Internalizing Explicit Reasoning into Latent Space for Dense Retrieval

Jiajie Jin
Gaoling School of Artificial Intelligence, Renmin University of China
BeiJing, China
jinjiajie@ruc.edu.cn

Yanzhao Zhang
Mingxin Li
Dingkun Long
Pengjun Xie
dingkun.ldk@alibaba-inc.com
Alibaba Group
HangZhou, China

Yutao Zhu
Zhicheng Dou*
Gaoling School of Artificial Intelligence, Renmin University of China
BeiJing, China
dou@ruc.edu.cn

Abstract

Large Language Models (LLMs) have fundamentally transformed dense retrieval, upgrading backbones from discriminative encoders to generative architectures. However, a critical disconnect remains: while LLMs possess strong reasoning capabilities, current retrievers predominantly utilize them as static encoders, leaving their potential for complex reasoning unexplored. To address this, existing approaches typically adopt "rewrite-then-retrieve" pipelines to generate explicit Chain-of-Thought (CoT) rationales before retrieval. However, this incurs prohibitive latency. Conversely, implicit reasoning methods utilizing latent tokens offer efficiency but often suffer from semantic degeneration due to the lack of explicit supervision. In this paper, we propose **LaSER**, a novel self-distillation framework that internalizes explicit reasoning into the latent space of dense retrievers. Operating on a shared LLM backbone, LaSER introduces a dual-view training mechanism: an Explicit view that explicitly encodes ground-truth reasoning paths, and a Latent view that performs implicit latent thinking. To bridge the gap between these views, we design a multi-grained alignment strategy. Beyond standard output alignment, we introduce a *trajectory alignment* mechanism that synchronizes the intermediate latent states of the latent path with the semantic progression of the explicit reasoning segments. This allows the retriever to "think" silently and effectively without autoregressive text generation. Extensive experiments on both in-domain and out-of-domain reasoning-intensive benchmarks demonstrate that LaSER significantly outperforms state-of-the-art baselines. Furthermore, analyses across diverse backbones and model scales validate the robustness of our approach, confirming that our unified learning framework is essential for eliciting effective latent thinking. Our method successfully combines the reasoning depth of explicit CoT pipelines with the inference efficiency of standard dense retrievers[1].

*Corresponding author.

[1]The code, model, and training data are available at https://github.com/RUC-NLPIR/LaSER.

SIGIR '26, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-2599-9/2026/07
https://doi.org/10.1145/3805712.3809575

CCS Concepts

• **Information systems** → **Language models**.

Keywords

dense retrieval, latent reasoning, self-distillation, query rewrite

ACM Reference Format:
Jiajie Jin, Yanzhao Zhang, Mingxin Li, Dingkun Long, Pengjun Xie, Yutao Zhu, and Zhicheng Dou. 2026. Internalizing Explicit Reasoning into Latent Space for Dense Retrieval. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '26), July 20–24, 2026, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3805712.3809575

1 Introduction

Dense retrieval has evolved into a fundamental component of modern Information Retrieval (IR) systems, widely deployed in web search, question answering [25, 26, 56], and Retrieval-Augmented Generation (RAG) systems [13, 15, 24, 30]. These retrievers effectively bridge the vocabulary mismatch gap via semantic matching in a shared embedding space [26, 50, 54]. With the advent of Large Language Models (LLMs) [1, 34, 61, 63], the field has witnessed a critical paradigm shift: retriever backbones are transitioning from small-scale discriminative models (e.g., BERT, RoBERTa) to large-scale generative LLMs [36, 51]. State-of-the-art LLM-based retrievers [29, 54, 60] have demonstrated superior semantic understanding capabilities compared to their predecessors. While LLM-based retrievers have demonstrated superior semantic understanding capabilities [5, 51, 60], a paradox remains: while these retrievers inherit the massive knowledge and reasoning potential of the underlying LLMs, they are primarily trained via contrastive learning to optimize representational distinctiveness [20]. This training objective largely leaves the inherent reasoning capabilities of the generative architecture dormant, treating the LLM merely as a stronger encoder rather than a reasoner [33, 45, 47, 49].

This limitation becomes particularly evident when handling complex user queries. Real-world search scenarios often involve implicit intents [41, 62], multi-hop logic [18, 21, 56], or ambiguous descriptions [40, 53] that require reasoning beyond superficial semantic matching. To bridge this gap, a prevalent solution adopts a "rewrite-then-retrieve" pipeline [14, 55], utilizing external LLMs to explicitly generate query expansions or Chain-of-Thought (CoT) rationales before retrieval [2, 8, 48, 57, 59]. While effective in enhancing performance, their dependency on verbose autoregressive generation

creates prohibitive latency bottlenecks [16, 35] (as shown in Figure 1). To mitigate these latency issues, recent studies have explored *implicit reasoning* approaches, allowing retrievers to generate "latent thinking tokens" within the embedding space before producing the final representation [49]. While promising, these methods face a critical challenge: **the lack of explicit semantic supervision.** Relying solely on the final contrastive learning loss, the generated latent tokens often lack sufficient guidance to accurately capture the critical information required for retrieval, particularly for queries involving reasoning.

We argue that **the key to resolving this dilemma lies in internalization**: leveraging the explicit reasoning capabilities of LLMs as privileged supervision to guide the lightweight latent thinking of the retriever. Our core insight is that while generating explicit CoT during inference is inefficient, the semantics carried by these reasoning paths are invaluable. Instead of treating explicit and implicit reasoning as separate paradigms, we propose to align them within a unified framework, allowing the retriever to "think" silently in latent space. Driven by this insight, we propose a dual-view framework where the model learns to internalize explicit reasoning capabilities within latent space. The Explicit-View learns to encode the reasoning logic into embedding, serving as a semantic upper bound. The **Latent-View** generates latent thinking tokens to simulate the reasoning process internally.

To effectively transfer reasoning capabilities from the explicit view to the latent view, we introduce a multi-grained alignment strategy. We align not only the final query representations (output alignment) but also apply auxiliary supervision to intermediate latent states (trajectory alignment), encouraging the latent tokens to capture the semantic progression of the explicit CoT. This framework offers significant versatility. During training, the explicit view acts as a semantic teacher; during inference, LaSER operates in a streamlined mode using latent reasoning, eliminating the need for text generation while maintaining performance comparable to computationally intensive pipeline methods. We evaluate LaSER on a comprehensive suite of embedding benchmarks, focusing on reasoning-intensive tasks such as Bright [47], BrowseComp-Plus [7, 23], and FollowIR [53]. Empirical results demonstrate that our method significantly enhances retrieval performance by introducing only a few latent tokens during inference, avoiding the latency of autoregressive decoding. Furthermore, LaSER consistently outperforms existing explicit and implicit reasoning baselines. Notably, in certain reasoning-intensive scenarios, it even achieves superior performance compared to computation-heavy pipelines that rely on external LLMs for query rewriting, validating the effectiveness of our proposed distillation mechanism.

Our main contributions are summarized as follows:

- We propose LaSER, a unified retrieval framework that bridges the gap between explicit CoT reasoning and implicit latent thinking via self-distillation within a shared backbone.
- We design a dual-view alignment mechanism, featuring both output-level and process-level trajectory alignment, which effectively distills reasoning semantics into latent tokens and prevents the degeneration often observed in implicit reasoning models.
- Extensive experiments demonstrate that LaSER consistently outperforms baselines across diverse backbone architectures and model scales, robustly validating the efficacy of our proposed self-distillation mechanism.
- We empirically validate that the semantics of explicit reasoning paths can be effectively distilled into the latent space. This allows LaSER to achieve performance comparable to computationally intensive "rewrite-then-retrieve" pipelines while maintaining the inference efficiency of standard dense retrievers.

2 Related Work

2.1 LLM-Based Retrieval and Query Expansion

The paradigm of dense retrieval has shifted significantly from small-scale discriminative backbones (e.g., BERT [27]) to large-scale generative LLMs [51, 60]. By leveraging the massive world knowledge and semantic understanding inherent in LLMs, these retrievers achieve superior performance in standard semantic matching. However, they continue to rely primarily on contrastive learning objectives, which often struggle with complex queries requiring deep reasoning or intent disambiguation. To bridge this gap, the "rewrite-then-retrieve" pipeline has become prevalent (as shown in Figure 1), utilizing external LLMs to expand queries [14, 32, 52] or generate reasoning rationales prior to retrieval [4, 22, 44]. While effective, this approach decouples reasoning from retrieval, introducing significant latency due to autoregressive decoding and complicating system deployment. Unlike these multi-stage pipelines, our work aims to *internalize* the rewriting and reasoning capabilities directly into the retriever's parameters, preserving the inference efficiency of a single-stage dual-encoder while retaining the semantic benefits of generative expansion.

2.2 Reasoning in Retrieval

Recent research has explored integrating reasoning capabilities directly into retrieval systems, generally categorized into data-centric [9, 38, 54] and mechanism-centric approaches. Data-centric methods enhance retrieval by synthesizing reasoning-intensive queries with corresponding hard negatives. Mechanism-centric approaches, conversely, modify the architecture of retriever to support reasoning, further dividing into *explicit* and *implicit* strategies. Explicit methods leverage the retriever itself to generate natural language CoT rationales before producing the final embedding [8, 16, 31, 43, 48, 57]. While this approach improves interpretability, it complicates training by requiring the model to learn two tasks simultaneously [42]. Furthermore, it suffers from the same computational bottlenecks as external rewriting. Implicit reasoning, alternatively, simulates this process within the embedding space by generating "latent thinking tokens" [49] or multi-vector representations [3] that capture query nuances without visible text generation. Despite their efficiency, existing implicit methods predominantly rely on the final contrastive loss for supervision, which acts as a bottleneck for capturing complex query semantics and hinders further performance improvements. Our framework addresses this limitation by using explicit CoT as a privileged supervisor, distilling the semantics of explicit reasoning paths into latent tokens to ensure the implicit process is semantically meaningful.

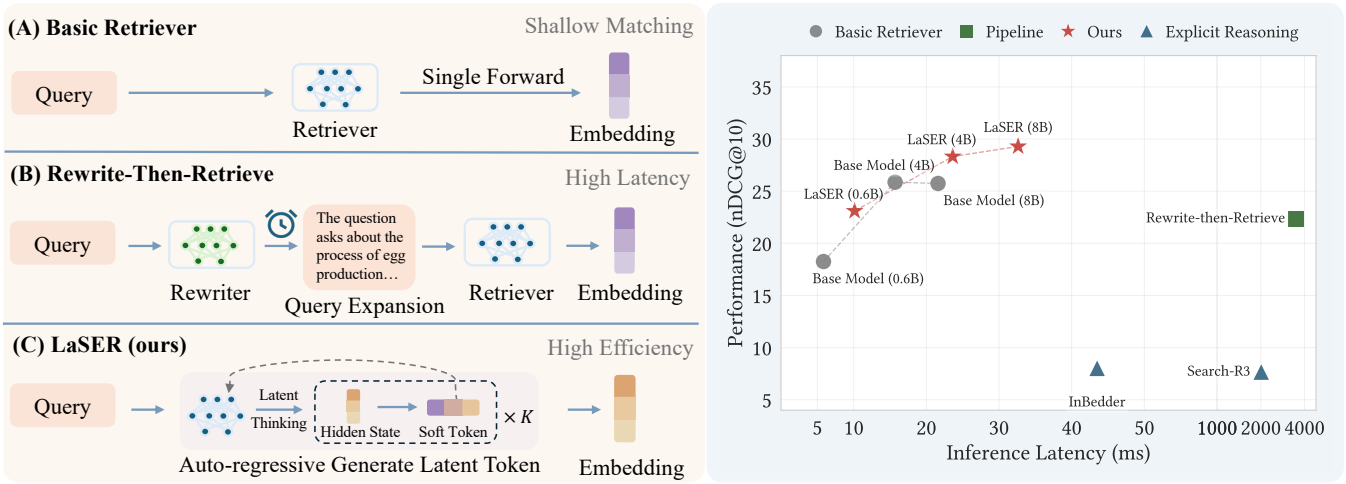

Figure 1: Left: Comparison between our proposed pipeline and conventional retriever and rewrite-then-retrieve method. Right: Comparison of performance and latency of different methods on the Bright benchmark. Latency represents the time consumption of processing a single query. The "Rewrite-then-Retrieve" method uses a 0.6B retriever and a 3B rewriter.

2.3 Knowledge Distillation in IR

Knowledge Distillation (KD) is widely used to transfer capabilities from powerful teachers to efficient students [39], generally fall into *output-level* and *feature-level* distillation in IR. Output-level distillation typically utilizes LLMs to generate synthetic training data, such as query expansions or pseudo-relevance labels [60], which the student mimics. Feature-level distillation focuses on aligning continuous representations, commonly training a student retriever to approximate the hidden space or relevance scores of a powerful teacher model [37, 58]. Critically, these methods are outcome-oriented: they improve relevance signals but do not fundamentally alter the student's information processing mechanism. Distinct from this paradigm, we draw inspiration from LLM latent reasoning [6, 17, 46] to internalize explicit reasoning steps into latent states [49]. Instead of merely distilling relevance scores, we leverage explicit Chain-of-Thought as "ground truth" logic to supervise the latent thinking. To our knowledge, LaSER the first work to apply explicit-to-latent reasoning distillation in dense retrieval.

3 Methodology

In this section, we present LaSER, a unified framework designed to internalize explicit reasoning capabilities into the latent space of dense retrievers. We first formulate the retrieval problem and give an overview of our framework. Then, we detail the training process and optimization objectives of our framework, highlighting our multi-grained self-distillation mechanism.

3.1 Problem Formulation

Given a user query q and a candidate document corpus $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$, the goal of dense retrieval is to learn an encoder $f(\cdot)$ that maps textual inputs into an m-dimensional embedding space $\mathbb{R}^m$. The relevance score between a query q and a document d is typically computed via the cosine similarity of their embeddings v_q and v_d, i.e., $s(q, d) = \cos(v_q, v_d)$.

In standard LLM-based retrievers, the query representation is obtained through a single forward pass. To enhance instruction following, a task-specific instruction I is often prepended to q. An [EOS] token is then appended to the input sequence, and the hidden state of the last layer corresponding to this token is extracted as the embedding:

$$v_q = f([I_{\text{task}}; q; [\text{EOS}]]). \tag{1}$$

However, this shallow processing often fails to capture the complex implicit intents required for reasoning-intensive queries.

To address this, existing "rewrite-then-retrieve" approaches introduce an external LLM reasoner $\mathcal{M}$ to generate an explicit reasoning path and the retriever then encodes this enriched context:

$$v_q^* = f([I_{\text{task}}; q; r_q; [\text{EOS}]]), \text{where } r_q = \mathcal{M}(I_{q2r}, q). \tag{2}$$

While v_q^* captures deeper semantics, the autoregressive generation of textual r_q incurs prohibitive latency and deployment costs.

In this work, we aim to bridge this gap by training a model to perform *latent reasoning*. Instead of generating explicit text r_q, the model autonomously generates a sequence of K continuous latent thinking tokens $T = \{t_1, \ldots, t_K\}$ within the embedding space. The final query representation v_q is derived from these latent states, aiming to approximate the semantic depth of v_q^* while maintaining the efficiency of a single-model architecture.

3.2 Overview of Framework

As illustrated in Figure 2, LaSER is built upon a unified generative LLM backbone with causal attention. To achieve the internalization of reasoning capabilities, we employ a dual-view training paradigm where the model learns through two aligned perspectives sharing the same parameters θ:

Explicit-View. This view acts as the semantic upper bound during training. It receives the query augmented with a high-quality, explicit CoT rationale generated by a superior reasoner. By performing a single forward pass on this enriched context, the explicit-view

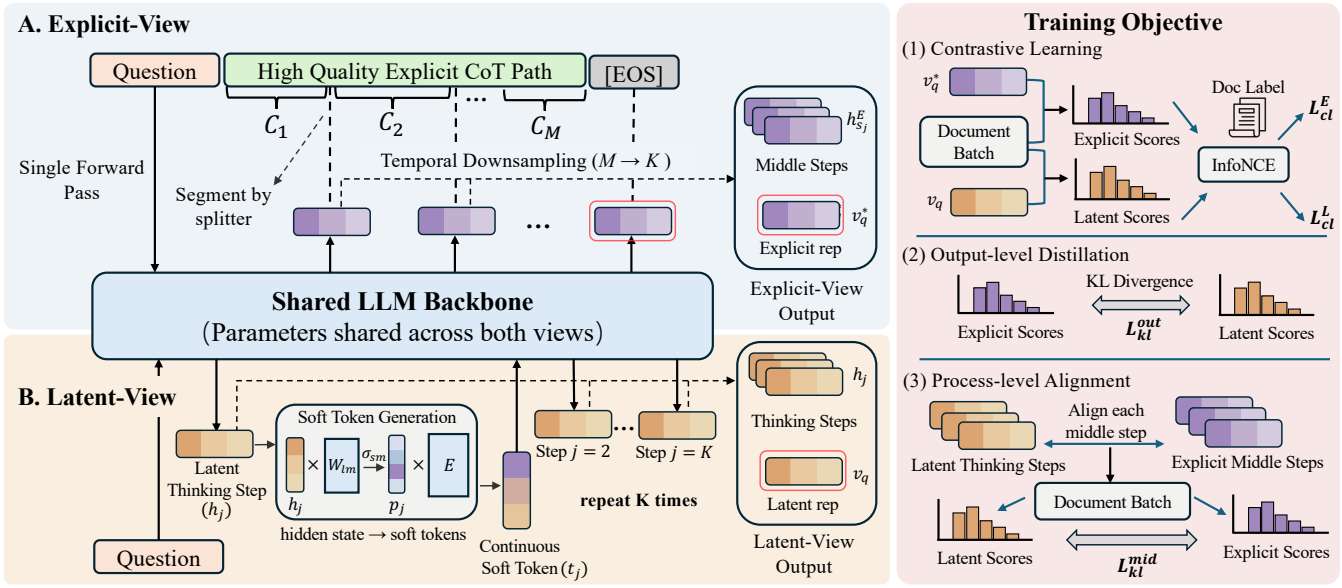

Figure 2: Illustration of the LaSER. During training, each step involves two inference pathways: the explicit-view and the latent-view (shown on the left). The inputs for these two paths differ, with the explicit-view incorporating additional information. However, during the inference phase, only the latent-view mode is utilized. The loss function is computed based on the results from both paths (shown on the right).

acquires a representation that comprehensively synthesizes the original query and the explicit reasoning path.

Latent-View. This view serves as the target mode for inference. It takes only the original query as input, and generate a sequence of *latent thinking tokens* in the latent embedding space before producing the final query representation. We design a self-distillation framework to facilitate the co-training of these two views, especially the latent-view, as detailed in Section 3.5.

Inference. During inference, LaSER operates exclusively in the ***latent-view***. It utilizes the learned latent thinking tokens to encode queries, ensuring the latency matches standard dense retrievers while benefiting from internalized reasoning. Optionally, the model retains the flexibility to accept rewritten queries from external LLMs if available, further adapting to diverse deployment scenarios.

3.3 Latent View

The primary objective of the Latent View is to enhance the model's reasoning capacity for complex queries while circumventing the latency overhead associated with explicit text generation. Inspired by recent advances in continuous reasoning, we propose a mechanism where the encoder model, parameterized by θ, performs "thinking" within a continuous embedding space. This approach replaces discrete token generation with an autoregressive sequence of continuous thought vectors, allowing the model to refine its understanding of the query intent in a fully differentiable manner.

As shown in Figure 2, given an input query q, the model first maps the discrete token sequence into dense embeddings $\mathbf{X}_0 = \mathbf{E}(q)$, where $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times m}$ is the embedding matrix and $|\mathcal{V}|$ is the vocabulary size. The backbone LLM f_θ processes this sequence to produce initial hidden states.

To facilitate latent reasoning, we extend the query with a sequence of K latent thinking tokens $T = \{t_1, \ldots, t_K\}$. Unlike standard decoding which samples discrete indices, we construct these tokens as continuous vectors to preserve semantic richness and differentiability. The process proceeds autoregressively: at the j-th thinking step ($1 \le j \le K$), let $h_{j-1} \in \mathbb{R}^m$ be the last hidden state output by the backbone. We project h_{j-1} to the vocabulary space via the language modeling head $W_{\text{lm}} \in \mathbb{R}^{m \times |\mathcal{V}|}$ to obtain logits l_j:

$$l_j = W_{\text{lm}} h_{j-1}. \tag{3}$$

Rather than performing a hard selection (e.g., arg max), we compute a probability distribution $p_j = \text{Softmax}(l_j)$ over the vocabulary. The soft latent token t_j is then computed as the expected embedding vector under this distribution:

$$t_j = p_j^\top \mathbf{E} = \left(\text{Softmax}(W_{\text{lm}} h_{j-1})\right)^\top \mathbf{E}. \tag{4}$$

This soft token t_j is appended to the input sequence for the next step. The model operates autoregressively, where the input at step j includes the original query and all preceding latent tokens:

$$\mathbf{H}_j = f_\theta\left([\mathbf{E}(q), t_1, \ldots, t_j]\right), \tag{5}$$

where $\mathbf{H}_j$ represents the sequence of hidden states. Through the causal attention mechanism, each latent token t_j can attend to the query context and previous thinking steps, mimicking the sequential logic of explicit Chain-of-Thought reasoning.

After K steps, we obtain a sequence of reasoning states $\mathcal{H}_{\text{think}} = \{h_1, \ldots, h_K\}$. To synthesize the global semantics of this reasoning process into a unified representation, we apply mean-pooling over

these states to derive the final query vector v_q:

$$v_q = \frac{1}{K}\sum_{j=1}^{K} h_j. \tag{6}$$

The representation of documents v_d is obtained via a standard single forward pass with the last hidden state as the embedding (*without* latent thinking tokens), following standard dense retriever practice. This design avoids additional indexing overhead, as latent reasoning is applied only on the query side. This latent thinking process maintains strict autoregressiveness, ensuring full compatibility with KV caching optimizations: at each step j, we cache the key-value states from all preceding tokens so that only the newly appended latent token t_j requires a single incremental forward pass rather than recomputing the full prefix. Furthermore, by limiting the reasoning horizon K (typically $K < 10$), the method achieves inference latency comparable to standard dense retrievers, significantly outperforming generative approaches that require decoding long textual chains.

3.4 Explicit-View

The Explicit-View serves as the semantic teacher, learning explicit reasoning processes through privileged knowledge derived from an advanced external reasoner. Specifically, for each query q, we prompt the external reasoner to generate a comprehensive CoT rationale r_q, which includes the intermediate reasoning logic and understanding of the potential intent behind the query (refer to Table 1 for the detailed prompt). We construct the augmented input sequence x_{aug} by concatenating the original query q, the generated rationale r_q, and [EOS] token:

$$x_{\text{exp}} = [q; r_q; [\text{EOS}]]. \tag{7}$$

The model then encodes this augmented input as in Equation 2. As the inputs of the explicit-view and latent-view are different, there is no mutual interference between the two tasks. Since the reasoning logic is explicitly included in the textual input, the encoder only performs a single forward pass and gets the last hidden state as the final representation v_q^*, as shown in Equation (2). Note that both views share the same backbone parameters, and gradients from the explicit loss $\mathcal{L}_{cl}^E$ actively update the shared parameters simultaneously with the latent view losses, enabling mutual adaptation (see Section 5.6 for empirical validation). Given that the explicit path itself can provide intermediate signals, we additionally extract the hidden states of certain intermediate tokens in the sequence to represent the intermediate reasoning process (as shown in Figure 2). These details will be discussed in Section 3.5.3.

3.5 Optimization via Self-Distillation

To transfer the reasoning capability from the Explicit-View to the Latent-View, we propose a joint training objective combining contrastive learning and multi-grained distillation.

3.5.1 Contrastive Training. Both two views must learn basic discriminative signals. We employ the standard InfoNCE loss [20] for both views using ground-truth positive documents d^+ and negatives d^-. For the latent-view:

$$\mathcal{L}_{cl}^L = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(v_q \cdot v_{d^+}/\tau)}{\sum_{d\in\{d^+\}\cup\{d^-\}}\exp(v_q \cdot v_d/\tau)}, \tag{8}$$

where N is the batch size, τ is the temperature hyperparameter. A similar loss $\mathcal{L}_{cl}^E$ is computed for the explicit view using v_q^*.

Table 1: The prompt used for annotating the explicit reasoning path of the given query.

Instruction:
Given a question, your mission is to follow the instructions below:
1. Identify the essential problem.
2. Think step by step to reason and describe what information could be relevant and helpful to address the questions in detail.
3. Draft an answer with as many thoughts as you have.
The given question:
`[Begin of Question]`
{Original Query}
`[End of Question]`

3.5.2 Output-Level Distillation. To ensure the Latent-View effectively internalizes the explicit reasoning capabilities, we distill the semantic knowledge from the Explicit-View to the Latent-View. Instead of strictly aligning the query embeddings v_q and v_q^*, which may over-constrain the latent view given the inherent information gap between the raw and reasoning-augmented inputs, we adopt a score-based distillation strategy. This approach focuses on transferring the ranking preferences encapsulated in the explicit view. Formally, for a query q and a document batch $\mathcal{B} = \{d_1, \ldots, d_N\}$, we align the probability distributions of relevance scores derived from both views.

Let $s_i^E = v_q^* \cdot v_{d_i}$ and $s_i^L = v_q \cdot v_{d_i}$ denote the similarity scores from the Teacher and Student, respectively. We apply a softmax function with a distillation temperature τ_{kd} to obtain the probability distributions:

$$p_i^E = \frac{\exp(s_i^E/\tau_{kd})}{\sum_{k=1}^{N}\exp(s_k^E/\tau_{kd})}, \quad p_i^L = \frac{\exp(s_i^L/\tau_{kd})}{\sum_{k=1}^{N}\exp(s_k^L/\tau_{kd})}. \tag{9}$$

The output-level distillation loss is then defined as the Kullback-Leibler (KL) divergence between these two distributions:

$$\mathcal{L}_{kl}^{out} = \frac{1}{N}\sum_{i=1}^{N} p_i^E \log\left(\frac{p_i^E}{p_i^L}\right). \tag{10}$$

By minimizing $\mathcal{L}_{kl}^{out}$, the latent view learns to mimic the fine-grained information of document relevance in explicit view, effectively capturing the reasoning-enhanced judgment logic without requiring identical embedding coordinates.

3.5.3 Process-Level Trajectory Alignment. Relying solely on output alignment often treats the reasoning process as a black box, which may lead to the degeneration of latent tokens where they fail to encode meaningful intermediate semantics. To prevent this, we introduce a *Trajectory Alignment* mechanism to explicitly supervise the intermediate thinking steps.

Our design operates on the hypothesis that the sequence of latent tokens should function as a compressed semantic trajectory of the verbose explicit CoT. While the explicit rationale r consists of

variable-length logical segments $\{s_1, ..., s_M\}$ (where M is the number of reasoning steps), the latent view operates with a fixed budget of K tokens. Specifically, we segment the explicit rationale r into M chunks by splitting on sentence boundaries (periods), resulting in roughly equal-length reasoning segments. Since typically $M \geq K$, we posit that each latent token t_i should correspond to a specific semantic keyframe within the explicit reasoning path. To realize this, we employ a **Temporal Downsampling** strategy to align the granularities of the two views. We map the i-th latent step to a corresponding explicit segment index j_i via uniform sampling:

$$j_i = \lfloor i \times \frac{M}{K} \rfloor. \tag{11}$$

This mapping ensures that the model synchronizes its latent thoughts with the semantic progression of the explicit reasoning chain. For each mapped segment s_{j_i}, we extract the hidden state of the last token in that segment from the Explicit-View's forward pass as the intermediate explicit representation $h^E_{j_i}$. We then align the intermediate states by minimizing the KL divergence between their projected distributions over the document batch:

$$\mathcal{L}^{mid}_{kl} = \frac{1}{K} \sum_{i=1}^{K} \mathrm{KL}\left(\sigma(h^E_{j_i} \cdot v_{\mathcal{D}}), \sigma(h_i \cdot v_{\mathcal{D}})\right), \tag{12}$$

where $v_{\mathcal{D}}$ denotes the stacked document embeddings from the current batch, and σ denotes the softmax function over the document batch. By enforcing this alignment, we ensure that the latent tokens do not drift but instead act as semantic checkpoints that rigorously approximate the explicit reasoning logic.

3.5.4 Overall Objective. The final training objective is a weighted sum of the components:

$$\mathcal{L} = \mathcal{L}^L_{cl} + \lambda_1 \mathcal{L}^E_{cl} + \lambda_2 \mathcal{L}^{out}_{kl} + \lambda_3 \mathcal{L}^{mid}_{kl}, \tag{13}$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters that used to balance the tasks.

4 Experiments Design

4.1 Datasets and Metrics

To improve the model's reasoning capabilities, we utilize the synthetic dataset from ReasonEmb [4] for training, which includes 81k training examples across 12 domains. Each training query is accompanied by a reasoning path, which is generated by GPT-4o-mini [19] as a reasoner, serving as the explicit input in our explicit-view.

For evaluation, we select three benchmarks requiring deep reasoning: Bright [47], FollowIR [53](out-of-domain) and BrowseComp-Plus [7](out-of-domain). Following standard practices, we report nDCG@10 for BRIGHT, and Recall@5, Recall@100 and Recall@1000 for BC-Plus. For FollowIR, the evaluation for each subset includes a standard metric alongside p-MRR (pairwise Mean Reciprocal Rank) [53], a pairwise metric that evaluates whether a retrieval model correctly re-ranks document pairs when their relevance ordering is altered by a change in the user instruction. p-MRR ranges from −1 to 1, where positive values indicate correct sensitivity to instruction changes and negative values suggest the model ranks documents contrary to the intended instruction. Adhering to official protocols, in addition to p-MRR, we report MAP@5 for the Robust'04 and Core'17 subsets, and nDCG@5 for the News'21 subset.

The statistics of both training datasets and evaluation datasets are shown in Table 2.

4.2 Baselines

We compare LaSER against four categories of baseline methods. To ensure a rigorous comparison, unless otherwise noted, all trainable baselines utilize the same backbone and training data as our method.

Table 2: Statistics of the datasets used in our experiments. "Q" and "C" denote Query and Corpus respectively. "Len." denotes average token count. "Reas. Len." denotes average reasoning length annotated by external LLM.

Split	Dataset	# Q	# C	Q Len.	Reas. Len.
Train	ReasonEmb	81,659	–	222.1	984.8
Test	Bright (ID)	1,384	1,145,164	240.8	2,412.8
	FollowIR (OOD)	104	98,312	76.6	–
	BC-Plus (OOD)	830	100,195	123.2	–

Standard Dense Retrievers: We include state-of-the-art encoder-based models such as BGE-M3 [5] and E5-Large-Instruct [50], as well as LLM-based retrievers including E5-Mistral [51] and Qwen3-Embedding [60]. Additionally, we train a "Fair Baseline" using standard contrastive learning on our composite dataset to isolate the gains attributed to our architecture.

Pipeline Methods (Rewrite-then-Retrieve): This category employs an external LLM to explicitly rewrite the query before retrieval. We utilize BGE-Reasoner-Rewriter-7B [4] to generate rewritten queries as input to the retriever (using the Fair Baseline model) during inference.

Explicit Reasoning Methods: These approaches integrate generation and retrieval within a single model, generating an explicit CoT prior to outputting the representation. We compare against Search-R3 [16] and GRACE [48]. As training codes for some baselines are not publicly available, we perform inference using their official checkpoints.

Implicit Reasoning Methods: We compare against GIRCSE, a representative method that also employs latent tokens to model reasoning steps before producing the final embedding.

4.3 Implementation Details

We conduct experiments using base version of Qwen3 series (0.6B, 4B, 8B) and LLaMA 3.2 series (1B, 3B, 8B). As LLaMA 3.2 does not offer an 8B variant, we utilize LLaMA 3.1-8B to evaluate performance at that scale. All models are fine-tuned for 1 epoch using LoRA ($r = 64, \alpha = 32$) on 4 A100 GPUs, with a global batch size of 8 (batch size per device set to 2 with 8 gradient accumulation steps). We use the AdamW optimizer with a learning rate of 1e-4 and a warmup ratio of 0.1. Each query is paired with one hard negative sample and in-batch negatives for training. The maximum sequence length for both query and documents is set to 512 during training and expanded to 8192 during testing to accommodate longer contexts. To accommodate the explicit reasoning paths processed by

Table 3: Main retrieval performance on the Bright benchmark. We report nDCG@10 for all subsets. The best results in the same backbone setting are formatted in bold and the second best are underlined. † indicates the pipeline method that uses an external LLM to rewrite queries during inference.

Models	Size	StackExchange							Coding		Theorem-based			Avg.
		Bio.	Earth.	Econ.	Psy.	Rob.	Stack.	Sus.	Leet.	Pony	AoPS	TheoQ.	TheoT.	
Standard Dense Retrievers														
BGE-M3	0.6B	7.2	13.3	12.5	12.8	12.6	10.0	10.1	15.6	30.7	1.5	5.6	5.0	11.4
Multilingual-E5-Large-Instruct	0.6B	15.0	24.6	14.0	14.6	16.1	10.3	13.8	15.1	2.1	3.0	11.6	6.9	12.3
E5-Mistral-7B-Instruct	7B	17.9	28.6	18.5	16.5	18.6	13.4	20.0	19.7	6.3	2.0	14.2	22.1	16.5
Qwen3-Embedding-0.6B	0.6B	12.7	26.3	17.9	16.5	12.5	12.4	12.2	14.3	0.7	3.1	17.2	26.5	14.4
Qwen3-Embedding-4B	4B	15.9	33.6	16.7	23.2	13.2	15.0	16.8	21.2	1.7	4.0	18.4	35.4	17.9
Qwen3-Embedding-8B	8B	14.7	17.9	15.5	19.9	9.1	12.9	16.5	17.4	0.8	2.5	16.8	24.5	14.0
Basic Contrastive Learning														
Fair Baseline (Qwen3-0.6B)	0.6B	28.8	31.6	25.2	28.1	15.1	22.3	24.2	8.8	**3.5**	**2.1**	14.1	15.4	18.3
Fair Baseline (Qwen3-8B)	8B	49.7	51.2	26.9	37.4	23.4	28.0	34.1	3.7	3.2	**2.8**	16.8	31.8	25.7
Fair Baseline (LLaMA3.1-8B)	8B	57.7	40.9	24.6	29.8	**20.1**	29.2	25.8	4.3	5.7	**1.6**	12.0	17.7	22.5
Explicit Reasoning														
Rewrite-then-Retrieve (Qwen3-0.6B) †	0.6B	57.8	51.6	14.1	39.4	15.0	19.8	23.7	0.6	14.6	1.2	14.9	15.5	22.4
Rewrite-then-Retrieve (Qwen3-8B) †	8B	53.1	54.3	32.1	34.8	20.5	31.1	32.2	3.2	15.2	4.1	17.4	38.8	28.1
Rewrite-then-Retrieve (LLaMA3.1-8B) †	8B	63.1	54.5	27.7	40.6	17.1	28.1	28.2	1.6	11.4	3.8	16.1	30.3	26.9
Search-R3 (Qwen2.5-1.5B)	1.5B	13.8	8.3	4.2	3.5	4.5	12.4	4.6	16.7	3.0	0.6	8.5	11.7	7.7
Latent Reasoning														
GIRCSE (Qwen3-0.6B)	0.6B	29.0	32.8	24.7	30.6	13.6	24.0	**26.6**	**11.1**	1.1	1.3	13.5	20.6	19.1
GIRCSE (Qwen3-8B)	8B	**59.0**	**56.5**	27.2	40.3	19.0	28.5	31.4	3.2	3.6	1.7	14.0	27.2	26.0
GIRCSE (LLaMA3.1-8B)	8B	50.6	43.4	**28.3**	35.7	14.3	26.9	26.4	**6.0**	5.2	0.5	11.6	15.1	22.0
LaSER (Qwen3-0.6B)	0.6B	**50.0**	**45.9**	**25.7**	**32.4**	**18.4**	**27.1**	26.5	9.1	2.7	1.2	**16.0**	**22.4**	**23.1**
LaSER (Qwen3-8B)	8B	58.0	51.8	**29.0**	**44.1**	**26.0**	**32.9**	**34.5**	**9.2**	**11.7**	1.6	**18.7**	**34.0**	**29.3**
LaSER (LLaMA3.1-8B)	8B	**58.4**	**48.1**	28.0	**40.9**	17.0	**29.9**	**28.3**	1.7	**5.9**	1.5	**14.6**	**19.2**	**24.4**

the Explicit-View, we set the maximum sequence length for this view to 1024 during training. The temperature hyperparameters for both the contrastive loss (τ) and the distillation loss (τ_{kd}) are set to 0.02. To balance the multi-task objective, we set the loss weights as follows: $\lambda_1 = 1$, $\lambda_2 = 10$, and $\lambda_3 = 0.1$. The number of latent thinking steps K is set to 3 for both training and inference phases.

Training Cost. Compared to standard contrastive training, LaSER introduces additional overhead from the dual-view forward passes. For the Qwen3-8B backbone, training with LoRA on 4 A100 GPUs takes approximately 8 hours (1 epoch), roughly 1.6× the time of standard contrastive training under the same configuration, primarily due to the longer Explicit-View sequence length (1024 tokens). Peak GPU memory consumption is approximately 72 GB per device.

5 Results

In this section, we present the experimental results of LaSER, followed by a detailed ablation study and efficiency analysis. Our evaluation aims to answer the following three research questions:

RQ1: How does our proposed framework perform compared to fair baselines utilizing standard contrastive learning and other training methods?

RQ2: What is the impact of the proposed multi-grained self-distillation strategies?

RQ3: Can LaSER achieve a superior trade-off between retrieval performance and latency compared to other baseline methods?

5.1 Main Results

In this section, we present a comprehensive analysis of LaSER's performance on the in-domain reasoning benchmark (BRIGHT) and out-of-domain benchmarks (FollowIR and BrowseComp-Plus). To ensure a fair comparison, we reproduced fair baselines using identical training data across all methods and conducted experiments using three model scales based on two distinct architectures. The main results are reported in Table 3 and Table 4 (experiments on the full range of parameter scales are detailed in Subsection 5.3). Our key observations are summarized below.

LaSER Shows Significant Improvements over Standard Dense Retrievers. First, LaSER demonstrates a significant improvement over fair baselines trained with standard contrastive learning. As shown in Table 3, LaSER (Qwen3-8B) achieves an average nDCG@10 of 29.3. This performance not only surpasses the original Qwen3-Embedding-8B (14.0) but also exceeds the fine-tuned Fair Baseline (25.7) by approximately 15%. This substantial performance gap indicates that standard contrastive training with conventional retriever architectures is insufficient for complex reasoning tasks, as it fails to leverage the model's underlying reasoning potential. By internalizing reasoning into the latent space, LaSER effectively activates the reasoning capabilities of the LLM backbone, bridging the gap between shallow semantic matching and complex reasoning.

LaSER Matches the Upper Bound of of Explicit Pipelines. A key motivation of our work is to match the performance of computationally heavy "rewrite-then-retrieve" pipelines without incurring

their high latency costs. Experimental results demonstrate that on both Qwen3-0.6B and Qwen3-8B backbones, our method outperforms explicit reasoning approaches based on query rewriting (surpassing them by 0.7 and 1.2 points, respectively). Furthermore, LaSER significantly outperforms explicit reasoning models that require generating multiple thinking tokens, such as Search-R3 and InBedder. Crucially, LaSER achieves this competitive performance using only a few latent tokens, thereby avoiding the prohibitive computational cost associated with decoding long explicit CoT rationales. This validates our core insight: the semantics of reasoning paths can be effectively compressed into latent states via self-distillation. With an appropriate backbone, our method more effectively elicits the model's inherent reasoning capabilities, leading to superior performance.

LaSER Superiority over Existing Implicit Reasoning Methods. We further compare LaSER with state-of-the-art implicit reasoning methods, specifically GIRCSE. Across 9 experimental settings involving different model scales and datasets, LaSER outperforms GIRCSE in 8 cases. Given that both methods utilize identical input contexts and computational budgets, these results highlight the effectiveness of our proposed self-distillation framework. This suggests that integrating supervision from the teacher model significantly aids the learning of latent thinking. Unlike GIRCSE, which relies solely on output contrastive signals, LaSER employs a dual alignment strategy which incorporate both process-level and output-level distillation to ensure that intermediate latent tokens capture richer semantic information.

5.2 Ablation Study

To address RQ2, we conduct a comprehensive ablation study utilizing Qwen3-0.6B as the backbone model. Given that LaSER functions as a self-distillation framework designed to internalize explicit reasoning, our analysis focuses on two critical dimensions:

(1) Architectural Effectiveness. We examine the contribution of the dual-view structure and the latent thinking mechanism. **w/o Explicit View** removes the explicit-view branch entirely, training the student to perform latent reasoning relying solely on contrastive loss without privileged supervision. **w/o Latent View** replaces the latent reasoning mechanism with a standard single-forward pass to obtain the final embedding, assessing whether the shared backbone learning benefits standard architectures even in the absence of latent tokens.

(2) Learning Strategy Effectiveness. We isolate the impact of the co-learning paradigm and our alignment objectives. **w/o Process KD** excludes the trajectory alignment loss ($\mathcal{L}_{kl}^{mid}$), isolating the impact of supervising intermediate latent states. **w/o Output-level Distillation** removes the final output alignment loss ($\mathcal{L}_{kl}^{out}$), retaining only process-level supervision. **w/o Co-Learning** replaces online self-distillation with a two-stage process, where a fixed explicit-view teacher (fine-tuned offline) generates static supervision labels, validating the efficacy of our dynamic shared-backbone training.

Architectural Effectiveness. We first validate the necessity of the proposed latent reasoning mechanism. As shown in Table 5, removing the latent view leads to the most significant performance degradation (e.g., from 23.10 to 19.93 on Bright). This suggests that the single-vector output form inherent to standard dense retrievers creates a representational bottleneck that constrains reasoning capabilities and hinders the model from effectively absorbing the additional information provided by the teacher input. In contrast, our latent thinking tokens successfully expand the semantic capacity. We also examine the role of the explicit teacher. Removing the explicit view similarly impairs performance. This indicates that without explicit CoT supervision, the learning efficacy of the intermediate latent tokens diminishes, likely due to the absence of explicit semantic guidance.

Figure 3: Performance comparison of LaSER and baselines across different backbones and model sizes on Bright.

Impact of Multi-grained Alignment. We further analyze the effectiveness of our learning strategies. Experimental results demonstrate that both output-level and process-level distillation are essential; removing either component significantly compromises reasoning capabilities and generalization performance across tasks. Notably, the exclusion of output distillation leads to a more pronounced decline, confirming that transferring fine-grained ranking preferences from the teacher is a prerequisite for effective student learning. We also observe that online learning has a substantial impact. On both in-domain and out-of-domain datasets, allowing the teacher to learn dynamically yields superior results. This validates that our shared backbone design promotes better mutual adaptation between views (further analysis of this claim is provided in Section 5.6).

5.3 Robust Analysis

We verify the robustness of LaSER across different backbone architectures, key observations are two fold:

Consistent Superiority over Baselines. We conduct experiments with various backbone model on Bright. As shown in Figure 3, LaSER maintains consistent superiority over all baselines across diverse model families in all 6 settings. Notably, on lightweight backbones (e.g., 0.6B and 1B), LaSER demonstrates superior effectiveness (23.10), even surpassing the computationally intensive rewrite pipeline (22.35). This indicates that our self-distillation framework enables small models to "think" effectively without the overhead of external reasoners.

Instability of Contrastive-only Latent Thinking. We observe that GIRCSE, which relies solely on the contrastive learning objective, exhibits instability across different architectures. In certain

Table 4: Retrieval performance. Columns under Robust04, News21, and Core17 belong to the FollowIR benchmark.

Models	Size	Robust04		News21		Core17		FollowIR-Avg		BrowseComp-Plus		
		MAP@5	p-MRR	nDCG@5	p-MRR	MAP@5	p-MRR	Score	p-MRR	R@5	R@100	R@1000
Standard Dense Retrievers												
BGE-M3	0.6B	1.5	-1.6	21.4	-1.3	7.5	-8.8	10.1	-3.9	4.1	21.7	47.2
Multilingual-E5-Large-Instruct	0.6B	2.4	2.7	25.0	1.2	8.5	-6.0	12.0	-0.7	10.1	36.8	68.6
E5-Mistral-7B-Instruct	7B	2.7	2.5	28.8	0.8	13.7	-7.8	15.1	-1.5	9.3	37.1	70.2
Qwen3-Embedding-0.6B	0.6B	2.8	8.9	27.3	3.6	11.4	2.7	13.8	5.1	8.0	29.1	64.8
Qwen3-Embedding-4B	4B	5.0	11.0	26.7	6.9	11.2	11.1	14.3	9.7	5.9	18.1	39.4
Qwen3-Embedding-8B	8B	3.1	6.2	25.5	8.8	10.1	6.6	12.9	7.2	7.7	31.6	61.3
Basic Contrastive Learning												
Fair Baseline (Qwen3-0.6B)	0.6B	2.1	3.5	13.4	-0.6	6.7	**-3.4**	7.4	-0.2	3.5	21.3	46.9
Fair Baseline (Qwen3-8B)	8B	2.8	4.4	18.9	**2.0**	11.2	**-1.3**	11.0	1.7	11.3	37.4	63.2
Fair Baseline (LLaMA3.1-8B)	8B	2.5	**2.8**	18.9	0.1	8.1	-2.6	9.8	0.1	6.1	25.4	50.8
Explicit Reasoning												
Search-R3 (Qwen2.5-1.5B)	1.5B	3.2	7.1	26.2	-0.1	10.1	2.7	13.2	3.2	0.0	0.3	1.1
Inbedder (LLaMA2-7B)	7B	2.6	3.2	8.5	3.1	1.6	-6.4	4.2	-0.1	1.4	8.1	27.4
Latent Reasoning												
GIRCSE (Qwen3-0.6B)	0.6B	**3.0**	**3.9**	12.8	-0.9	5.7	-7.7	7.2	-1.6	**6.9**	25.2	52.8
GIRCSE (Qwen3-8B)	8B	3.0	4.2	**22.6**	0.7	8.5	1.0	11.4	2.0	**13.0**	**40.8**	**68.1**
GIRCSE (LLaMA3.1-8B)	8B	2.9	1.0	19.3	-1.3	**12.2**	**-1.2**	11.5	-0.5	6.7	23.5	51.0
LaSER (Qwen3-0.6B)	0.6B	2.0	3.4	**25.0**	**1.0**	**7.4**	-4.5	**11.5**	**0.0**	6.8	**26.8**	**54.9**
LaSER (Qwen3-8B)	8B	**4.1**	**5.8**	21.8	0.2	**11.4**	1.3	**12.5**	**2.4**	11.7	38.4	66.9
LaSER (LLaMA3.1-8B)	8B	**3.6**	2.7	**22.0**	**1.6**	11.1	-1.9	**12.2**	**0.8**	**6.8**	**25.7**	**52.9**

Table 5: Ablation study of our proposed method. For FollowIR, we report the average score on all subsets.

Method	Bright	FollowIR		BrowseComp-Plus	
	nDCG@10	Score	p-MRR	R@5	R@1000
LaSER (Qwen3-0.6B)	**23.10**	11.49	-0.02	6.76	**54.88**
w/o Explicit View	20.59	9.78	-1.71	6.48	50.14
w/o Latent View	19.93	8.62	-0.11	5.93	52.25
w/o Process Align.	22.33	**11.63**	-1.58	6.18	52.43
w/o Output Distill.	19.97	7.55	**0.43**	**7.03**	53.02
w/o Co-Learning	20.98	10.87	-0.54	5.56	52.57
Basic Contrastive Learn.	18.25	7.38	-0.17	3.54	46.9

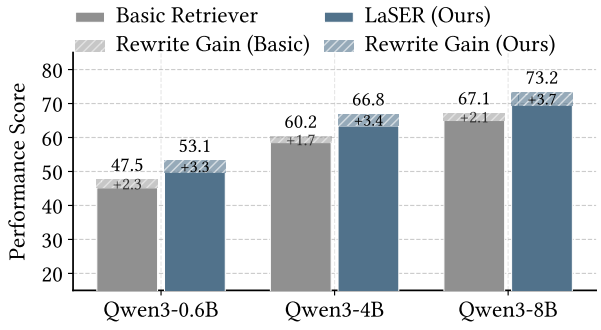

Figure 4: Comparison of the performance of using rewrite path on Bright between our method and basic training.

settings (e.g., LLaMA3.2-3B), it yields performance degradations compared to the standard retriever baseline, while showing only marginal improvements in others. In contrast, by incorporating distillation signals, our method achieves substantial gains on similar architectures. This suggests that latent thinking mechanisms may require privileged supervision to facilitate effective learning.

5.4 Latency Analysis

To address RQ3, we evaluate inference latency on a subset of 80 queries from the BRIGHT benchmark, utilizing a single NVIDIA A100 (80GB) GPU. To ensure a rigorous comparison, we employ vLLM [28] for the inference of external rewriter models in explicit reasoning pipelines, while the native Hugging Face Transformers implementation is used for all retriever backbones (including ours). A batch size of 8 is maintained for all baselines and our method.

We note that the use of different inference stacks (vLLM for the rewriter vs. HF Transformers for the retriever) may slightly favor the pipeline baseline's rewriting stage; however, it reflects realistic deployment where generative models leverage optimized serving frameworks. As demonstrated in Figure 1, LaSER drastically reduces computational costs, incurring only 0.3% of the latency associated with the "rewrite-then-retrieve" pipeline while maintaining comparable retrieval effectiveness. Although LaSER introduces a modest latency overhead compared to the standard base retriever (approximately 1.7×) due to the autoregressive generation of latent thinking tokens, this overhead decreases as model scale increases. For larger backbones, the relative cost of vector operations becomes marginal compared to the model's forward pass time, allowing the system to benefit more significantly from KV caching mechanisms.

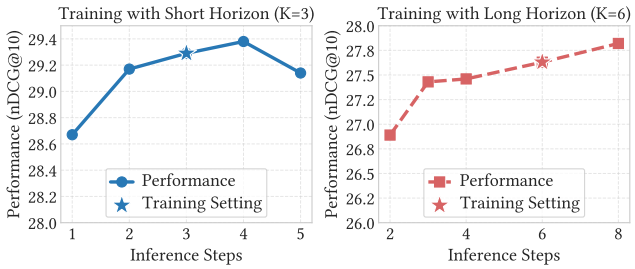

Figure 5: Effect of latent thinking steps during training and inference.

5.5 Impact of Latent Reasoning Steps

We investigate the impact of the number of latent thinking steps (K) during both the training and inference on Qwen3-8B, with results illustrated in Figure 5.

Training Efficiency. We observe that increasing the training steps from $K = 3$ to $K = 6$ yields negligible performance gains or even slight degradation. We attribute this phenomenon to the high semantic density of the latent space facilitated by our framework. Unlike standard implicit reasoning, the introduction of the Explicit-View provides privileged supervision, serving as a clear optimization target. This allows LaSER to effectively compress the reasoning process, where a few latent steps are sufficient to capture the core logic of the reasoning path. Consequently, enforcing a larger K during training may introduce noise by compelling the model to align with granular, non-informative segments of the explicit text.

Inference Scaling. Conversely, increasing K during inference demonstrates consistent performance improvements, which highlights that LaSER has acquired a robust mechanism for iterative refinement of embedding. Rather than merely memorizing fixed step trajectories or copying preceding states, the model actively utilizes additional computing steps to deepen its semantic understanding and optimize the query representation.

5.6 Analysis of Co-Learning of Both Views

Since our framework involves the joint training of two distinct views sharing a common backbone, we further investigate the synergistic interaction between them. Specifically, we evaluate the backbone's capability to process explicit reasoning by comparing the performance gains achieved when providing explicit rewritten queries to both the Basic Retriever and LaSER during inference. As shown in Figure 4, LaSER demonstrates a significantly larger performance boost from explicit rewrites compared to the Basic baseline across all model sizes (e.g. 3.4 v.s. 1.7 in Qwen3-4B). This substantial difference indicates that our Explicit-View training objective effectively optimizes the shared backbone, enabling it to better capture and encode the semantic information within CoT rationales for retrieval tasks. Consequently, this enhanced explicit processing capability serves as a superior semantic upper bound, thereby facilitating more effective distillation into the Latent-View.

6 Conclusion & Future Work

In this paper, we propose LaSER, a novel self-distillation framework that internalizes explicit CoT reasoning into the latent space of dense retrievers. By employing a multi-grained alignment strategy that synchronizes latent thinking tokens with explicit reasoning trajectories, LaSER effectively bridges the gap between deep semantic reasoning and retrieval efficiency. Extensive experiments demonstrate that LaSER significantly outperforms state-of-the-art baselines, achieving performance comparable to computationally expensive "rewrite-then-retrieve" pipelines while maintaining low inference latency. Our work validates that the semantics of explicit reasoning can be effectively compressed into latent states, offering an efficient solution for complex query understanding. Despite these promising results, our method represents only a preliminary step toward fully autonomous latent reasoning. In future work, we plan to explore reinforcement learning techniques [10–12] to further optimize the intermediate reasoning process by directly optimizing the latent reasoning trajectory based on retrieval utility.

Acknowledgements

This work was supported by National Natural Science Foundation of China No. 62272467. The work was partially done at the Engineering Research Center of Next-Generation Intelligent Search and Recommendation, MOE.

References

[1] Qingyao Ai, Ting Bai, Zhao Cao, Yi Chang, Jiawei Chen, Zhumin Chen, Zhiyong Cheng, Shoubin Dong, Zhicheng Dou, Fuli Feng, Shen Gao, Jiafeng Guo, Xiangnan He, Yanyan Lan, Chenliang Li, Yiqun Liu, Ziyu Lyu, Weizhi Ma, Jun Ma, Zhaochun Ren, Pengjie Ren, Zhiqiang Wang, Mingwen Wang, Ji-Rong Wen, Le Wu, Xin Xin, Jun Xu, Dawei Yin, Peng Zhang, Fan Zhang, Weinan Zhang, Min Zhang, and Xiaofei Zhu. 2023. Information Retrieval meets Large Language Models: A strategic report from Chinese IR community. *AI Open* 4 (2023), 80–90. doi:10.1016/J.AIOPEN.2023.08.001

[2] Maarten Bosma, Ed Chi, Brian Ichter, Quoc V. Le, Dale Schuurmans, Xuezhi Wang, Jason Wei, Fei Xia, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems 35*. Neural Information Processing Systems Foundation, 24824–24837. doi:10.52202/068431-1800

[3] Hung-Ting Chen, Xiang Liu, Shauli Ravfogel, and Eunsol Choi. 2025. Beyond Single Embeddings: Capturing Diverse Targets with Multi-Query Retrieval. *CoRR* abs/2511.02770 (2025). doi:10.48550/ARXIV.2511.02770 arXiv:2511.02770

[4] Jianlyu Chen, Junwei Lan, Chaofan Li, Defu Lian, and Zheng Liu. 2025. ReasonEmbed: Enhanced Text Embeddings for Reasoning-Intensive Document Retrieval. *CoRR* abs/2510.08252 (2025). doi:10.48550/ARXIV.2510.08252 arXiv:2510.08252

[5] Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 2318–2335. doi:10.18653/v1/2024.findings-acl.137

[6] Xinghao Chen, Anhao Zhao, Heming Xia, Xuan Lu, Hanlin Wang, Yanjun Chen, Wei Zhang, Jian Wang, Wenjie Li, and Xiaoyu Shen. 2025. Reasoning Beyond Language: A Comprehensive Survey on Latent Chain-of-Thought Reasoning. *CoRR* abs/2505.16782 (2025). doi:10.48550/ARXIV.2505.16782 arXiv:2505.16782

[7] Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. 2025. BrowseComp-Plus: A More Fair and Transparent Evaluation Benchmark of Deep-Research Agent. *CoRR* abs/2508.06600 (2025). doi:10.48550/ARXIV.2508.06600 arXiv:2508.06600

[8] Xuanming Cui, Jianpeng Cheng, Hong-you Chen, Satya Narayan Shukla, Abhijeet Awasthi, Xichen Pan, Chaitanya Ahuja, Shlok Kumar Mishra, Yonghuan Yang, Jun Xiao, Qi Guo, Ser-Nam Lim, Aashu Singh, and Xiangjun Fan. 2025. Think Then Embed: Generative Context Improves Multimodal Embedding. *CoRR* abs/2510.05014 (2025). doi:10.48550/ARXIV.2510.05014 arXiv:2510.05014

[9] Debrup Das, Sam O'Nuallain, and Razieh Rahimi. 2025. RaDeR: Reasoning-aware

Dense Retrieval Models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 19981–20008. doi:10.18653/v1/2025.emnlp-main.1011

[10] Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. Agentic Entropy-Balanced Policy Optimization. *CoRR* abs/2510.14545 (2025). doi:10.48550/ARXIV.2510.14545 arXiv:2510.14545

[11] Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. 2025. Tool-Star: Empowering LLM-Brained Multi-Tool Reasoner via Reinforcement Learning. *CoRR* abs/2505.16410 (2025). doi:10.48550/ARXIV.2505.16410 arXiv:2505.16410

[12] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. Agentic Reinforced Policy Optimization. *CoRR* abs/2507.19849 (2025). doi:10.48550/ARXIV.2507.19849 arXiv:2507.19849

[13] Guanting Dong, Yutao Zhu, Chenghao Zhang, Zechen Wang, Zhicheng Dou, and Ji-Rong Wen. 2024. Understand What LLM Needs: Dual Preference Alignment for Retrieval-Augmented Generation. *CoRR* abs/2406.18676 (2024). doi:10.48550/ARXIV.2406.18676 arXiv:2406.18676

[14] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise Zero-Shot Dense Retrieval without Relevance Labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, 1762–1777. doi:10.18653/V1/2023.ACL-LONG.99

[15] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2024. Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997 [cs.CL]

[16] Yuntao Gui and James Cheng. 2025. Search-R3: Unifying Reasoning and Embedding Generation in Large Language Models. *CoRR* abs/2510.07048 (2025). doi:10.48550/ARXIV.2510.07048 arXiv:2510.07048

[17] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training Large Language Models to Reason in a Continuous Latent Space. *CoRR* abs/2412.06769 (2024). doi:10.48550/ARXIV.2412.06769 arXiv:2412.06769

[18] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, Donia Scott, Núria Bel, and Chengqing Zong (Eds.). International Committee on Computational Linguistics, 6609–6625. doi:10.18653/V1/2020.COLING-MAIN.580

[19] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276* (2024).

[20] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. *Trans. Mach. Learn. Res.* 2022 (2022). https://openreview.net/forum?id=jKN1pXi7b0

[21] Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Qi Ye, and Zhicheng Dou. 2025. Hierarchical Document Refinement for Long-context Retrieval-augmented Generation. *CoRR* abs/2505.10413 (2025). doi:10.48550/ARXIV.2505.10413 arXiv:2505.10413

[22] Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yang Zhao, Hongjin Qian, and Zhicheng Dou. 2025. HiRA: A Hierarchical Reasoning Framework for Decoupled Planning and Execution in Deep Search. *CoRR* abs/2507.02652 (2025). doi:10.48550/ARXIV.2507.02652 arXiv:2507.02652

[23] Jiajie Jin, Yuyao Zhang, Yimeng Xu, Hongjin Qian, Yutao Zhu, and Zhicheng Dou. 2025. FinSight: Towards Real-World Financial Deep Research. *CoRR* abs/2510.16844 (2025). doi:10.48550/ARXIV.2510.16844 arXiv:2510.16844

[24] Jiajie Jin, Yutao Zhu, Xinyu Yang, Chenghao Zhang, and Zhicheng Dou. 2024. FlashRAG: A Modular Toolkit for Efficient Retrieval-Augmented Generation Research. *CoRR* abs/2405.13576 (2024). doi:10.48550/ARXIV.2405.13576 arXiv:2405.13576

[25] Jiajie Jin, Yutao Zhu, Yujia Zhou, and Zhicheng Dou. 2024. BIDER: Bridging Knowledge Inconsistency for Efficient Retrieval-Augmented LLMs via Key Supporting Evidence. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 750–761. doi:10.18653/V1/2024.FINDINGS-ACL.42

[26] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *EMNLP*. 6769–6781.

[27] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[28] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, Jason Flinn, Margo I. Seltzer, Peter Druschel, Antoine Kaufmann, and Jonathan Mace (Eds.). ACM, 611–626. doi:10.1145/3600006.3613165

[29] 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. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=lgsyLSsDRe

[30] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[31] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic Search-Enhanced Large Reasoning Models. *CoRR* abs/2501.05366 (2025). doi:10.48550/ARXIV.2501.05366 arXiv:2501.05366

[32] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yutao Zhu, and Zhicheng Dou. 2025. RAG-Critic: Leveraging Automated Critic-Guided Agentic Workflow for Retrieval Augmented Generation. *CoRR* abs/2501.00000 (2025).

[33] Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. DeepAgent: A General Reasoning Agent with Scalable Toolsets. *CoRR* abs/2510.21618 (2025). doi:10.48550/ARXIV.2510.21618 arXiv:2510.21618

[34] Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Shijian Wang, Guanting Dong, Jiajie Jin, Hao Wang, Yinuo Wang, Ji-Rong Wen, and Zhicheng Dou. 2026. OmniGAIA: Towards Native Omni-Modal AI Agents. *CoRR* abs/2602.22897 (2026). doi:10.48550/ARXIV.2602.22897 arXiv:2602.22897

[35] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. 2025. WebThinker: Empowering Large Reasoning Models with Deep Research Capability. *CoRR* abs/2504.21776 (2025). doi:10.48550/ARXIV.2504.21776 arXiv:2504.21776

[36] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *CoRR* abs/2308.03281 (2023). doi:10.48550/ARXIV.2308.03281 arXiv:2308.03281

[37] Zhengyang Liang, Meiyu Liang, Wei Huang, Yawen Li, and Zhe Xue. 2024. Dynamic Self-adaptive Multiscale Distillation from Pre-trained Multimodal Large Model for Efficient Cross-modal Representation Learning. *CoRR* abs/2404.10838 (2024). doi:10.48550/ARXIV.2404.10838 arXiv:2404.10838

[38] Wenhan Liu, Xinyu Ma, Weiwei Sun, Yutao Zhu, Yuchen Li, Dawei Yin, and Zhicheng Dou. 2025. ReasonRank: Empowering Passage Ranking with Strong Reasoning Ability. *CoRR* abs/2508.07050 (2025). doi:10.48550/ARXIV.2508.07050 arXiv:2508.07050

[39] Amir M. Mansourian, Rozhan Ahmadi, Masoud Ghafouri, Amir Mohammad Babaei, Elaheh Badali Golezani, Zeynab Yasamani Ghamchi, Vida Ramezanian, Alireza Taherian, Kimia Dinashi, Amirali Miri, and Shohreh Kasaei. 2025. A Comprehensive Survey on Knowledge Distillation. *Trans. Mach. Learn. Res.* 2025 (2025). https://openreview.net/forum?id=3cbJzdR78B

[40] Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 5783–5797. doi:10.18653/V1/2020.EMNLP-MAIN.466

[41] Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. 2025. A Survey of Conversational Search. *ACM Trans. Inf. Syst.* 43, 6 (2025), 167:1–167:50. doi:10.1145/3759453

[42] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2024. Generative Representational Instruction Tuning. *CoRR* abs/2402.09906 (2024). doi:10.48550/ARXIV.2402.09906 arXiv:2402.09906

[43] Letian Peng, Yuwei Zhang, Zilong Wang, Jayanth Srinivasa, Gaowen Liu, Zihan Wang, and Jingbo Shang. 2024. Answer is All You Need: Instruction-following Text Embedding via Answering the Question. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 459–477. doi:10.18653/V1/2024.ACL-LONG.27

[44] Xubo Qin, Jun Bai, Jiaqi Li, Zixia Jia, and Zilong Zheng. 2025. TongSearch-QR: Reinforced Query Reasoning for Retrieval. *CoRR* abs/2506.11603 (2025). doi:10.48550/ARXIV.2506.11603 arXiv:2506.11603

[45] Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. ReasonIR: Training Retrievers for Reasoning Tasks. *CoRR* abs/2504.20595 (2025). doi:10.48550/ARXIV.2504.20595 arXiv:2504.20595

[46] Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. CODI: Compressing Chain-of-Thought into Continuous Space via Self-Distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 677–693. doi:10.18653/v1/2025.emnlp-main.36

[47] Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han-yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan Ö. Arik, Danqi Chen, and Tao Yu. 2025. BRIGHT: A Realistic and Challenging Benchmark for Reasoning-Intensive Retrieval. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net. https://openreview.net/forum?id=ykuc5q381b

[48] Jiashuo Sun, Shixuan Liu, Zhaochen Su, Xianrui Zhong, Pengcheng Jiang, Bowen Jin, Peiran Li, Weijia Shi, and Jiawei Han. 2025. GRACE: Generative Representation Learning via Contrastive Policy Optimization. *CoRR* abs/2510.04506 (2025). doi:10.48550/ARXIV.2510.04506 arXiv:2510.04506

[49] Yu-Che Tsai, Kuan-Yu Chen, Yuan-Chi Li, Yuan-Hao Chen, Ching-Yu Tsai, and Shou-De Lin. 2025. Let LLMs Speak Embedding Languages: Generative Text Embeddings via Iterative Contrastive Refinement. *CoRR* abs/2509.24291 (2025). doi:10.48550/ARXIV.2509.24291 arXiv:2509.24291

[50] Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text Embeddings by Weakly-Supervised Contrastive Pre-training. *CoRR* abs/2212.03533 (2022). doi:10.48550/ARXIV.2212.03533 arXiv:2212.03533

[51] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving Text Embeddings with Large Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, 11897–11916. doi:10.18653/V1/2024.ACL-LONG.642

[52] Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query Expansion with Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, 9414–9423. doi:10.18653/V1/2023.EMNLP-MAIN.585

[53] Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn J. Lawrie, and Luca Soldaini. 2025. FollowIR: Evaluating and Teaching Information Retrieval Models to Follow Instructions. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, Luis Chiruzzo, Alan Ritter, and Lu Wang (Eds.). Association for Computational Linguistics, 11926–11942. doi:10.18653/V1/2025.NAACL-LONG.597

[54] Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed Resources For General Chinese Embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, Grace Hui Yang, Hongning Wang, Sam Han, Claudia Hauff, Guido Zuccon, and Yi Zhang (Eds.). ACM, 641–649. doi:10.1145/3626772.3657878

[55] Seunghan Yang, Juntae Lee, Jihwan Bang, Kyuhong Shim, Minsoo Kim, and Simyung Chang. 2025. Learning Contextual Retrieval for Robust Conversational Search. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 11991–12003. doi:10.18653/v1/2025.emnlp-main.602

[56] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*. Association for Computational Linguistics, Brussels, Belgium, 2369–2380. doi:10.18653/v1/D18-1259

[57] Sijia Yao, Pengcheng Huang, Zhenghao Liu, Yu Gu, Yukun Yan, Shi Yu, and Ge Yu. 2025. ExpandR: Teaching Dense Retrievers Beyond Queries with LLM Guidance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 19036–19054. doi:10.18653/v1/2025.emnlp-main.963

[58] Dun Zhang and FulongWang. 2024. Jasper and Stella: distillation of SOTA embedding models. *CoRR* abs/2412.19048 (2024). doi:10.48550/ARXIV.2412.19048 arXiv:2412.19048

[59] Yichi Zhang, Jun Bai, Zhixin Cai, Shuhan Qin, Zhuofan Chen, Jinghua Guan, and Wenge Rong. 2025. Your Dense Retriever is Secretly an Expeditious Reasoner. *CoRR* abs/2510.21727 (2025). doi:10.48550/ARXIV.2510.21727 arXiv:2510.21727

[60] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *CoRR* abs/2506.05176 (2025). doi:10.48550/ARXIV.2506.05176 arXiv:2506.05176

[61] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *CoRR* abs/2303.18223 (2023). doi:10.48550/ARXIV.2303.18223 arXiv:2303.18223

[62] Ziliang Zhao, Changle Qu, Zhicheng Dou, Haonan Chen, and Jiajie Jin. 2025. Retrieving Intent-covering Demonstrations for Clarification Generation in Conversational Search Systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.2, KDD 2025, Toronto ON, Canada, August 3-7, 2025*, Luiza Antonie, Jian Pei, Xiaohui Yu, Flavio Chierichetti, Hady W. Lauw, Yizhou Sun, and Srinivasan Parthasarathy (Eds.). ACM, 3992–4001. doi:10.1145/3711896.3737104

[63] Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Haonan Chen, Zheng Liu, Zhicheng Dou, and Ji-Rong Wen. 2026. Large Language Models for Information Retrieval: A Survey. *ACM Transactions on Information Systems* 44, 1 (2026), 1–54. doi:10.1145/3748304