Back to Basics: A Simple Recipe for Improving Out-of-Domain Retrieval in Dense Encoders

Hyunji Lee[κ,*] **Luca Soldaini**[α] **Arman Cohan**[γ,α] **Minjoon Seo**[κ] **Kyle Lo**[α]
[κ]KAIST AI [α]Allen Institute for AI [γ]Yale University
hyunji.amy.lee@kaist.ac.kr {lucas, kylel}@allenai.org

Abstract

Prevailing research practice today often relies on training dense retrievers on existing large datasets such as MSMARCO and then experimenting with ways to improve zero-shot generalization capabilities to unseen domains. While prior work has tackled this challenge through resource-intensive steps such as data augmentation, architectural modifications, increasing model size, or even further base model pretraining, comparatively little investigation has examined whether the training procedures themselves can be improved to yield better generalization capabilities in the resulting models. In this work, we recommend a simple recipe for training dense encoders: Train on MSMARCO with parameter-efficient methods, such as LoRA, and opt for using in-batch negatives unless given well-constructed hard negatives. We validate these recommendations using the BEIR benchmark and find results are persistent across choice of dense encoder and base model size and are complementary to other resource-intensive strategies for out-of-domain generalization such as architectural modifications or additional pretraining. We hope that this thorough and impartial study around various training techniques, which augments other resource-intensive methods, offers practical insights for developing a dense retrieval model that effectively generalizes, even when trained on a single dataset.

github.com/amy-hyunji/lora-for-retrieval

1 Introduction

Dense neural retrieval methods have been proven to be generally effective in many Information Retrieval (IR) tasks (Karpukhin et al., 2020; Izacard et al., 2021; Ni et al., 2021a). These methods use learned neural encoders to obtain dense vector representations of text and the relevance of passages for any given query is estimated by computing the dot product between their encodings. Dense approaches can outperform traditional retrieval techniques (*e.g.*, BM25 (Robertson & Jones, 1976)), as they estimate similarity beyond syntactic matching (Lin et al., 2022).

Neural retrieval models are effective rankers in domains for which large supervised datasets exist (*e.g.*, MSMARCO (Campos et al., 2016) or Google NQ (Kwiatkowski et al., 2019)). Conversely, they might struggle to generalize to settings they have not been trained on, leading to challenges in handling out-of-domain tasks (Thakur et al., 2021a; Ren et al., 2022; Lupart et al., 2023). In most real-world applications, supervision data is not available; whereas, retrieval models play a key role in the nascent field of augmented language models across many new exciting scenarios (Mialon et al., 2023). Thus, it is essential to analyze techniques that can improve generalization to unseen domains.

[*] Work performed during internship at AI2.

Many approaches have been proposed to tackle out-of-domain generalization. For example, **data augmentation** approaches use weak supervision or auxiliary systems to bridge to unseen tasks (Dai et al., 2022; Bonifacio et al., 2022; Saad-Falcon et al., 2023; Lin et al., 2023). Other works introduce **novel architectures** that assess relevance at the token-level multi embeddings rather than employing a single embedding per passage (Khattab & Zaharia, 2020; Formal et al., 2021; Lee et al., 2023). Moreover, empirical observations suggest that **increasing the model size** leads to better out-of-domain performance (Ni et al., 2021b). While these approaches show significant improvements, they require *additional* resource-intensive steps: data augmentation requires additional steps of generating new datasets, fine-grained token-level interaction requires higher inference costs with a large storage footprint, and larger model size requires more GPU memory during training and inference. Finally, recently proposed approaches use contrastive losses to **pretrain** domain-specific encoders without explicit supervision (Izacard et al., 2021; Xu et al., 2022). These methods, while more effective than statistical IR techniques, still underperform supervised rankers unless they are then also fine-tuned on large supervised datasets like MSMARCO. In fact, despite being out-of-distribution for many real-world tasks, large supervised collections remain critical to improving zero-shot retrieval, particularly for larger and well-trained rankers (Ni et al., 2021b; Rosa et al., 2022; Lin et al., 2023; Weller et al., 2023).

Despite the vast body of work on improving out-of-domain generalization through resource-intensive steps like data augmentation, novel architectures, and pretraining, we notice comparatively less work has been done on the **training strategies** themselves commonly used to fine-tune rankers on large supervised datasets. In this work, we aim to answer the following question: **when training dense models on large data collections, what procedures lead to better out-of-domain retrieval performance?** In particular, we aim to address the following research questions:

- **(RQ1)** Do parameter-efficient fine-tuning (PEFT) methods, such as LoRA (Hu et al., 2021), improve performance on out-of-domain tasks?
- **(RQ2)** How might we modify the design of our batches for better out-of-domain performance?
- **(RQ3)** To what extent do our recommendations complement other resource-intensive techniques that improve out-of-domain generalization?

by identifying key design decisions for training dense retrieval models and conducting a series of carefully designed experiments that isolate the effects of these various decisions (§3).

Addressing RQ1 in §4, we find that **LoRA**, one of the most widely-used PEFT techniques, leads to better out-of-domain generalization performance compared to **full parameter tuning**. Simultaneously, we validate an intuitive tradeoff—full parameter tuning still outperforms LoRA on in-domain settings. Nevertheless, we provide further analysis showing that even when considering this **tradeoff** between in- and out-of-domain performance, LoRA may provide more than it gives up. We recommend LoRA as a suitable training approach when training a model that expects high performance in both in-domain and out-of-domain settings.

Further, addressing RQ2 in §5, we find that contrary to their well-established benefit in in-domain settings, mined **hard negatives** may hurt out-of-domain retrieval performance unless selected with great care. On the other hand, increasing the number of **in-batch negatives** is consistently beneficial for out-of-domain performance, a finding that can be opportunistically employed by adopting PEFT as our fine-tuning strategy. Specifically, under identical GPU configurations, increasing the in-batch size typically yields more robust performance compared to adding hard negatives.

Finally, addressing RQ3 in §6, we find that our learnings complement other popular yet resource-intensive techniques for enhancing out-of-domain performance, such as adopting **larger base models**, novel retriever architectures (e.g., **late interaction models**), and additional **contrastive pretraining** of the base model.

Our results show consistent trends across several encoder-only base models and common dual-encoder retriever architectures. Combining the findings from RQ1 and RQ2, we speculate that common full parameter fine-tuning practices are prone to **overfitting** large popular datasets like MSMARCO. Finally, taking all our findings together, our work provides simple, actionable takeaways that yield better out-of-domain generalization for neural retrieval models that we recommend as complements to other resource-intensive methods.

2 Background and Related Work

Out-of-domain generalization in information retrieval Many data augmentation techniques have been proposed as a means to offset limited training data availability (Dai et al., 2022; Bonifacio et al., 2022; Saad-Falcon et al., 2023; Lin et al., 2023). Fully unsupervised techniques can also be used to circumvent the lack of domain-specific supervised data (Izacard et al., 2021; Xu et al., 2022). Finally, modifications to the model itself have been explored, such as combining sparse retrieval (Formal et al., 2021; Gao et al., 2021a), late-interaction learning (Lee et al., 2023; Khattab & Zaharia, 2020), or using larger encoder models (Ni et al., 2021b; Neelakantan et al., 2022; Ma et al., 2023). However, as mentioned in §1, all methods come with computational trade-offs: they might require additional (expensive) training steps or have slower inference speeds. Therefore, our study aims to investigate various approaches that maximize the advantages of the dense retrieval approach while improving its generalizability performance, addressing these practical challenges.

Parameter efficient fine-tuning in information retrieval While standard fine-tuning of neural models typically entails training all model parameters, recent studies highlight the advantages of parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019; Hu et al., 2021; Ben-Zaken et al., 2021). PEFT selectively updates a subset of model parameters or adds additional ones while keeping existing parameters fixed. These approaches offer several benefits, including reduced storage requirements, shorter training times, and lower GPU memory costs. Moreover, by not updating or only partially updating original parameters, PEFT helps prevent catastrophic forgetting and maintains robust performance in continual training (Wang et al., 2020; Jin et al., 2021; Yoon et al., 2023; Jang et al., 2021). Due to the benefits and its competitive performance compared to full parameter tuning across various tasks, PEFT is widely used in machine learning (Liu et al., 2022a; Ustun & Stickland, 2022).

In information retrieval, the standard of full parameter tuning (Karpukhin et al., 2020; Lee et al., 2022b; Izacard et al., 2021; Formal et al., 2021; Xiong et al., 2020) is also giving way as PEFT gains traction. Litschko et al. (2022) examined the use of PEFT for multilingual information retrieval; Ma et al. (2022) studied the use of PEFT to improve in-domain search capabilities; Pal et al. (2023) applied PEFT to sparse retrieval systems, while Jung et al. (2021) investigated improving hybrid retrieval; Yoon et al. (2023) showed that PEFT can help adapt generative retrieval systems to new corpora. Tam et al. (2022) is closest to our work in that they also study PEFT for out-of-domain generalization. We distinguish our work from theirs in several ways: (1) the scope of our study extends beyond the application of PEFT as we also consider the role of batch design (e.g., in-batch and hard negatives, §5) and the effect of base models (e.g., model size, continued pretraining, §6), (2) our experimental methodology (§3) controls for differing amounts of training data seen by the base model; that is, we train different dense retrievers within an experiment group from the same base model whereas Tam et al. (2022) use different public model checkpoints fine-tuned on different datasets for different retrievers, and (3) for PEFT method, we focus on LoRA, which they did not include in their work; they instead focus on P-tuning v2 (Liu et al., 2022b) for zero-shot out-of-domain retrieval evaluations, as well as perform a wider sweep of different PEFT methods for in-domain evaluation, which we did not perform.

Batch design in information retrieval When training dense encoders in a contrastive manner, it is not feasible to compute the loss across a corpus at each training step; instead, the loss is computed over a smaller subset of positive and negative pairs. Consequently, many works adopt sampling strategies aimed at improving how training batches are constructed (Zhong et al., 2022; Lee et al., 2019; Min et al., 2022; Lee et al., 2022a; Qu et al., 2020). When organizing each batch, two aspects are widely known to be key for retrieval performance: (1) using relevant passages for one query as contrastive samples for other queries in the same batch, known as **"in-batch" negatives** (Lindgren et al., 2021; Xiong et al., 2020; Gillick et al., 2019), and (2) mining additional passages that are challenging to distinguish from relevant passages, known as **"hard" negatives** (Karpukhin et al., 2020; Izacard et al., 2021; Luan et al., 2020; Qu et al., 2020; Wu et al., 2019). Although the role of batch design has been widely studied for in-domain scenarios, there has not been an exploration of how such strategies translate to out-of-domain performance. Many works focusing on out-of-domain generalization tend to report the use of hard negatives without associated investigation to validate their use (Lee et al., 2023; Izacard et al., 2021; Khattab & Zaharia, 2020; Gao et al., 2021a). We believe our work is one of the first to question this practice as the de facto standard.

3 Experimental Methodology

We carefully design an experimental procedure to study our various hypotheses. Specifically, we identify several key decision points when building a dense retrieval model. We make sure to define decision points such that: (1) we believe different design options will meaningfully impact end retrieval performance, and (2) we have the ability to experiment with different design variations at a single decision point while keeping others fixed, thereby allowing us to study the *isolated* effect of those decisions at that single decision point. In this work, our selected decision points are: (1) pretrained base model, (2) dense retriever architecture, (3) fine-tuning strategy, (4) how batches are constructed, and (5) datasets for training and testing.[1]

Pretrained base models We center our experimentation around BERT (Devlin et al., 2019), the most popular choice of encoder-only base model (Karpukhin et al., 2020; Izacard et al., 2021; Khattab & Zaharia, 2020). Additionally, focusing on BERT gives us the ability to study how controlled changes in the base model affect retrieval performance. For example, one can repeat an experimental run using different variants of BERT weights: (1) Different model sizes (e.g., Tiny, Small, Base, Large) to study the effect of scale, (2) RoBERTa (Liu et al., 2019) to study variation as a result of a different pretraining strategy rather than significant model architectural changes, and (3) Contriever (Izacard et al., 2021) to study variation as a result of retrieval-motivated continued pretraining using a contrastive loss.

Dense retriever architectures There are various architectural designs of encoder-only retrievers worth studying. In particular, we consider **`dual encoder`** architectures, which use an encoder-only model to embed queries and documents such that their pairwise relevance can be derived by proximity (e.g., cosine similarity) in the shared embedding space. In our work, we focus on three widely used designs. In the **`asymmetric`** dual encoder, the weights of the query and document encoders are not shared. Following the architecture design of Karpukhin et al. (2020), we use the first token embedding (the CLS token embedding) as the representative embedding. In the **`symmetric`** dual encoder, the weights of the query and document encoders are shared. Following the architecture design of Izacard et al. (2021), we use the mean embedding (average of all token embeddings) as the representative embedding. In the **`late interaction`** dual encoder (Lee et al., 2023; Khattab & Zaharia, 2020), we use *multiple* token embeddings as representative embeddings, unlike the symmetric and asymmetric dual encoders which use a *single* representative embedding. We follow the architecture design of Khattab & Zaharia (2020) closely, including sharing the weights of the query and document encoders and use of the MaxSim operation to score similarity between each query against a bag of documents.

Fine-tuning strategies We consider both full parameter tuning (FT) and PEFT for fine-tuning experiments. Among various PEFT methods (Ben-Zaken et al., 2021; Liu et al., 2022a; Ma et al., 2023), we focus on the low-rank adaptation (LoRA) (Hu et al., 2021) method due to its wide usage (Dettmers et al., 2023; Chen

[1] We recognize that many of these design options are tightly coupled and difficult to fully study in isolation of each other. For example, a particular choice of retriever architecture will preclude certain base model choices as well as certain fine-tuning strategies.

et al., 2023; Xu et al., 2023). LoRA keeps the pretrained model parameters fixed and integrates trainable rank decomposition matrices into each layer of the Transformer architecture. A key advantage of LoRA over other PEFT methods is that it does not increase inference latency, as it combines the trained parameters with the original weights during inference. We chose a rank of 7 and an alpha of 32, approximately 0.25% of the original parameter count as trainable parameters.

Batch designs We first consider our options for handling **mined hard negatives**. While there is a line of research that shows adding hard negatives mined through a heavy distillation process improves performance (Santhanam et al., 2021; Formal et al., 2021; Ren et al., 2021), this is resource-intensive and not broadly accessible. In this work, we focus on simple yet widely-used techniques: (1) **BM25**, (2) **model self-distillation** during training (Karpukhin et al., 2020; Izacard et al., 2021; Khattab & Zaharia, 2020), which have consistently shown to improve in-domain performance and does not have high dependency on other dual encoder models, (3) a combination of the two, (4) using **dataset-provided** hard negatives,[2] and (5) of course, the option to use **no hard negatives**.

Regarding **in-batch negatives**, this is driven by adjusting per-GPU batch size where each example in the batch is a positive query. PEFT methods take up less GPU memory to hold the model, thereby freeing up more space for larger per-GPU batch sizes. While per-GPU memory limitations can be overcome using gradient accumulation in many settings when training retrieval models, the use of in-batch negatives is practically limited to per-GPU batch size and not easily overcome through techniques like gradient accumulation without significant engineering and computation overhead (Gao et al., 2021b). As such, we define a fixed per-GPU batch size **(B)** (see hyperparameters below), as well as settings for twice **(2B)** and four times **(4B)** larger per-GPU batch sizes.

Datasets Focusing on evaluating whether models trained on large supervised datasets can generalize to out-of-domain tasks, for **training**, we use NaturalQuestions (Kwiatkowski et al., 2019) and MSMARCO (Campos et al., 2016), two popular large datasets that have been used successfully used for this purpose (Thakur et al., 2021a; Lee et al., 2023; Izacard et al., 2021; Gao et al., 2021a).

For **testing**, we evaluate over 14 different datasets from the BEIR benchmark, which have been used in works studying out-of-domain generalization of retrieval models (Khattab & Zaharia, 2020; Ni et al., 2021b; Weller et al., 2023; Tam et al., 2022). We evaluate over TREC-COVID (TR) (Roberts et al., 2020), NFCorpus (NF) (Boteva et al., 2016), NaturalQuestions (NQ) (Kwiatkowski et al., 2019), HotpotQA (HO) (Yang et al., 2018), FIQA-2018 (FI) (Maia et al., 2018), ArguAna (AR) (Wachsmuth et al., 2018), Touche-2020 (TO) (Bondarenko et al., 2020), Quora (QU), DBpedia (DB) (Hasibi et al., 2017), MSMARCO (MS) (Campos et al., 2016), SCIDOCS (SD) (Cohan et al., 2020), FEVER (FE) (Thorne et al., 2018), Climate-FEVER (CL) (Diggelmann et al., 2020), and SciFact (SF) (Wadden et al., 2020). Of course, when training on MSMARCO, we treat MSMARCO evaluation as **in-domain** and all others as **out-of-domain**; similarly when training on NQ.

Hyperparameters Following Karpukhin et al. (2020), we trained with effective batch sizes of 128 for 40 epochs with 10% as warmup steps for both asymmetric and symmetric dual encoders unless otherwise specified[3]. All experiments are conducted using 8 or fewer A6000 GPUs (40GB memory), making the per-GPU batch size of 16. We use checkpoints for all pretrained models from Huggingface (Wolf et al., 2019).

[2] For NaturalQuestions, we use the version of the dataset released by Karpukhin et al. (2020) to utilize their hard negatives. For MSMARCO, we use the version from its official website and similarly use its provided hard negatives.

[3] In the case of symmetric dual encoder, due to resource limitations, we conducted experiments with the same hyperparameters as Karpukhin et al. (2020) rather than Izacard et al. (2021), as Izacard et al. (2021) is trained using a much larger batch size (1024) and a longer training duration (approximately 77 epochs). Also, please note that replicating the official contriever with MSMARCO in Izacard et al. (2021) is challenging because the optimizer code and negatives used during the training step are not released.

Since the number of training parameters differs for LoRA and FT, we perform early hyperparameter search over different learning rates $\in \{$1e-4, 2e-4, 1e-5, 2e-5, 5e-5$\}$ for both the FT and LoRA settings, following various configurations from previous works (Maillard et al., 2021; Karpukhin et al., 2020; Izacard et al., 2021; Xiong et al., 2020; Ni et al., 2021b). We found that the optimal learning rate for LoRA tends to be higher than FT—2e-5 for FT and 2e-4 for LoRA. For late interaction dual encoder, we follow the configuration from Khattab & Zaharia (2020).

Evaluation Metrics Following the widely used metrics in the BEIR benchmark, we report results in nDCG@10 which calculates the ranking of the top 10 retrieved documents. All results are calculated with the official BEIR evaluation code (Thakur et al., 2021b).

4 How should we train? Comparing LoRA with full fine-tuning

		Out-Of-Domain (OOD)														**In-Domain**
		FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
Asymmetric Dual Encoder																
w/o Neg	FT	18.6	64.2	25.7	72.4	7.2	45.3	26.8	31.1	44.1	**44.1**	57.5	13.8	**35.7**	37.4	**31.1**
	LoRA	**19.5**	**67.9**	**27.1**	**73.4**	**8.0**	**49.4**	**30.0**	**33.8**	**45.4**	43.5	**58.7**	**15.8**	34.2	**39.0**	30.2
w/ Neg	FT	17.8	63.0	23.5	69.7	6.8	40.6	20.6	26.0	34.8	32.8	54.1	12.6	**36.9**	33.8	**33.2**
	LoRA	**19.5**	**66.4**	**27.1**	**73.0**	**7.8**	**48.6**	**29.2**	**33.5**	**44.7**	**42.2**	**57.8**	**15.2**	34.1	**38.4**	30.9
Symmetric Dual Encoder																
w/o Neg	FT	22.9	36.4	27.7	83.4	**13.2**	54.6	13.5	**41.0**	**49.4**	30.3	60.7	**18.7**	33.7	37.3	**31.3**
	LoRA	**30.1**	**39.0**	**33.8**	**88.2**	11.5	**65.0**	**21.7**	34.7	41.8	**30.9**	**70.9**	18.2	**34.8**	**40.0**	29.4
w/ Neg	FT	19.0	**34.7**	24.5	**67.1**	9.6	44.4	11.9	32.4	**38.4**	20.2	56.6	10.3	**35.4**	31.1	**36.9**
	LoRA	**25.5**	34.3	**27.9**	62.8	**10.0**	**55.7**	**16.5**	**33.8**	37.8	**22.7**	**64.9**	**13.8**	30.2	**33.5**	34.6

Table 1: Overall performance of dual encoder models trained on MSMARCO on BEIR benchmark tasks. The highest scores between the pair of full fine-tuning (FT) and LoRA experiments are in **bold**. For all encoder architectures, we see two trade-offs between in-domain and out-of-domain tasks (*"Avg"* column): (1) FT exhibits higher in-domain performance but lower out-of-domain performance, and (2) incorporating hard negative (*"w/ Neg"* rows) consistently boosts in-domain performance but reduces out-of-domain performance.

In this section, we examine the impact of training techniques on both out-of-domain and in-domain performance; namely, we compare LoRA, a parameter-efficient training method, to the traditional approach of fully fine-tuning all model parameters (FT). We conduct our experiments on two dense retrieval architectures: asymmetric and symmetric dual encoders.

LoRA consistently shows higher performance in out-of-domain over FT Table 1, compares the performance of dual encoder models trained using FT and LoRA techniques on the BEIR benchmarks. Results show that, on average, retrieval models trained with LoRA outperform FT by 1.6 to 4.6 absolute points (4.3% to 13.7% relative) on out-of-domain datasets. This result is consistent across different architecture, and suggest that LoRA is a more effective training technique for maximizing out-of-domain performance; conversely, models fully fine-tuned exhibit better in-domain performance. We speculate that this difference is due FT models overfitting to in-domain distribution, thus making them less versatile across out-of-domain datasets.

LoRA offers a better trade-off between in-domain and out-of-domain performance Table 2 quantifies the trade-off between better in-domain performance afforded by FT techniques against better out-of-domain

		Out-Of-Domain (OOD)		In-Domain	
		Avg	Rel. Diff (%)	MS	Rel. Diff (%)
Asymmetric Dual Encoder					
w/o Neg	FT	37.4	**+4.1%**	31.1	-3.0%
	LoRA	39.0		30.2	
w/ Neg	FT	33.8	**+12.0%**	33.2	-7.4%
	LoRA	38.4		30.9	
Symmetric Dual Encoder					
w/o Neg	FT	37.3	**+6.8%**	31.3	-6.5%
	LoRA	40.0		29.4	
w/ Neg	FT	31.1	**+7.2%**	36.9	-6.6%
	LoRA	33.5		34.6	

Table 2: Rel. Diff (%) is the percentage change between LoRA and FT trained models on in-domain and out-of-domain datasets. Results show that the out-of-domain performance increase of LoRA over FT always more than offset the decrease in in-domain performance. This suggests that LoRA is a more effective approach for training models that consistently perform well in both in-domain and out-of-domain scenarios.

generalization with LoRA. It is evident that in all scenarios, the decrease in average performance for out-of-domain datasets is more pronounced than the gains in in-domain datasets. These findings indicate that LoRA (PEFT) is a more suitable approach for training models that performs well in both in-domain and out-of-domain settings.

Asymmetric and Symmetric Encoders achieve similar performance When comparing the two dual encoder models, we note that the two architectures perform similarly regardless of the training technique used. While on individual datasets one might significantly outperform the other, their average performance on out-of-domain tasks is within one absolute point. Therefore, due to their popularity (Ni et al., 2021b;a; Lin et al., 2023; Karpukhin et al., 2020), we choose to use asymmetric dual encoders for the remainder of our work, unless otherwise noted.

5 How should we train? Designing optimal batches for fine-tuning

In this section, we focus on the influence of training batch design on both in-domain and out-of-domain performance. This includes examining the effects of incorporating hard negatives, a technique commonly acknowledged for enhancing in-domain performance, as well as the impact of selecting different batch sizes.

Mined hard negatives degrade out-of-domain performance for FT and LoRA models. Table 3 shows that hard negatives, despite generally enhancing in-domain performance, consistently degrade out-of-domain nDCG scores. We hypothesize this is due to the fact that models tend to over-adapt to the training dataset when adding hard negatives, making it challenging to generalize to datasets with differing distributions. In the experiments, we use negatives from the official MSMARCO dataset following Khattab & Zaharia (2020)[4].

We note that FT models are more influenced by hard negatives than models trained with LoRA. while they benefit more from the inclusion of mined negatives on in-domain tasks, their performance decreases more severely on out-of domain tasks. The finding suggests that since FT trains a larger number of parameters than

[4] As some queries are missing negatives, we fill the queries with BM25 negatives.

		Out-Of-Domain (OOD)		**In-Domain**	
		Avg	Rel. Diff (%)	MS	Rel. Diff (%)
Asymmetric Dual Encoder					
FT	w/o Neg	37.4	**-10.7%**	31.1	**+6.3%**
	w/Neg	33.8		33.2	
LoRA	w/o Neg	39.0	-1.6%	30.2	+2.3%
	w/ Neg	38.4		30.9	
Symmetric Dual Encoder					
FT	w/o Neg	37.3	**-19.9%**	31.3	**+15.7**
	w/ Neg	31.1		36.9	
LoRA	w/o Neg	40.0	-19.4%	29.4	+12.9%
	w/ Neg	33.5		34.6	

Table 3: Rel. Diff (%) represents the percentage change in performance due to using mined negatives. FT shows a more significant reduction in out-of-domain (OOD) performance and a higher increase in in-domain (IN) performance relative to LoRA.

	Asymmetric Dual Encoder		**Symmetric Dual Encoder**	
	FT	LoRA	FT	LoRA
Similar Distribution (%)	-8.6%	-1.4%	-16.7%	-11.8%
Different Distribution (%)	**-11.1%**	**-1.7%**	**-18.8%**	**-18.8%**

Table 4: Relative change in performance from using hard negatives. We partition out-of-domain datasets in BEIR by how similar they are to MSMARCO. Similarity is assesses by sampling 50 documents from each dataset, and comparing their average Contriever embeddings. Datasets that are most dissimilar to MSMARCO have consistently higher relative decrease in performance than the group of most similar datasets.

LoRA, it becomes more attuned to the given datasets, and further gets more affected by hard negatives. This results in more pronounced improvements in in-domain performance, but at the cost of a larger decrease in out-of-domain performance.

Experimental evidence suggests hard negatives encourage overfitting to training data distribution. We set out to investigate our hypothesis—performance decline when incorporating hard negatives is due to over-specialization to a specific training dataset—by empirically assessing whether dateset that are most *dissimilar* from training data are more severely affected. To assess similarity between datasets, we randomly select 50 instances from a corpus of each dataset and compute the inner product of embeddings from contriever (Izacard et al., 2021). The dataset most frequently identified as top-ranked, excluding its own dataset, was considered the most similar. After identifying the most relevant dataset for each dataset, we then grouped them together. Repeating this process five times, we categorize datasets grouped with MSMARCO more than three times as similar distribution. The BEIR datasets most similar to MSMARCO are trec-covid, NFcorpus, scidocs, scifacts, and arguana (see Appendix B for details).

We summarize our findings in Table 4. Results show that datasets that are most similar to MSMARCO exhibit a smaller performance drop when hard negatives are added. When comparing the average drop rates between these groups, we observed that those in similar domains showed a lesser reduction (average of 3%) compared to those grouped in different distributions. This suggests that training with hard negatives tends to overfit the model to its training dataset, reducing its effectiveness on datasets with different distributions.

		Out-Of-Domain (OOD)														In-Domain
	Batch Size	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
FT	B	18.6	64.2	25.7	72.4	7.2	45.3	26.8	31.1	44.1	44.1	57.5	13.8	**35.7**	37.4	31.1
LoRA	B	19.5	67.9	27.1	73.4	8.0	49.4	30.0	33.8	45.4	43.5	58.7	15.8	34.2	39.0	30.2
	2 × B	20.7	71.1	27.5	**76.1**	8.0	51.5	**30.8**	33.9	46.3	44.2	58.7	14.2	34.1	39.8	31.0
	4 × B	**20.9**	**71.4**	**28.3**	75.5	**8.1**	**52.6**	30.2	**34.3**	**47.2**	**44.8**	**59.7**	**14.7**	35.0	**40.2**	**31.8**

Table 5: Increasing batch size (increasing the number of in-batch negatives) consistently helps both in-domain and out-of-domain performance.

Unlike mined negatives, using larger batch size increases both in-domain and out-of-domain performance. In Table 5, we observe that using larger batch sizes, which include a greater number of in-batch negatives, enhances performance both within and outside the domain. This observation suggests that, under certain GPU configurations, to boost performance across both domains, increasing batch size could be a more effective strategy than incorporating hard negatives. We hypothesize that this is because hard negatives often lead the model to over-adapt to certain distributions. On the other hand, in-batch negatives, which are typically random negatives, do not exhibit such a tendency. Moreover, since LoRA demands less GPU memory, it enables the use of larger batch sizes under the same GPU constraints compared to FT. In our experiments, LoRA with a batch size quadruple that of FT consumes a similar amount of GPU memory.

6 How complementary are our findings with other resource-intensive methods for out-of-domain retrieval?

In this section, we investigate (1) the impact of increasing model size, (2) the use of late-interaction retrieval architectures, and (3) whether our findings still add benefit on top of more powerful base models that have undergone additional pretraining.

		Out-Of-Domain (OOD)														**In-Domain**
		FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
Bert Large																
w/o Neg	FT	19.3	70.0	26.5	76.7	7.5	47.6	**28.1**	31.5	42.5	43.5	57.0	14.1	36.1	38.5	32.9
	LoRA	**23.2**	**76.7**	**28.2**	**79.6**	**8.5**	**54.9**	26.1	**34.1**	**45.9**	**45.0**	**57.3**	15.4	**36.6**	**40.9**	31.4
w/ Neg	FT	17.5	69.5	23.3	74.6	7.2	44.4	22.7	30.7	40.5	42.8	54.5	11.7	37.3	36.7	**34.4**
	LoRA	**22.7**	**75.0**	**28.2**	**78.5**	**8.4**	**54.3**	**25.2**	**31.9**	**42.3**	**44.3**	**56.5**	**15.5**	**34.6**	**39.8**	32.1
RoBERTa Large																
w/o Neg	FT	24.5	65.9	27.9	77.8	9.0	48.1	**30.2**	32.1	41.5	40.6	58.7	14.7	34.5	38.9	34.0
	LoRA	**29.0**	**73.7**	**28.8**	**78.6**	**9.4**	**55.2**	26.2	**37.8**	**43.7**	**42.7**	**59.9**	**18.4**	35.5	**41.5**	32.1
w/ Neg	FT	24.5	55.9	25.9	**77.8**	7.0	48.1	22.2	32.1	38.5	36.6	56.7	13.7	32.5	36.3	**35.9**
	LoRA	**28.0**	**70.2**	**28.7**	76.1	**8.3**	**53.2**	**25.9**	**36.7**	**42.5**	**41.9**	**58.9**	**16.8**	**36.8**	**40.3**	34.5

Table 6: Overall BEIR performance of RoBERTa-large and BERT-large, two similar base model architectures and size but with different training strategies. (1) Performance tends to increase when using RoBERTa-large. (2) Our findings about the benefits of LoRA (§4) and the possible detriment of hard negatives (§5) hold here as well.

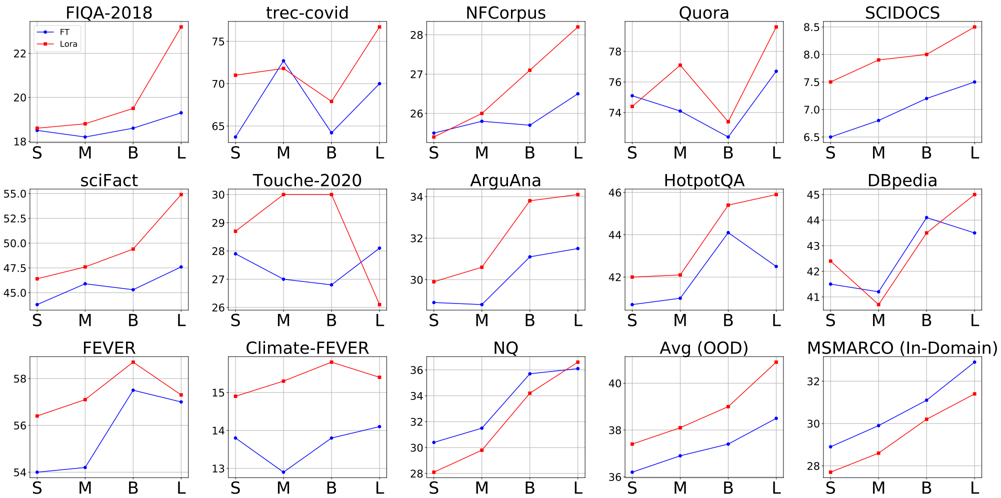

Figure 1: Overall BEIR performance of different base model sizes of asymmetric dual encoder trained with MSMARCO (without hard negatives). Performance consistently increases with larger encoder models for both in-domain and out-of-domain. Notation in x-axis indicates **S**: Bert Small, **M**: Bert Medium, **B**: Bert Base, **L**: Bert Large

Impact of increasing model size: out-of-domain performance improves with better base model especially in LoRA We study whether our findings hold when increasing the size of the base model, such as in Ni et al. (2021b). Figure 1 shows results using BERT models in four different sizes (small, medium, base, large) finetuned on MSMARCO. We can see that larger base model consistently leads to higher performance in all cases See Appendix Table 11 for numerical results. From this table, we also validate our earlier findings about the trade-offs between in-domain versus OOD performance when using LoRA versus FT and the possible detrimental effects of using hard negatives.

We further experiment by switching the base model from BERT-large to RoBERTa-large, a model trained with more optimized hyperparameters (Table 6). We observe that when training with RoBERTa-large instead of BERT-large, both in-domain and out-of-domain performance show improvement. And of course, we re-validate our consistent findings about LoRA versus FT and hard negatives from earlier.

Looking deeper into the LoRA versus FT tradeoff, we can see that LoRA tends to benefit more by using a better base model (Table 7). This matches intuition as PEFT does not update many of the base model parameters; therefore, as the base model improves, so does the performance of a LoRA trained model as it can use much more information from the frozen parameters. FT, on the other hand, updates the whole base model and is more likely affected by forgetting and shifts in the distribution across all parameters.

	OOD Avg			In-domain		
	Medium	Base	Large	Medium	Base	Large
FT	**1.9%**	3.3%	6.4%	**3.5%**	7.6%	11.6%
LoRA	**1.9%**	**4.3%**	**9.4%**	3.2%	**9.0%**	**13.4%**

Table 7: Performance using larger BERT model sizes relative to performance using BERT-small weights. (1) Performance improves monotonically with model size. (2) LoRA tends to benefit more from a larger base model compared to FT.

	Out-Of-Domain (OOD)														In-Domain
	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
FT	31.9	**65.1**	31.8	**83.8**	15.1	67.5	19.5	24.0	58.4	**38.7**	77.9	18.2	**51.7**	44.9	**39.2**
LoRA	**32.5**	64.4	**32.7**	83.3	**15.6**	**68.5**	**21.6**	**37.4**	**61.1**	33.6	**78.6**	**19.1**	51.2	**46.1**	37.2

Table 8: Overall BEIR performance of a token-level late-interaction dual encoder following Khattab & Zaharia (2020) and trained on MSMARCO. Like all other models tested, we see a clear performance trade-off between FT, which is better for in-domain performance, and LoRA, which is better for out-of-domain performance.

Use of token-level late-interaction models We experiment with a late interaction dual encoder model following Khattab & Zaharia (2020)to see whether our findings persist in retrieval architectures that involve more resource-intensive steps.[5] Table 8 shows a similar trend to what we observed in asymmetric and symmetric dual encoders; LoRA surpasses FT in out-of-domain settings, whereas FT demonstrates superior performance in in-domain settings.

	Out-Of-Domain (OOD)														In-Domain
	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
Bert Base															
FT	22.9	36.4	27.7	83.4	**13.2**	54.6	13.5	**41.0**	**49.4**	30.3	60.7	**18.7**	33.7	37.3	**31.3**
LoRA	**30.1**	**39.0**	**33.8**	**88.2**	11.5	**65.0**	**21.7**	34.7	41.8	**30.9**	**70.9**	18.2	**34.8**	**40.0**	29.4
Bert Base with Contrastive Pretraining (Contriever)															
FT	**26.9**	40.1	30.5	84.4	14.9	64.4	13.4	**40.9**	60.6	37.5	68.8	**20.4**	**38.9**	41.7	**32.8**
LoRA	**26.9**	**44.6**	**33.9**	**88.7**	**15.8**	**65.7**	**20.6**	36.2	**62.3**	**38.3**	**71.5**	19.0	37.1	**43.1**	31.5

Table 9: Comparison between two encoder models derived from a BERT-base and Contriver checkpoint. Contriver was obtained by further training a BERT-base model on a large unlabeled collection using a contrasive loss. Performance tends to increase when changing the base encoder model to that pretrained with contrastive loss.

The effectiveness of employing models pre-trained on contrastive loss as the initial base model We conduct experiments to determine if our findings are consistent when using encoder-only base models pretrained with contrastive loss. Specifically, we use the Contriever pretrained model weights (Izacard et al., 2021). Table 9 shows that switching to this base model with additional pretraining significantly improves performance, especially in average OOD performance.

Importantly, we see again that our earlier findings persist even with this more powerful base model. We see again that LoRA shows superior performance in OOD settings but worse performance in in-domain settings. However, we notice that there is LoRA benefits less when trained on Contriever compared to BERT-base.

When comparing the rate of improvement for average OOD performance and the rate of degradation for in-domain between FT and LoRA, we can see that (1) the rate of improvement in average OOD performance using Contriever (3.36%) is not as substantial as that using BERT-base (7.24%) and (2) the improvement rate in OOD (3.36%) is smaller than the degradation rate than on in-domain settings (3.96%). We hypothesize that this is due to Contriever's adaptation to various domains during its pretraining with a massive unsupervised

[5] Our FT numbers are obtained through a replication of Khattab & Zaharia (2020) experiments, which was necessary to make a fair comparison between LoRA and FT, as opposed to using their released weights. Nevertheless, our results are very close to the ones reported in Thakur et al. (2021b), indicating our successful replication. For more details, see Appendix A.

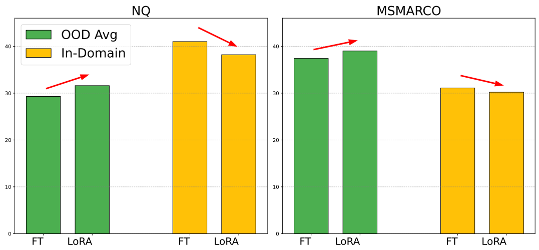

Figure 2(a): Performance of Asymmetric Dual Encoder when trained with NQ (left). Results tend to be similar to that of MSMARCO (right): LoRA tends to show higher performance in out-of-domain (OOD) but lower performance in in-domain.

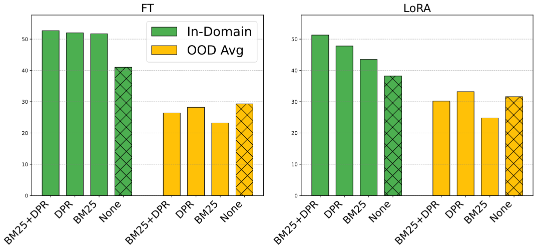

Figure 2(b): Performance of dual encoders trained on the NQ dataset with different negative sampling strategies. Adding hard negatives improves in-domain performance but degrades out-of-domain (OOD) performance for both FT and LoRA.

dataset, making it less likely to over-adapt to a specific training dataset's distribution and maintaining its generalizability.

7 Do Findings Generalize to Other Training Datasets? A Case Study on Google Natural Questions

To analyze whether findings presented in § 4–6 generalize to different training datasets, we experiment over another widely used retrieval collection, Google Natural Questions (Kwiatkowski et al., 2019) (NQ). While NQ is significantly smaller than MSMARCO (NQ is comprised of 307k training examples, while MSMARCO contains over 1M queries and 8.8M passages) and only contains passages extracted from Wikipedia, we note that trends observed on models trained on MSMARCO generally hold constants for NQ, suggesting that they generalize across training datasets.

Models trained on MSMARCO or NQ have similar in- and out-of-domain performance characteristics Figure 2a compares the performance of a dual encoder model trained on MSMARCO and NQ. Results show similar trends across the datasets: full fine-tuning generally improves in-domain performance, but LoRA achieves better nDCG@10 on out-of-domain tasks in BEIR.

Many hard negative mining approaches remain detrimental to out-of-domain performance Similarly to MSMARCO in §5, using mined hard negative when training on NQ seems to negatively affect out-of-domain performance. Figure 2b compares the effect of negatives on in-domain and out-of-domain performance for FT and LoRA trained asymmetric dual encoders. Compared to §5, we experiment with two different rankers to select hard negatives: BM25 and model-based negatives (DPR[6]). We also evaluate using a combination of both ("BM25+DPR" in fig. 2b). While all three mining approaches yield improvements in in-domain performance, they are rivaled or bested by using in-batch negative only ("None" in fig. 2b). This finding is consistent for both FT and LoRA-trained models. Overall, our observations support previous research calling for careful selection of hard negatives (Santhanam et al., 2021). Further, they highlight how the wrong mining approach is far more likely to hurt out-of-domain performance than in-domain performance, as, in the latter case, incorporating negative examples is consistently beneficial.

Advantages of LoRA on out-of-domain generalization is consistent across model sizes. Figure 3 presents the in-domain and out-of-domain performance across various model sizes. We can see that the trend

[6] We use the negatives provided from `https://github.com/facebookresearch/DPR`

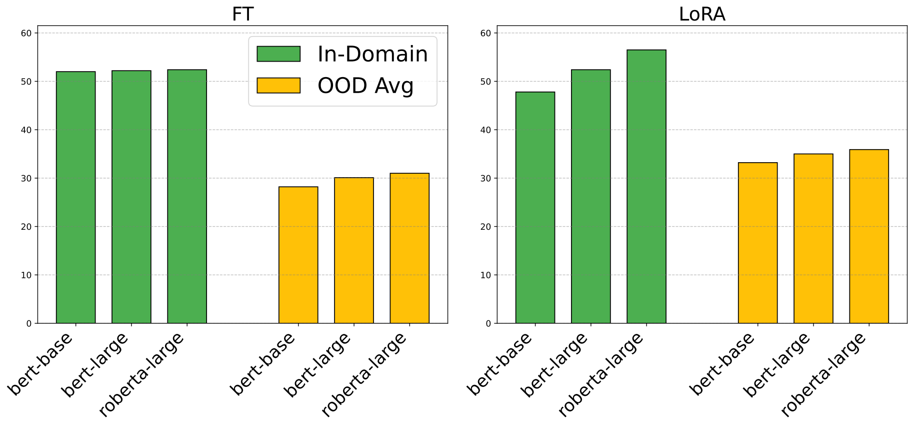

Figure 3: Impact of model size on the performance of dual encoders trained on NQ. Larger models consistently lead to higher performance in both in-domain and out-of-domain (OOD). LoRA shows larger gains when moving to larger models.

when trained with MSMARCO persists; a superior base model consistently yields enhanced performance in out-of-domain scenarios, with LoRA especially benefiting more from an improved base model. Surprisingly, the data shows that using RoBERTa-large as the base model enables LoRA to exhibit higher performance than fine-tuning (FT), even in the in-domain setting.

MSMARCO vs. NQ as training dataset We observed that fine-tuning with MSMARCO consistently yields robust performance (Table 1), surpassing that achieved with NQ (Table 12 in Appendix D.1), confirming trends observed by Thakur et al. (2021a) and Ni et al. (2021b). In terms of the average out-of-domain performance reduction rate, models trained with LoRA exhibit a lower reduction rate of 17.4% compared to when training full parameter (FT), which shows a 20.2% reduction. To calculate the average out-of-domain performance, we averaged the reduction rate excluding MSMARCO and NQ data. This pattern indicates that LoRA experiences a smaller performance drop with fewer training datasets, a trend consistent with previous studies in parameter efficient methods (Ustun & Stickland, 2022; Litschko et al., 2022).

8 Conclusion

In this work, we investigate the impact of training strategies on the generalizability of dense retrieval models. Our focus is on scenarios that avoid extra resource-intensive steps, such as (1) comparing LoRA with full fine-tuning and (2) designing optimal batch sizes for fine-tuning. We further examine how these findings align with other resource-intensive methods for out-of-domain retrieval (*i.e.,* token-level late-interaction models, scaling model size). Across various experiments, we observe a consistent trend: (1) LoRA invariably enhances generalizability, and (2) under identical GPU configurations, increasing the in-batch size typically yields more robust performance compared to adding hard negatives. Furthermore, we find that our insights complement popular techniques for boosting out-of-domain performance. Our study offers practical, actionable insights for developing dense retrieval models with high generalizability.

9 LIMITATIONS

Most of our experiments are conducted on smaller base models relative to some larger choices like Llama (Ma et al., 2023). We have demonstrated some robustness of our findings under scaling (§6), but further investigation is needed.

Our study does not explore the compatibility of our approach with data augmentation methods for out-of-domain generalization. Also, we have not investigated whether our approach maintains its effectiveness when trained on a diverse combination of domains, rather than a single training dataset (NQ, MSMARCO).

Our focus is primarily on widely-used negative sampling strategies that do not involve resource-intensive steps like distillation (Santhanam et al., 2021; Formal et al., 2021; Ren et al., 2021), leading to a lack of exploration in various other negative sampling methods.

REFERENCES

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv*, abs/2106.10199, 2021. URL `https://api.semanticscholar.org/CorpusID:231672601`.

Alexander Bondarenko, Maik Fröbe, Meriem Beloucif, Lukas Gienapp, Yamen Ajjour, Alexander Panchenko, Christian Biemann, Benno Stein, Henning Wachsmuth, Martin Potthast, and Matthias Hagen. Overview of touché 2020: Argument retrieval. In *Conference and Labs of the Evaluation Forum*, 2020. URL `https://api.semanticscholar.org/CorpusID:225073856`.

Luiz Henrique Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Data augmentation for information retrieval using large language models. *ArXiv*, abs/2202.05144, 2022. URL `https://api.semanticscholar.org/CorpusID:246705967`.

Vera Boteva, Demian Gholipour Ghalandari, Artem Sokolov, and Stefan Riezler. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*, 2016. URL `https://api.semanticscholar.org/CorpusID:14355670`.

Daniel Fernando Campos, Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, Li Deng, and Bhaskar Mitra. Ms marco: A human generated machine reading comprehension dataset. *ArXiv*, abs/1611.09268, 2016. URL `https://api.semanticscholar.org/CorpusID:1289517`.

Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *ArXiv*, abs/2309.12307, 2023. URL `https://api.semanticscholar.org/CorpusID:262084134`.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. Specter: Document-level representation learning using citation-informed transformers. *ArXiv*, abs/2004.07180, 2020. URL `https://api.semanticscholar.org/CorpusID:215768677`.

Zhuyun Dai, Vincent Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. *ArXiv*, abs/2209.11755, 2022. URL `https://api.semanticscholar.org/CorpusID:252519173`.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314, 2023. URL `https://api.semanticscholar.org/CorpusID:258841328`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *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)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Thomas Diggelmann, Jordan L. Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. Climate-fever: A dataset for verification of real-world climate claims. *ArXiv*, abs/2012.00614, 2020. URL https://api.semanticscholar.org/CorpusID:227239135.

Thibault Formal, C. Lassance, Benjamin Piwowarski, and Stéphane Clinchant. Splade v2: Sparse lexical and expansion model for information retrieval. *ArXiv*, abs/2109.10086, 2021. URL https://api.semanticscholar.org/CorpusID:237581550.

Luyu Gao, Zhuyun Dai, and Jamie Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *North American Chapter of the Association for Computational Linguistics*, 2021a. URL https://api.semanticscholar.org/CorpusID:233241070.

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. Scaling deep contrastive learning batch size under memory limited setup. In Anna Rogers, Iacer Calixto, Ivan Vulić, Naomi Saphra, Nora Kassner, Oana-Maria Camburu, Trapit Bansal, and Vered Shwartz (eds.), *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pp. 316–321, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.repl4nlp-1.31. URL https://aclanthology.org/2021.repl4nlp-1.31.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. In *Conference on Computational Natural Language Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:202718954.

Faegheh Hasibi, Fedor Nikolaev, Chenyan Xiong, Krisztian Balog, Svein Erik Bratsberg, Alexander Kotov, and Jamie Callan. Dbpedia-entity v2: A test collection for entity search. *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017. URL https://api.semanticscholar.org/CorpusID:3675602.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:59599816.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021. URL https://api.semanticscholar.org/CorpusID:235458009.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Towards unsupervised dense information retrieval with contrastive learning. *ArXiv*, abs/2112.09118, 2021. URL https://api.semanticscholar.org/CorpusID:245218527.

Joel Jang, Seonghyeon Ye, Sohee Yang, Joongbo Shin, Janghoon Han, Gyeonghun Kim, Stanley Jungkyu Choi, and Minjoon Seo. Towards continual knowledge learning of language models. *ArXiv*, abs/2110.03215, 2021. URL https://api.semanticscholar.org/CorpusID:238419458.

Xisen Jin, Dejiao Zhang, Henghui Zhu, Wei Xiao, Shang-Wen Li, Xiaokai Wei, Andrew O. Arnold, and Xiang Ren. Lifelong pretraining: Continually adapting language models to emerging corpora. *ArXiv*, abs/2110.08534, 2021. URL https://api.semanticscholar.org/CorpusID:239016173.

Euna Jung, Jaekeol Choi, and Wonjong Rhee. Semi-siamese bi-encoder neural ranking model using lightweight fine-tuning. *Proceedings of the ACM Web Conference 2022*, 2021. URL https://api.semanticscholar.org/CorpusID:240070656.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2020. URL https://api.semanticscholar.org/CorpusID:215737187.

O. Khattab and Matei A. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. URL https://api.semanticscholar.org/CorpusID:216553223.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc V. Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019. URL https://api.semanticscholar.org/CorpusID:86611921.

Hyunji Lee, Jaeyoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vladimir Karpukhin, Yi Lu, and Minjoon Seo. Nonparametric decoding for generative retrieval. In *Annual Meeting of the Association for Computational Linguistics*, 2022a. URL https://api.semanticscholar.org/CorpusID:258959550.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. Generative multi-hop retrieval. In *Conference on Empirical Methods in Natural Language Processing*, 2022b. URL https://api.semanticscholar.org/CorpusID:249049410.

Jinhyuk Lee, Minjoon Seo, Hannaneh Hajishirzi, and Jaewoo Kang. Contextualized sparse representations for real-time open-domain question answering. In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:219058995.

Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Zhao. Rethinking the role of token retrieval in multi-vector retrieval. *ArXiv*, abs/2304.01982, 2023. URL https://api.semanticscholar.org/CorpusID:257921404.

Jimmy Lin, Rodrigo Nogueira, and Andrew Yates. *Learned Dense Representations for Ranking*, pp. 195–238. Springer International Publishing, Cham, 2022. ISBN 978-3-031-02181-7. doi: 10.1007/978-3-031-02181-7_5. URL https://doi.org/10.1007/978-3-031-02181-7_5.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oğuz, Jimmy Lin, Yashar Mehdad, Wen tau Yih, and Xilun Chen. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. *ArXiv*, abs/2302.07452, 2023. URL https://api.semanticscholar.org/CorpusID:256868909.

Erik M. Lindgren, Sashank J. Reddi, Ruiqi Guo, and Surinder Kumar. Efficient training of retrieval models using negative cache. In *Neural Information Processing Systems*, 2021. URL https://api.semanticscholar.org/CorpusID:245018271.

Robert Litschko, Ivan Vulic, and Goran Glavavs. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In *International Conference on Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:247958074.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638, 2022a. URL https://api.semanticscholar.org/CorpusID:248693283.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, Dublin, Ireland, May 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.8. URL https://aclanthology.org/2022.acl-short.8.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692, 2019. URL https://api.semanticscholar.org/CorpusID:198953378.

Y Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2020. URL https://api.semanticscholar.org/CorpusID:218470027.

Simon Lupart, Thibault Formal, and Stéphane Clinchant. Ms-shift: An analysis of msmarco distribution shifts on neural retrieval. In Jaap Kamps, Lorraine Goeuriot, Fabio Crestani, Maria Maistro, Hideo Joho, Brian Davis, Cathal Gurrin, Udo Kruschwitz, and Annalina Caputo (eds.), *Advances in Information Retrieval*, pp. 636–652, Cham, 2023. Springer Nature Switzerland. ISBN 978-3-031-28244-7.

Xinyu Ma, J. Guo, Ruqing Zhang, Yixing Fan, and Xueqi Cheng. Scattered or connected? an optimized parameter-efficient tuning approach for information retrieval. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022. URL https://api.semanticscholar.org/CorpusID:251718746.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. Fine-tuning llama for multi-stage text retrieval. *ArXiv*, abs/2310.08319, 2023. URL https://api.semanticscholar.org/CorpusID:263908865.

Macedo Maia, Siegfried Handschuh, André Freitas, Brian Davis, Ross McDermott, Manel Zarrouk, and Alexandra Balahur. Www'18 open challenge: Financial opinion mining and question answering. *Companion Proceedings of the The Web Conference 2018*, 2018. URL https://api.semanticscholar.org/CorpusID:13866508.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen tau Yih, Barlas Oğuz, Veselin Stoyanov, and Gargi Ghosh. Multi-task retrieval for knowledge-intensive tasks. *ArXiv*, abs/2101.00117, 2021. URL https://api.semanticscholar.org/CorpusID:230435546.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. Augmented language models: A survey. February 2023.

Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:254220735.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas A. Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David P. Schnurr, Felipe Petroski Such, Kenny Sai-Kin Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005, 2022. URL https://api.semanticscholar.org/CorpusID:246275593.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B. Hall, Daniel Matthew Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *ArXiv*, abs/2108.08877, 2021a. URL https://api.semanticscholar.org/CorpusID:237260023.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith B. Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. *ArXiv*, abs/2112.07899, 2021b. URL https://api.semanticscholar.org/CorpusID:245144556.

Vaishali Pal, Carlos Lassance, Hervé Déjean, and Stéphane Clinchant. Parameter-efficient sparse retrievers and rerankers using adapters. *ArXiv*, abs/2303.13220, 2023. URL https://api.semanticscholar.org/CorpusID:257642038.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. Rocketqa: An optimized training approach to dense passage retrieval for open-domain question answering. In *North American Chapter of the Association for Computational Linguistics*, 2020. URL https://api.semanticscholar.org/CorpusID:231815627.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji rong Wen. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking. *ArXiv*, abs/2110.07367, 2021. URL https://api.semanticscholar.org/CorpusID:238857121.

Ruiyang Ren, Yingqi Qu, J. Liu, Wayne Xin Zhao, Qifei Wu, Yuchen Ding, Hua Wu, Haifeng Wang, and Ji rong Wen. A thorough examination on zero-shot dense retrieval. *ArXiv*, abs/2204.12755, 2022. URL https://api.semanticscholar.org/CorpusID:248405719.

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen M. Voorhees, Lucy Lu Wang, and William R. Hersh. Trec-covid: rationale and structure of an information retrieval shared task for covid-19. *Journal of the American Medical Informatics Association : JAMIA*, 27: 1431 – 1436, 2020. URL https://api.semanticscholar.org/CorpusID:218504088.

S E Robertson and K Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science. American Society for Information Science*, 27(3):129–146, May 1976. ISSN 1097-4571,0002-8231. doi: 10.1002/asi.4630270302.

Guilherme Moraes Rosa, Luiz Henrique Bonifacio, Vitor Jeronymo, Hugo Abonizio, Marzieh Fadaee, Roberto de Alencar Lotufo, and Rodrigo Nogueira. In defense of cross-encoders for zero-shot retrieval. *ArXiv*, abs/2212.06121, 2022. URL https://api.semanticscholar.org/CorpusID:254564419.

Jon Saad-Falcon, O. Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Arafat Sultan, and Christopher Potts. Udapdr: Unsupervised domain adaptation via llm prompting and distillation of rerankers. *ArXiv*, abs/2303.00807, 2023. URL https://api.semanticscholar.org/CorpusID:257279774.

Keshav Santhanam, O. Khattab, Jon Saad-Falcon, Christopher Potts, and Matei A. Zaharia. Colbertv2: Effective and efficient retrieval via lightweight late interaction. In *North American Chapter of the Association for Computational Linguistics*, 2021. URL https://api.semanticscholar.org/CorpusID:244799249.

Weng Lam Tam, Xiao Liu, Kaixuan Ji, Lilong Xue, Xing Zhang, Yuxiao Dong, Jiahua Liu, Maodi Hu, and Jie Tang. Parameter-efficient prompt tuning makes generalized and calibrated neural text retrievers. *ArXiv*, abs/2207.07087, 2022. URL https://api.semanticscholar.org/CorpusID:250526456.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663, 2021a. URL https://api.semanticscholar.org/CorpusID:233296016.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 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)*, 2021b. URL `https://openreview.net/forum?id=wCu6T5xFjeJ`.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. Fever: a large-scale dataset for fact extraction and verification. *ArXiv*, abs/1803.05355, 2018. URL `https://api.semanticscholar.org/CorpusID:4711425`.

A. Ustun and Asa Cooper Stickland. When does parameter-efficient transfer learning work for machine translation? In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL `https://api.semanticscholar.org/CorpusID:248986389`.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. Retrieval of the best counterargument without prior topic knowledge. In *Annual Meeting of the Association for Computational Linguistics*, 2018. URL `https://api.semanticscholar.org/CorpusID:51880268`.

David Wadden, Kyle Lo, Lucy Lu Wang, Shanchuan Lin, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. *ArXiv*, abs/2004.14974, 2020. URL `https://api.semanticscholar.org/CorpusID:216867133`.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. K-adapter: Infusing knowledge into pre-trained models with adapters. In *Findings*, 2020. URL `https://api.semanticscholar.org/CorpusID:211031933`.

Orion Weller, Kyle Lo, David Wadden, Dawn J Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. *ArXiv*, abs/2309.08541, 2023. URL `https://api.semanticscholar.org/CorpusID:262012661`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019. URL `https://api.semanticscholar.org/CorpusID:204509627`.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke S. Zettlemoyer. Scalable zero-shot entity linking with dense entity retrieval. In *Conference on Empirical Methods in Natural Language Processing*, 2019. URL `https://api.semanticscholar.org/CorpusID:263877300`.

Wenhan Xiong, Xiang Lorraine Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Wen tau Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oğuz. Answering complex open-domain questions with multi-hop dense retrieval. *ArXiv*, abs/2009.12756, 2020. URL `https://api.semanticscholar.org/CorpusID:221970302`.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings*, 2022. URL `https://api.semanticscholar.org/CorpusID:247411106`.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. *ArXiv*, abs/2309.14717, 2023. URL `https://api.semanticscholar.org/CorpusID:262825568`.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL `https://api.semanticscholar.org/CorpusID:52822214`.

Soyoung Yoon, Chaeeun Kim, Hyunji Lee, Joel Jang, and Minjoon Seo. Continually updating generative retrieval on dynamic corpora. *ArXiv*, abs/2305.18952, 2023. URL `https://api.semanticscholar.org/CorpusID:258967398`.

Zexuan Zhong, Tao Lei, and Danqi Chen. Training language models with memory augmentation. In *Conference on Empirical Methods in Natural Language Processing*, 2022. URL `https://api.semanticscholar.org/CorpusID:249062699`.

A COLBERT

	Out-Of-Domain (OOD)														In-Domain
	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
Beir	31.7	67.7	30.5	85.4	14.5	67.1	20.2	23.3	59.3	39.2	77.1	18.4	52.4	45.1	40.1
Ours	31.9	65.1	31.8	83.8	15.1	67.5	19.5	24.0	58.4	38.7	77.9	18.2	51.7	44.9	39.2

Table 10: "Ours" is the performance of Colbert we replicate for fair comparison and "Beir" is the performance provided from Table 8, which is widely used. The performance tends to be similar.

To study each training configuration's impact and ensure a fair comparison, we replicated the experiment for consistent results (Ours in Table 8). The result aligns closely with the result of Colbert in Thakur et al. (2021b) (Beir in Table 8).

B CALCULATING DISTRIBUTION SIMILARITY

To explore the impact of hard negatives on datasets within similar distribution, we conduct an analysis using the BEIR dataset, grouping them based on distribution similarity. To assess similarity, our methodology is as follows: First, we sample 50 instances from the corpus of each dataset. Second, we generate embeddings for each instance using contriever (Izacard et al., 2021). Third, for each instance, we compute its similarity (dot product) with other embeddings, identifying the most relevant dataset, and excluding its original dataset. Fourth, for each dataset, we determine which dataset appears most frequently (out of 50) and regard this as the dataset with the most similar distribution. Last, using this information, we cluster datasets that are interconnected. This process is repeated five times, and we observe that the resulting groupings tend to be consistently similar. We categorize datasets grouped with MSMARCO more than three times as having a similar distribution. Consequently, datasets such as trec-covid, NFcorpus, scidocs, scifacts, and arguana are considered to share a similar distribution with MSMARCO. One thing we notice is that datasets with Wikipedia as their source consistently tend to be grouped together, leading us to assume that the grouping shows a high tendency of distribution.

C PERFORMANCE OF DIFFERENT MODEL SIZES WHEN TRAINED WITH MSMARCO

Table 11 presents the overall BEIR performance of various base model sizes of an asymmetric dual encoder trained on MSMARCO, without adding hard negatives. The performance of both in-domain and out-of-domain generally improves with the use of larger encoder models.

	Out-Of-Domain (OOD)														In-Domain
	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
Bert Small															
FT	18.5	63.7	25.5	75.1	6.5	43.8	27.9	28.9	40.7	41.5	54.0	13.8	30.4	36.2	28.9
LoRA	18.6	71.0	25.4	74.4	7.5	46.4	28.7	29.9	42.0	42.4	56.4	14.9	28.1	37.4	27.7
Bert Medium															
FT	18.2	72.7	25.8	74.1	6.8	45.9	27.0	28.8	41.0	41.2	54.2	12.9	31.5	36.9	29.9
LoRA	18.8	71.8	26.0	77.1	7.9	47.6	**30.0**	30.6	42.1	40.7	57.1	15.3	29.8	38.1	28.6
Bert Base															
FT	18.6	64.2	25.7	72.4	7.2	45.3	26.8	31.1	44.1	44.1	57.5	13.8	35.7	37.4	31.1
LoRA	19.5	67.9	27.1	73.4	8.0	49.4	**30.0**	33.8	45.4	43.5	**58.7**	**15.8**	34.2	39.0	30.2
Bert Large															
FT	19.3	70.0	26.5	76.7	7.5	47.6	28.1	31.5	42.5	43.5	57.0	14.1	36.1	38.5	**32.7**
LoRA	**23.2**	**76.7**	**28.2**	**79.6**	**8.5**	**54.9**	26.1	**34.1**	**45.9**	**45.0**	57.3	15.4	**36.6**	**40.9**	31.4

Table 11: Overall BEIR performance of different base model sizes of asymmetric dual encoder trained with MSMARCO (without hard negatives). The best and second best over all the model sizes in *bold* and underline respectively.

	Out-Of-Domain (OOD)														In-Domain
	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	MS	*Avg*	NQ
Asymmetric Dual Encoder															
FT	16.0	61.2	20.5	31.0	7.3	41.8	24.6	21.6	34.1	40.2	46.4	15.1	**21.2**	29.3	**41.0**
LoRA	**16.6**	**63.1**	**21.2**	**37.4**	**8.3**	**42.6**	**25.6**	**22.4**	**35.0**	**41.0**	**52.4**	**24.6**	20.0	**31.6**	38.2
Symmetric Dual Encoder															
FT	6.1	**26.8**	8.6	**75.0**	4.9	34.2	2.1	3.1	**26.5**	5.7	23.1	**13.9**	6.9	18.2	**28.0**
LoRA	**7.5**	22.9	**9.8**	73.7	**6.5**	**36.0**	**4.9**	**5.9**	25.3	**7.9**	**23.4**	12.8	**8.7**	**18.8**	25.9

Table 12: BEIR performance of asymmetric and symmetric dual encoders trained without hard negatives using NQ as the training dataset. Best from FT and LoRA in **bold**.

D RESULTS WITH NQ AS TRAINING DATASET

D.1 TRAINING METHOD

Table 12 shows that parameter-efficient training (PEFT) consistently achieves higher performance in out-of-domain compared to the traditional approach of training full parameters (FT).

D.2 BATCH DESIGN

Adding Hard Negatives Table 13 demonstrates that incorporating hard negatives consistently improves in-domain performance, yet it often reduces out-of-domain performance. Our experiments use negatives from (Karpukhin et al., 2020), including BM25 negatives (R_{BM25}) and model-based negatives (R_{DPR}). Please note that the model used for R_{DPR} is DPR (Karpukhin et al., 2020), a dense retrieval model known for its superior in-domain performance compared to BM25. R_{None} is when there is no hard negatives used during

		Out-Of-Domain (OOD)														In-Domain
		FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	MS	*Avg*	NQ
Full	$R_{BM25+DPR}$	13.2	42.1	17.6	28.3	5.8	32.1	24.4	20.8	33.8	40.2	48.6	16.1	19.6	26.4	**52.7**
	R_{DPR}	11.8	54.0	19.3	30.2	6.1	35.9	**25.6**	**25.0**	**34.1**	**40.7**	48.6	16.4	19.5	28.2	52.0
	R_{BM25}	10.4	35.5	15.8	19.2	4.9	25.9	19.1	10.8	32.9	38.0	**52.6**	**18.7**	18.1	23.2	51.7
	R_{None}	**16.0**	**61.2**	**20.5**	**31.0**	**7.3**	**41.8**	24.6	21.6	**34.1**	40.2	46.4	15.1	**21.2**	**29.3**	41.0
LoRA	$R_{BM25+DPR}$	13.9	54.5	20.5	22.9	7.5	43.5	26.4	20.8	40.4	45.4	**55.0**	21.6	20.8	30.2	**51.3**
	R_{DPR}	15.5	**63.9**	20.7	37.8	7.8	**44.5**	**31.8**	**25.8**	**40.8**	**48.4**	50.0	21.6	**22.4**	**33.2**	47.8
	R_{BM25}	10.2	51.0	17.2	18.5	6.3	28.2	17.8	13.9	35.8	39.1	49.1	18.1	17.1	24.8	43.5
	R_{None}	**16.6**	63.1	**21.2**	**37.4**	**8.3**	42.6	25.6	22.4	35.0	41.0	52.4	**24.6**	20.0	31.6	38.2

Table 13: Adding hard negatives consistently enhances in-domain performance (NQ). However, for out-of-domain tasks, adding hard negatives tends to degrade performance unless they are selected very carefully (LoRA R_{DPR}). Such results suggest that adding hard negatives makes the model adapt strongly to specific datasets, making it challenging to generalize effectively across different domains. All experiments in the table use the asymmetric dual encoder with NQ as the training dataset.

		Out-Of-Domain (OOD)														In-Domain
	batch size	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	MS	*Avg*	NQ
Full	B	16.0	61.2	20.5	31.0	**7.3**	41.8	**24.6**	21.6	34.1	40.2	46.4	15.1	21.2	29.3	41.0
	2 × B	**17.3**	**64.6**	**22.4**	**33.0**	6.9	**45.2**	23.5	**21.9**	**34.3**	**41.9**	**47.1**	**16.8**	**21.6**	**30.5**	**46.1**
LoRA	B	16.6	63.1	21.2	37.4	**8.3**	42.6	25.6	22.4	35.0	41.0	52.4	24.6	20.0	31.6	38.2
	2 × B	17.0	63.9	21.7	39.7	7.7	44.6	**32.1**	**23.1**	35.7	**46.7**	**52.6**	25.4	21.3	33.2	41.6
	4 × B	**17.6**	**64.8**	**22.5**	**42.8**	**8.3**	**45.8**	29.9	**23.1**	**35.8**	43.9	51.8	**26.0**	**22.4**	**33.4**	**42.4**

Table 14: Increasing batch size consistently improves both in-domain and out-of-domain performance. Experiments are conducted with asymmetric dual encoders, without hard negatives, and using NQ as a training dataset.

the training step and only with in-batch negatives. $R_{BM25+DPR}$ is when using a mix of DPR and BM25 negatives hard negatives, effectively doubling the training dataset size[7].

In both PEFT and FT experiments, we observe that $R_{BM25+DPR}$ achieves the best in-domain dataset performance, while R_{None} shows the lowest, confirming that hard negatives enhance in-domain dataset performance. However, for out-of-domain datasets, R_{None} performs best, and adding negatives seems to worsen performance. This indicates that hard negatives may cause the model to overfit to a single training dataset distribution, limiting its generalization to out-of-domain datasets.

As observed in various studies on hard negative selection (Santhanam et al., 2021; Formal et al., 2021), we could also see case where hard negatives mined with high-performance models improve out-of-domain performance (LoRA performance with R_{DPR}). Conversely, R_{BM25}, which utilizes BM25 for negatives, consistently lowers performance, even when combined with R_{DPR}. We speculate that this significant drop with R_{BM25} relates to BEIR's tendency to show high performance with BM25, leading to a higher incidence of false negatives. This finding highlights the critical importance of selecting appropriate hard negatives for out-of-domain performance.

[7] $R_{BM25+DPR}$ contains twice as many training datasets compared to others since each query includes two instances, one negative from BM25 and one from DPR

		Out-Of-Domain (OOD)														In-Domain
		FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	MS	*Avg*	NQ
Bert Base																
Full	R_{DPR}	11.8	54.0	19.3	30.2	6.1	35.9	25.6	25.0	34.1	40.7	48.6	16.4	19.5	28.2	**52.0**
	R_{None}	16.0	61.2	20.5	31.0	7.3	41.8	24.6	21.6	34.1	40.2	46.4	15.1	21.2	29.3	41.0
LoRA	R_{DPR}	15.5	**63.9**	20.7	**37.8**	7.8	**44.5**	**31.8**	**25.8**	**40.8**	**48.4**	50.0	21.6	**22.4**	**33.2**	47.8
	R_{None}	**16.6**	63.1	**21.2**	37.4	**8.3**	42.6	25.6	22.4	35.0	41.0	**52.4**	**24.6**	20.0	31.6	38.2
Bert Large																
Full	R_{DPR}	14.9	46.6	20.1	59.0	6.1	40.5	20.9	21.3	34.6	37.8	46.1	15.2	19.8	30.1	52.2
	R_{None}	14.8	41.1	22.2	57.6	7.7	44.3	24.3	31.1	34.9	38.9	44.7	16.9	21.5	30.8	43.6
LoRA	R_{DPR}	**18.2**	**58.9**	**23.3**	**60.5**	**7.8**	**48.2**	**28.2**	**31.6**	**42.2**	**40.2**	**51.4**	22.2	**22.9**	**35.0**	**52.4**
	R_{None}	18.1	58.4	22.3	58.9	7.5	46.9	26.9	31.3	35.7	34.7	50.0	**24.9**	20.6	33.6	42.8
RoBERTa Large																
Full	R_{DPR}	18.1	40.7	22.0	70.2	6.4	39.4	23.5	35.1	30.4	35.2	42.8	18.6	20.4	31.0	52.4
	R_{None}	19.5	41.4	24.4	72.6	**7.4**	46.2	22.0	32.3	32.2	35.3	40.8	17.9	22.1	31.9	44.4
LoRA	R_{DPR}	**21.8**	**51.8**	**25.4**	71.8	7.2	**48.5**	**24.8**	**37.8**	**38.8**	**41.2**	**48.4**	25.6	**23.2**	**35.9**	**56.5**
	R_{None}	21.6	51.7	24.3	**73.4**	7.2	43.4	23.9	37.2	35.7	40.7	46.6	**25.7**	21.8	34.9	46.1

Table 15: Performance of different sizes of asymmetric dual encoder trained with NQ without hard negatives.

	Out-Of-Domain (OOD)														In-Domain
	FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	MS	*Avg*	NQ
FT	13.5	**30.4**	23.0	**72.8**	9.2	64.6	4.4	**37.9**	**42.0**	14.1	38.8	**11.3**	10.9	28.7	**25.7**
LoRA	**16.9**	28.3	**25.3**	**72.8**	**9.3**	**64.8**	**5.4**	35.2	40.1	**15.7**	**40.3**	11.7	**12.8**	**29.1**	23.7

Table 16: Performance of Colbert trained with hard negatives sampled from DPR model and using NQ as the training dataset.

Increasing Batch Size Table 14 shows that increasing batch size improves both in-domain and out-of-domain performance. Such results suggest that when given the same GPU configuration, increasing batch size would be a good option to further improve performance rather than adding hard negatives.

D.3 Robustness of results across different resource-intensive methods

(1) Increasing model size Table 15 shows that both in-domain and out-of-domain performance tends to increase with model size and better base model. Having a better base model consistently leads to higher performance in out-of-domain. LoRA tends to gain more benefits from a better base model, demonstrating higher improvements when considering the average over total scores and even outperforming in-domain scenarios.

(2) Use of token-level late-interaction models Table 16 shows the result of Colbert (Khattab & Zaharia, 2020), a widely used token-level late-interaction model, performance when trained with NQ. The trend tends to persist; LoRA shows higher performance on out-of-domain whereas lower performance in the in-domain dataset.