Align Then Adapt: Label-Efficient Adapter Learning for Asymmetric Dense Retrieval

Anonymous ACL submission

Abstract

Dense retrieval systems increasingly face an asymmetry between complex instruction-like queries and relatively simple, static document collections. While stronger embedders can better understand such queries, re-embedding large corpora or fine-tuning large models is often impractical. We propose Efficient Retrieval Adapter (ERA), a query-side adapter learning framework for re-index-free retrieval adaptation. ERA first aligns the embedding spaces of a strong query embedder and a lightweight document embedder using unlabeled corpus documents, and then adapts the aligned query representation with a small number of labeled query-document pairs. Across 126 MAIR retrieval tasks from six domains, ERA improves average nDCG@10 by up to 8.2 points in symmetric settings and by more than 12 points in asymmetric settings, while using substantially fewer labels than supervised adapter training. These results show that retrieval systems can benefit from stronger query understanding without updating backbone embedders or rebuilding document indexes.

1 Introduction

Dense retrieval systems (Karpukhin et al., 2020) are increasingly expected to handle queries that go far beyond short keyword-like inputs. In many realistic scenarios, users express complex intent through long instructions, multi-step constraints, or task-specific descriptions, while the target documents remain comparatively simple and static. Recent studies (Sun et al., 2024; Weller et al., 2025; van den Elsen et al., 2025) have introduced benchmarks that reflect this asymmetry, where queries are often long and instruction-like. This asymmetry creates a fundamental mismatch in retrieval: understanding the query may require strong reasoning and instruction-following ability, whereas representing the document collection efficiently

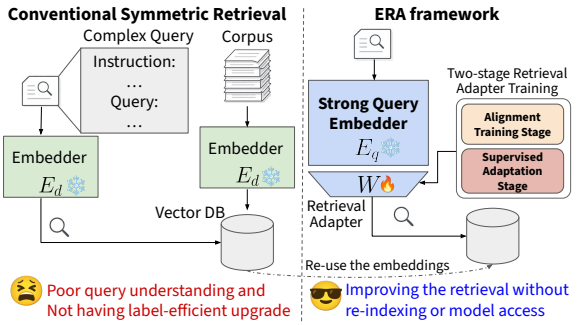


Figure 1: ERA enables re-index-free asymmetric dense retrieval by aligning a strong query embedder with a lightweight document index. Unlike conventional symmetric retrieval, ERA trains a lightweight query-side adapter in two stages: self-supervised alignment using unlabeled corpus documents and supervised adaptation using limited labeled query-document pairs.

demands a lightweight indexing pipeline. However, most existing retrieval systems treat the query and document sides symmetrically, using the same model, the same adaptation strategy, or the same optimization assumptions for both, as illustrated in the left part of Figure 1. As a result, they either fail to account for the complexity and semantic gaps between queries and documents, or do so only at the substantial cost of fine-tuning.

A natural solution would be to fine-tune a LLM-based retrieval model (Zhang et al., 2025; Lee et al., 2025) so that it can better follow complex instructions. In practice, however, this is difficult to deploy. Fine-tuning large embedding models is computationally expensive, memory-intensive, and operationally burdensome, especially when multiple tasks or domains must be supported. Beyond training cost, supervised adaptation itself depends on collecting query-document relevance labels, which is often the real bottleneck. Such labels are expensive to create, highly task-dependent, and difficult to scale across domains. Recent work (Sanjeev and Troynikov, 2024; Yoon et al., 2024) has explored

lightweight alternatives to fine-tuning, such as training retrieval adapters. However, these methods still require a large number of labeled examples per task and do not effectively address the representation gap between different embedding models or the semantic gap between complex queries and simple documents. Moreover, empirical evidence (Sanjeev and Troynikov, 2024) suggests that adapter-based training alone may not yield noticeable gains unless each task has at least around 1,500 labeled query-document pairs, making low-resource adaptation challenging even when full fine-tuning is avoided.

In many realistic settings, however, practitioners do not have enough labeled data to support heavy supervised adaptation, nor the computational budget to repeatedly fine-tune large embedders or re-index a potentially large-volume document corpus. This motivates a different question: *can retrieval be efficiently adapted to complex queries without updating the backbone model and without re-indexing the potentially large-volume document corpus?* To the best of our knowledge, addressing query-document complexity mismatch using a lightweight retrieval adapter alone, while keeping the underlying embedding models frozen, has not been systematically studied.

In this work, we propose Efficient Retrieval Adapter (ERA), a novel framework for training retrieval adapters in a label-efficient manner. Inspired by the pre-training and supervised fine-tuning stages of LLMs, the key idea of ERA is to decouple retrieval adapter training into two stages: a self-supervised alignment stage and a supervised adaptation stage. In the alignment stage, we train the retrieval adapter to align the embedding spaces between a query embedder and a document embedder using self-supervised learning on unlabeled data. This step reduces the semantic gap between the two embedding spaces, making subsequent supervised training more effective and efficient, allowing the retrieval adapter to achieve performance gains even with fewer labeled examples, as observed in LLM fine-tuning after pre-training (Devlin et al., 2019). In the supervised adaptation stage, we tune the retrieval adapter using a small number of labeled query-document pairs, capturing semantic nuances between the query and document embedding spaces that are not fully captured by the alignment stage. This design avoids the prohibitive cost of re-embedding the corpus with a large model, while allowing the system to benefit from richer query understanding at inference time.

The key contributions are as follows:

- We propose ERA, a novel framework for training retrieval adapters in a label-efficient manner, which decouples the training into a self-supervised alignment stage and a supervised adaptation stage.
- We conduct extensive experiments with five embedding models from three model families on the MAIR benchmark (Sun et al., 2024), spanning 126 retrieval tasks across six domains. We show that ERA, despite being a linear adapter, effectively combines stronger query embedders with weaker document embedders, including across different model families, by bridging both the representation gap between embedding models and the semantic gap between complex queries and simple documents, improving average **nDCG@10 by up to 12%** over zero-shot retrieval of the document embedder without expensive corpus re-indexing.
- We observe that the alignment stage significantly improves the label efficiency of adapter training, allowing ERA to achieve substantial performance gains with fewer than 100 labeled examples per task, which is a significant reduction compared to prior methods that require around 1,500 labeled examples for noticeable gains.
- We analyze the robustness of ERA against out-of-domain tasks and its hyperparameters, showing that it is a practical solution for real-world retrieval scenarios.

2 Related Work

LLM Embedders. While traditional BERT-based embedders (Karpukhin et al., 2020; Chen et al., 2024) have long been the standard for retrieval, recent LLM-based embedders (Lee et al., 2025; Zhang et al., 2025) often achieve stronger performance due to their superior reasoning and instruction-following abilities. However, these gains come at a substantial cost: LLM-based embedders are expensive to fine-tune and deploy, making them less practical for real-world settings that must support many tasks or domains, especially given their large model sizes (e.g., 4B+ parameters).

Embedding Adapters. While prior work has explored fine-tuning for retrieval improvement

Methods	No Parameter Access	< 1000 Label Train	Asymmetric Embeddings	Domain Generality	Adaptation Signal
Zero-shot Retrieval	✓	✗	✗	✓	✗
Embedding Adapter (Sanjeev and Troynikov, 2024)	✓	✗	✗	✗	Relevance labels
Search Adaptor (Yoon et al., 2024)	✓	✗	✗	✗	Relevance labels
Drift-Adapter (Vejendla, 2025)	✓	✗	✓	✓	Alignment Only
ERA (Ours)	✓	✓	✓	✓	Alignment + Relevance

Table 1: Comparison of the proposal, ERA, with existing methods. "Adaptation Signal" summarizes the supervision used to train the adapter. ERA combines cross-embedder alignment with relevance-aware supervised adaptation.

(Kim and Baek, 2025; Dai et al., 2023; Schlatt et al., 2025), Embedding Adapters (Sanjeev and Troynikov, 2024) and Search Adaptor (Yoon et al., 2024) provide lightweight alternatives to full fine-tuning, but remain limited in our setting: both rely on substantial task-specific supervision and require at least 1,500 labeled examples per task for noticeable gains. Moreover, neither is designed to bridge the representation gap between heterogeneous query and document embedders. Drift-Adapter (Vejendla, 2025) shows that adapters can align embedding spaces across versions of the same model for re-index-free retrieval adaptation, but does not study supervised label-efficient retrieval improvement. Overall, although embedding adapters are promising for retrieval adaptation, as summarized in Table 1, their label efficiency and ability to bridge both cross-model representation gaps and query-document semantic gaps remain underexplored.

Transferability of Embeddings. Recent work suggests that representations from different models share meaningful geometric structure. Chen et al. (2025) show that affine stitches can transfer linear features across language models, Huh et al. (2024) argue that representations increasingly converge as models scale, and Jha et al. (2025) show that embeddings from different text encoders can be translated through a shared latent structure even without paired data. However, these studies mainly support the assumption that cross-model geometry is transferable, rather than showing how to use it for retrieval adaptation under query-document asymmetry.

3 Efficient Retrieval Adapter, ERA

To address the challenges of retrieval adaptation under query-document complexity asymmetry, we propose Efficient Retrieval Adapter (ERA), a novel framework for training retrieval adapters in a label-efficient manner. In this section, we first formulate the problem of retrieval with an adapter, and then describe the design principles of ERA, followed by our training recipe.

3.1 Task Formulation

Zero-shot Retrieval. In a traditional retrieval setting (Karpukhin et al., 2020; Maekawa et al., 2024; Taguchi et al., 2025), we have an embedding model E that maps a query q and a document d to their respective embeddings, $E(q) \in \mathbb{R}^h$ and $E(d) \in \mathbb{R}^h$, where h is the embedding dimension. The retrieval process typically involves computing the similarity between the query and document embeddings $\text{sim}(E(q), E(d))$, often using cosine similarity, to rank documents based on their relevance to the query.

Retrieval with Adapter. Embedding Adapter (Sanjeev and Troynikov, 2024) introduces a lightweight adapter module $\mathbf{W} \in \mathbb{R}^{h \times h}$ that transforms the query embedding space. The adapted query embedding is computed as $E(q)\mathbf{W}$, which is then L2-normalized, and the retrieval process is based on the similarity $\text{sim}(E(q)\mathbf{W}, E(d))$. This allows for task-specific adaptation without updating the full embedding model, but it typically requires a large number of labeled examples to train effectively.

Asymmetric Retrieval with Adapter. In our setting, we consider a more general scenario where the query and document embeddings are generated by different models, denoted as E_q and E_d, respectively. The retrieval process is based on the similarity $\text{sim}(E_q(q)\mathbf{W}, E_d(d))$, where $\mathbf{W} \in \mathbb{R}^{h_q \times h_d}$ is the retrieval adapter that transforms the query embedding space to better align with the document embedding space. This formulation allows for asymmetric embedding spaces, which is more realistic in scenarios where queries require powerful reasoning models while documents are indexed with lightweight encoders.

3.2 Design Principles

ERA is designed with the following principles:

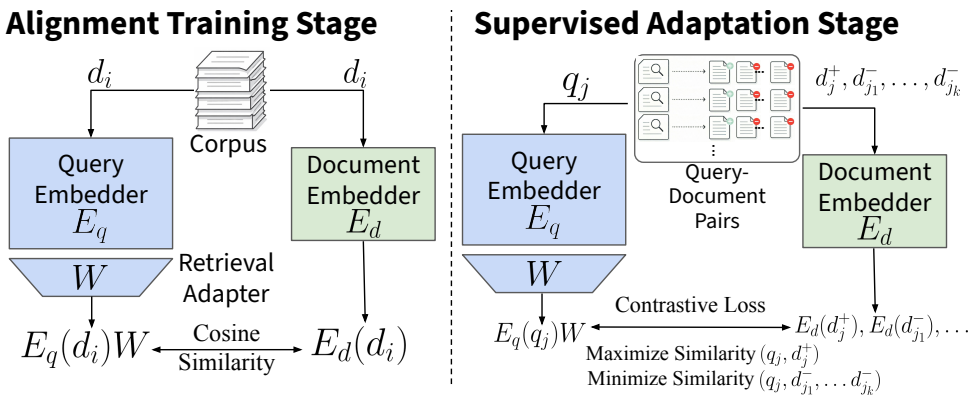


Figure 2: The overview of Efficient Retrieval Adapter. In the alignment stage, the same unlabeled document is encoded by both embedders and used as a positive pair. In the adaptation stage, labeled query-document pairs and hard negatives train the same adapter for retrieval.

- **No Parameter Access**: ERA requires no access to the base embedding model parameters, making it compatible with black-box APIs.
- **Low Label Requirement**: ERA is designed to be effective with fewer than 100 labeled query-document pairs per task, making it practical for real-world scenarios where labeled data is scarce.
- **Asymmetric Embeddings**: ERA supports heterogeneous encoder pairs, including embedders from entirely different model families, reflecting realistic deployments where queries demand powerful reasoning models while documents are efficiently indexed with lightweight encoders.
- **Domain Generality**: ERA is designed to be effective across a wide range of domains and tasks, without requiring domain-specific adaptations or assumptions.

To achieve these design goals, ERA uses a retrieval adapter, a lightweight module, to transform the query embedding space toward the document embedding space. ERA trains this adapter in two stages: a *self-supervised alignment stage* with no supervised labels and a *supervised adaptation stage* with limited labeled data, as illustrated in Figure 2. In the alignment stage, we train the retrieval adapter on unlabeled data to align the embedding spaces of the query and document embedders. This helps bridge the representation gap between the two asymmetric spaces, reduces the semantic gap, and makes subsequent supervised training more effective and efficient. In the adaptation stage, we fine-tune the adapter on limited labeled query-document pairs to capture semantic nuances not fully addressed during alignment, improving understanding of complex queries and retrieval performance without re-indexing the corpus.

3.3 Training Recipe

Alignment Training Stage. The core idea of the alignment training stage is to match the embeddings from the query embedder and the document embedder by using identical input documents. Specifically, we first collect a large set of unlabeled documents from the target corpus. For each document d, we compute its embedding using the query embedder, $E_q(d)$ and its embedding using the document embedder, $E_d(d)$. Then, we train the retrieval adapter $\mathbf{W}$ to maximize the similarity between the adapted query-encoder representation of the document and the document embedding, i.e., $\text{sim}(E_q(d)\mathbf{W}, E_d(d))$. We use cosine similarity as the similarity function sim in our experiments since it is widely-used and employed in our retrieval process. This training process encourages the retrieval adapter to learn a transformation that aligns the query embedding space with the document embedding space, effectively bridging the representation gap between the two models.

Supervised Adaptation Stage. After the alignment training stage, we fine-tune the retrieval adapter using a small number of labeled query-document pairs. Specifically, we have a set of labeled examples $\{(q_i, d_i^+, \{d_{i,j}^-\}_{j=1}^k)\}_{i=1}^N$, where q_i is a query, d_i^+ is a relevant document, and $\{d_{i,j}^-\}_{j=1}^k$ are irrelevant documents. We compute the adapted query embedding as $E_q(q_i)\mathbf{W}$, the positive document embedding as $E_d(d_i^+)$, and the negative document embeddings as $\{E_d(d_{i,j}^-)\}_{j=1}^k$. Then, we optimize the adapter $\mathbf{W}$ to maximize the similarity

between the adapted query embedding and the positive document embedding while minimizing the similarity with the negative document embeddings. This can be achieved using a contrastive loss function, such as the InfoNCE loss (Oord et al., 2018) or the triplet loss (Reimers and Gurevych, 2019). This supervised training process allows the adapter to capture the semantic nuances between the query and document embedding spaces that are not captured by the alignment stage.

Domain Generality. To support general-domain coverage, we train a single retrieval adapter across all tasks and domains. Specifically, we sample training examples from the full dataset in both the alignment and adaptation stages, enabling the adapter to capture shared structure across diverse retrieval scenarios without requiring domain-specific adaptation.

4 Experiments

We apply ERA to conduct extensive experiments to explore the following research questions: **(RQ1)** *Can we effectively combine a stronger query embedder with a weaker document embedder using ERA, and can query and document embedders come from different model families?* **(RQ2)** *How does ERA perform under different label resource settings?* **(RQ3)** *Is a general adapter obtained by ERA effective across different domains?* **(RQ4)** *How sensitive is ERA to key design and training choices?*

4.1 Setup

Embedders. We use the Qwen3-embedding-8B/0.6B (Zhang et al., 2025), OpenAI embeddings (OpenAI, 2024), and BGE-M3 (Chen et al., 2024) as the base embedders for all methods, including ERA and baselines. We refer to these models as Qwen3-8B/-0.6B, OpenAI-small, and BGE-M3. The details are summarized in Appendix A.

Datasets. We evaluate ERA and the baselines on the MAIR benchmark (Sun et al., 2024), which contains 126 retrieval tasks across six domains: Academic, Code, Finance, Legal, Medical, and Web. Built on widely used benchmarks such as BEIR (Thakur et al., 2021) and KILT (Petroni et al., 2021), MAIR provides broad coverage across domains and tasks. At the same time, it contains only about 80 labeled examples per task on average (10,038 queries in total), making it an ideal testbed for evaluating label efficiency. To simulate explicit user intent, we use the queries with instructions in the MAIR benchmark. We split labeled examples into 5%–40% for training (roughly 500–4,000 queries across all tasks), 10% for validation, and 50% for testing. As we vary the training label percentage, we keep the validation and test sets fixed for fair comparison across label settings.

Methods. We compare methods under the constraints targeted by ERA: frozen backbone embedders, a fixed document index, and limited labeled query-document pairs. These constraints exclude full fine-tuning, LoRA-style adaptation, and document-side adaptation such as Search Adaptor from our primary comparison. We instantiate them as follows. **Zero-shot Retrieval** uses the frozen document embedder without adaptation. **Embedding Adapter** trains a query-side linear adapter using only labeled pairs, serving as the closest supervised-only fixed-index baseline[1]. **ERA w/o adapt.** uses only self-supervised alignment on unlabeled corpus documents. **ERA** is the full method, combining self-supervised alignment with supervised adaptation. In addition, to evaluate the effectiveness of asymmetric embedding spaces, we compare **Symmetric ERA**, where the same model is used for both query and document encoding, with **Asymmetric ERA**, where a stronger model (Qwen3-8B) is used for query encoding and a weaker model is used for document encoding.

Training Details. For the alignment stage, we sample 1,000 unlabeled documents from the corpus of each training task, which corresponds to only 5.2% of the full MAIR corpus in terms of token cost. We report the detailed token cost analysis in Appendix B.2. We train the retrieval adapter for 100 epochs with a learning rate of 1e-3, a weight decay of 1e-2, and a batch size of 256. We then train the adapter on labeled query-document pairs in the adaptation stage for up to 1,000 epochs with early stopping (patience of 5 epochs based on validation loss), a learning rate of 1e-5, a weight decay of 1e-4, and a batch size of 256. For negative sampling, we use TopK-PercPos (Moreira et al., 2025), which selects hard negatives based on retrieval scores after the alignment stage. We also evaluate naive top-k negative sampling and random negative sampling as ablations.[2] We provide more details about training setup in Appendix B.1.

[1] Embedding Adapter corresponds to ERA without the alignment stage.

[2] For Embedding Adapter, we use random negatives because, without alignment, hard negatives are less effective for bridging the representation gap.

Base model	Variant	**Avg**	Academic	Code	Finance	Legal	Medical	Web
BGE-M3	Zero-shot	34.03	34.81	34.85	52.44	37.01	25.25	33.28
	Embedding Adapter	26.18	21.66	30.44	42.09	25.24	18.76	26.55
	Sym. ERA w/o adapt.	34.17	35.01	33.79	53.58	38.41	25.54	33.35
	Sym. ERA	42.14	40.28	40.25	**60.77**	41.78	32.35	44.08
	Asym. ERA w/o adapt.	43.69	39.15	40.16	58.66	40.65	34.49	47.84
	Asym. ERA	**46.59**	**42.50**	**46.19**	59.27	**45.20**	**38.24**	**49.28**
OpenAI-small	Zero-shot	36.74	35.62	41.35	50.36	31.77	30.64	36.67
	Embedding Adapter	29.67	23.68	34.10	38.17	25.34	26.64	30.66
	Sym. ERA w/o adapt.	37.19	36.32	41.01	50.13	30.79	31.09	37.71
	Sym. ERA	44.95	42.34	48.33	**56.88**	37.16	38.02	46.87
	Asym. ERA w/o adapt.	46.63	41.69	49.48	52.68	34.72	40.40	50.86
	Asym. ERA	**49.10**	**45.45**	**52.81**	54.73	**39.75**	**42.34**	**52.40**
Qwen3-0.6B	Zero-shot	50.37	48.48	58.34	60.56	48.58	44.55	49.17
	Embedding Adapter	38.94	34.49	48.68	51.02	35.15	32.51	38.26
	Sym. ERA w/o adapt.	51.12	49.28	57.28	62.17	49.85	45.53	50.20
	Sym. ERA	52.19	49.41	58.49	63.57	**49.99**	**46.28**	51.76
	Asym. ERA w/o adapt.	51.44	50.11	57.53	63.13	48.35	44.70	51.07
	Asym. ERA	**53.33**	**50.72**	**59.14**	**63.86**	49.66	45.68	**54.06**
Qwen3-8B	Zero-shot	57.91	56.94	61.85	69.38	57.89	**53.75**	56.65
	Embedding Adapter	46.30	43.23	53.44	58.26	42.75	37.12	47.01
	Sym. ERA w/o adapt.	58.51	58.85	61.53	70.03	58.25	53.39	57.56
	Sym. ERA	**59.38**	**59.13**	**63.03**	**71.83**	**58.58**	53.50	**58.63**

Table 2: nDCG@10 (%) on MAIR using 20% labeled query-document pairs. **Bold** indicates the best result per model and domain. Embedding Adapter and Symmetric ERA use the same model for both query and document embedders. Asymmetric ERA uses Qwen3-8B as the query embedder and the base model as the document embedder.

4.2 Main Results (RQ1)

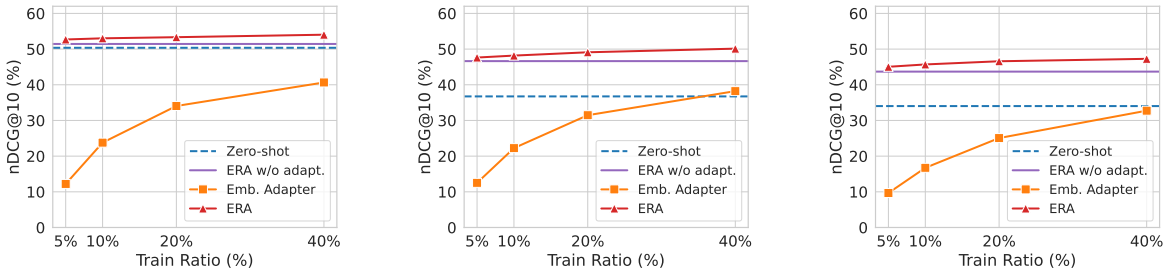
Table 2 shows the nDCG@10 results of Symmetric/Asymmetric ERA and baselines on the MAIR benchmark using 20% of labeled query-document pairs for training. We report a widely used retrieval metric, nDCG@10, as the main metric in the paper. We also evaluate other training ratios in Appendix C.1 and other metrics such as Recall@100 and MAP@100 in Appendix C.2, which show the same trend as the results in this section.

ERA enables effective asymmetric retrieval without updating the backbone models or re-indexing the corpus. As shown in the table, asymmetric ERA retrieval consistently outperforms zero-shot retrieval and symmetric ERA retrieval across all document embedders, demonstrating the effectiveness of ERA in enabling asymmetric retrieval even when the query and document embedders are from different model families. Notably, the performance improvement is more significant when the document embedder is weaker, i.e., more than **12% nDCG@10 improvement for OpenAI-small and BGE-M3**, which suggests that ERA can effectively bridge the representation gap between a stronger query embedder and a weaker document embedder, allowing for more effective retrieval adaptation without the need for re-indexing the corpus.

ERA's alignment stage is a key driver of its performance, and supervised adaptation further enhances the results. Comparing ERA w/o adapt. with zero-shot retrieval, we see that the alignment stage alone already provides consistent improvements across models and domains, highlighting the value of bridging the representation gap between the query and document embedders. This is especially evident for asymmetric ERA, where alignment substantially improves retrieval even before supervised adaptation. Full ERA then adds further gains on top of this aligned starting point, indicating that supervised adaptation captures semantic distinctions between queries and documents that alignment alone does not model. We conjecture that part of the alignment-only gain comes from training on unlabeled documents from the target corpus itself, which adapts the query-side representation to the corpus geometry and makes retrieval more compatible even before using labeled pairs.

4.3 Label Efficiency (RQ2)

To evaluate the label efficiency of ERA, we conduct experiments with varying percentages of labeled query-document pairs for training, ranging from 5% to 40%. We use Qwen3-8B as the query embedder and Qwen3-0.6B, OpenAI-small, and BGE-M3 as the document embedders for ERA and Embed-



(a) Qwen3-8B – Qwen3-0.6B (b) Qwen3-8B – OpenAI-small (c) Qwen3-8B – BGE-M3

Figure 3: nDCG@10 of zero-shot, ERA w/o adaptation, Embedding Adapter, and ERA retrieval at varying train ratios, where we use Qwen3-8B as the query embedder.

Held-out Domain \ Evaluation Domain	Academic	Code	Finance	Legal	Medical	Web
Academic	49.14	59.31	63.82	49.34	45.12	54.01
Code	50.66	55.52	63.79	49.47	45.40	54.20
Finance	51.07	59.02	60.73	48.55	45.46	54.03
Legal	50.78	58.43	63.94	48.24	45.86	54.01
Medical	50.79	59.36	63.98	49.66	43.80	54.20
Web	51.30	59.34	64.57	51.09	45.98	51.82
Zero-shot	48.48	58.34	60.56	48.58	44.55	49.17
All	50.72	59.14	63.86	49.66	45.68	54.06

(a) nDCG@10 comparison between zero-shot, ERA trained on all domains, and ERA trained on all domains except the target domain.

Training Domain \ Evaluation Domain	Academic	Code	Finance	Legal	Medical	Web
Academic	+1.01	-0.75	-1.39	-0.51	-0.49	-2.66
Code	-1.29	+0.20	-1.08	-1.27	-0.36	-2.09
Finance	-1.14	-0.92	+1.47	-1.37	-0.13	-2.81
Legal	-0.77	-0.82	-1.85	+2.11	-0.06	-2.95
Medical	-1.06	-0.77	-1.61	-1.65	+0.73	-2.88
Web	-1.49	-1.79	-2.23	-2.31	-0.75	+0.02

(b) nDCG@10 difference (ERA trained on only the target domain − ERA trained on all domains); positive values indicate that training on only the target domain yields better performance.

Figure 4: Domain generality analysis using Qwen3-8B as the query embedder and Qwen3-0.6B as the document embedder, trained on 20% of labeled query-document pairs.

ding Adapter, while zero-shot retrieval uses the same embedder for both query and document encoding. Figure 3 shows the results of asymmetric ERA and baselines across different label resource settings. Since zero-shot and ERA w/o adaptation do not use any labeled examples, their performance remains constant thus is shown as horizontal lines.

ERA consistently improves retrieval performance across low-label resource settings. ERA consistently outperforms zero-shot retrieval and Embedding Adapter across all training ratios. Even with only 5% of labeled query-document pairs, i.e., 500 queries in total, ERA still yields clear gains over zero-shot retrieval.

Alignment matters most in the lowest-label settings. Across all training ratios, ERA w/o adaptation remains above zero-shot retrieval, and the large gap between ERA and Embedding Adapter in the 5% and 10% settings indicates that alignment is what makes the supervised stage effective in the low-label regime.

4.4 Domain Generality (RQ3)

To evaluate the domain generality of ERA, we conduct two sets of experiments with Qwen3-8B as the query embedder and Qwen3-0.6B as the document embedder: **1) Out-of-Domain Experiment**, where we train ERA on all domains except the target domain, and **2) Domain-specific Experiment**, where we train ERA using only the target domain. We use 20% of labeled query-document pairs for training in both sets of experiments. The results of other train ratios are provided in Appendix C.3, which show the same trends as the results with 20% of labeled query-document pairs.

(1. Out-of-Domain) ERA is robust against out-of-domain tasks. As shown in Figure 4a, while ERA trained on all domains except the target domain shows comparable or slightly lower performance than zero-shot retrieval for the target domain (see the diagonal elements in the top six rows), it improves the performance in other domains over zero-shot. This indicates that training on other

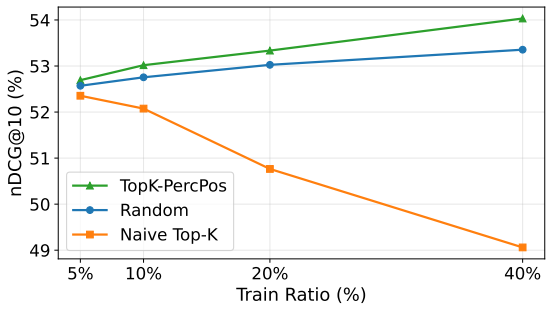


Figure 5: Ablation on negative sampling strategy with varying train ratios.

domains does not significantly degrade the performance on out-of-domain tasks. We conjecture that ERA's linear transformation is simple enough to avoid overfitting to the training domains, which allows it to maintain its generalization ability to out-of-domain tasks.

(2. Domain-specific) A general adapter trained across all domains can perform comparably with a domain-specific adapter while excelling in other domains. As shown in Figure 4b, while the performance of domain-specific adapters trained on only the target domain can be slightly better than the general adapter trained on each domain where they are trained, the performance difference is generally small across all domains. This encourages the use of a single general adapter for retrieval adaptation across different domains, avoiding the need for domain-specific adaptations and switching between different adapters for different domains, which can increase the complexity of the system, introduce maintenance overhead, and scale poorly with the number of domains.

4.5 Ablation Study (RQ4)

We conduct ablation studies on the negative sampling strategy and the number of negative samples. The hyperparameter analysis on weight decay and learning rate is provided in Appendix C.4.

4.5.1 Negative Sampling Strategy

We use Qwen3-8B as the query embedder and Qwen3-0.6B as the document embedder. We compare TopK-PercPos, naive top-k negative sampling, and random negative sampling. Naive top-k selects the top-k most similar documents to the query as negatives based on the base embedder's similarity scores, while random sampling selects k documents uniformly from the corpus.

TopK-PercPos negative sampling is more effective than naive top-k and random negative sampling. As shown in Figure 5, TopK-PercPos consistently outperforms both naive top-k and random negative sampling across different train ratios, demonstrating the importance of selecting hard negatives based on retrieval scores after the alignment stage for effective supervised adaptation. As reported by Moreira et al. (2025), naive top-k negatives can degrade performance by selecting false negatives, i.e., relevant documents mislabeled as negatives.

# Negatives	5	10	50	100	200
nDCG@10 (%)	**53.02**	52.80	52.89	52.65	52.66

Table 3: Number of negative samples ablation. We set the train ratio to 10%.

4.5.2 Number of Negative Samples

We evaluate the impact of the number of negative samples on the performance of ERA under TopK-PercPos negative sampling strategy. We vary the number of negative samples from 5 to 200 and evaluate the nDCG@10 with a train ratio of 10%.

ERA is robust against the number of negative samples. As shown in Table 3, the performance of ERA is relatively stable across different numbers of negative samples, with the best performance achieved with 5 negative samples. This suggests that ERA can be effective even with a small number of negative samples, which can reduce the computational cost of training while still maintaining strong retrieval performance.

5 Conclusion

We propose Efficient Retrieval Adapter (ERA), a novel framework for training retrieval adapters in a label-efficient manner to address the challenges of retrieval adaptation under query-document complexity asymmetry. ERA employs a two-stage training process, *self-supervised alignment training* and *supervised adaptation*, to effectively bridge the representation gap between different embedding models and the semantic gap between complex queries and simple documents. Our experiments demonstrate that ERA efficiently and effectively improves retrieval performance across various models, domains, and label resource settings, and enables asymmetric retrieval with different query and document embedders without the need for re-indexing the corpus.

Limitations

While ERA improves retrieval performance without updating backbone embedders or re-indexing the document corpus, it has several limitations. First, ERA uses a linear adapter, and we do not investigate whether non-linear or deeper adapters further improve cross-model alignment. Second, although ERA avoids full-corpus re-indexing, the alignment stage still requires embedding a small sampled set of unlabeled documents with both query and document embedders. As we mentioned in Section 4, we sample only 1,000 documents for each training task, which is a substantially smaller number than the size of the full corpus. Third, while we evaluate ERA on a wide range of retrieval tasks across different domains, we do not evaluate it on other retrieval settings such as multimodal retrieval and cross-lingual retrieval, which can be interesting directions for future work. Lastly, while there are many potential strategies for the retrieval performance improvement under the setting of frozen backbone embedders and fixed document index such as sparse and dense fusion, re-ranking, document-side adaptation, and few-shot prompting, we focus on the retrieval adapter approach in this work. We leave the exploration of the combination of retrieval adapters with other strategies for future work because they are complementary to each other and can potentially lead to further retrieval performance improvement rather than a direct comparison between them.

Ethical Considerations

Our work focuses on improving retrieval performance by training retrieval adapters in a label-efficient manner. While we do not foresee any direct ethical concerns with our method, it is important to consider the potential downstream applications of retrieval systems that utilize ERA, such as information retrieval, question answering, and recommendation systems. These applications may have ethical implications related to user privacy, bias in retrieved results, and the potential for misuse in spreading misinformation. Therefore, it is crucial for practitioners to carefully evaluate and mitigate any ethical concerns when deploying retrieval systems that incorporate ERA, and to ensure that the benefits of improved retrieval performance are balanced with responsible and ethical use.

We made use of AI tools such as ChatGPT and Copilot to support coding and refining this paper, but all content was carefully reviewed and edited by us to ensure it adheres to our standards and aligns with our research objectives.

References

Alan Chen, Jack Merullo, Alessandro Stolfo, and Ellie Pavlick. 2025. Transferring linear features across language models with model stitching. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. 2023. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. 2024. Position: The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*.

Rishi Dev Jha, Collin Zhang, Vitaly Shmatikov, and John Xavier Morris. 2025. Harnessing the universal geometry of embeddings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Minsang Kim and Seung Jun Baek. 2025. Syntriever: How to train your retriever with synthetic data from LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2523–2539, Albuquerque, New Mexico. Association for Computational Linguistics.

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*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Seiji Maekawa, Hayate Iso, Sairam Gurajada, and Nikita Bhutani. 2024. Retrieval helps or hurts? a deeper dive into the efficacy of retrieval augmentation to language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5506–5521, Mexico City, Mexico. Association for Computational Linguistics.

Gabriel de Souza P. Moreira, Radek Osmulski, Mengyao Xu, Ronay Ak, Benedikt Schifferer, and Even Oldridge. 2025. Improving text embedding models with positive-aware hard-negative mining. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, page 2169–2178, New York, NY, USA. Association for Computing Machinery.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

OpenAI. 2024. New embedding models and api updates. *Technical report*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Suvansh Sanjeev and Anton Troynikov. 2024. Embedding adapters. *Chroma Technical Report*.

Ferdinand Schlatt, Maik Fröbe, Harrisen Scells, Shengyao Zhuang, Bevan Koopman, Guido Zuccon, Benno Stein, Martin Potthast, and Matthias Hagen. 2025. Rank-distillm: Closing the effectiveness gap between cross-encoders and llms for passage re-ranking. In *Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part III*, page 323–334, Berlin, Heidelberg. Springer-Verlag.

Weiwei Sun, Zhengliang Shi, Wu Jiu Long, Lingyong Yan, Xinyu Ma, Yiding Liu, Min Cao, Dawei Yin, and Zhaochun Ren. 2024. MAIR: A massive benchmark for evaluating instructed retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14044–14067, Miami, Florida, USA. Association for Computational Linguistics.

Chihiro Taguchi, Seiji Maekawa, and Nikita Bhutani. 2025. Efficient context selection for long-context QA: No tuning, no iteration, just adaptive-k. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20105–20130, Suzhou, China. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Coen van den Elsen, Francien Barkhof, Thijmen Nijdam, Simon Lupart, and Mohammad Aliannejadi. 2025. Reproducing nevir: Negation in neural information retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 3346–3356, New York, NY, USA. Association for Computing Machinery.

Harshil Vejendla. 2025. Drift-adapter: A practical approach to near zero-downtime embedding model upgrades in vector databases. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15938–15949, Suzhou, China. Association for Computational Linguistics.

Orion Weller, Benjamin Chang, Sean MacAvaney, Kyle Lo, Arman Cohan, Benjamin Van Durme, Dawn 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 (Volume 1: Long Papers)*, pages 11926–11942, Albuquerque, New Mexico. Association for Computational Linguistics.

Jinsung Yoon, Yanfei Chen, Sercan Arik, and Tomas Pfister. 2024. Search-adaptor: Embedding customization for information retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12230–12247, Bangkok, Thailand. Association for Computational Linguistics.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang,

Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Embedding Models

We summarize the details of the models used in our experiments in Table 4. All models are accessed through HuggingFace except for OpenAI-small, which is accessed through the OpenAI API. We adhere their intended usage and licenses for all models since the models are used for retrieval purposes in our experiments.

B Experiment Details

We provide additional details of our experiments in this section, including the hyperparameters used for training ERA and baselines, the negative sampling strategy, and the evaluation metrics.

B.1 Training Details

We use the AdamW optimizer (Loshchilov and Hutter, 2019) in both stages. For the label-based training stage, we apply linear warmup for the first 10% of training steps. When using the InfoNCE loss, we set the temperature to 0.05.

For the adaptation stage, we test different learning rates (1e-3, 5e-4, 1e-4, 5e-5, 1e-5) and weight decays (1e-2, 1e-3, 1e-4) with Qwen3-8B as a query embedder and Qwen3-0.6B as a document embedder, trained on 10% of labeled query-document pairs. While we find that performance is generally stable across these hyperparameters, we use for all experiments the hyperparameter set described in Section 4.1, which yields the best retrieval evaluation performance. We also tune the hyperparameters for the Embedding Adapter and use the best set, 1e-3 for the learning rate and 1e-4 for weight decay, which we use for all experiments. We show the hyperparameter search results in the Appendix C.4.

B.2 Token Cost Analysis

The alignment stage requires encoding only a small subset of the MAIR corpus rather than the full document collection. We compute token counts using the `cl100k_base` tokenizer. Across MAIR, the full corpus contains 8,958,270 documents with 2,549,625,873 tokens in total, whereas the alignment sample used in our experiments contains 114,858 documents with 132,526,211 tokens. This corresponds to 1.28% of the documents and 5.20% of the total corpus tokens. Therefore, the additional cost of the alignment stage is substantially small relative to full-corpus processing.

C Additional Results

C.1 Main Results with Varying Train Ratios

We show the nDCG@10 results of ERA and baselines on the MAIR benchmark using varying percentages of labeled query-document pairs for training, including 5%, 10%, and 40% in Tables 5, 6, and 7, respectively. We observe the same trend that ERA consistently improves retrieval performance across all models in most domains, compared with zero-shot retrieval and Embedding Adapter, demonstrating the effectiveness of ERA in improving retrieval performance under different label resource settings.

C.2 Main Results with Recall@100 and MAP@100

In addition to nDCG@10, we also evaluate the performance of ERA and baselines using Recall@100 and MAP@100, which are shown in Table 8 and Table 9, respectively.

C.3 Domain Generality with Varying Train Ratios

C.3.1 Out-of-Domain Experiment.

We show the nDCG@10 results of domain generality analysis with varying percentages of labeled query-document pairs for training, including 5%, 10%, and 40% in Figures 6 and 7, respectively. We observe the same trend that ERA trained on all domains except the target domain shows comparable or slightly lower performance than zero-shot retrieval for the target domain while largely improving the performance in other domains.

C.3.2 Domain-specific Experiment.

We show the nDCG@10 difference between ERA trained on all domains and ERA trained on the target domain with varying percentages of labeled query-document pairs for training, including 5%, 10%, and 40% in Figures 7a, 7b, and 7c, respectively. The experimental results show the same trend that while the performance of domain-specific adapters trained on only the target domain can be slightly better than the general adapter trained on each domain where they are trained, the

[1] https://openai.com/policies/services-agreement/ [Accessed: May 22, 2026]

Model	Size	Context	HuggingFace / API	License
Qwen3-8B (Zhang et al., 2025)	8B	32k	`Qwen/Qwen3-Embedding-8B`	Apache license 2.0
Qwen3-0.6B (Zhang et al., 2025)	0.6B	32k	`Qwen/Qwen3-Embedding-0.6B`	Apache license 2.0
OpenAI-small (OpenAI, 2024)	-	8k	`text-embedding-3-small`	OpenAI Service Terms[1]
BGE-M3 (Chen et al., 2024)	0.6B	8k	`BAAI/bge-m3`	MIT

Table 4: Models used in experiments. Model sizes are not publicly disclosed (-).

Base model	Variant	**Avg**	Academic	Code	Finance	Legal	Medical	Web
BGE-M3	Zero-shot	34.03	34.81	34.85	52.44	37.01	25.25	33.28
	Emb. Adapter	7.48	6.59	14.36	8.92	6.65	4.04	6.60
	Sym. ERA w/o adapt.	34.17	35.01	33.79	53.58	38.41	25.54	33.35
	Sym. ERA	39.80	38.75	39.62	58.45	39.60	29.65	41.02
	Asym. ERA w/o adapt.	43.69	39.15	40.16	**58.66**	40.65	34.49	47.84
	Asym. ERA	**45.04**	**41.25**	**42.48**	58.47	**43.31**	**35.72**	**48.65**
OpenAI-small	Zero-shot	36.74	35.62	41.35	50.36	31.77	30.64	36.67
	Emb. Adapter	8.36	6.47	15.62	4.85	5.82	5.75	8.46
	Sym. ERA w/o adapt.	37.19	36.32	41.01	50.13	30.79	31.09	37.71
	Sym. ERA	41.51	40.65	44.44	**53.63**	34.21	35.38	42.65
	Asym. ERA w/o adapt.	46.63	41.69	49.48	52.68	34.72	40.40	50.86
	Asym. ERA	**47.63**	**43.61**	**50.11**	52.80	**37.65**	**40.90**	**51.63**
Qwen3-0.6B	Zero-shot	50.37	48.48	**58.34**	60.56	48.58	44.55	49.17
	Emb. Adapter	14.83	12.21	23.04	17.22	9.31	9.53	15.49
	Sym. ERA w/o adapt.	51.12	49.28	57.28	62.17	**49.85**	**45.53**	50.20
	Sym. ERA	51.38	48.96	57.35	62.46	49.67	45.38	50.93
	Asym. ERA w/o adapt.	51.44	50.11	57.53	63.13	48.35	44.70	51.07
	Asym. ERA	**52.69**	**50.26**	57.86	**64.09**	49.62	45.36	**53.21**
Qwen3-8B	Zero-shot	57.91	56.94	61.85	69.38	57.89	**53.75**	56.65
	Emb. Adapter	19.77	17.72	26.67	19.25	12.84	12.31	22.18
	Sym. ERA w/o adapt.	58.51	**58.85**	61.53	70.03	**58.25**	53.39	57.56
	Sym. ERA	**58.53**	58.66	**62.03**	**70.61**	57.38	52.44	**57.91**

Table 5: nDCG@10 (%) on the MAIR benchmark with 5% of labeled query-document pairs.

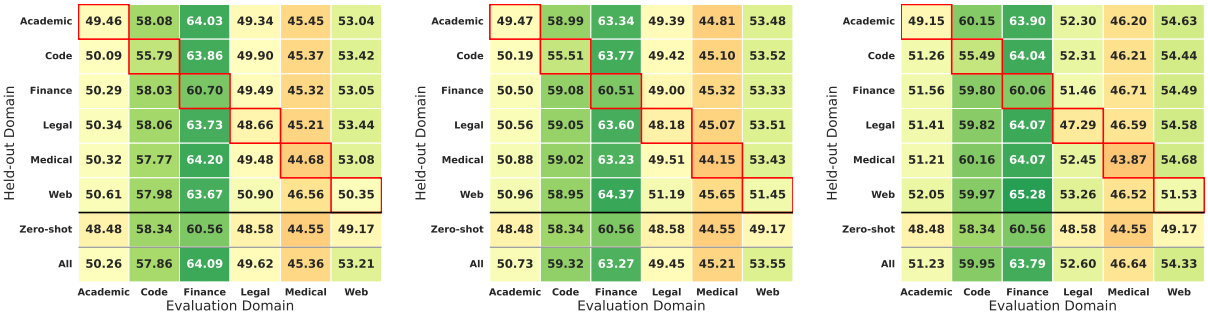
performance difference is generally small across all domains.

C.4 Hyperparameter Search Results

We show the hyperparameter search results for the learning rate and weight decay of the adaptation stage in Table 10. We observe that the performance is generally stable, i.e., 0.27% nDCG@10 difference between the best and worst configurations, which suggests that ERA is not sensitive to the choice of these hyperparameters.

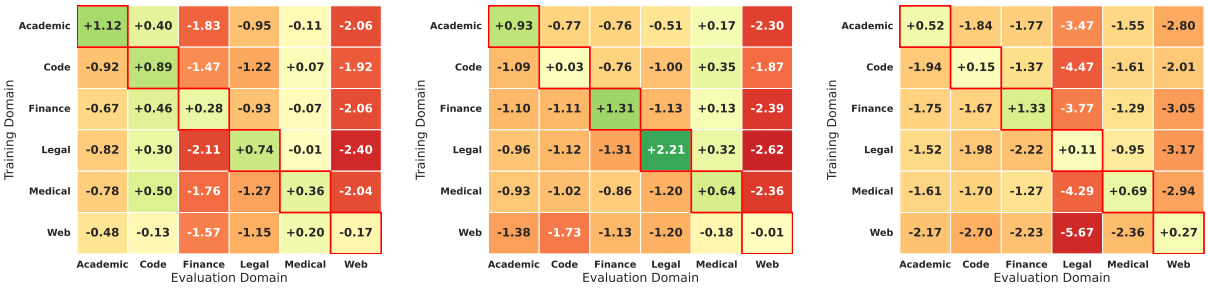
Base model	Variant	**Avg**	Academic	Code	Finance	Legal	Medical	Web
BGE-M3	Zero-shot	34.03	34.81	34.85	52.44	37.01	25.25	33.28
	Embedding Adapter	17.37	14.68	22.16	27.52	15.27	11.67	17.51
	Sym. ERA w/o adapt.	34.17	35.01	33.79	53.58	38.41	25.54	33.35
	Sym. ERA	40.77	39.57	38.94	**59.36**	40.47	31.15	42.43
	Asym. ERA w/o adapt.	43.69	39.15	40.16	58.66	40.65	34.49	47.84
	Asym. ERA	**45.70**	**42.01**	**44.20**	59.33	**44.13**	**37.16**	**48.59**
OpenAI-small	Zero-shot	36.74	35.62	41.35	50.36	31.77	30.64	36.67
	Embedding Adapter	20.63	17.98	26.01	21.56	17.35	17.88	21.12
	Sym. ERA w/o adapt.	37.19	36.32	41.01	50.13	30.79	31.09	37.71
	Sym. ERA	43.26	41.56	46.89	**56.05**	35.91	36.84	44.42
	Asym. ERA w/o adapt.	46.63	41.69	49.48	52.68	34.72	40.40	50.86
	Asym. ERA	**48.18**	**44.57**	**50.93**	53.52	**38.34**	**41.65**	**51.84**
Qwen3-0.6B	Zero-shot	50.37	48.48	58.34	60.56	48.58	44.55	49.17
	Embedding Adapter	28.71	23.47	39.04	37.71	22.41	22.62	28.92
	Sym. ERA w/o adapt.	51.12	49.28	57.28	62.17	**49.85**	45.53	50.20
	Sym. ERA	51.62	49.49	57.85	62.74	49.52	**45.96**	50.95
	Asym. ERA w/o adapt.	51.44	50.11	57.53	63.13	48.35	44.70	51.07
	Asym. ERA	**53.02**	**50.73**	**59.32**	**63.27**	49.45	45.21	**53.55**
Qwen3-8B	Zero-shot	57.91	56.94	61.85	69.38	57.89	**53.75**	56.65
	Embedding Adapter	35.32	33.45	43.34	43.45	29.77	24.64	36.88
	Sym. ERA w/o adapt.	58.51	**58.85**	61.53	70.03	**58.25**	53.39	57.56
	Sym. ERA	**58.92**	58.61	**63.26**	**70.75**	57.72	53.33	**58.03**

Table 6: nDCG@10 (%) on the MAIR benchmark with 10% of labeled query-document pairs.



(a) 5% train ratio (b) 10% train ratio (c) 40% train ratio

Figure 6: nDCG@10 comparison between zero-shot, ERA trained on all domains, and ERA trained on all domains except the target domain with varying train ratios, where we use Qwen3-8B as the query embedder and Qwen3-0.6B as the document embedder.



(a) 5% train ratio (b) 10% train ratio (c) 40% train ratio

Figure 7: nDCG@10 difference between ERA trained on all domains and ERA trained on the target domain with varying train ratios, where we use Qwen3-8B as the query embedder and Qwen3-0.6B as the document embedder.

Base model	Variant	**Avg**	Academic	Code	Finance	Legal	Medical	Web
BGE-M3	Zero-shot	34.03	34.81	34.85	52.44	37.01	25.25	33.28
	Embedding Adapter	33.24	28.17	37.72	48.70	33.55	24.28	34.05
	Sym. ERA w/o adapt.	34.17	35.01	33.79	53.58	38.41	25.54	33.35
	Sym. ERA	43.65	41.18	43.27	**61.57**	43.39	34.46	45.15
	Asym. ERA w/o adapt.	43.69	39.15	40.16	58.66	40.65	34.49	47.84
	Asym. ERA	**47.27**	**42.91**	**47.89**	60.50	**48.03**	**38.05**	**49.49**
OpenAI-small	Zero-shot	36.74	35.62	41.35	50.36	31.77	30.64	36.67
	Embedding Adapter	37.21	30.74	42.68	47.38	31.47	32.23	38.71
	Sym. ERA w/o adapt.	37.19	36.32	41.01	50.13	30.79	31.09	37.71
	Sym. ERA	46.37	42.37	50.59	**57.32**	39.14	39.98	48.24
	Asym. ERA w/o adapt.	46.63	41.69	49.48	52.68	34.72	40.40	50.86
	Asym. ERA	**50.12**	**46.25**	**54.67**	55.81	**42.62**	**43.40**	**52.81**
Qwen3-0.6B	Zero-shot	50.37	48.48	58.34	60.56	48.58	44.55	49.17
	Embedding Adapter	43.76	38.92	54.98	54.03	39.69	36.76	43.22
	Sym. ERA w/o adapt.	51.12	49.28	57.28	62.17	49.85	45.53	50.20
	Sym. ERA	52.57	49.46	59.13	**64.03**	51.73	**46.85**	51.80
	Asym. ERA w/o adapt.	51.44	50.11	57.53	63.13	48.35	44.70	51.07
	Asym. ERA	**54.03**	**51.23**	**59.95**	63.79	**52.60**	46.64	**54.33**
Qwen3-8B	Zero-shot	57.91	56.94	61.85	69.38	57.89	53.75	56.65
	Embedding Adapter	51.79	49.51	59.85	63.16	50.43	43.10	51.42
	Sym. ERA w/o adapt.	58.51	58.85	61.53	70.03	58.25	53.39	57.56
	Sym. ERA	**59.99**	**59.20**	**64.57**	**71.82**	**59.16**	**54.75**	**58.96**

Table 7: nDCG@10 (%) on the MAIR benchmark with 40% of labeled query-document pairs.

Base model	Variant	**Avg**	Academic	Code	Finance	Legal	Medical	Web
BGE-M3	Zero-shot	55.50	69.34	68.05	84.97	58.20	20.99	54.44
	Embedding Adapter	51.49	63.09	65.85	75.59	56.98	19.53	49.82
	Sym. ERA w/o adapt.	55.28	70.35	66.55	84.28	58.43	20.76	54.27
	Sym. ERA	64.17	78.53	73.59	89.03	66.86	26.56	65.78
	Asym. ERA w/o adapt.	65.91	76.80	79.19	89.52	65.44	26.14	68.86
	Asym. ERA	**68.62**	**80.58**	**81.22**	**90.20**	**69.77**	**30.65**	**70.81**
OpenAI-small	Zero-shot	60.99	74.67	75.96	88.69	62.11	27.82	59.29
	Embedding Adapter	57.71	65.21	73.27	81.62	60.62	27.14	56.91
	Sym. ERA w/o adapt.	61.18	75.88	75.26	87.99	61.02	27.01	60.23
	Sym. ERA	67.83	81.18	81.18	92.02	69.25	32.90	67.85
	Asym. ERA w/o adapt.	68.62	79.34	83.84	91.87	66.55	33.35	69.75
	Asym. ERA	**70.98**	**83.92**	**84.50**	**93.52**	**71.59**	**36.00**	**71.48**
Qwen3-0.6B	Zero-shot	70.24	82.39	88.53	91.87	71.11	36.58	69.01
	Embedding Adapter	63.52	75.18	81.86	85.36	66.96	28.84	62.22
	Sym. ERA w/o adapt.	70.44	83.05	88.10	91.54	70.99	36.46	69.54
	Sym. ERA	72.00	84.48	88.03	91.77	73.48	**37.39**	71.90
	Asym. ERA w/o adapt.	71.33	83.95	88.31	93.35	71.97	35.24	71.24
	Asym. ERA	**73.56**	**86.31**	**89.02**	**93.51**	**74.06**	37.13	**74.39**
Qwen3-8B	Zero-shot	76.22	87.62	90.57	96.69	76.78	**43.74**	76.34
	Embedding Adapter	70.24	81.84	85.47	93.12	72.58	34.82	70.31
	Sym. ERA w/o adapt.	76.36	87.57	90.42	97.46	76.83	42.77	76.95
	Sym. ERA	**77.08**	**88.10**	**90.73**	**97.87**	**79.05**	43.06	**77.76**

Table 8: Recall@100 (%) on the MAIR benchmark with 20% of labeled query-document pairs.

Base model	Variant	**Avg**	Academic	Code	Finance	Legal	Medical	Web
BGE-M3	Zero-shot	26.93	30.23	29.99	47.80	30.66	11.83	26.39
	Embedding Adapter	19.31	17.34	25.66	37.17	21.29	7.43	18.91
	Sym. ERA w/o adapt.	27.01	30.49	29.40	49.03	30.22	11.70	26.66
	Sym. ERA	33.29	35.28	35.05	**56.08**	**34.70**	14.60	35.03
	Asym. ERA w/o adapt.	33.34	33.43	34.43	53.55	30.05	13.44	37.63
	Asym. ERA	**36.20**	**36.70**	**40.34**	54.55	33.94	**16.20**	**39.44**
OpenAI-small	Zero-shot	28.97	30.25	36.35	46.19	27.65	14.12	29.07
	Embedding Adapter	22.04	19.19	29.29	33.80	20.66	11.96	22.56
	Sym. ERA w/o adapt.	29.29	30.83	36.05	46.01	26.59	13.43	30.23
	Sym. ERA	35.80	36.43	42.58	**52.50**	32.07	17.81	37.97
	Asym. ERA w/o adapt.	36.70	35.48	43.96	48.46	30.15	18.25	40.74
	Asym. ERA	**39.22**	**39.15**	**47.22**	50.17	**35.11**	**19.76**	**42.64**
Qwen3-0.6B	Zero-shot	39.97	42.48	52.62	55.83	39.11	21.88	39.21
	Embedding Adapter	29.90	28.59	43.32	46.57	27.99	13.51	29.50
	Sym. ERA w/o adapt.	40.59	43.26	52.00	57.76	39.40	21.61	40.38
	Sym. ERA	41.88	43.60	53.62	58.77	**40.68**	**22.07**	42.17
	Asym. ERA w/o adapt.	40.71	44.01	51.78	58.45	37.87	20.61	41.06
	Asym. ERA	**42.70**	**44.58**	**54.14**	**59.11**	39.89	21.51	**43.92**
Qwen3-8B	Zero-shot	47.08	50.76	56.21	63.82	47.24	**28.38**	47.02
	Embedding Adapter	36.45	36.72	47.69	52.14	35.39	16.64	37.50
	Sym. ERA w/o adapt.	47.65	52.53	56.01	64.75	47.02	27.21	48.21
	Sym. ERA	**48.55**	**52.56**	**57.80**	**66.99**	**47.64**	27.14	**49.26**

Table 9: MAP@100 (%) on the MAIR benchmark with 20% of labeled query-document pairs.

WD \ LR	10^{-5}	5×10^{-5}	10^{-4}	5×10^{-4}	10^{-3}
10^{-2}	52.83	52.59	52.60	52.58	52.60
10^{-3}	52.83	52.59	52.57	52.57	52.58
10^{-4}	**52.84**	52.59	52.56	52.57	52.60

Table 10: Hyperparameter search results (nDCG@10 %). Rows: weight decay (WD), columns: learning rate (LR). **Bold** = best configuration.