the level of random chance—roughly three orders of magnitude below that of SFR-Embedding-Mistral. It is thereby confirmed that the generation-oriented training of OSCAR produces an embedding space that is essentially uninformative for retrieval without additional alignment.

OSCAR. OSCAR (Online Soft Compression And Reranking) (Louis et al., 2026) is a neural context compression model built on top of a pre-trained large language model. In this work, naver/oscar-qwen2-7B was used as the base model, which employs Qwen2-7B (Yang et al., 2024) (7B parameters, hidden dimension $h = 3,584$) as the compressor backbone. The OSCAR compressor processes an input document d_i and produces $k = 8$ special MEM-token representations. These are the final hidden-state vectors at the positions of 8 designated MEM-token positions in the compressor's output sequence:

$$MEM(d_i) = [m_1, ..., m_8], m_j \in \mathbb{R}^{3584} \quad (2)$$

The original OSCAR system is designed to be query-dependent: both the query and the document are taken as input by the compressor model, and MEM-tokens are generated conditioned on both. However, query-dependent compression is incompatible with offline indexing: if the compressed representation depends on the query, a new set of MEM-tokens must be computed for each (query, document) pair at inference time, eliminating the storage and latency advantages of pre-computation. In this work, it is investigated whether OSCAR's MEM-token embeddings, generated in a query-independent mode (i.e., when the document is passed to the compressor without any query context), carry sufficient retrieval signal to support passage-level retrieval. The central empirical question addressed by this research is whether the information encoded in query-independent OSCAR MEM-tokens can be aligned with a retrieval objective through a lightweight learned projector.

The compressor is fine-tuned with a combined loss that encourages the MEM-tokens to retain information sufficient for both document reconstruction and query-relevance ranking. In the original paper, the compressor is conditioned on both the query and the document; in the query-independent setup adopted here, only the document is passed, using a fixed prompt template that omits the query component. The 8 MEM-tokens are then fed to the downstream LLM generator (also Qwen2-7B, but with separate OSCAR-specific fine-tuning) as a soft prefix, thereby replacing the raw document text in its context window. The generator is kept frozen during projector training; only the projector weights are updated.

Pooling Strategies for Baseline OSCAR Retrieval. Before training any projector, the retrieval quality of raw OSCAR MEM-token embeddings is evaluated. Since each document is represented by 8 MEM-token vectors (each in $\mathbb{R}^{3584}$), we must aggregate these into a single vector for cosine-similarity retrieval. Four pooling strategies are considered. First, mean pooling calculates the average across the tokens:

$$\bar{e} = \frac{1}{8}\sum_{i=1}^{8} m_i \in \mathbb{R}^{3854} \quad (3)$$

Second, the last token strategy uses only the final MEM-token vector $m_8 \in \mathbb{R}^{3584}$. This approach is motivated by the causal attention structure of decoder-based models, where the last token attends over all previous tokens. Third, the first token strategy relies exclusively on the first MEM-token

vector $m_1 \in \mathbb{R}^{3584}$. Finally, max pooling extracts the maximum value:

$$\hat{e} = \max_j m_j \in \mathbb{R}^{3854}, \tag{4}$$

where the maximum is taken element-wise across all 8 MEM-token vectors.

Appendix **??** provides additional analysis of pooling strategies, including mean-max combination and effective rank of the embedding covariance matrix. For raw retrieval, first-token pooling marginally outperforms others, but all strategies yield near-random performance.

MLP Projector Architecture. The principal methodological contribution of this work is an MLP projector that transforms OSCAR's 8 MEM-token representation into an embedding space compatible with retrieval. The projector is implemented as a feed-forward network, trained upon the frozen MEM-token representations obtained from OSCAR. For the input representation, the 8 MEM-tokens from OSCAR are flattened into a single vector:

$$x = \text{flatten}([m_1, \ldots, m_8]) \in \mathbb{R}^{8\times 3584} = \mathbb{R}^{28672} \tag{5}$$

Following this input formulation, three variants of projector architectures were considered, summarized in Table 7. The first is a 1-layer linear projection ($\mathbb{R}^{28672} \rightarrow \mathbb{R}^d$) with no nonlinearity, which serves as the ablation baseline for projector expressiveness. The second variant is a 2-layer MLP structured as Linear(28672 → 8192) → LayerNorm → GELU → Linear(8192 → d). This is the primary architecture used in the distillation experiments. The third variant is a deeper 3-layer MLP structured as Linear(28672 → 8192) → LN → GELU → Linear(8192 → 4096) → LN → GELU → Linear(4096), which is used in the ablation experiments.

The output dimension d is determined by the training strategy: $d = 768$ for contrastive training aligned with `msmarco-distilbert-base-v3`, and $d = 4,096$ for distillation training aligned with SFR-Embedding-Mistral. Furthermore, regarding LayerNorm placement, LayerNorm is applied between linear layers to stabilise training with the high-dimensional input (28,672 dimensions). Without LayerNorm, we observe training instability and poor convergence in preliminary experiments, likely due to the large dynamic range of OSCAR hidden states across different document types.

Projector Training Strategies. Two approaches to projector training were evaluated in this study.

Contrastive Learning on MS-MARCO. This projector training strategy uses contrastive learning on the MS-MARCO passage ranking dataset. The objective is to train the projector to produce document embeddings that rank relevant passages above irrelevant passages for a given query. In this configuration, the teacher and query encoder is *MS MARCO DistilBERT-base v3* (providing 768-dimensional embeddings), which is used to encode queries at inference time. The dataset comprises MS-MARCO passage triplets (query, positive passage, in-batch negatives) and contains approximately 500,000 training triples. The model is optimized using the MultipleNegativesRankingLoss (equivalent to InfoNCE). For a batch of B query-passage pairs $\{(q_i, p_i^+)\}_{i=1}^B$, the loss is formulated as:

$$\mathcal{L}_{contrastive} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\cos(Q(q_i), P(x_i^+))/\tau)}{\sum_{j=1}^{B} \exp(\cos(Q(q_i), P(x_j))/\tau)}, \tag{6}$$

where $P(\cdot)$ denotes the projector, $Q(\cdot)$ denotes the query encoder, and τ is a temperature parameter.

This strategy is conceptually natural: it trains the projector to align OSCAR's document representations with the embedding space in which they should be most similar to relevant queries. However, as reported further, this strategy suffers from domain shift: it generalises to indistribution tasks (MS-MARCO, Natural Questions) but fails on entity-centric QA (PopQA, HotpotQA) where entity identity, rather than topical relevance, determines document relevance.

Knowledge Distillation from SFR. This projector training strategy is motivated by the domain shift failure of contrastive training. Instead of training on query-passage relevance pairs, we train the projector to reproduce the document embeddings produced by SFR-Embedding-Mistral — a model that performs strongly across a wide range of benchmarks including multi-hop QA and entity-centric retrieval.

The full training pipeline follows a four-step process. First, during offline teacher computation, all passages in the training corpus (100k MS-MARCO Document corpus passages) are encoded through SFR-Embedding-Mistral using last-token pooling and L2 normalisation. The resulting 4,096-dimensional embeddings are stored in HDF5 for-

mat for efficient random-access loading during training. Second, for the OSCAR encoding step, the same passages are encoded through OSCAR (`naver/oscar-qwen2-7B`) in query-independent mode, producing 8 MEM-token representations per passage. Third, the projector is trained to minimise the MSE between its output (L2-normalised) and the SFR teacher embedding (L2-normalised):

$$\begin{aligned}\mathcal{L}_{\text{distillation}} &= \|\text{norm}(P(x)) - \text{norm}(\text{SFR}(d))\|_2^2 \\ &= 2 - 2\cos(P(x), \text{SFR}(d)),\end{aligned} \tag{7}$$

where $x = \text{flatten}(\text{MEM}(d))$. Finally, at inference time, queries are encoded by SFR-Embedding-Mistral directly (without OSCAR processing). Documents are encoded by the OSCAR+projector pipeline offline. Ultimately, both representations live in the same $\mathbb{R}^{4096}$ cosine-similarity space.

The distillation strategy produces an asymmetric encoding scheme at inference: queries are encoded by SFR-Embedding-Mistral (a forward pass of a 7B model over a short query, typically 10–30 tokens), while documents are encoded by the OSCAR+MLP projector (an offline pre-computed pipeline). This asymmetry is unusual but principled.

The validity rests on the distillation training objective: the projector is explicitly trained to reproduce SFR's document embedding. If training succeeds, then for any document d:

$$P(\text{flatten}(\text{MEM}(d))) \approx \text{SFR}(d) \tag{8}$$

This means that the document embeddings produced by the projector and the query embeddings produced by SFR both live in an approximately shared $\mathbb{R}^{4096}$ space, where cosine similarity reflects query-document relevance in the same way it does for SFR-encoded query-document pairs. The asymmetry is therefore a consequence of the training objective, not an inconsistency.

Baseline Retriever Models. The following retrieval baselines were used throughout the experimental evaluation:

SFR-Embedding-Mistral (Rui Meng, 2024): The embedding model is a 7-billion-parameter decoder-based architecture, adapted from Mistral-7B (Jiang et al., 2023a) through contrastive fine-tuning on retrieval datasets. Embeddings are produced via last-token pooling over the final hidden states, followed by L2 normalisation. This model, denoted SFR, achieves state-of-the-art performance on the MTEB leaderboard (Muennighoff et al., 2023) and is employed as the upper-bound retrieval baseline in the experiments.

BM25 (Robertson and Zaragoza, 2009): Best Match 25 (BM25) is defined as the classical probabilistic sparse retrieval model, grounded in term frequency–inverse document frequency (TF–IDF) weighting. BM25 is still employed as a strong baseline, particularly in entity-centric question answering where queries contain unique entity names that are present verbatim in the relevant documents. It is utilized as the sparse retrieval lower bound.

Model Architecture and Training. The resulting best projector model (**OSCAR E11**)was obtained with the following settings and parameters based on previously conducted experiments and their results.

Base model. We use OSCAR (naver/oscar-qwen2-7B), a 7B-parameter document compression model based on Qwen2-7B, which encodes each document into a sequence of 8 memory tokens. The hidden state dimension of OSCAR is 28,672 (= 4,096 × 7, or equivalently the concatenation of the 8 memory token hidden states of dimension 3,584 each).

Projector. To map OSCAR's compressed memory representations into a dense retrieval embedding space, we introduce a lightweight per-token gated projector (MEMProjectorGated). Given the concatenated hidden states of the 8 memory tokens $\mathbf{m} \in \mathbb{R}^{B\times 28672}$, the projector first reshapes the input into per-token representations $\mathbf{x} \in \mathbb{R}^{B\times 8\times 3584}$ and applies a shared linear token projection:

$$\mathbf{h}_i = \text{ReLU}(\mathbf{W}_{\text{proj}}\,\mathbf{x}_i + \mathbf{b}), \quad \mathbf{h}_i \in \mathbb{R}^{1024} \tag{9}$$

A scalar gate is computed per token via a linear layer without bias:

$$g_i = \frac{\exp(\mathbf{w}_g^\top \mathbf{h}_i)}{\sum_{j=1}^{8} \exp(\mathbf{w}_g^\top \mathbf{h}_j)} \tag{10}$$

The gated pool aggregates token representations as a weighted sum:

$$\mathbf{p} = \sum_{i=1}^{8} g_i \cdot \mathbf{h}_i \tag{11}$$

A two-layer MLP head then projects $\mathbf{p}$ to the target embedding space:

$$\mathbf{e} = \text{LayerNorm}(\mathbf{W}_2\,\text{Dropout}(\text{ReLU}(\mathbf{p}))) \in \mathbb{R}^{768} \tag{12}$$

The final embedding is L2-normalized. All projector weights are initialized with Xavier uniform; the projector is trained in bfloat16. The projector contains approximately 47M parameters.

Training Objective. Training proceeds in two stages, separated at step $t_1 = 1{,}500$.

Stage 1 (steps 0–1,499): MSE anchoring. The model is trained to align query and document embeddings with pre-computed teacher embeddings from BGE-base-en-v1.5:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}^q_{\text{MSE}} + \mathcal{L}^d_{\text{MSE}} \tag{13}$$

where $\mathcal{L}^q_{\text{MSE}} = \|\mathbf{e}_q - \hat{\mathbf{e}}_q\|^2$ and $\mathcal{L}^d_{\text{MSE}}$ is the analogous loss for the positive document embedding. This stage ensures geometrically stable initialization of the embedding space before contrastive learning begins.

Stage 2 (steps 1,500+): Contrastive distillation. The full loss combines four terms:

$$\mathcal{L}_{\text{stage2}} = \lambda_{\text{KL}}\,\mathcal{L}_{\text{KL}} + \lambda_{\text{NCE}}\,\mathcal{L}_{\text{InfoNCE}} + \lambda_{\text{rank}}\,\mathcal{L}_{\text{margin}} + \lambda_{\text{MSE}}\,\mathcal{L}^q_{\text{MSE}} \tag{14}$$

with weights $\lambda_{\text{KL}} = 0.3$, $\lambda_{\text{NCE}} = 0.05$, $\lambda_{\text{rank}} = 0.02$, $\lambda_{\text{MSE}} = 1.0$.

Listwise KL distillation ($\mathcal{L}_{\text{KL}}$). For each query q with K candidates, teacher and student score distributions are computed at temperature $\tau = 0.07$:

$$\mathcal{L}_{\text{KL}} = \sum_{k=1}^{K} p_k^{\text{teacher}} \log \frac{p_k^{\text{teacher}}}{p_k^{\text{student}}} \tag{15}$$

Global in-batch InfoNCE ($\mathcal{L}_{\text{InfoNCE}}$). All $B \times K$ candidate documents in a batch are used as negatives. The positive for query i is the candidate with the highest teacher score:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(s_{i,k^*}/\tau)}{\sum_{j=1}^{BK}\exp(s_{i,j}/\tau)} \tag{16}$$

where $k^* = \arg\max_k \hat{s}^{\text{teacher}}_{i,k}$.

Hard negative margin loss ($\mathcal{L}_{\text{margin}}$). Given the hardest in-batch negative s_i^-:

$$\mathcal{L}_{\text{margin}} = \frac{1}{B}\sum_{i=1}^{B} \max\big(0,\ m - s_i^+ + s_i^-\big)\,, \quad m = 0.15 \tag{17}$$

Training Data. The projector is trained on six IR datasets with teacher embeddings pre-computed offline using BGE-base-en-v1.5. Each dataset provides query–document pairs with BM25-mined hard negatives ($K = 50$ candidates per query). Dataset sampling follows fixed weights during training: MS MARCO (0.85), SciFact (0.03), TREC-COVID (0.03), FiQA (0.03), NFCorpus (0.02), ArguAna (0.02), Quora (0.02). The heavy weighting toward MS MARCO ensures broad generalization, while the domain-specific datasets provide signal on specialized retrieval tasks.

Training Procedure. The projector is trained with AdamW (lr $= 2 \times 10^{-5}$, min_lr $= 10^{-6}$, weight decay $= 0.01$) using a cosine learning rate schedule with 300 warmup steps, for 5 epochs with batch size 64. The OSCAR backbone is frozen throughout; only the projector weights are updated. Validation is performed every 500 steps on a held-out 2

4 Numerical Experiments

4.1 Experimental Setup

All experiments are conducted in Python 3.10 using PyTorch (Paszke et al., 2019) and the HuggingFace Transformers library (Wolf et al., 2020). The OSCAR model (`naver/oscar-qwen2-7B`) is loaded in bfloat16 precision for memory efficiency. The teacher model for distillation (`Salesforce/SFR-Embedding-Mistral`) is loaded in float16 precision. Retrieval evaluation uses the BEIR (Thakur et al., 2021) framework for standardised benchmark evaluation and metric computation. All experiments are run on 2×NVIDIA H100 80GB GPUs.

The key pipeline components are configured as follows. For the OSCAR compressor, `naver/oscar-qwen2-7B` is employed in query-independent mode, meaning the document is passed without a query prefix. The teacher model is Salesforce/SFR-Embedding-Mistral, which uses a last-token pooling, L2 normalised model. As the sparse baseline, BM25 is utilised with the parameters $k_1 = 0.9$ and $b = 0.4$. The evaluation framework relies on BEIR for retrieval metrics and custom evaluation scripts for QA generation metrics. Finally, the selected QA metric is Exact Match (EM) evaluated with standard normalisation, which involves lowercasing and stripping punctuation.

4.2 Datasets and Benchmarks

HotpotQA Distractor Setting (Yang et al., 2018b). HuggingFace hub (`hotpot_qa, distractor split`). The dataset contains 7,405 multi-hop questions. For each question, 10 candidate paragraphs are provided: 2 gold supporting paragraphs and 8 distractors. The retrieval task is to rank the 2 gold paragraphs in the top-2 positions.

HotpotQA Fullwiki Setting (Yang et al., 2018b). BEIR benchmark (Thakur et al., 2021) (BEIR/hotpotqa). Global retrieval over approximately 5.2 million Wikipedia paragraphs.

Natural Questions (NQ). BEIR benchmark (Thakur et al., 2021) (BEIR/nq). The dataset contains 3,452 test queries, with a corpus of approximately 2.7 million Wikipedia passages.

PopQA (Mallen et al., 2023). HuggingFace hub (akariasai/PopQA), 14,267 entity-centric questions. Entity popularity (Wikidata pageviews) is available for each entity, enabling popularity-stratified analysis.

MS-MARCO Passage Ranking (Bajaj et al., 2018). The MS-MARCO passage ranking benchmark contains approximately 8.8 million passages and a development set of 6,980 queries with positive relevance labels. MS-MARCO represents general-purpose web-query retrieval as opposed to the entity-centric QA tasks.

4.3 Retrieval Results for raw OSCAR Embeddings

The four OSCAR pooling strategies (mean, last, first, max) were evaluated on the HotpotQA distractor corpus (14,232 candidate passages for 7,405 queries). Table 1 presents the results alongside the SFR-Embedding-Mistral upper bound

The results in Table 1 are unambiguous: all four OSCAR pooling strategies yield near random retrieval performance. The best-performing strategy (first token) achieves an MRR of 0.0011 — approximately 693× lower than SFR-Embedding-Mistral's MRR of 0.7621. The Recall@10 for the best OSCAR strategy (0.0023) is 409× lower than SFR's (0.9420). The same results can be seen in Table 2. These results confirm that OSCAR MEM-token embeddings, as produced in query-independent mode, carry essentially no retrieval signal in their raw form.

4.4 Retrieval Results for MLP Projector

Table 7 presents results on the MS-MARCO benchmark. The 1-layer linear projector trained with contrastive loss achieves MRR@10 = 0.8680, which approaches the teacher model (DistilBERT, MRR@10 = 0.9993) and substantially outperforms BM25 (0.9424) and raw OSCAR (0.1384). However, as shown in Table 8, this performance does not transfer to out-of-domain BEIR datasets. The contrastive projector collapses to near-random performance on SciFact (NDCG@10 = 0.0096), NFCorpus (0.0099), and FiQA-2018 (0.0009). This confirms that contrastive training on MS-MARCO overfits to topical relevance and fails on entity-centric tasks.

Tables 9 and 10 evaluate distillation using E5-base and BGE-base teacher models (output dimension 768). These achieve low BEIR-3 averages (MRR 0.017), indicating that smaller teacher models transfer less effectively. Our primary distillation setup uses SFR-Embedding-Mistral (7B parameters, output dim 4096). Table 4 (OSCAR E11) shows results on BEIR-6: the projector achieves macro-averaged NDCG@10 = 0.223, MRR@10 = 0.272, Recall@10 = 0.232. While still far below BM25 (0.486 MRR) and SFR (0.629 MRR), this represents a 10× improvement over raw OSCAR (0.019 MRR) and demonstrates that retrieval is feasible from only 8 compressed tokens per document. This answers RQ2 affirmatively: a lightweight projector can bridge the retrieval gap.

Table 8 and Table 3 together answer RQ3: knowledge distillation from a strong, domain-agnostic teacher (SFR) demonstrates superior generalization across heterogeneous benchmarks compared to contrastive learning on MS-MARCO. The contrastive strategy achieves high in-domain performance (MS-MARCO MRR@10 = 0.868) but fails on entity-centric tasks (PopQA, HotpotQA). Distillation sacrifices some MS-MARCO performance (0.730 MRR@10) but retains cross-domain transfer (BEIR-6 macro avg 0.272 MRR vs. contrastive's 0.084 MRR). We hypothesize that distillation preserves the teacher's semantic geometry, whereas contrastive learning tailors the space to lexical/topical relevance patterns that do not generalize.

5 Discussion

The experimental results highlight the fundamental geometric difference between embedding spaces

Model	MRR	NDCG@1	NDCG@10	Recall@10
SFR-Embedding-Mistral	0.7621	0.6573	0.8053	0.9420
BM25	0.6330	0.5280	—	0.8570
OSCAR (mean pooling)	0.0003	0.0001	0.0004	0.0009
OSCAR (last token)	0.0007	0.0002	0.0010	0.0022
OSCAR (first token)	0.0011	0.0006	0.0013	0.0023
OSCAR (max pooling)	0.0004	0.0001	0.0004	0.0008

Table 1: OSCAR raw pooling strategies vs. SFR baseline on HotpotQA distractor corpus (n = 14,232 passages)

Dataset	BM25 MRR	BM25 NDCG@10	BM25 R@10	OSCAR MRR	OSCAR NDCG@10	OSCAR R@10
SciFact	0.637	0.665	0.785	0.001	0.000	0.000
NFCorpus	0.524	0.311	0.153	0.051	0.017	0.002
FiQA-2018	0.296	0.232	0.298	0.004	0.002	0.001
Average	0.486	0.402	0.412	0.019	0.006	0.001

Table 2: OSCAR raw embeddings (flattened) retrieval results on BeIR benchmark.

optimized for text generation and those optimized for vector retrieval. OSCAR MEM-tokens, trained to reconstruct document content, lack the properties required for cosine-similarity retrieval. The MLP projector successfully bridges this gap, but the choice of training objective dictates its ability to generalize. Contrastive learning on MS-MARCO optimizes for topical relevance, resulting in poor transfer to entity-centric QA tasks (PopQA, HotpotQA) where specific entity identity is required. Knowledge distillation from a robust teacher (SFR-Embedding-Mistral) overcomes this domain shift by transferring a broader, domain-agnostic semantic representation. While the MLP projector's retrieval metrics on multi-hop datasets do not match the heavy teacher model, it successfully extracts a functional retrieval signal from 8 compressed tokens, validating the dual-purpose use of MEM-tokens.

6 Conclusion

This work establishes the feasibility of GPU-resident, server-free Retrieval-Augmented Generation (RAG) using soft context compression. While raw MEM-tokens from models like OSCAR retain factual information for high-quality text generation, they are uninformative for direct passage retrieval. By introducing a lightweight MLP projector trained via knowledge distillation, we aligned these generative tokens into a retrieval-compatible space. This approach allows a single set of compressed tokens to serve both as a retrieval index and a generation prompt. Consequently, the memory footprint and inference infrastructure requirements of RAG are reduced, enabling deployment on a single consumer GPU without external retrieval servers or heavy dense encoders.

7 Limitations

OSCAR's compression architecture processes documents up to 128 tokens. For longer documents, the current system must truncate or chunk, with potential information loss for passages where key factual content appears after the 128-token boundary. Operating OSCAR in query-independent mode means the compressor cannot selectively retain query-relevant information. In the original, query-dependent setting, OSCAR can prioritise information relevant to the query when deciding what to compress into the 8 MEM-tokens. The query-independent mode trades off some compression efficiency for the practical advantage of offline indexing. The distillation training strategy requires pre-computing SFR teacher embeddings for all training documents offline. For a corpus of 100k passages, this is manageable; for a full Wikipedia-scale corpus (5M+ passages), the computational cost is substantial and must be amortised over the system's operational lifetime.

References

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. Ms marco: A human generated machine reading comprehension dataset. *Preprint*, arXiv:1611.09268.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, An-

drew McNamara, Bhaskar Mitra, Tri Nguyen, and 1 others. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Xin Cheng, Xun Wang, Xingxing Zhang, Tao Ge, Si-Qing Chen, Furu Wei, Huishuai Zhang, and Dongyan Zhao. 2024. xrag: Extreme context compression for retrieval-augmented generation with one token. *Preprint*, arXiv:2405.13792.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. *Preprint*, arXiv:2305.14788.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade: Sparse lexical and expansion model for first stage ranking. *Preprint*, arXiv:2107.05720.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. 2024. In-context autoencoder for context compression in a large language model. *Preprint*, arXiv:2307.06945.

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. 2023a. Mistral 7b. *Preprint*, arXiv:2310.06825.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023b. Llmlingua: Compressing prompts for accelerated inference of large language models. *Preprint*, arXiv:2310.05736.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Preprint*, arXiv:2004.04906.

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. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Maxime Louis, Hervé Déjean, and Stéphane Clinchant. 2025. Pisco: Pretty simple compression for retrieval-augmented generation. *Preprint*, arXiv:2501.16075.

Maxime Louis, Thibault Formal, Hervé Dejean, and Stéphane Clinchant. 2026. Oscar: Online soft compression and reranking. *Preprint*, arXiv:2504.07109.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. *Preprint*, arXiv:2212.10511.

Michael McCloskey and Neal J Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *Preprint*, arXiv:2210.07316.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, and 2 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

S. Robertson and H. Zaragoza. 2009. *The Probabilistic Relevance Framework: BM25 and Beyond*. Foundations and trends in information retrieval. Now Publishers.

Shafiq Rayhan Joty Caiming Xiong Yingbo Zhou Semih Yavuz Rui Meng, Ye Liu. 2024. Sfr-embedding-mistral:enhance text retrieval with transfer learning. Salesforce AI Research Blog.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *Preprint*, arXiv:2104.08663.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2024. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *Preprint*, arXiv:2007.00808.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan

Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018a. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *Preprint*, arXiv:1809.09600.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *Preprint*, arXiv:2305.16739.

8 Use of AI assistants.

We used GPT-5.5 and Grammarly to improve and proofread the text of this paper, correcting grammatical, spelling, and stylistic errors, as well as rephrasing sentences and shortening passages. Consequently, some portions of text may be identified as AI-generated, AI-edited, or combination of human and AI contributions.

A Compression Method Selection Details

The space of context compression methods is large and heterogeneous. To focus the experimental effort, three selection criteria are applied, designed to identify methods compatible with the target resource-constrained deployment:

1. **Query-independent operation**: the compressor must be able to process documents offline, without knowledge of the query at index time. This property is essential for any system where documents are pre-indexed and queries arrive dynamically at inference time.

2. **Vector output**: the compressed representation must take the form of a fixed-size vector (or a small set of vectors) in a continuous embedding space. Representations that encode knowledge as weight updates or structured text are not directly compatible with cosine-similarity retrieval.

3. **Empirical functionality**: the method must demonstrate basic capability on QA tasks. A paraphrase reconstruction test is adopted as a proxy: a method that cannot reproduce the source document's factual content from its compressed representation cannot, in principle, answer questions about that document.

When these criteria are applied to the literature, five candidate methods emerge:

- **xRAG** (Cheng et al., 2024): Uses a dense retrieval embedding as the sole document representation, projected via a lightweight MLP into the LLM's token space. Query independent and produces a vector output

- **PISCO** (Louis et al., 2025): Pretty Simple Context Compression. Uses sequence-level knowledge distillation to compress documents into a fixed number of soft tokens. Query-independent and produces vector output

- **OSCAR** (Louis et al., 2026): Online Soft Compression And Reranking. Compresses documents into 8 MEM-tokens using an LLM-based compressor. Can be operated in query-independent mode

The paraphrase reconstruction diagnostic serves as a critical quality gate for compression methods. The test protocol is as follows: the compressed representation of the corresponding Wikipedia page for each PopQA benchmark entity is provided as context, and the language model is instructed to reproduce the original passage verbatim. The generated output is evaluated against the source using Align-Score (Zha et al., 2023), a hallucination-detection metric grounded in natural language inference, together with standard exact-match scoring.

The qualitative observation that motivates the choice of method becomes apparent even before final quantitative metrics are obtained. When prompted to reproduce the source passage (Figure 1, Table 11), the xRAG model generates text that seems plausible but is factually incorrect. Entity names, dates, and specific quantities are routinely hallucinated or omitted. This confirms a fundamental compression failure: a single retrieval token does not carry sufficient mutual information with a 128-token factual passage to enable entity-level reconstruction.

Architecturally, xRAG constitutes the most elegant candidate for the dual-purpose requirement:

Dataset	BM25	BGE	OSCAR E11
SciFact	0.665	0.738	0.124
NFCorpus	0.310	0.369	0.075
FiQA-2018	0.232	0.391	0.023
Arguana	0.315	0.453	0.137
Trec-COVID	0.651	0.730	0.240
Quora	0.745	0.889	0.741
Macro Avg(6)	0.486	0.595	0.223

Table 3: NDCG@10 values for OSCAR E11 on BeIR-6.

Dataset	BM25	BGE	OSCAR E11
SciFact	0.633	0.700	0.104
NFCorpus	0.520	0.565	0.166
FiQA-2018	0.287	0.474	0.032
Arguana	0.204	0.317	0.089
Trec-COVID	0.794	0.819	0.494
Quora	0.740	0.882	0.737
Macro Avg(6)	0.530	0.626	0.272

Table 4: MRR@10 values for OSCAR E11 on BeIR-6.

Dataset	BM25	BGE	OSCAR E11
SciFact	0.785	0.866	0.200
NFCorpus	0.153	0.176	0.027
FiQA-2018	0.298	0.457	0.036
Arguana	0.670	0.870	0.289
Trec-COVID	0.016	0.020	0.005
Quora	0.846	0.957	0.836
Macro Avg(6)	0.461	0.558	0.232

Table 5: Recall@10 values for OSCAR E11 on BeIR-6.

Dataset	Corpus docs	Queries	#rows in test
SciFact	5183	1109	340
NFCorpus	3633	3237	12.335
FiQA-2018	57638	6648	1.707
Arguana	8674	1406	1.407
Trec-COVID	171332	50	493.5
Quora	522931	15000	15.676

Table 6: BeIR-6 benchmark structure.

Method	MRR	MRR@10	NDCG@10	R@10
SFR	0.9993	0.9993	0.9995	1.000
BM25	0.9430	0.9424	0.9521	0.9810
Projector Contrastive 1L	0.8689	0.8680	0.8971	0.9854
Projector Contrastive 2L	0.8294	0.8285	0.8674	0.9849
Projector Contrastive 3L	0.8423	0.8415	0.8778	0.9873
Projector Distillation 1L	0.7305	0.7278	0.7842	0.9590
OSCAR raw	0.1633	0.1384	0.1842	0.3374

Table 7: MLP projector retrieval results on MS-MARCO benchmark.

Method	MRR avg	NDCG@10 avg	R@10 avg
SFR	0.629	0.541	0.539
BM25	0.486	0.402	0.412
OSCAR + Projector	0.084	0.054	0.064
OSCAR flatten	0.019	0.006	0.001

Table 8: Contrastive MLP projector retrieval results on BEIR benchmark averaged by datasets. Contrastive training on MS-MARCO does not transfer — projector collapses out-of-domain

Model	SciFact NDCG@10	SciFact MRR@10	NFCorpus NDCG@10	NFCorpus MRR@10	FiQA-2018 NDCG@10	FiQA-2018 MRR@10
OSCAR+ Distillation (E5-base)	0.0096	0.0093	0.0099	0.028	0.0009	0.0016
OSCAR+ Distillaion (BGE-base)	0.0174	0.0161	0.0084	0.0223	0.0007	0.0015

Table 9: Projector distillation retrieval results on BEIR benchmark for E5-base and BGE-base teacher models.

the dense retrieval embedding (SFR-Embedding-Mistral), already optimised for cosine-similarity retrieval, is reused and projected into the LLM's residual stream via a single MLP layer. Where xRAG to function successfully, the retrieval problem would be trivially solved. However, it is demonstrated by the paraphrase reconstruction experiments that xRAG fails completely at the basic task of reproducing the source document from its compressed representation. When a model operating under xRAG's single-token bottleneck is instructed to reconstruct the source passage, factual entities such as names, dates, or specific quantities cannot be recovered. This failure is not a matter of fine-tuning; rather, it reflects a fundamental information-theoretic constraint: a single 4,096-dimensional vector cannot encode sufficient mutual information with a 128-token document for reliable entity-level reconstruction. The exact match score on the paraphrase reconstruction task for xRAG is near zero across all test passages, confirming that the compression is destructive with respect to factual content.

In contrast, OSCAR passes the reconstruction test at 16× compression — 8 MEM-tokens produced by the OSCAR compressor contain enough information for the OSCAR-fine-tuned generator to reproduce the source passage with high fidelity.

B Qualitative Context Reconstruction Examples

Figures 2 and 3 show representative examples of context reconstruction from the paraphrase reconstruction experiment. In each example, the original Wikipedia passage (left) is compressed by the respective model, and the decompressed output (right) is compared to the source.

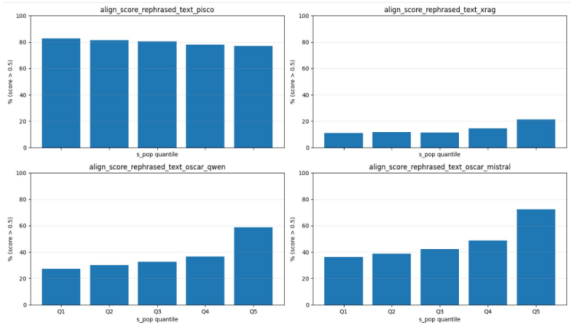

Figure 1: AlignScore > 0.5 rate (%) for paraphrase reconstruction, stratified by PopQA entity popularity quintile (Q1=rarest, Q5=most popular). PISCO achieves ≈ 80 % across all bins. OSCAR-Qwen and OSCAR-Mistral show a positive popularity gradient (27–59 % and 37–73 % respectively): popular entities are easier to reconstruct because their descriptions follow predictable patterns. xRAG scores ≈ 10 % uniformly — effectively random — confirming catastrophic compression failure.

Model	MRR	MRR@10	NDCG@10
OSCAR+ Distillation (E5-base)	0.0172	0.0131	0.0172
OSCAR+ Distillaion (BGE-base)	0.0171	0.0133	0.0088
E5-base (teacher)	0.5334	0.5258	0.4473
BGE-base (teacher)	0.5858	0.5799	0.4992

Table 10: Projector distillation retrieval results averaged by BEIR3 benchmark subsets for E5-base and BGE-base teacher models.

Method	AlignScore > 0.5(%)	Note
PISCO	≈ 80	Stable across all popularity bins
OSCAR-Mistral	37-73	Increases with entity popularity
OSCAR-Qwen	27-59	Increases with entity popularity
xRAG	≈ 10	Near-random, fails reconstruction

Table 11: Paraphrase reconstruction metrics for context compression methods, PopQA benchmark

Pooler	Mean Cossim	Std Cossim	Effective rank
mean	0.905	0.016	200
max	0.901	0.008	0.516
mean_max	0.903	0.010	0.416
flatten	0.714	0.016	609
first	0.948	0.009	204
last	0.674	0.042	230

Table 12: Pooling strategies evaluation for projector

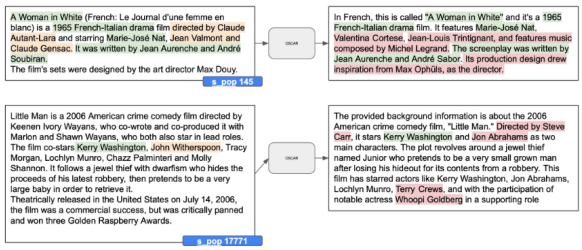

Figure 2: OSCAR context reconstruction examples. The model faithfully recovers most factual content: names, dates, roles, and plot details are largely preserved (green highlights). Some minor hallucinations appear (red highlights), but entity-level information is retained. Top: "A Woman in White" (entity popularity s_{pop} = 145). Bottom: "Little Man" (2006 film, s_{pop} = 17,771).

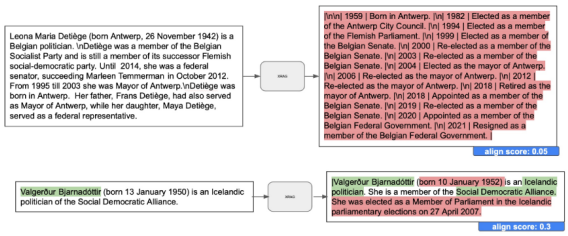

Figure 3: xRAG context reconstruction examples demonstrating catastrophic compression failure. Top: input is a biography of Leona Maria Detiège; xRAG hallucinates an entirely different structured timeline (AlignScore = 0.05). Bottom: input about Valgerður Bjarnadóttir (born 13 Jan- uary 1950); xRAG reproduces a similar but factually wrong birth date and different parliament election date (AlignScore = 0.3). Both outputs show the characteristic failure mode of single-token bottleneck compression: surface fluency without factual grounding.