Lost in Embedding: Reconstruction Supervision for Information-Preserving Dense Retrieval

Prayas Agrawal
Microsoft Research
t-agrawalpra@microsoft.com

Niket Tandon
Microsoft Research
niket.tandon@microsoft.com

Abstract

Standard dense retrievers optimize to bring relevant query document pairs closer, but do not preserve which task-critical information (e.g., entities, dates, concepts, or logical relationships) survives compression into embeddings. We observe that supplementing queries with such information at inference time improves retrieval, motivating the question of how to train information-preserving dense retrievers. We propose a new training strategy that augments contrastive learning with a reconstruction loss to provide supervision over embedding content. Our method, **RECOVIR**, is a decoder-only LLM retriever that appends learned latent tokens and uses their representations for both cosine retrieval and autoregressive reconstruction. Across multiple architectures, RECOVIR consistently improves nDCG@10 by 1–2 points on MTEB-English retrieval benchmarks over standard baselines. We find that retrieval failures often coincide with reconstruction errors, and show theoretically that when targets are relevance-sufficient and recoverable, reconstruction lower-bounds the relevance information retained in embeddings.

1 Introduction

Dense retrieval has become the dominant paradigm for RAG (Lewis et al., 2021), that powers knowledge-intensive NLP (Bajaj et al., 2018). Current embedding models (Wang et al., 2024; Xiao et al., 2024; Lee et al., 2025) produce high-quality embeddings, but when a retrieved document is wrong, we cannot inspect the embedding to see what input information it retained or lost, making them *opaque*. Standard contrastive training only reveals if a pair of embedding is close or far but does not expose what information the embedding used to make that decision, nor what was lost in the compression from text to vector.

However, dense embeddings have been shown to contain far more information than their surface opacity suggests e.g. Morris et al. (2023) show that embeddings can reveal information about the original text. To reveal this information, post-hoc inverters such as Vec2Text (Morris et al., 2023) train a separate model to invert embeddings back into text. They show evidence that embeddings are information-rich, but they are not ideal evidence of what a retriever actually encoded: a separate inverter may introduce content absent in the retrieval model's representation.

This motivates us to ask: *can the retriever be trained so that its embeddings can be directly decoded and inspected?* RECOVIR answers this question by making the embedding a latent bottleneck inside a decoder-only LLM. Given an input, we append learned latent tokens after a boundary marker, extract their final-layer hidden states, and use the same latent state for two purposes: cosine retrieval and autoregressive reconstruction. The reconstruction target need not be the raw input. Instead, it can be a task-specific description of the input's retrieval-relevant content, such as topic, entities, claims, or domain-specific fields. Thus, the reconstruction objective becomes a direct way to specify what the embedding should preserve. Reconstruction objective can target either verbatim or task-specific construction. Verbatim reconstruction would spend bottleneck capacity on surface details such as exact phrasing, punctuation, and formatting, which are often irrelevant for retrieval. Task-specific reconstruction instead pressures the embedding to preserve the semantic variables that should transfer: entities, relations, stance, evidence, or other information required by the model to retrieve relevant documents. See Figure 1 for an overview.

Empirically, across evaluated backbone and attention configurations, adding latent tokens and the reconstruction objective improves average MTEB nDCG@10 over the corresponding contrastive-only baseline.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 49498–49517
October 24-29, 2026 ©2026 Association for Computational Linguistics

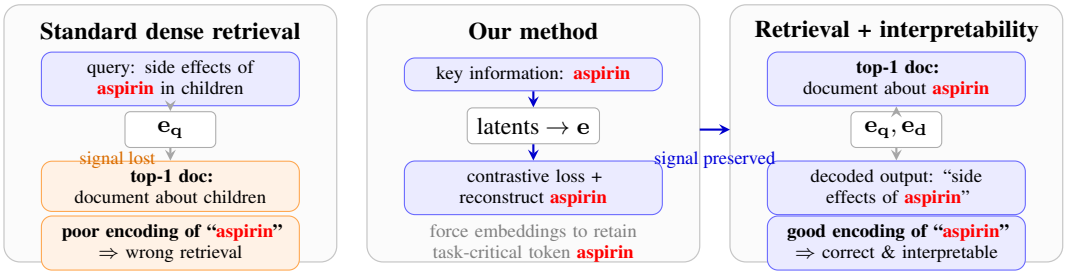


Figure 1: Standard dense retrieval can fail when compression discards task-critical information (see left). RECOVIR trains with retrieval loss and task-specific critical information reconstruction loss, encouraging the embedding to retain entities, constraints, numbers, stance, evidence, and other retrieval signals (see middle). At test time, RECOVIR provides decoded target "aspirin" along with the relevant document (see right).

Contributions.

- We introduce **RECOVIR**, a latent-token bottleneck for decoder-only LLM retrievers in which the same hidden state is used for cosine retrieval and intrinsic autoregressive decoding to controllable reconstruction targets as a supervision signal over embedding content (Sec. 3).
- We show that adding RECOVIR improves average nDCG@10 on the MTEB retrieval benchmark, over contrastive-only baselines across the evaluated model and attention settings, while producing embeddings that can be decoded for auditability (Sec. 6).
- We provide a theoretical explanation of why reconstruction and contrastive learning are complementary: reconstruction pressures the bottleneck to retain relevance-sufficient information, while contrastive training extracts it through the retrieval score (Sec. 4).

2 Task-Specific Information and Embedding Loss

To test whether retrieval failures come from lost information, we need a concrete target for what the embedding should have preserved.

2.1 Getting task-specific information

For a query q, let q_{info} denote a compact textual proxy for the information in q that is useful for retrieving its relevant documents. We construct q_{info} with a task-specific teacher prompt. In the experiments below, the teacher may use the benchmark relevance signal to select which parts of the query are retrieval-critical; q_{info} is therefore not an inference-time input, but a probe for measuring what information a retriever could have used.

The remaining section asks two empirical questions to motivate the architecture: (i) is task-specific information q_{info} useful for retrieval? and (ii) when retrieval fails, is the model's encoding of q_{info} poor?

2.2 Generated Retrieval Information Improves Retrieval

To validate q_{info} as a proxy, we perform a controlled query-substitution experiment. We encode either the original query q or the generated query q_{info}, keep the document corpus unchanged, and retrieve with the same bi-encoder. If q_{info} improves nDCG metric, then the teacher has extracted a view of the query that is more directly aligned with the benchmark relevance relation. This does not claim that q_{info} is a deployable query rewrite or a unique minimal explanation; it only tests whether the generated proxy contains retrieval-sufficient information.

The gains in Table 1(a) validate q_{info} as a high-signal proxy for task-specific information. They also motivate a sharper question: when the original query fails, did the embedding preserve the information captured by q_{info}, or was that information lost before ranking?

2.3 Retrieval Failures Often Reflect Lost Key Information

We next analyze complete retrieval failures, defined as queries with nDCG@10 equal to zero. For each failed query, we inspect decoded reconstructions from the query and document embeddings produced by our model (Section 3) and assign the failure to a small set of categories. "Recon error" means that at least one decoded representation omits or distorts task-critical information needed to identify the relevant document. "Ranking er-

Table 1: Left: replacing the original query q with generated information query q_{info} improves nDCG@10 over the original document corpus. Right: complete failures (nDCG@10 = 0) are often accompanied by reconstructed-information errors; n is the number of failed queries, rates are fractions over failures, Recon aggregates query/document reconstruction errors, and dominant modes show the largest task-specific subtypes.

(a) Query substitution

Dataset	q	q_{info}	Δ
ArguAna	57.5	58.3	+0.8
SCIDOCS	22.7	24.3	+1.6
FiQA	40.8	41.3	+0.5

(b) Failure analysis

Dataset	n	Recon	Rank	Noise	Dominant modes
ArguAna	230	.93	.05	.02	Evidence .33, Topic .19
SCIDOCS	362	.51	.22	.27	Label .26, Acronym .23
FiQA	213	.66	.23	.11	Concept .23, Rank .23
HotpotQA	46	.93	.07	.00	Entity .37, Bridge .22

ror" means the reconstructions appear faithful, but nearest-neighbor ranking still misses the positives. "Label noise" covers cases where the labeled positives are incomplete or questionable. Table 1(b) reports the resulting rates.

Across ArguAna, FiQA2018, and HotpotQA, reconstruction errors account for the majority of complete failures, while pure ranking errors account for a smaller fraction. SCIDOCS is noisier because many failures are judged to be label noise or hard ranking cases, but roughly half of failures still show missing or distorted reconstructed information. Thus, decoded reconstructions do not reduce all retrieval errors to information loss; rather, they separate information-loss failures from ranking failures and dataset-noise cases.

Appendix B breaks the complete failures into task-specific subtypes. The dominant subtypes match the benchmark semantics: ArguAna failures often lose evidence, topic, or stance information; HotpotQA failures concentrate on entity, bridge, and attribute preservation; SCIDOCS frequently exposes citation-label ambiguity alongside acronym or method corruption; and FiQA2018 mixes financial concept drift with hard-ranking and answer ambiguity. Fine-grained numerical errors in FiQA2018 are about 4.2% of complete failures, and they are highly revealing when they occur, as in Table 2. Prior work has likewise shown that dense retrieval can struggle when relevance depends on quantities, units, and comparison intent (Agrawal et al., 2025).

2.4 Motivating example

Table 2 shows one audited FiQA2018 failure. The original query is about holding more than the FDIC insurance limit in a bank account, but the decoded query embedding drops a zero and changes the amount from $250,000 to $25,000. After this loss, retrieval falls back to generic bank-account mistake answers rather than FDIC-limit answers about splitting deposits across insured accounts. A query-only q_{info} that preserves the FDIC-limit information raises this query's nDCG@10 from 0.0 to 0.316, illustrating how a small reconstruction error can correspond to a real retrieval miss.

Table 2: A FiQA2018 failure where the decoded query embedding loses retrieval-critical numerical information.

Original query	I have more than $250,000 in a US Bank account… mistake?
Query-only q_{info}	Is holding over $250,000 in a US bank account a mistake? Evaluating FDIC insurance limits, deposit insurance coverage, and strategies for protecting large cash balances.
Decoded query	I have more than $25,000 in US bank account mistake?
Top retrieved answer	Banks make mistakes. Reconciling your account with your bank statement is the way to catch the errors.
Relevant answer	Split the money among multiple banks so no account exceeds the FDIC insurance limit.

The rest of the paper introduces RECOVIR, a training procedure that uses reconstruction of task-specific retrieval information as a supervision signal so the embedding is pressured to retain information of the kind represented by q_{info}.

3 Method

3.1 Architecture

RECOVIR builds on a pretrained decoder-only LLM fine-tuned with LoRA (Hu et al., 2021). We add K special *latent tokens* to the vocabulary, each with a unique learned embedding, along with a <|stop|> boundary token. For an input text x with token sequence $(x_1, \ldots, x_n)$, the model input is:

$$\mathbf{x}^{\text{enc}} = [x_1, \ldots, x_n, <|\texttt{stop}|>, \ell_1, \ldots, \ell_K] \quad (1)$$

where $\ell_1, \ldots, \ell_K$ are the latent tokens. A single forward pass through the LLM produces hidden states at all positions. We extract the hidden states

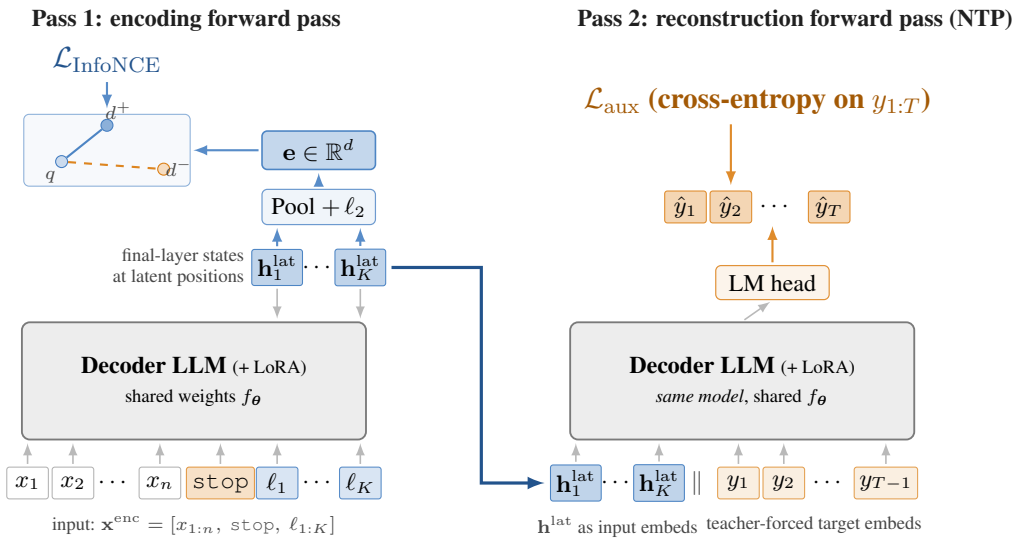


Figure 2: RECOVIR training step (cf. Alg. 1). Both passes use the *same* decoder-only LLM with shared weights $f_{\boldsymbol{\theta}}$ (LoRA-adapted), drawn upright with token-level inputs at the bottom and outputs at the top. **Pass 1 (left, encoding).** The input $\mathbf{x}^{\text{enc}} = [x_{1:n},\ \texttt{stop},\ \ell_{1:K}]$ is encoded in one forward pass; the final-layer states at the K latent positions form $\mathbf{h}^{\text{lat}}$, which is pooled and ℓ_2-normalised into the retrieval embedding $\mathbf{e}$ and trained with $\mathcal{L}_{\text{InfoNCE}}$ against in-batch and hard negatives. **Pass 2 (right, reconstruction).** $\mathbf{h}^{\text{lat}}$ is fed as the first K input embeddings to the *same* model, followed by teacher-forced target-token embeddings $y_{1:T-1}$; the LM head produces next-token predictions $\hat{y}_{1:T}$ trained with $\mathcal{L}_{\text{aux}}$ against the task-specific target y (e.g. for a FiQA query, y might be "q_{info}: more than $250,000 in a US bank account; FDIC insurance limit"). The arrow between the two passes carries $\mathbf{h}^{\text{lat}}$ from Pass 1 into Pass 2 as input embeddings, so the reconstruction signal pressures the same bottleneck that supplies $\mathbf{e}$.

at the K latent positions from the final layer:

$$\mathbf{h}^{\text{lat}} = [\mathbf{h}^{(L)}_{n+2}, \ldots, \mathbf{h}^{(L)}_{n+K+1}] \in \mathbb{R}^{K \times d_h} \quad (2)$$

where L is the number of transformer layers and d_h is the hidden dimension. A pooling function $\text{Pool}(\mathbf{h}^{\text{lat}}) \in \mathbb{R}^{d_h}$ (last-token pooling by default) followed by ℓ_2-normalisation produces the retrieval embedding $\mathbf{e} \in \mathbb{R}^{d_h}$.

3.2 Training Objective

RECOVIR is trained with a composite loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{InfoNCE}} + \lambda\, \mathcal{L}_{\text{aux}}$, combining contrastive and reconstruction objectives.

Contrastive loss ($\mathcal{L}_{\text{InfoNCE}}$). Standard InfoNCE with in-batch negatives and hard negatives:

$$\begin{aligned} s(q,d) &= \cos(\mathbf{e}_q, \mathbf{e}_d)/\tau, \\ Z_q &= \sum_{d' \in \mathcal{D}_q} \exp s(q,d'), \\ \mathcal{L}_{\text{InfoNCE}} &= -\frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \log \frac{\exp s(q,d^+)}{Z_q}. \end{aligned} \quad (3)$$

where $\mathcal{D}_q = \{d^+\} \cup \{d^-_1, \ldots, d^-_N\} \cup \mathcal{D}_{\text{in-batch}}$ includes the positive document, N hard negatives, and all in-batch negatives. Temperature $\tau = 0.02$.

Reconstruction loss ($\mathcal{L}_{\text{aux}}$). After the encoding forward pass, we extract the final-layer hidden states $\mathbf{h}^{\text{lat}}$ at the latent positions. These are fed as input embeddings — concatenated with the token embeddings of the reconstruction target — and the decoder is trained via cross-entropy:

$$\mathcal{L}_{\text{aux}} = -\frac{1}{T} \sum_{t=1}^{T} \log p_{\boldsymbol{\theta}}(y_t \mid \mathbf{h}^{\text{lat}}, y_{1:t-1}) \quad (4)$$

where $(y_1, \ldots, y_T)$ is the reconstruction target (see Section 3.3). This is the key component that makes embeddings decodable: by choosing what the model must reconstruct from $\mathbf{h}^{\text{lat}}$, practitioners directly control what information the embedding retains — a form of supervision unavailable in purely contrastive training.

3.3 Controllable Reconstruction Targets

Contrastive loss specifies *whom* to match but is silent on *what* the embedding should capture. The reconstruction target fills this gap: it defines a **second, orthogonal axis of supervision** that directly controls the information content of the bottleneck. By choosing what the model must reconstruct from $\mathbf{h}^{\text{lat}}$, practitioners specify what the embedding must retain.

Verbatim text reconstruction is a poor choice for this axis: it forces the bottleneck to preserve surface-form details — exact wording, formatting,

Algorithm 1 RECOVIR Training (one step)

Require: Batch $\{(q_i, d_i^+, \{d_{i,j}^-\}_j)\}_i$, model $f_{\boldsymbol{\theta}}$

Pass 1: encode all retrieval texts
for each text $x \in \{q_i\} \cup \{d_i^+\} \cup \{d_{i,j}^-\}$ **do**
  $\mathbf{x}^{\text{enc}} \leftarrow [x, <|\texttt{stop}|>, \ell_{1:K}]$
  $\mathbf{h}^{(1:L)} \leftarrow f_{\boldsymbol{\theta}}(\mathbf{x}^{\text{enc}})$
  $\mathbf{h}^{\text{lat}} \leftarrow \mathbf{h}^{(L)}_{\text{latent positions}}$
  $\mathbf{e} \leftarrow \ell_2\text{-norm}(\text{Pool}(\mathbf{h}^{\text{lat}}))$
end for
$\mathcal{L}_{\text{InfoNCE}} \leftarrow$ Eq. (3) using query/document embeddings
Pass 2: reconstruct targets from final-layer latents
for each (text x, target y) pair **do**
  $\mathbf{e}_{\text{input}} \leftarrow [\mathbf{h}_x^{\text{lat}}; \text{Embed}(y_{<T})]$
  $\text{logits} \leftarrow f_{\boldsymbol{\theta}}(\mathbf{e}_{\text{input}})$
  $\mathcal{L}_{\text{aux}} \mathrel{+}= -\sum_t \log p(y_t \mid \mathbf{h}_x^{\text{lat}}, y_{<t})$
end for
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{InfoNCE}} + \lambda\, \mathcal{L}_{\text{aux}}$
Update $\boldsymbol{\theta}$ via AdamW on $\mathcal{L}_{\text{total}}$

punctuation — that are often irrelevant for retrieval, and documents can be too long for K tokens to faithfully reproduce. Instead, we use a teacher LLM to generate *task-specific descriptions* that extract the retrieval-relevant content of each input, serving as $(y_1, \ldots, y_T)$ in Eq. (4).

Crucially, different retrieval tasks require different information to be preserved. A multi-hop reasoning task needs entity names and their relationships; an argument retrieval task needs stance polarity and supporting evidence; a biomedical task needs population, intervention, and outcome. We design task-specific reconstruction prompts that extract exactly these signals (full prompts in Appendix D), so the reconstruction axis teaches each embedding to prioritise what matters for *its* retrieval task. The framework is general: any target that encodes a useful signal — structured JSON, natural-lang. summaries, or domain-specific extractions — can serve as the reconstruction objective.

3.4 Training Algorithm

Algorithm 1 summarises one training step. Pass 1 encodes all texts with latent tokens and computes the contrastive loss; Pass 2 reconstructs targets from the extracted final-layer latent hidden states.

4 Theoretical Analysis

We view the embedding as an information bottleneck (Tishby et al., 2000): the encoder compresses text into a vector that must still support relevance decisions. InfoNCE-style contrastive learning (van den Oord et al., 2019) trains the geometry of this space through pairwise scores, but it does not specify which information should survive compression. The reconstruction target provides that specification, and the auxiliary prediction loss gives a variational mechanism for enforcing it (Alemi et al., 2019).

We analyze a simplified setting. Let $X = (X_q, X_d)$ be a query–document text pair, $R \in \{0, 1\}$ its relevance label, $E = (E_q, E_d)$ the encoded latent bottleneck states for the query and document, $S = s(E_q, E_d)$ the retrieval score after deterministic pooling and normalization, and $Y = (Y_q, Y_d)$ the reconstruction target. Thus E denotes the representation from which both the retrieval score and the reconstruction decoder read out information. The teacher is fixed, so $Y = g(X)$ on the training distribution. Full proofs are in Appendix A.

4.1 Reconstruction Preserves Relevance Information

Two quantities determine whether reconstruction can help retrieval: target sufficiency and target recoverability from the embedding.

Definition 1 (δ-sufficiency)**.** A reconstruction target Y is *δ-sufficient for relevance* if

$$I(R; X \mid Y) \le \delta. \tag{5}$$

Small δ means that the target omits little relevance information.

Definition 2 (ϵ-reconstructability)**.** An embedding E is *ϵ-reconstructive for Y* if

$$H(Y \mid E) \le \epsilon. \tag{6}$$

The auxiliary loss controls ϵ through the standard cross-entropy decomposition. For a decoder distribution $p_\theta(Y \mid E)$,

$$\begin{aligned}\mathbb{E}[-\log p_\theta(Y \mid E)] &= H(Y \mid E) \\ &\quad + \mathbb{E}\,\mathrm{KL}(p(Y \mid E) \,\|\, p_\theta(Y \mid E)) \\ &\ge H(Y \mid E).\end{aligned} \tag{7}$$

Thus lowering reconstruction cross-entropy lowers an upper bound on ϵ. For the token-averaged loss in Eq. (4), the corresponding sequence-level bound is obtained by summing over target tokens.

Theorem 1 (Relevance information retained by reconstruction)**.** *Assume $Y = g(X)$ and $E = f(X)$ are deterministic, Y is δ-sufficient for relevance, and E is ϵ-reconstructive for Y. Then*

$$I(R; E) \geq I(R; X) - \delta - \epsilon. \quad (8)$$

Moreover, if $S = s(E_q, E_d)$ is deterministic and

$$\eta = I(R; E \mid S), \quad (9)$$

then

$$I(R; S) \geq I(R; X) - \delta - \epsilon - \eta. \quad (10)$$

The bound separates three failure modes. The target can be incomplete (δ), the embedding can fail to retain the target (ϵ), or the score can fail to extract information present in the embedding (η). Reconstruction affects the second term directly; contrastive training acts on the score.

4.2 Why Task-Specific Targets Matter

For a target Y, define

$$B(Y, E) = \delta(Y) + \epsilon(Y, E), \quad (11)$$

where $\delta(Y) = I(R; X \mid Y)$ and $\epsilon(Y, E) = H(Y \mid E)$. Theorem 1 lower-bounds retained relevance information by $I(R; X) - B(Y, E)$.

Corollary 2 (Target comparison)**.** *For two reconstruction targets Y_a and Y_b with embeddings E_a and E_b, if*

$$\delta(Y_a) + \epsilon(Y_a, E_a) < \delta(Y_b) + \epsilon(Y_b, E_b), \quad (12)$$

then Y_a gives a strictly stronger lower bound on retained relevance information than Y_b.

This criterion explains why the target should be task-specific. Verbatim reconstruction may have small δ but large ϵ, because surface form and irrelevant details are costly to reproduce. A generic summary may have small ϵ but large δ, because it can drop the entity, relation, stance, or condition needed for retrieval. Task-specific targets are useful when they reduce $\delta + \epsilon$.

5 Experimental Setup

Models. We evaluate three decoder-only backbones spanning two model families: Qwen3-0.6B, Qwen3-1.7B (Yang et al., 2025), and Llama-3.2-1B (Grattafiori et al., 2024). All models are fine-tuned with LoRA (rank 64, $\alpha = 32$, dropout 0.05) applied to all attention projections (Q, K, V, O) and feed-forward layers (gate, up, down). Flash attention (Dao, 2023) and Liger fused kernels (Hsu et al., 2025) are used for efficiency.

Attention modes. To demonstrate that RECOVIR is complementary to the choice of attention pattern, we evaluate each backbone under three attention configurations:

- **Causal**: standard left-to-right attention. The embedding is extracted from K latent tokens appended after a `<|stop|>` boundary token, pooled via last-token pooling.
- **Echo**: the input is duplicated as $[\text{content}, \text{content}, \texttt{EOS}]$ under causal attention, providing pseudo-bidirectional context in the second copy (Springer et al., 2025). The embedding is pooled from the EOS position.
- **Bidirectional attention**: causal masks are removed entirely, following bidirectional adaptation for decoder-only language models (BehnamGhader et al., 2024). Mean pooling over all content positions yields the embedding.

For RECOVIR, latent tokens and the reconstruction loss are added on top of each configuration; the baseline uses the same attention mode and pooling but without latent tokens or reconstruction ($\lambda = 0$).

Training. We train each run on a single GPU with HuggingFace Accelerate. AdamW optimiser with $\text{lr} = 3 \times 10^{-4}$, linear warmup (100 steps) and linear decay. Batch size 64 with 7 hard negatives per query. $\lambda = 0.5$ (reconstruction loss weight). Training for 5000 steps on the F2LLM data recipe (Zhang et al., 2025), spanning diverse retrieval datasets including MS MARCO (Bajaj et al., 2018), NQ (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), and others. Maximum lengths are 128 query tokens and 256 document tokens.

Evaluation. We evaluate on held-out MTEB retrieval splits (Muennighoff et al., 2023): ArguAna, CQADupstack (Gaming, Unix), ClimateFEVER, FEVER, FiQA-2018, HotpotQA, and SCIDOCS. Primary metric: nDCG@10.

Baselines. For each (model, attention mode) pair, the baseline is the *same model trained with the same data and hyperparameters*, using only the contrastive loss ($\lambda = 0$) and standard pooling (no latent tokens). This controlled comparison isolates the combined effect of the latent-token bottleneck and reconstruction supervision within each matched setting. The $\lambda = 0$ ablation in Table 6 instead retains the latent tokens and removes only reconstruction, isolating the contribution of the auxiliary loss.

Table 3: nDCG@10 on MTEB retrieval tasks across different architectures and attention configurations. RECOVIR outperforms the base architecture.

	Method	ArguAna	CQA-G	CQA-U	ClimaF	FEVER	FiQA	HotpQA	SciD	Avg
Causal attention										
Qwen3-0.6B	Contrastive	57.0	58.6	46.7	41.8	90.3	38.6	69.8	21.6	53.1
	+RECOVIR	**59.4**	**60.3**	**47.4**	**43.1**	**90.6**	**40.5**	**72.8**	**22.6**	**54.6**
Qwen3-1.7B	Contrastive	60.6	**64.1**	**52.0**	44.8	92.0	47.7	75.0	24.4	57.6
	+RECOVIR	**62.4**	63.8	51.8	**45.8**	**92.3**	**48.3**	**75.7**	**25.5**	**58.2**
Llama-3.2-1B	Contrastive	59.2	62.6	49.3	43.4	92.0	44.4	75.5	22.8	56.1
	+RECOVIR	**61.3**	**63.9**	**50.6**	**46.6**	**93.2**	**46.1**	**76.6**	**24.5**	**57.9**
Echo attention										
Qwen3-0.6B	Contrastive	**59.1**	**61.2**	49.2	42.0	**91.5**	41.9	71.4	22.5	54.9
	+RECOVIR	59.0	61.1	**49.5**	**42.7**	91.0	**42.4**	**72.7**	**22.8**	**55.2**
Qwen3-1.7B	Contrastive	62.8	65.3	53.9	43.4	92.3	49.7	75.8	25.5	58.6
	+RECOVIR	**63.3**	**65.8**	**54.2**	**45.2**	**92.6**	**50.5**	**76.5**	**25.9**	**59.3**
Llama-3.2-1B	Contrastive	60.9	64.1	51.3	44.8	93.0	46.7	76.6	24.4	57.7
	+RECOVIR	**62.5**	**65.2**	**52.7**	**47.7**	**94.6**	**48.2**	**78.8**	**25.7**	**59.4**
Bidirectional attention										
Qwen3-0.6B	Contrastive	56.7	60.7	49.4	42.8	91.2	40.2	71.6	22.3	54.4
	+RECOVIR	**58.9**	**61.7**	**49.5**	**44.0**	**91.3**	**41.2**	**73.4**	**23.0**	**55.4**
Llama-3.2-1B	Contrastive	59.1	63.9	50.4	44.0	92.4	45.0	75.9	22.9	56.7
	+RECOVIR	**61.3**	**65.6**	**51.5**	**45.9**	**94.3**	**46.8**	**78.0**	**24.7**	**58.5**

Table 4: Broader text-retrieval evaluation using matched baselines.

Benchmark	Metric	RECOVIR	Base
BEIR-15 avg.	nDCG@10	0.5837	0.5741
MS MARCO dev	MRR@10	0.6874	0.6802

6 Results

6.1 Retrieval Quality

Table 3 reports nDCG@10 across the eight MTEB retrieval tasks used in the main comparison. Each row pair is a controlled comparison: same backbone, attention mode, data recipe, and training setup; the difference is whether the latent-token bottleneck and reconstruction objective are used.

Across backbones and attention modes, adding reconstruction supervision improves retrieval over the matched contrastive baseline. The largest gains appear on tasks where relevance depends on preserving specific entities, conditions, evidence, or multi-hop cues, such as ArguAna, ClimateFEVER, HotpotQA, FiQA, SCIDOCS. This pattern matches the information-loss analysis in Section 2: reconstruction is useful when it pressures the embedding to retain the information that retrieval later needs.

Table 4 reports results on BEIR and MS MARCO dev. Under a stronger, longer matched recipe, RECOVIR also reaches 59.4 average nDCG@10 versus 58.1 for the EOS-pooling baseline, a gain of 1.3 points.

6.2 Code Retrieval

CoIR is a comprehensive benchmark for evaluating code information retrieval across diverse retrieval tasks and code domains (Li et al., 2025). To evaluate a domain distinct from the text benchmarks, we evaluate nine CoIR tasks using a Qwen-1.5B backbone and a matched EOS-pooling contrastive baseline, both trained for 8k LoRA steps (Appendix E.5).

Table 5 shows that RECOVIR improves average nDCG@10 from 58.1 to 58.8, outperforming the matched baseline on seven tasks, tying on CodeEditSearchRetrieval, and trailing on AppsRetrieval.

6.3 Ablation Studies

Sensitivity to λ. The reconstruction loss is useful as a pressure on the bottleneck, but it should not dominate the retrieval objective. Table 6 varies its weight in the 2k-step Qwen3-0.6B causal setup. The $\lambda = 0$ control retains the latent-token architecture but removes reconstruction, isolating the auxiliary loss from the effect of adding tokens. At the tuned $\lambda = 0.5$, reconstruction adds 2.8 nDCG@10 and 1.5 Recall@10 over this control; increasing the

Table 5: nDCG@10 on nine CoIR code-retrieval tasks.

Method	Apps	CEdit	CF-MT	CF-ST	CTO-C	CTO-DL	CosQA	SO-QA	Text2SQL	Avg
Contrastive (EOS)	**30.0**	**81.2**	45.8	78.8	75.6	25.8	35.5	89.1	60.8	58.1
+RECOVIR	28.2	**81.2**	**47.1**	**79.2**	**76.2**	**26.7**	**35.8**	**89.5**	**65.7**	**58.8**

Table 6: Ablation over reconstruction-loss strength for the Qwen3-0.6B causal model trained for 2k steps. Scores are averages over the evaluation tasks.

Reconstruction weight	nDCG@10	Recall@10
$\lambda = 0.0$ (no reconstruction)	50.0	49.1
$\lambda = 0.5$	52.8	50.6
$\lambda = 1.0$	51.7	49.9
$\lambda = 2.0$	50.2	49.4

Table 7: Eval-set ablation comparing verbatim reconstruction targets with task-specific reconstruction targets. Scores and Δ are nDCG@10 / Recall@10;

Task	Verbatim	Task-spec	Δ
ArguAna	57.8 / 85.0	**60.1 / 86.9**	+2.3 / +1.9
CQA-U	**49.3 / 62.6**	**49.3** / 62.0	+0.0 / −0.6
ClimaF	43.2 / 49.3	**44.4 / 51.3**	+1.2 / +2.0
FiQA	42.6 / 50.8	**43.3 / 51.1**	+0.7 / +0.3
HotpQA	72.8 / 75.3	**75.0 / 77.7**	+2.2 / +2.4
SciD	23.2 / 24.5	**23.7 / 24.9**	+0.5 / +0.4
Avg	48.2 / 57.9	**49.3 / 59.0**	+1.1 / +1.1

weight to 1.0 and 2.0 steadily reduces both metrics. This is consistent with the role of reconstruction in RECOVIR: shape what the embedding retains and let contrastive learning optimize retrieval geometry.

Target choice. Reconstruction target also matters. Table 7 compares verbatim-reconstruction with task-specific LLM-generated targets. Task-specific targets improve the average by 1.1 points in both nDCG@10 and Recall@10, with the clearest gains on ArguAna, ClimateFEVER, and HotpotQA. The marginal recall drop on CQA-U indicates that gains are not automatic; the target must preserve task-relevant info. Overall, this supports the supervision-axis view: reconstruction is most useful when it asks the bottleneck to retain task-relevant info, not merely reproduce input text.

6.4 Efficiency

RECOVIR separates retrieval from reconstruction (Table 8). The standard path encodes embeddings and ranks by cosine similarity, adding no embedding-time overhead, index storage during inference. During training, RECOVIR reconstruction adds approximately 20% wall-clock time per training step.

Inference-time Decoding is invoked on demand as a diagnostic and interpretability signal, revealing which task-critical information remains recoverable from the embedding and helping distinguish information-loss from retrieval-geometry errors; Table 2 shows a dropped zero ($250,000 → $25,000). Each inspection costs one standard autoregressive generation but adds no retrieval latency.

7 Related Work

Dense retrieval. Dense retrieval has shifted from BERT-scale dual encoders (Karpukhin et al., 2020; Reimers and Gurevych, 2019; Ni et al., 2021) to decoder-only LLM backbones, which now set the state of the art on standard benchmarks (Thakur et al., 2021; Muennighoff et al., 2023). Recent work improves such models through large contrastive training recipes (Wang et al., 2024; Xiao et al., 2024), latent or repeated pooling (Lee et al., 2025; Springer et al., 2025), and bidirectional adaptation of decoder LLMs (BehnamGhader et al., 2024). Other approaches use reconstruction pre-training, joint sparse–dense training, unified generation and embedding, or task-specific scoring architectures (Su et al., 2025; Zeng et al., 2025; Muennighoff et al., 2025; Agrawal et al., 2025; Nair et al., 2026). In contrast, RECOVIR trains task-specific reconstruction *jointly* on the same latent states used for contrastive retrieval, so decoding is a property of the deployed retriever rather than a separate pre-training stage or task switch.

Interpretability and inversion of retrieval embeddings. Sparse retrievers are inherently interpretable: BM25 and learned sparse models such as SPLADE (Formal et al., 2021) expose vocabulary-indexed term weights, while multi-vector approaches such as ColBERT (Khattab and Zaharia, 2020) provide token-level interaction scores but require per-token storage and do not yield a single decoded representation. For dense retrieval, most interpretability methods operate after training: sparse autoencoders can decompose

Table 8: Cost profile of RECOVIR relative to the matched dense baseline.

Phase	Operation	Invocation	Additional cost
Training	Short-target reconstruction	Every step	∼20% wall-clock / step
Retrieval	Encoding, indexing, cosine ranking	Every request	None
Diagnosis	Autoregressive reconstruction	On demand	One generation / inspected embedding

embeddings from a fixed DPR model into named latent concepts (Park et al., 2025), while DIME attributes retrieval decisions to individual embedding dimensions (Campagnano et al., 2025). The sparse-autoencoder concept retriever is a post-hoc discretisation whose retrieval effectiveness trails the dense target model. These methods explain or discretise an existing vector or score, but they do not train the retriever's own representation to preserve decodable task information. Related context-compression methods such as Gisting (Mu et al., 2024) and ICAE (Ge et al., 2024) also use learned soft tokens to bottleneck text, but optimise them for downstream model generation rather than retrieval inspection. A complementary line treats embeddings as text-bearing artefacts and reconstructs their inputs with separate inverter models (Morris et al., 2023; Chen et al., 2024), showing that dense vectors can leak substantial input information but also introducing content from a model other than the retriever itself. RECOVIR bakes decodability into training rather than analyzing it later: the same model that produces $\mathbf{e}$ also decodes from latent states behind it, so the decoded text reflects what the retriever's bottleneck actually retains and can be inspected per query at test time.

Reconstruction objectives and the information bottleneck. Reconstruction has been used to pressure a latent code to retain semantic content: text VAEs such as RNN VAE (Bowman et al., 2016), Optimus (Li et al., 2020) pair a stochastic latent with a generative decoder, motivated by the information-bottleneck principle (Shannon, 1948; Tishby et al., 2000) and its variational realizations (Alemi et al., 2019). RECOVIR does not impose a prior over embedding space or use a separate inference network; instead, it uses cross-entropy on a task-specific target through the same decoder to tighten a lower bound on the relevance information retained by the latent bottleneck (Section 4), so reconstruction acts as controllable sufficiency pressure complementary to contrastive geometry trained by InfoNCE (van den Oord et al., 2019).

8 Conclusion

We introduce RECOVIR which brings a new perspective on dense retrieval training, producing interpretable representations that utilizes and produces key information in the input to embed. This provides a diagnostic lens for identifying when critical signals are lost or preserved in the embedding. By training the same latent bottleneck for retrieval and task-specific reconstruction, the model is pressured to retain retrieval-relevant content. We show that RECOVIR improves retrieval performance across architectures, and we provide theoretical justifications of this improvement. Broadly, we show that embeddings can be trained as compact, partially decodable representations whose information content is part of the objective, improving retrieval while making some failures easier to inspect.

9 Limitations

For very long inputs, a small set of latent tokens placed only after the full sequence may not be enough to preserve all retrieval-relevant information. This is most likely to matter for long documents, multi-hop evidence, or inputs whose crucial details appear far apart. A natural extension is to insert latent slots periodically through long contexts, so the model can compress information locally before pooling it into the retrieval representation.

Our experiments use small-to-mid-sized decoder backbones. Larger models may change the tradeoff between retrieval accuracy, reconstruction quality, and bottleneck capacity, so the scaling behaviour should be measured directly rather than inferred from the present runs.

The reconstruction target is another design choice. Targets that are too generic can fail to add useful retrieval signal, while query-only evaluations require targets constructed without answer-side information. For new domains, the target prompts should therefore be validated and reported as part of the experimental setup.

Finally, the decodability of embeddings raises privacy considerations: embeddings should be treated with the same protections as raw text.

References

Prayas Agrawal, Nandeesh Kumar K M, Muthusamy Chelliah, Surender Kumar, and Soumen Chakrabarti. 2025. Dense retrieval with quantity comparison intent. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23825–23839, Vienna, Austria. Association for Computational Linguistics.

Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. 2019. Deep variational information bottleneck. *Preprint*, arXiv:1612.00410.

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.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *Preprint*, arXiv:2404.05961.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. *Preprint*, arXiv:1511.06349.

Cesare Campagnano, Antonio Mallia, and Fabrizio Silvestri. 2025. Unveiling dime: Reproducibility, generalizability, and formal analysis of dimension importance estimation for dense retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, New York, NY, USA. Association for Computing Machinery.

Yiyi Chen, Heather Lent, and Johannes Bjerva. 2024. Text embedding inversion security for multilingual language models. *Preprint*, arXiv:2401.12192.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *Preprint*, arXiv:2307.08691.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval. *Preprint*, arXiv:2109.10086.

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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pin-Lun Hsu, Yun Dai, Vignesh Kothapalli, Qingquan Song, Shao Tang, Siyu Zhu, Steven Shimizu, Shivam Sahni, Haowen Ning, and Yanning Chen. 2025. Liger kernel: Efficient triton kernels for llm training. *Preprint*, arXiv:2410.10989.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

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.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Preprint*, arXiv:2004.12832.

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

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

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.

Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing sentences via pre-trained modeling of a latent space. *Preprint*, arXiv:2004.04092.

Xiangyang Li, Kuicai Dong, Yi Quan Lee, Wei Xia, Hao Zhang, Xinyi Dai, Yasheng Wang, and Ruiming Tang. 2025. CoIR: A comprehensive benchmark for code information retrieval models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22074–22091, Vienna, Austria. Association for Computational Linguistics.

John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. 2023. Text embeddings reveal (almost) as much as text. *Preprint*, arXiv:2310.06816.

Jesse Mu, Xiang Lisa Li, and Noah Goodman. 2024. Learning to compress prompts with gist tokens. *Preprint*, arXiv:2304.08467.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative representational instruction tuning. *Preprint*, arXiv:2402.09906.

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

Kartik Nair, Pritish Chakraborty, Atharva Tambat, Indradyumna Roy, Soumen Chakrabarti, Anirban Dasgupta, and Abir De. 2026. A dense subset index for collective query coverage. In *International Conference on Learning Representations*, volume 2026, pages 67280–67319.

Jianmo Ni, Gustavo Hernández Ábrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *Preprint*, arXiv:2108.08877.

Seongwan Park, Taeklim Kim, and Youngjoong Ko. 2025. Decoding dense embeddings: Sparse autoencoders for interpreting and discretizing dense retrieval. *Preprint*, arXiv:2506.00041.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *Preprint*, arXiv:1908.10084.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. 2025. Repetition improves language model embeddings. *Preprint*, arXiv:2402.15449.

Chang Su, Dengliang Shi, Siyuan Huang, Jintao Du, Changhua Meng, Yu Cheng, Weiqiang Wang, and Zhouhan Lin. 2025. Training llms to be better text embedders through bidirectional reconstruction. *Preprint*, arXiv:2509.03020.

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.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *Preprint*, arXiv:1803.05355.

Naftali Tishby, Fernando C. Pereira, and William Bialek. 2000. The information bottleneck method. *Preprint*, arXiv:physics/0004057.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

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.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. *Preprint*, arXiv:2309.07597.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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

Hansi Zeng, Julian Killingback, and Hamed Zamani. 2025. Scaling sparse and dense retrieval in decoder-only llms. *Preprint*, arXiv:2502.15526.

Ziyin Zhang, Zihan Liao, Hang Yu, Peng Di, and Rui Wang. 2025. F2llm technical report: Matching sota embedding performance with 6 million open-source data. *Preprint*, arXiv:2510.02294.

Supplementary Material: Lost in Embedding: Reconstruction Supervision for Information-Preserving Dense Retrieval

Contents

Appendices 12
A Proofs for Theoretical Analysis 12
A.1 Proof of Theorem 1 12
A.2 Cross-Entropy Upper Bound 13
A.3 Proof of Corollary 2 13
B Failure Analysis Subtypes 13
C Significance Tests 14
D Task-Specific Reconstruction Prompts 14
E Implementation Details 19
E.1 Training Datasets 19
E.2 Evaluation Instructions 20
E.3 Full Hyperparameters 20
E.4 Compute 20
E.5 CoIR Evaluation 20
F Broader Impact 20

A Proofs for Theoretical Analysis

This appendix proves the information-retention statements in Section 4. All random variables are defined on the training distribution: X is the query–document pair, R is the relevance label, Y is the fixed reconstruction target, E is the encoded latent bottleneck pair, and $S = s(E_q, E_d)$ is the retrieval score after deterministic pooling and normalization. The proof abstracts away tokenization, pooling, and the number of latent tokens; these details only determine the conditional distribution of E given X. We write the argument for discrete variables for notational simplicity. The same identities hold for continuous embeddings using the usual measure-theoretic form of mutual information, since the only operations used are the chain rule, non-negativity of conditional mutual information, and the data-processing consequence of deterministic maps.

The key point is that the theorem does not assume reconstruction directly optimizes retrieval. Instead, it proves an information-retention statement: if the target contains the information needed for relevance and the embedding retains the target, then the embedding must retain relevance information. The retrieval score then has to extract that information, which is the role of the contrastive objective.

A.1 Proof of Theorem 1

Proof. We prove the two inequalities in the theorem separately.

Step 1: a sufficient target preserves relevance information. Because $Y = g(X)$ is deterministic, observing both (X, Y) is equivalent, for predicting R, to observing X alone. Formally,

$$H(R \mid X, Y) = H(R \mid X). \tag{13}$$

Therefore adding Y to X does not change the mutual information between X and R:

$$I(R; X) = I(R; X, Y). \tag{14}$$

Applying the chain rule to the right-hand side gives

$$I(R; X, Y) = I(R; Y) + I(R; X \mid Y). \tag{15}$$

By δ-sufficiency, $I(R; X \mid Y) \le \delta$. Combining this with Eqs. (14)–(15),

$$I(R; Y) \ge I(R; X) - \delta. \tag{16}$$

Thus Y carries all but at most δ of the relevance information in X.

Step 2: reconstructability transfers target information to the embedding. We next lower-bound $I(R; E)$ in terms of $I(R; Y)$ and the reconstruction uncertainty $H(Y \mid E)$. Apply the chain rule to the joint variable (Y, E) in two orders:

$$I(R; Y) + I(R; E \mid Y) = I(R; E) + I(R; Y \mid E). \tag{17}$$

Since conditional mutual information is nonnegative,

$$I(R; Y) \le I(R; E) + I(R; Y \mid E). \tag{18}$$

The final term is bounded by the conditional entropy of the target:

$$I(R; Y \mid E) \le H(Y \mid E) \le \epsilon, \tag{19}$$

where the last inequality is ϵ-reconstructability. This step uses no assumption about how E is trained; it only says that if Y remains uncertain after observing E, then at most that much relevance information can fail to transfer from Y to E. Combining Eqs. (16)–(19) gives

$$I(R; E) \ge I(R; X) - \delta - \epsilon. \tag{20}$$

Step 3: the retrieval score extracts information from the embedding. The first bound concerns information present in the embedding pair. Retrieval, however, ranks documents using the scalar score $S = s(E_q, E_d)$. Since S is deterministic given E, adding S to E does not add information:

$$I(R; E, S) = I(R; E). \tag{21}$$

Applying the chain rule,

$$I(R;E) = I(R;S) + I(R;E \mid S). \tag{22}$$

With $\eta = I(R;E \mid S)$,

$$I(R;S) = I(R;E) - \eta \geq I(R;X) - \delta - \epsilon - \eta. \tag{23}$$

□

A.2 Cross-Entropy Upper Bound

For completeness, we justify Eq. (7) and connect it to the implemented loss. Let $p(y \mid e)$ be the true conditional distribution of targets given embeddings and $p_\theta(y \mid e)$ the decoder distribution. For a fixed embedding value e, the cross-entropy between the true conditional target distribution and the decoder distribution decomposes as

$$H(p, p_\theta) = H(p) + \mathrm{KL}(p \| p_\theta) \geq H(p). \tag{24}$$

Substituting $p = p(\cdot \mid e)$ and then taking expectation over E gives

$$\mathbb{E}_{E,Y}[-\log p_\theta(Y \mid E)] \geq H(Y \mid E). \tag{25}$$

Thus the sequence-level reconstruction cross-entropy upper-bounds the conditional entropy term ϵ used in Theorem 1.

In the model, the target $Y = (Y_1, \ldots, Y_T)$ is decoded autoregressively from the latent representation. The loss in Eq. (4) is written as a token average:

$$\mathcal{L}_{\text{aux}} = -\frac{1}{T} \sum_{t=1}^{T} \log p_\theta(Y_t \mid E, Y_{<t}). \tag{26}$$

Multiplying this average by T gives the sequence-level cross-entropy. Therefore, lower token-averaged reconstruction loss corresponds to a lower upper bound on $H(Y \mid E)$ for fixed target length, and to a length-normalized version of the same bound when target lengths vary.

Interpretation. The theorem uses ϵ as an abstract conditional-entropy quantity, while training observes cross-entropy under a parameterized decoder. The KL term above is the approximation cost of the decoder family. If the decoder is expressive enough and optimization lowers cross-entropy, the upper bound on $H(Y \mid E)$ tightens; if the decoder is weak or poorly optimized, cross-entropy may stay high even when the target is well chosen. This is why the theorem separates target quality (δ) from reconstructability (ϵ).

A.3 Proof of Corollary 2

Proof. Theorem 1 gives the lower bound

$$I(R;E) \geq I(R;X) - \delta(Y) - \epsilon(Y,E). \tag{27}$$

For Y_a this lower bound is

$$I(R;X) - \delta(Y_a) - \epsilon(Y_a, E_a), \tag{28}$$

and for Y_b it is

$$I(R;X) - \delta(Y_b) - \epsilon(Y_b, E_b). \tag{29}$$

The term $I(R;X)$ is the same in both comparisons because the underlying retrieval problem is the same. The condition in Eq. (12) therefore makes the first lower bound strictly larger than the second. □

The corollary is not a claim that the shorter target always wins. A short target can be easy to reconstruct but omit the information that decides relevance, increasing δ. A verbatim target can preserve relevance information but be too expensive to reconstruct through a small bottleneck, increasing ϵ. The useful target is the one that minimizes the sum of the two terms for the retrieval task.

B Failure Analysis Subtypes

This appendix gives the full audit protocol behind Table 1(b). We analyze all complete retrieval failures, defined as queries with nDCG@10 equal to zero, from the recon-eval output of the checkpoint used in the main analysis. For calibration, we also sample up to 100 high-scoring controls per task with nDCG@10 at least 0.5. The judge is Qwen/Qwen3.5-9B served through vLLM; judging uses the original query, decoded query reconstruction, labeled positive documents, decoded positive-document reconstructions, and top retrieved documents with their decoded reconstructions. The judge sees at most five retrieved documents and five labeled positives per query. The run used thinking mode with a 2048-token thinking budget, an 8192-token final-answer budget, and a 5888-token fitted prompt budget; malformed individual judge outputs are recorded as judge errors rather than dropping the task. The final run produced 851 failure judgments and 400 high-scoring control judgments, with no remaining parse or API errors in the saved summaries. We reran the failure split with GPT-4o and Llama-3.3-70B under the identical prompt and rubric. Table 9 reports substantial agreement among the three judges.

Failure-analysis judge prompt

```
You are a retrieval information-loss
analyst for MTEB recon-eval outputs.
Task: {task}. Record outcome:
{failure or high-scoring control}.
```

Table 9: Agreement on primary failure categories under an identical prompt and rubric.

Judges	Failures κ
Qwen3.5-9B vs. GPT-4o	0.72
Qwen3.5-9B vs. Llama-3.3-70B	0.66
GPT-4o vs. Llama-3.3-70B	0.75
All three (Fleiss' κ)	0.71

```
Task family, matching signal, query
reconstruction check,
positive-document check, and
label-noise check are filled with the
task-specific guidance from Table 11.

Choose exactly one primary category:
Query, Positive, Collapse, Label,
Input, Rank, Successful Retrieval, or
Unknown.
Choose exactly one subtype from the
task-specific subtype list in
Table 12.  Use Other only when no
named subtype fits, and explain the
rare pattern in tail_detail.

Rules:  use labeled positives as the
anchor for intended relevance; use
reconstruction-error categories only
when visible text drops, changes,
contradicts, or hallucinates
retrieval-critical information; do
not call a term dropped if it is
still present; do not blame
positive-document reconstruction for
harmless summarization; use Label
only when unlabeled retrieved
documents satisfy the task-specific
relevance relation at least as well
as labeled positives; prefer Rank
when reconstructions preserve the
main retrieval signals but ranking
still misses positives; treat
[TRUNCATED] and repetition markers as
analyzer clipping or malformed-text
markers rather than unseen evidence.

Return only valid JSON with fields:
primary_category, query_recon_error,
positive_doc_recon_error,
retrieved_doc_recon_error,
label_noise, failure_subtype,
subtype_location, subtype_evidence,
tail_detail, failure_reason,
evidence, missing_query_info,
positive_requirements, and action.
```

The user payload contains the task, split, query id, nDCG key/value, original query text, query text used for evaluation, decoded query reconstruction, labeled positive documents, and top retrieved documents. For each document the judge sees rank, document id, score, relevance flag, title, original text preview, and decoded reconstruction after prompt-budget fitting. Table 13 reports the resulting task-specific subtype rates used to interpret complete retrieval failures. Rates are computed within the set of queries with nDCG@10 equal to zero. The subtype labels are intentionally task-specific: for example, *Evidence* and *Stance* are meaningful for counterargument retrieval, while *Bridge* and *Attribute* are meaningful for multi-hop evidence retrieval.

We also audit high-scoring controls, defined as queries with nDCG@10 at least 0.5. These controls are not a matched causal analysis, but they help calibrate the interpretation of visible reconstruction errors with enough examples from every dataset. As Table 14 shows, visible reconstruction defects can occur even when retrieval is substantially successful. Thus, subtype labels should be read as explanations of complete failures, not as sufficient conditions for failure across all queries.

C Significance Tests

For the Qwen3-0.6B causal comparison in Table 3, we compute per-query nDCG@10 differences on the same evaluation queries and use a paired bootstrap with 10,000 resamples to estimate the mean gain and its 95% confidence interval. We correct the resulting dataset-level p-values with Benjamini–Hochberg. Table 15 shows positive gains on all eight datasets, seven of which remain significant after correction.

Across all eight model–attention configurations in Table 3, RECOVIR improves 59 of 64 dataset cells. A one-sided sign test gives $p = 4.5 \times 10^{-13}$; all eight per-configuration averages also improve, giving $p = 0.0039$.

D Task-Specific Reconstruction Prompts

A key design choice in RECOVIR is using *task-specific* reconstruction prompts rather than a single generic extraction template. Each prompt is designed to force the latent bottleneck to preserve the information most relevant for that task's retrieval signal. Table 16 summarises the dataset-to-prompt mapping; the full prompt text follows.

Design rationale. The prompt design follows a key principle: *the reconstruction target should encode the retrieval signal, not the surface form*. For example:

- **Multi-hop** (HotpotQA): The document prompt produces a natural-language summary preserving entity names and relationships — the bridge entities needed for multi-hop reasoning chains. The

Table 10: Primary categories used by the failure-analysis judge. The main paper aggregates Query, Positive, and Collapse into the Recon column.

Category	Meaning
Query	The decoded query embedding drops, changes, or hallucinates retrieval-critical information from the original query.
Positive	The labeled positive document contains the needed signal, but its decoded reconstruction drops or corrupts it.
Collapse	Retrieved wrong documents and positives look too similar in reconstruction, suggesting bottleneck collapse or indistinguishable hard negatives.
Rank	Query and positive reconstructions appear faithful, but nearest-neighbor ranking still prefers hard negatives.
Label	Retrieved unlabeled documents appear genuinely relevant, or benchmark labels are sparse/questionable.
Input	Inputs or reconstructions are empty, truncated, repetitive, malformed, or otherwise unusable.
Unknown	The visible evidence is insufficient to assign a cause.

Table 11: Task-specific relevance guidance supplied to the judge. This prevents the judge from treating all retrieval datasets as keyword matching.

Dataset	Matching signal	Query reconstruction check	Label-noise check
ArguAna	Retrieve the best counterargument: same debate topic, opposing or corrective stance, and same rebuttal target.	Preserve topic, stance, key claims, evidence, and vulnerable angle.	Use label noise only when the positive is not a real counterargument, has the same stance, or no visible positive is plausible.
SCIDOCS	Retrieve cited scientific paper abstracts, not merely same-title-keyword papers.	Preserve field, problem, method, contribution, and named technical entities.	A retrieved paper is label noise only if it plausibly shares citation/recommendation relevance, not just topical overlap.
FiQA2018	Retrieve user replies that answer the financial question with the right entity, concept, condition, and reasoning direction.	Preserve financial entities, quantities, comparison direction, and causal or opinion intent.	A retrieved answer must directly provide the same decision-relevant advice, not merely finance vocabulary.
HotpotQA	Retrieve documents that provide a bridge entity or supporting fact needed for multi-hop answering.	Preserve both hops, bridge entities, relation type, and requested answer constraint.	Label noise requires an unlabeled retrieved document to provide a required bridge fact or answer-supporting hop.

query prompt decomposes the multi-hop question into a sentence incorporating specific entities from the evidence passage.

- **Argument** (ArguAna): The document prompt captures the position and supporting evidence as direct prose, while the query prompt identifies the specific rhetorical angle the passage responds to. This forces the bottleneck to preserve *stance polarity*, not just topic overlap.
- **Biomedical** (BioASQ, PubMedQA): The document prompt extracts Population, Intervention, Outcome, and finding direction — the structured elements that distinguish two abstracts about the same drug but different patient populations.
- **Financial** (FiQA): The document prompt captures the causal logic ("DJIA is price-weighted, so high-priced stocks would dominate") rather than just the entities mentioned.
- **Duplicate question**: The prompt strips forum noise (code, HTML, verbose context) to extract the canonical problem intent, forcing surface-invariant embeddings.

Below we provide the full system prompts for each task type. All prompts instruct the teacher to output *only* the extraction result (JSON or NL) with no reasoning or meta-commentary.

Generic retrieval document prompt

```
Used for: MS MARCO, Natural
Questions, SQuAD, TriviaQA, ELI5,
S2ORC title-abstract, NFCorpus.

You are an information extraction
assistant. Given a document passage,
extract the specific details that
must be preserved for accurate
retrieval. Output valid JSON with
exactly these fields:

{
  "topic": "<main topic/domain in
```

Table 12: Subtype labels and descriptions. Shared labels apply to every dataset; task-specific labels are only used where meaningful.

Scope	Label	Description
Shared	OK	No visible retrieval-critical reconstruction loss; used mainly for high-scoring controls.
Shared	Input	Inputs or reconstructions are empty, clipped, malformed, or pathological.
Shared	Label	Benchmark label appears sparse, ambiguous, or questionable under the task relevance relation.
Shared	Rank	Right information is preserved, but a plausible hard negative wins the ranking.
Shared	Other	Rare or mixed tail pattern not covered by the named subtype set.
ArguAna	Topic	Debate topic, policy, or domain is lost or replaced.
ArguAna	Stance	Pro/con stance or claim polarity flips.
ArguAna	Target	Specific vulnerable claim a counterargument should attack is omitted.
ArguAna	Evidence	Statistics, examples, legal references, or named evidence needed for the rebuttal are corrupted.
ArguAna	Broad	Reconstruction keeps only a broad moral/topic summary and loses the counterargument angle.
SCIDOCS	Method	Named methods, models, algorithms, or technical terms are corrupted.
SCIDOCS	Task	Research task, problem setting, or application domain is lost.
SCIDOCS	Lineage	Cited-work relation, intellectual lineage, or foundational method signal is lost.
SCIDOCS	Setting	Datasets, metrics, modalities, or experimental settings are dropped or changed.
SCIDOCS	Acronym	Acronyms, title tokens, or morphology are garbled into another technical phrase.
SCIDOCS	Broad	Reconstruction preserves only broad field words and loses citation-level specificity.
FiQA2018	Number	Money amounts, rates, limits, thresholds, dates, or increments are wrong.
FiQA2018	Entity	Financial entities, accounts, securities, instruments, institutions, or products are changed.
FiQA2018	Condition	Tax, legal, timing, jurisdiction, insurance, eligibility, or scenario constraints are lost.
FiQA2018	Role	Who owes, owns, pays, receives, is liable, or is the customer/dependent changes.
FiQA2018	Direction	Preferred option, comparison direction, or causal/decision polarity flips.
FiQA2018	Concept	Finance concept such as FDIC, delta, tick size, margin, premium, or diversification drifts to another concept.
HotpotQA	Entity	Named entities or aliases are misspelled, swapped, or hallucinated.
HotpotQA	Bridge	Bridge entity or supporting fact needed to connect the hops is missing.
HotpotQA	Relation	Relation between entities or hop direction is reversed or changed.
HotpotQA	Attribute	Dates, counts, rankings, locations, occupations, or other attributes are wrong.
HotpotQA	Constraint	Comparison, superlative, shared-property, or requested answer type is lost.

```
3-8 words>",
  "entities": ["<named entity 1>",
...],
  "facts": ["<fact 1>", "<fact 2>",
...],
  "numbers":
["<number/date/measurement 1>", ...],
  "ideal_query": ["<hypothetical
search query 1>", ...],
  "description": "<1-2 sentence
summary preserving entities and key
facts>"
}

Rules:
- "topic": core subject in 3-8
words.
- "entities": proper nouns only
(people, places, orgs, products).
Max 8.
- "facts": core claims, each < 15
words. Max 3.
- "numbers": exact values (dates,
percentages, measurements). Max 6.
- "ideal_query": 1-3 realistic
search queries a user would type.
- "description": natural-language
summary preserving entity names and
key facts.
```

Generic retrieval query prompt

```
Given a search query and its
gold-standard relevant passage,
restate the query incorporating
specific details from the passage
that clarify what makes this passage
the right match. The restated query
should ground the original question
in the passage's specific entities,
facts, and terminology.

Output ONLY: {"generated_query":
"<your restated query>"}
```

Table 13: Subtype breakdown of complete retrieval failures. Percentages are over failed queries for each dataset. "Other" denotes rare or mixed patterns that did not fit the named subtype set.

Dataset	Failures	Subtype rates
ArguAna	230	Evidence 32.6%; Topic 18.7%; Other 14.3%; Stance 11.3%; Broad 8.7%; Target 4.3%; Rank 4.3%; Input 3.9%; Label 1.7%
SCIDOCS	362	Label 26.2%; Acronym 22.9%; Rank 21.0%; Other 11.9%; Method 7.7%; Broad 7.2%; Input 1.7%; Lineage 0.8%; Task 0.6%
FiQA2018	213	Concept 23.0%; Rank 22.5%; Other 20.7%; Label 11.3%; Entity 8.5%; Input 7.0%; Number 4.2%; Condition 1.4%; Direction 1.4%
HotpotQA	46	Entity 37.0%; Bridge 21.7%; Attribute 17.4%; Other 6.5%; Rank 6.5%; Constraint 4.3%; Input 4.3%; Relation 2.2%

Table 14: Subtype audit for high-scoring controls (nDCG@10 ≥ 0.5). Percentages are over 100 sampled controls for each dataset.

Dataset	Controls	Subtype rates
ArguAna	100	OK 29.0%; Evidence 24.0%; Broad 20.0%; Topic 10.0%; Other 6.0%; Stance 5.0%; Rank 4.0%; Input 2.0%
SCIDOCS	100	OK 29.0%; Acronym 22.0%; Method 21.0%; Broad 11.0%; Other 10.0%; Label 4.0%; Task 2.0%; Rank 1.0%
FiQA2018	100	OK 60.0%; Other 13.0%; Concept 12.0%; Entity 7.0%; Direction 3.0%; Input 2.0%; Number 2.0%; Condition 1.0%
HotpotQA	100	OK 36.0%; Entity 33.0%; Bridge 13.0%; Attribute 11.0%; Constraint 3.0%; Input 3.0%; Other 1.0%

HotpotQA document prompt

```
Given a passage, rewrite its key
facts as a concise 1-3 sentence
summary.

Rules:
- Name the main subject and key
entities by exact name.
- State how entities relate to each
other.
- Include dates, measurements, and
specific facts.
- NEVER write "The passage describes"
or any meta-commentary.
```

HotpotQA query prompt

```
Given a multi-hop question and one of
its gold evidence passages, rewrite
the question so it incorporates
specific entities from the evidence
and phrases the unanswered part as a
natural follow-up.

Rules:
- Incorporate entity names, dates,
and relationships from the evidence.
- End with the still-unanswered part
as a natural question or clause.
- ONE sentence, plain prose.
- NEVER start with "The evidence" or
any phrase referring to the evidence
as an object.
```

FEVER/SciFact document prompt

```
Given a passage (typically Wikipedia
or a scientific abstract), extract
verifiable claims.  Output JSON:

{
  "subject":  "<main entity or
topic>",
  "claims":  ["<verifiable claim 1>",
...],
  "entities":  ["<entity1>", ...],
  "qualifiers":
["<temporal/spatial/numerical
qualifier>", ...],
  "description":  "<1-2 sentence
summary preserving exact claims>"
}

Rules:
- "claims":  explicit, verifiable
factual assertions.  Prioritize
claims with dates, numbers,
measurements, named entities.
Preserve directionality of findings.
Max 5.
- "qualifiers":  dates, numbers,
locations, conditions that
distinguish this fact.  Max 5.
```

FEVER/SciFact query prompt

```
Given a claim to verify and its gold
evidence passage, rewrite the claim
by merging it with specific facts
from the evidence.  State the facts
directly, as a corrected or enriched
version of the claim.  Do NOT use
framing phrases like "Claim that" or
```

Table 15: Paired-bootstrap significance for Qwen3-0.6B with causal attention. Scores and confidence intervals use the $[0, 1]$ nDCG@10 scale; p-values are Benjamini–Hochberg corrected.

Dataset	RECOVIR	Base	Gain (95% CI)	p (BH)
ArguAna	0.594	0.570	+0.024 $[+0.022, +0.028]$	1.0×10^{-4}
CQA-G	0.603	0.586	+0.017 $[+0.009, +0.025]$	5.0×10^{-5}
CQA-U	0.474	0.467	+0.007 $[+0.004, +0.012]$	8.0×10^{-4}
ClimateFEVER	0.431	0.418	+0.013 $[+0.007, +0.019]$	5.0×10^{-5}
FEVER	0.906	0.903	+0.003 $[+0.001, +0.005]$	3.8×10^{-3}
FiQA	0.405	0.386	+0.019 $[+0.010, +0.024]$	4.0×10^{-7}
HotpotQA	0.728	0.698	+0.030 $[+0.016, +0.044]$	5.0×10^{-5}
SCIDOCS	0.226	0.216	+0.010 $[-0.006, +0.016]$	0.075

Table 16: Dataset-to-prompt mapping. Each prompt type targets the retrieval signal required by that task family. "Doc" prompts generate reconstruction targets for passages; "Query" prompts generate expanded query targets from (query, gold passage) pairs.

Prompt type	**Datasets**	**Doc format**	**Query format**
Generic retrieval	MS MARCO, NQ, SQuAD, TriviaQA, ELI5, S2ORC title–abstract, NFCorpus	JSON	JSON
Multi-hop	HotpotQA	NL	NL
Fact verification	FEVER, SciFact	JSON	JSON
Argument	ArguAna	NL	NL
Duplicate question	StackExchange Dup, StackOverflow Dup	NL	NL
Citation	SPECTER, S2ORC title citation, S2ORC abstract citation	NL	NL
Biomedical	BioASQ, PubMedQA	JSON	JSON
Financial	FiQA	JSON	JSON

```
"according to evidence".

Output ONLY: {"generated_query":
"<your rewritten claim>"}
```

ArguAna document prompt

```
Given an argumentative passage,
rewrite its core claim and supporting
evidence as a concise 1-3 sentence
summary.

Rules:
- Write in DIRECT voice -- state the
position and evidence as facts, not
as observations about the passage.
- NEVER start with "The passage
argues" or any meta-commentary.
- Preserve specific examples,
statistics, or references.
- IGNORE citation markers, debate
labels, HTML tags.
```

ArguAna query prompt

```
Given a debatable question and an
argumentative passage that addresses
it, write a single sentence that
names the topic and the specific
angle the passage responds to.

Rules:
- Write in DIRECT voice.
- NEVER start with "The passage
addresses" or any meta-commentary.
- Mention key terms or examples from
the passage.
- If the question is vague, infer the
debate topic from the passage
content.
```

Duplicate-question document and query prompt

```
Given a forum post (which may contain
code, HTML, math, verbose context),
extract the minimal core question
being asked.

Rules:
- Output ONLY the bare question --
the shortest form that preserves the
intent.
- Strip all code, markup, noise, and
context.
- Keep only what someone would type
as a search query.
- Preserve key technical terms but
nothing else.
- Do NOT add interpretation or
details not in the original.
```

Citation document prompt

```
Given a paper title (and optionally
abstract), write a 1-2 sentence
summary of the research contribution
that captures what a citing paper
would reference.

Rules:
```

```
- Name the specific research field,
problem, and approach.
- Include key technical terms,
material/system names, and method
names.
- NEVER write "This paper studies" or
any meta-commentary.
- State the contribution directly.
```

Citation query prompt

```
Given a paper and one it cites, write
a single sentence capturing the
shared intellectual contribution --
the method, problem, or finding that
creates the citation link.

Rules:
- Name the specific shared element
(technique, system, dataset,
finding).
- ONE sentence, plain prose.
- NEVER start with "Both papers" or
any phrase referring to documents as
objects.
- State the shared research content
directly.
```

BioASQ/PubMedQA document prompt

```
Given a paper abstract or biomedical
passage, extract elements for
matching with research questions or
scientific claims.  Output JSON:

{
  "population":  "<who/what is
studied>",
  "intervention":  "<treatment, drug,
procedure, or variable>",
  "outcome":  "<what was measured or
found>",
  "entities":
["<gene/protein/drug/disease 1>",
...],
  "finding_direction":
"<positive/negative/neutral/mixed>",
  "description":  "<1-2 sentences
with specific entities and result
direction>"
}

Rules:
- "population":  precise group or
system.
- "intervention":  treatment or
independent variable.
- "outcome":  measured effect with
direction.
- "entities":  biomedical named
entities, standard nomenclature.  Max
8.
```

BioASQ/PubMedQA query prompt

```
Given a research question or
scientific claim AND its
gold-standard abstract, restate the
query incorporating specific
biomedical details -- population,
intervention, measurements, and
result direction.  Bridge the claim's
general language to the abstract's
experimental vocabulary.

Output ONLY: {"generated_query":
"<your restated query>"}
```

FiQA document prompt

```
Given a passage from a financial Q&A
forum, extract the financial
reasoning.  Output JSON:

{
  "concepts":  ["<financial
concept/mechanism 1>", ...],
  "entities":
["<company/ticker/index/regulation
1>", ...],
  "reasoning":  "<core logical chain:
why/how X leads to Y>",
  "opinion_direction":  "<bullish/
bearish/neutral/explanatory>",
  "description":  "<1-2 sentences
with specific concepts and entities>"
}

Rules:
- "concepts":  financial mechanisms
or principles (e.g., "price-weighted
index").  Max 6.
- "reasoning":  cause-and-effect
logic in ONE sentence -- the signal
that distinguishes this answer from
others about the same entities.
```

FiQA query prompt

```
Given a financial question and its
gold-standard answer, restate the
question incorporating the specific
financial reasoning and concepts from
the answer.  Capture the reasoning
chain -- the "why" -- not just the
topic.

Output ONLY: {"generated_query":
"<your restated question>"}
```

E Implementation Details

E.1 Training Datasets

We train on a curated composite of open-source retrieval datasets spanning diverse domains and task types. Table 17 reports per-dataset statistics. Hard negatives (7 per query) are mined using a frozen Qwen3-Embedding-0.6B model with margin-based filtering following F2LLM (Zhang et al., 2025).

Table 17: Training data composition. Datasets are grouped by reconstruction prompt type (see Appendix D).

Task type	Dataset	Samples
Generic retrieval	MS MARCO	365,503
	Natural Questions	97,209
	SQuAD	89,509
	TriviaQA	60,025
	ELI5	161,345
	S2ORC (title–abstract)	250,000
	NFCorpus	1,283
Multi-hop	HotpotQA	120,528
Fact verification	FEVER	106,605
	SciFact	859
Argument	ArguAna	22,848
Duplicate question	StackExchange Dup (P2P)	203,060
	StackExchange Dup (S2S)	183,559
	StackOverflow Dup	19,847
NLI	SNLI	54,585
	MNLI	112,075
Citation	SPECTER	24,717
	S2ORC (title citation)	132,879
	S2ORC (abstract citation)	231,587
Biomedical	BioASQ	125,248
	PubMedQA	60,227
Financial	FiQA	7,452
Forum QA	StackExchange	534,705
	StackExchange (part 2)	220,000
Total		**3,185,655**

E.2 Evaluation Instructions

Following F2LLM (Zhang et al., 2025), we prepend task-specific instructions to queries during evaluation. Table 18 lists the instructions used for each MTEB retrieval task.

Table 18: Task-specific query instructions used during MTEB retrieval evaluation.

Task	Instruction
ArguAna	Given a claim, find documents that refute the claim.
CQADupstack	Retrieve duplicate questions.
ClimateFEVER	Retrieve a Wikipedia passage to verify this climate claim.
FEVER	Retrieve a Wikipedia passage to verify this claim.
FiQA-2018	Retrieve a passage that answers this financial question.
HotpotQA	Retrieve Wikipedia passages to answer this multi-hop question.
SCIDOCS	Retrieve scientific papers cited by this paper.

E.3 Full Hyperparameters

E.4 Compute

All experiments are run on a single NVIDIA B200 GPU (80 GB HBM). Evaluation uses batch size 64 for validation retrieval loss and batch size 2048 for MTEB retrieval scoring.

Table 19: Full hyperparameter settings for RECOVIR training.

Hyperparameter	Value
Backbone model	Qwen3-1.7B / 0.6B, Llama-3.2-1B
LoRA rank r	64
LoRA α	32
LoRA dropout	0.05
LoRA target modules	q, k, v, o, gate, up, down
Number of latent tokens K	4
Query max length	128
Document max length	256
Number of hard negatives	7
Optimiser	AdamW
Learning rate	3×10^{-4}
Weight decay	0.01
Warmup steps	100
LR schedule	Linear decay
Batch size	64
Max training steps	5000
Reconstruction weight λ	0.5
Temperature τ	0.02
Training hardware	Single GPU
Gradient checkpointing	Yes
Flash attention	Yes
Liger fused kernels	Yes

E.5 CoIR Evaluation

CoIR is a comprehensive benchmark designed to evaluate code information retrieval models across diverse retrieval tasks and code domains (Li et al., 2025). We evaluate nine CoIR tasks to test a domain distinct from the MTEB text-retrieval benchmarks used in the main experiments. We use a Qwen-1.5B backbone and compare RECOVIR with a matched EOS-pooling contrastive baseline; both models are trained for 8k LoRA steps. The evaluation metric is nDCG@10, reported on a 0–100 scale in Table 5.

F Broader Impact

Interpretable embeddings can improve trust and auditability in retrieval-augmented systems used for high-stakes decisions (medical, legal, financial). By providing a textual explanation of what the model captured, RECOVIR helps practitioners inspect retrieval failures at the embedding level.

However, the decodability of embeddings means they should be treated with the same privacy protections as raw text (Morris et al., 2023). The K-token bottleneck provides a natural privacy–utility trade-off (smaller K discards more information), but we caution against treating lossy reconstruction as a privacy guarantee.

Efficient LoRA training with reconstruction on 0.5B–1.7B backbones can reduce compute costs and democratise access to retrieval models.