A Simple Recipe for Improving Out-of-Domain Retrieval in Dense Encoders

Anonymous ACL submission

Abstract

When developing retrievers for low-supervision domains, prevailing research practice relies on out-of-domain (OOD) generalization of retrievers trained on large supervised datasets. In this work, we investigate the extent to which retriever training strategies (e.g., parameter-efficient fine-tuning, hard negative mining, in-batch negatives) which have been validated for in-domain retrieval settings, translate well to the OOD setting (e.g., train on MSMARCO, test elsewhere). Through extensive experiments training dense encoders on large datasets and testing zero-shot capabilities on the BEIR benchmark, we recommend a simple recipe: First, parameter-efficient methods like LoRA yield better OOD performance than full parameter tuning. Next, with the reduced memory footprint of LoRA, we can afford larger batch sizes (and thus more in-batch negatives), which further improves OOD performance. Finally, we caution against mining hard negatives despite their common mainstay for in-domain retrieval, as they have a tendency to harm OOD performance. We find this recipe is complementary to other methods for OOD generalization, including base model scaling, further pretraining, and novel retriever architectures.[1]

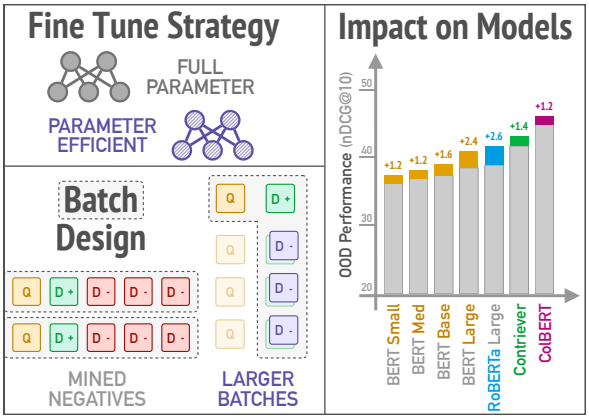

Figure 1: Overview of strategies to improve out-of-domain (OOD) retrieval. We investigate the effect of different fine-tuning methods and batch design; we find that a simple recipe using parameter-efficient methods (§3) and increased batch size rather than mining hard negatives (§4). We find this recipe is complementary to other methods for OOD generalization (§5).

1 Introduction

Dense neural retrieval models have been shown to be effective in many information retrieval (IR) settings (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 embeddings. 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 (OOD) test settings (Thakur et al., 2021b; Ren et al., 2022; Lupart et al., 2023). In most real-world applications, creating a large supervised dataset is impractical or infeasible.

Many approaches have been proposed to tackle OOD generalization. For example, some works propose pretraining with contrastive loss for encoders that can generalize without explicit supervision (Izacard et al., 2021; Xu et al., 2022). Others introduce novel architectures that assess relevance at the token-level with multi embeddings rather than employing a single embedding per passage (Khattab and Zaharia, 2020a; Formal et al., 2021; Lee et al., 2023). Empirical observations have suggested that increasing the model size leads to better out-of-domain performance (Ni et al., 2021b). These methods, while more effective than statistical IR techniques, still underperform super-

[1] Code available at `anonymous.4open.science/r/simple-recipe-to-improve-OOD-35EE`

vised retrievers unless they are then also fine-tuned on large supervised datasets. In fact, despite being OOD for many retrieval settings, large supervised collections like MSMARCO remain critical to improving zero-shot retrieval, particularly for larger and well-trained retrievers (Ni et al., 2021b; Rosa et al., 2022; Lin et al., 2023; Weller et al., 2023).

Complementary to these directions, we focus our attention on the training strategies themselves commonly used to fine-tune retrievers on large supervised datasets. Rich literature exists, of course, validating select training methods for the in-domain retrieval setting. For example, (1) training retrievers with parameter-efficient fine-tuning (PEFT) (Litschko et al., 2022; Ma et al., 2022; Pal et al., 2023; Jung et al., 2021; Yoon et al., 2023), (2) mining higher-quality hard negatives (Qu et al., 2020; Lindgren et al., 2021; Cai et al., 2022) and (3) increasing in-batch negatives (Zhan et al., 2021; Xiong et al., 2020; Karpukhin et al., 2020; Gao et al., 2021b) have been shown to improve in-domain performance. In contrast, comparatively fewer studies have examined whether these techniques also perform well in OOD settings (Tam et al., 2022). 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 out-of-domain generalization?
- **(RQ2)** How might we modify the design of our batches, such as through hard negative mining and increasing in-batch negatives, for better out-of-domain performance?
- **(RQ3)** To what extent do our recommendations complement other approaches for improving out-of-domain generalization, such as larger base models, pretraining, and novel retriever architectures?

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 (§2).

Addressing RQ1 in §3, we validate an intuitive tradeoff—LoRA, one of the most widely-used **PEFT** techniques, leads to better OOD generalization performance compared to full parameter tuning, but full tuning still outperforms LoRA on in-domain (ID) settings. Nevertheless, we provide a further analysis showing that even when considering this tradeoff between ID and OOD 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 ID and OOD settings.

Further, addressing RQ2 in §4, we find that contrary to their well-established benefit in ID settings, mined **hard negatives** tends to hurt OOD 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 due to its lower memory footprint. Specifically, under identical GPU configurations, increasing the in-batch size typically yields better OOD performance compared to adding hard negatives.

Finally, addressing RQ3 in §5, we find that our learnings (i.e., using LoRA and prioritizing in-batch negatives over hard negatives) lead to consistent improvements even atop other popular techniques for enhancing OOD performance, such as adopting larger base models, using novel retriever architectures (e.g., late interaction models), or performing on top of base model with contrastive pretraining.

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 on the supervised training dataset. Finally, taking all our findings together, our work provides simple, actionable takeaways that yield better OOD generalization for neural retrieval models that we recommend as complements to other methods.

2 Experimental Methodology

There are many important decisions to be made when training a dense retriever. To study the effects of the training strategies in our research questions, we design a careful experimental procedure that aims to isolate the effects of five key retriever design considerations: (1) pretrained base model, (2) dense retriever architecture, (3) fine-tuning strategy, (4) batch design, and (5) datasets for training and testing.

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 and Zaharia, 2020a). To study how controlled changes in the base model affect retrieval performance, we experiment over different variations of BERT: (1) Different model sizes (Small, Medium, Base, Large) to study the effect of scale, (2) RoBERTa (Liu et al., 2019) and Contriever (Izacard et al., 2021) to study variation as a result of pretraining strategy without significant base model size or architectural changes. For example, comparing retrievers trained on Contriever pretrained weights to retrievers trained on BERT-Base allows us to isolate the effect of retrieval-motivated continued pretraining using a contrastive loss.

Dense retriever architectures There are various architectural designs of encoder-only retrievers worth studying. We consider dual encoder architectures, which 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 Karpukhin et al. (2020), we use the first token embedding (`[CLS]`) as the representative embedding. In the **symmetric** dual encoder, the weights of the query and document encoders are shared. Following Izacard et al. (2021), we average all token embeddings as the representative embedding. In the **late interaction** dual encoder (Lee et al., 2023; Khattab and Zaharia, 2020a), we use *multiple* token embeddings as representative embeddings, unlike the other two architectures which use a *single* representative embedding. Following Khattab and Zaharia (2020a), we share the weights of the query and document encoders and use MaxSim to score the similarity between each query against a bag of tokens.

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 low-rank adaptation (LoRA) (Hu et al., 2021) due to its wide usage (Dettmers et al., 2023; Chen 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, only training about 0.25% number of parameters compared to FT.

Batch designs When given certain GPU memory, two key factors when designing batches for training dense retrievers are (1) **batch size**, which corresponds to the number of query-positive document pairs, and (2) **hard negatives**, which are documents specifically mined to be negative examples for a query. Increasing either of these will increase the number of **in-batch negatives**. Additionally, we note batch size has an important consequence in our work: due to lower memory footprint of PEFT, one can afford batch size while keeping within same overall GPU memory budget. In fact, we find using LoRA can afford about four times larger batch size compared to FT given the same GPU specification. As such, we define a fixed per-GPU batch size **(B)**, as well as settings for twice **(2B)** and four times **(4B)** larger per-GPU batch sizes.[2]

Regarding mined hard negatives, we focus on a representative set of widely-used techniques: (1) BM25, (2) **model self-distillation**[3] during training (Karpukhin et al., 2020; Izacard et al., 2021; Khattab and Zaharia, 2020a), in which we use a retriever of identical architecture trained without hard negatives, (3) an additional **denoising** step in which a more powerful model filters false negatives from an earlier hard negative mining step; we use the MSMARCO-trained cross encoder from (Qu et al., 2020), (4) using hard negatives that come provided with the training dataset (see Appendix§A.1), and finally (5) no hard negatives.

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

[2]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 overhead (Gao et al., 2021b). We believe there is benefit to studying implications of PEFT on batch design under GPU memory constraints, though we welcome future work to consider other settings.

[3]This technique, while widely employed in literature, does not appear to have established nomenclature and may be known under different names such as "warmup model." We follow the setup in Karpukhin et al. (2020) and use our best "no hard negatives" model to perform negative sampling.

		Out-Of-Domain (OOD)														In-Domain (ID)
		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: Performance of dual encoder models trained on MSMARCO and (zero-shot) tested on BEIR benchmark datasets. For each dataset and model, we **bold** highest score between full fine-tuning (FT) and LoRA. For all encoders, we see two trade-offs: (1) FT achieves higher in-domain (ID) but lower out-of-domain (OOD) performance; (2) mined hard negative (*w/ Neg*) boosts ID but reduces OOD performance.

MSMARCO (Campos et al., 2016), two popular large datasets that have been used successfully used for this purpose (Thakur et al., 2021b; Lee et al., 2023; Izacard et al., 2021; Gao et al., 2021a). See exact details of dataset provenance and size in Appendix §A.1. 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 and Zaharia, 2020a; Ni et al., 2021b; Weller et al., 2023; Tam et al., 2022). We provide references for all test datasets in Appendix §A.2. Of course, when training on MSMARCO, we treat MSMARCO evaluation as **in-domain** and all others as **out-of-domain**; analogously when training on NQ. All datasets are publicly available on Huggingface.

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[4]. We always train for the full 40 epochs, checkpointing after each epoch, and select optimal checkpoints using the provided validation set (see Appendix§A.1); models converged typically after 30-40 epochs. 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). 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 and Zaharia (2020a).

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., 2021a).

3 Favor LoRA over FT when seeking OOD 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. Here we report results from training on MSMARCO but note training on NQ also yields the same findings (Appendix §F.1).

[4]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.

We use dataset-provided hard negatives.

LoRA shows higher OOD performance 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 OOD datasets. This result is consistent across different architecture, and suggest that LoRA is a more effective training technique for maximizing OOD performance; conversely, models fully fine-tuned exhibit better ID performance. We speculate that this difference is due to FT models overfitting to ID distributions, thus making them less versatile across OOD datasets.

		Out-Of-Domain (OOD)		In-Domain (ID)	
		Avg	Rel. Diff (%)	MS	Rel. Diff (%)
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	

Table 2: Rel. Diff (%) is the percentage change between LoRA and FT trained models on ID and OOD datasets. Using LoRA over FT yields substantially better OOD performance (+4.1% to +12.0%) while incurring into a smaller ID penalty (-3.0% to -7.4%), suggesting better generalization.

LoRA offers a better trade-off between ID and OOD performance Table 2 quantifies the trade-off between better ID performance afforded by FT techniques against better OOD generalization with LoRA in the asymmetric dual encoder. It is evident that in all scenarios, the decrease in average performance for OOD datasets is more pronounced than the gains in ID datasets. These findings suggest LoRA is a more suitable approach for training models that perform well in both in-domain and OOD settings. The same result holds for the symmetric dual encoder (Table 8 in Appendix).

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 OOD 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 use asymmetric dual encoders for the remainder of our work, unless otherwise noted.

		Out-Of-Domain (OOD)		In-Domain (ID)	
		Avg	Rel. Diff (%)	MS	Rel. Diff (%)
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	

Table 3: Rel. Diff (%) is the percentage change in nDCG when using mined negatives. FT shows a more substantial reduction in OOD (-10.7% vs -1.6%) and a higher increase in ID performance (+6.3% vs +1.3%) than LoRA, indicating that mined negatives are less effective on LoRA and hurt OOD performance.

4 Favor in-batch negatives and avoid hard negative mining when seeking OOD performance

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. Here we report results from training on MSMARCO but note training on NQ also yields the same findings (Appendix §F.2).

Mined hard negatives often degrade OOD performance for both FT and LoRA models. Table 3 shows that hard negatives, despite generally enhancing ID performance, consistently degrade OOD nDCG scores (Numbers are derived from Table 1).[5] We hypothesize this is because models tend to over-adapt to the training dataset when adding hard negatives, making it challenging to generalize to datasets with differing distributions.

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 OOD tasks. The finding suggests that since FT trains a larger number of parameters than LoRA, it becomes more attuned to the given datasets, and further gets more affected by hard negatives. This results in more pro-

[5] Recall in these experiments, we use MSMARCO-provided hard negatives following Khattab and Zaharia (2020a). As some queries are missing negatives, we fill the queries with BM25 negatives.

Neg Source	Denoised	FT		LoRA	
		OOD	ID	OOD	ID
w/o Neg	x	37.4	31.1	39.0	30.2
MSMARCO	x	33.8	33.2	38.4	30.9
Random	x	36.6	30.8	38.3	30.4
BM25	x	26.7	21.4	28.6	26.6
Self-distillation	x	22.1	16.8	24.0	21.9
BM25	o	33.5	32.2	34.7	30.6
Self-distillation	o	37.3	35.2	37.1	33.7
RocketQA	o	37.9	36.9	39.6	36.4

Table 4: Effect of different models for hard negative mining, with (o) and without (x) denoising step. "MSMARCO" refers to the official negatives included in the dataset. "Self-distillation" indicates negative mined using a model trained with no negatives ("w/o Neg"). "RocketQA" is negatives provided by Qu et al. (2020).

nounced improvements in ID performance, but at the cost of a larger decrease in OOD performance. The trend persists for symmetric dual encoders as in Appendix Table 10.

OOD sees little to no benefit from well-known ID hard negative mining techniques. Previous works have demonstrated the importance of mining hard negatives for in-domain training (Qu et al., 2020; Lindgren et al., 2021). Table 4 examines the impact of mining hard negatives from various sources on ID and OOD performance. In particular, we evaluate several sources for hard negatives (MSMARCO, BM25, random sampling, self-distillation[6], and RocketQA), as well as, *denoising*, an additional filtering step to remove false negatives from the mined hard negatives. We adopt the cross-encoder method from Qu et al. (2020) for the denoising process. Our findings indicate that, in most scenarios, the OOD performance tends to decline when trained with hard negatives, whereas ID performance improves. Even when employing state-of-the-art mined hard negatives from RocketQA, the average improvement in OOD (+0.6) is minor compared to that in ID (+5.8).

Besides the limited effectiveness of mined hard negatives for OOD, the results show consistent gains when denoising mined candidates. This finding holds for both in-domain performance, confirming findings of Qu et al. (2020), as well as on out-of-domain tasks. We conclude that denoising might be a necessary—albeit not sufficient—step toward hard negative mining for OOD.

[6]hard negatives mined by using the model itself (Karpukhin et al., 2020)

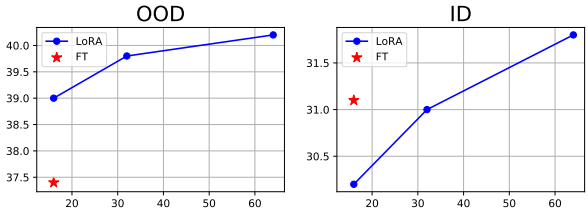

Figure 2: Increasing batch size (x-axis) increases the number of in-batch negatives, consistently improving nDCG (y-axis) in both in-domain (ID) and out-of-domain (OOD) tasks. Memory efficient models, like LoRA, can utilize up to 4× larger batch size compared to FT given the same GPU configuration.

Unlike mined negatives, using larger batch size increases both ID and OOD performance. In Figure 2, 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.

5 Our simple recipe combines well with other approaches for OOD 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 models that have undergone additional contrastive pretraining. Appendix §F.3 shows that the findings also hold with NQ as our training dataset.

Larger base models improve OOD performance and our recipe still yields improvements We study whether our findings hold when increasing the size of the base model, such as in Ni et al. (2021b). Figure 3 shows results using BERT mod-

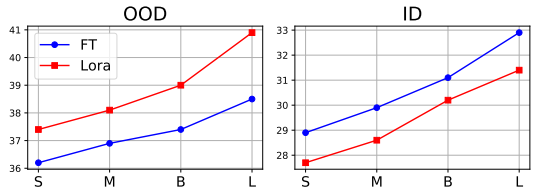

Figure 3: BEIR performance of different base model sizes of asymmetric dual encoder trained with MS-MARCO (no mined negatives). Performance increases with larger encoder models for both in-domain and out-of-domain. X-axis: BERT **S**mall, **M**edium, **B**ase, **L**arge.

els in four different sizes (small, medium, base, large) fine-tuned on MSMARCO. We can see that larger base model consistently leads to higher performance in all cases[7]. Results further confirm the trade-off between LoRA and FT we introduce in §3: namely, at all model sizes, we note that FT leads to better ID performance, but LoRA yields better OOD performance.

Interestingly, we also note that LoRA tends to benefit larger models more. For example, while BERT Small fine-tuned with LoRA outperforms FT by +1.2 nDCG points on average, BERT Large improves by +2.4 points. One possible explanation for this finding is that LoRA trained models are more likely to effectively leverage more powerful model without overfitting to query and document distribution in MSMARCO.

We further experiment by switching the base model from BERT-large to RoBERTa-large, as the latter generally outperforms BERT in many NLP tasks (Liu et al., 2019). We observe that when training with RoBERTa-large instead of BERT-large, both ID and OOD performance show improvement: average nDCG on OOD datasets improves from 38.5 to 38.9 (+0.4) when using FT, 40.9 to 41.5 (+0.6) when using LoRA; for ID, nDCG improves from 32.9 to 34.0 (+1.1) when using FT, and 31.4 to 32.1 (+0.7) with LoRA. Like in the case of different sizes of BERT models, FT outperforms LoRA for ID, while the opposite is true for OOD. See Appendix Table 20 for detailed results.

Our recipe improves OOD performance atop token-level late-interaction models We experiment with a late interaction dual encoder model following Khattab and Zaharia (2020a) to see whether our findings hold on diverse retrieval architectures[8]. Table 21 in Appendix shows a similar trend (FT: OOD Avg. 44.9, ID 39.2, LoRA: OOD Avg. 46.1, ID 37.2) to what we observed in asymmetric and symmetric dual encoders; LoRA surpasses FT in OOD settings, whereas FT demonstrates superior performance in ID settings.

Contrastive pretraining of base models improves OOD performance and our recipe still yields additional improvements 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 14 in Appendix shows that switching to this base model with additional pretraining significantly improves performance when compared with BERT Base. For FT, OOD nDCG increases from 37.3 to 41.7 (+4.4), and ID nDCG increses from 31.3 to 32.8 (+1.5). Similar gains can be observed for LORA: average OOD nDCG increases from 40.0 to 43.1 (+3.1), while ID increases from 29.4 to 31.5 (+2.1).

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

6 Related Work

Out-of-domain generalization in information retrieval Many data augmentation techniques have

[7] See Appendix Table 12 for numerical results.

[8] Our FT numbers are obtained through a replication of Khattab and Zaharia (2020a) 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. (2021a), indicating our successful replication. For more details, see Appendix B.3.

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 and Zaharia, 2020a), or using larger encoder models (Ni et al., 2021b; Neelakantan et al., 2022; Ma et al., 2023). However, these methods come with computational trade-offs: some require additional (expensive) training steps while others have higher computational burden at inference time. Our work studies training strategies that are more broadly accessible while still being complementary with more resource-intensive solutions.

PEFT for information retrieval 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 giving way as PEFT gains traction. Litschko et al. (2022) examined PEFT for multilingual information retrieval; Ma et al. (2022) studied PEFT to improve in-domain search capabilities; Pal et al. (2023) and Jung et al. (2021) applied PEFT to sparse retrieval systems and improving hybrid retrieval, respectively; Yoon et al. (2023) showed that PEFT helps adapt generative retrieval systems to new corpora. Tam et al. (2022) is closest to our work in that they also study PEFT for OOD generalization.[9] Our work differs from (Tam et al., 2022) in two key ways: (1) we validate PEFT amid other considerations like batch design (§4) and choice of base models (§5), and (2) we train different dense retrievers within an experiment group from the same base model whereas Tam et al. (2022) use different public base model checkpoints trained on different amounts of data for different retrievers.

Batch design for dense 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) "in-batch" negatives, gold passages of other queries in the same batch (Lindgren et al., 2021; Xiong et al., 2020; Gillick et al., 2019), and (2) "hard" negatives, additional passages that are challenging to distinguish from relevant passages (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 (Karpukhin et al., 2020; Qu et al., 2020; Xiong et al., 2020), 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 and Zaharia, 2020a; Gao et al., 2021a). We believe our work is one of the first to question this practice as the de facto standard.

7 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.

8 Limitations

Most of our experiments are conducted on smaller pretrained base models relative to some larger choices like Llama (Ma et al., 2023). While we

[9] Tam et al. (2022) study P-tuning v2 (Liu et al., 2022b) for zero-shot out-of-domain retrieval evaluations, while we focus on LoRA in our work. Despite different techniques, we view our combined findings as complementary evidence validating PEFT for OOD retrieval.

have demonstrated some consistency in our findings under model scaling (§5), we believe further investigation is needed for even larger models.

Further, while we performed investigation over various negative sampling strategies and denoising from (Qu et al., 2020), there is a line of research that mines hard negatives through various techniques (Santhanam et al., 2021; Formal et al., 2021; Ren et al., 2021). Further investigation is needed whether these intensive procedures can help overcome the costs of hard negative mining on OOD performance we observed in our study.

Finally, our study does not explore the compatibility of our recommended recipe with data augmentation methods for OOD generalization (Dai et al., 2022; Bonifacio et al., 2022; Saad-Falcon et al., 2023; Lin et al., 2023), as well as other approaches to OOD generalization such as training retrievers on a diverse combination of domains, rather than a single training dataset (NQ, MSMARCO).

References

Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *ArXiv*, abs/2106.10199.

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

Luiz Henrique Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. 2022. Inpars: Data augmentation for information retrieval using large language models. *ArXiv*, abs/2202.05144.

Vera Boteva, Demian Gholipour Ghalandari, Artem Sokolov, and Stefan Riezler. 2016. A full-text learning to rank dataset for medical information retrieval. In *European Conference on Information Retrieval*.

Yinqiong Cai, J. Guo, Yixing Fan, Qingyao Ai, Ruqing Zhang, and Xueqi Cheng. 2022. Hard negatives or false negatives: Correcting pooling bias in training neural ranking models. *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

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

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

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. *ArXiv*, abs/2004.07180.

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

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

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.

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

Thibault Formal, C. Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2021. Splade v2: Sparse lexical and expansion model for information retrieval. *ArXiv*, abs/2109.10086.

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

Luyu Gao, Yunyi Zhang, Jiawei Han, and Jamie Callan. 2021b. Scaling deep contrastive learning batch size under memory limited setup. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 316–321, Online. Association for Computational Linguistics.

Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. 2019. Learning dense representations for entity retrieval. In *Conference on Computational Natural Language Learning*.

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

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

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Towards unsupervised dense information retrieval with contrastive learning. *ArXiv*, abs/2112.09118.

Euna Jung, Jaekeol Choi, and Wonjong Rhee. 2021. Semi-siamese bi-encoder neural ranking model using lightweight fine-tuning. *Proceedings of the ACM Web Conference 2022*.

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

O. Khattab and Matei A. Zaharia. 2020a. 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*.

Omar Khattab and Matei Zaharia. 2020b. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT.

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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Hyunji Lee, Jaeyoung Kim, Hoyeon Chang, Hanseok Oh, Sohee Yang, Vladimir Karpukhin, Yi Lu, and Minjoon Seo. 2022a. Nonparametric decoding for generative retrieval. In *Annual Meeting of the Association for Computational Linguistics*.

Hyunji Lee, Sohee Yang, Hanseok Oh, and Minjoon Seo. 2022b. Generative multi-hop retrieval. In *Conference on Empirical Methods in Natural Language Processing*.

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

Jinhyuk Lee, Minjoon Seo, Hannaneh Hajishirzi, and Jaewoo Kang. 2019. Contextualized sparse representations for real-time open-domain question answering. In *Annual Meeting of the Association for Computational Linguistics*.

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

Erik M. Lindgren, Sashank J. Reddi, Ruiqi Guo, and Surinder Kumar. 2021. Efficient training of retrieval models using negative cache. In *Neural Information Processing Systems*.

Robert Litschko, Ivan Vulic, and Goran Glavavs. 2022. Parameter-efficient neural reranking for cross-lingual and multilingual retrieval. In *International Conference on Computational Linguistics*.

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

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022b. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

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

Y Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2020. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Simon Lupart, Thibault Formal, and Stéphane Clinchant. 2023. Ms-shift: An analysis of msmarco distribution shifts on neural retrieval. In *Advances in Information Retrieval*, pages 636–652, Cham. Springer Nature Switzerland.

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

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *ArXiv*, abs/2310.08319.

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

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

Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Nonparametric masked language modeling. In *Annual Meeting of the Association for Computational Linguistics*.

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. 2022. Text and code embeddings by contrastive pre-training. *ArXiv*, abs/2201.10005.

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

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. 2021b. Large dual encoders are generalizable retrievers. *ArXiv*, abs/2112.07899.

Vaishali Pal, Carlos Lassance, Hervé Déjean, and Stéphane Clinchant. 2023. Parameter-efficient sparse retrievers and rerankers using adapters. *ArXiv*, abs/2303.13220.

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

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

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

Kirk Roberts, Tasmeer Alam, Steven Bedrick, Dina Demner-Fushman, Kyle Lo, Ian Soboroff, Ellen M. Voorhees, Lucy Lu Wang, and William R. Hersh. 2020. 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.

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

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

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

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

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021a. 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)*.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. 2021b. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663.

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

A. Ustun and Asa Cooper Stickland. 2022. When does parameter-efficient transfer learning work for machine translation? In *Conference on Empirical Methods in Natural Language Processing*.

Henning Wachsmuth, Shahbaz Syed, and Benno Stein. 2018. Retrieval of the best counterargument without prior topic knowledge. In *Annual Meeting of the Association for Computational Linguistics*.

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

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

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

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke S. Zettlemoyer. 2019. Scalable zero-shot entity linking with dense entity retrieval. In *Conference on Empirical Methods in Natural Language Processing*.

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. 2020. Answering complex open-domain questions with multi-hop dense retrieval. *ArXiv*, abs/2009.12756.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval. In *Findings*.

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

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

Soyoung Yoon, Chaeeun Kim, Hyunji Lee, Joel Jang, and Minjoon Seo. 2023. Continually updating generative retrieval on dynamic corpora. *ArXiv*, abs/2305.18952.

Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, M. Zhang, and Shaoping Ma. 2021. Optimizing dense retrieval model training with hard negatives. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Zexuan Zhong, Tao Lei, and Danqi Chen. 2022. Training language models with memory augmentation. In *Conference on Empirical Methods in Natural Language Processing*.

A Dataset

A.1 MSMARCO and NQ

For MSMARCO, we use the official released dataset.[10] It contains 503k training and 56k validation examples. For NQ, we use the dataset released by Karpukhin et al. (2020). It contains 59k training and 7k validation examples. These two datasets also come with provided hard negatives which we use for experiments described in our setup §2.

[10] https://microsoft.github.io/msmarco/

A.2 Test datasets from BEIR

Table 5 provides references to the 14 datasets in the BEIR benchmark, a publicly available dataset with Apache-2.0 license.

B Experimental Methodology

B.1 Model size of base models

Table 6 shows the number of parameters and the time it takes to train a single epoch. All reported results from experiments are from a single training run.

B.2 Code base

All our code bases for experiments are built on top of the released code from Karpukhin et al. (2020), Izacard et al. (2021), and Khattab and Zaharia (2020b) each for dense retrieval architecture of asymmetric dual encoder, symmetric dual encoder, and late interaction dual encoder, respectively. For RocketQA (Qu et al., 2020), we used the released checkpoint and code base from its official repo.

B.3 Reproducing Colbert

To study each training configuration's impact and ensure a fair comparison, we replicated the experiment for consistent results (Ours in Table 7). The result aligns closely with the result of Colbert in Thakur et al. (2021a) (Beir in Table 7).

C LoRA improves OOD retrieval over FT

Table 8 shows the trade-off between in-domain and out-of-domain performance for both results of asymmetric and symmetric dual encoder.

D Favor in-batch negatives and avoid hard negative mining when seeking OOD performance

Table 9 show full results of each dataset by various methods to mine hard negatives and by adding a denoising step. Table 11 show full results when increasing batch size (increasing number of in-batch negatives). Table 10 show the degradation of OOD and improvement in ID for both asymmetric and symmetric dual encoders.

	Task	Domain	Num of Test dataset	Num of Corpus
TREC-COVID (TR) (Roberts et al., 2020)	Bio-Medical Information Retrieval	Bio-Medical	50	171,332
NFCorpus (NF) (Boteva et al., 2016)	Bio-Medical Information Retrieval	Bio-Medical		
NaturalQuestions (NQ) (Kwiatkowski et al., 2019)	Question Answering	Wikipedia	3,452	2,681,468
HotpotQA (HO) (Yang et al., 2018)	Question Answering	Wikipedia	7,405	5,233,329
FIQA-2018 (FI) (Maia et al., 2018)	Question Answering	Finance	648	57,638
ArguAna (AR) (Wachsmuth et al., 2018)	Argument Retrieval	Misc.	1,406	8,674
Touche-2020 (TO) (Bondarenko et al., 2020)	Argument Retrieval	Misc.	49	382,545
Quora (QU)	Duplicate-Question Retrieval	Quora	10,000	522,931
DBpedia (DB) (Hasibi et al., 2017)	Entity-Retrieval	Wikipedia	400	4,635,922
MSMARCO (MS) (Campos et al., 2016)	Passage-Retrieval	Misc.	6,980	8,841,823
SCIDOCS (SD) (Cohan et al., 2020)	Citation-Prediction	Scientific	1,000	25,657
FEVER (FE) (Thorne et al., 2018)	Fact Checking	Wikipedia	6,666	5,416,568
Climate-FEVER (CL) (Diggelmann et al., 2020)	Fact Checking	1,535	5,416,593	
SciFact (SF) (Wadden et al., 2020)	Fact Checking	300	5,183	

Table 5: Data statics of 14 datasets in BEIR benchmark

	Number of parameters	time
BERT-small	29.2M	0.3H
BERT-medium	41.7M	0.5H
BERT-base	110M	1H
BERT-large	340M	2.5H
RoBERTa-large	355M	2.5H
Contriever	110M	1H

Table 6: Number of parameters for base models we experiment on and time it took to run a single epoch (H means hours).

E Our simple recipe combines well with other approaches for OOD retrieval

E.1 Performance of different model sizes when trained with MSMARCO

Table 12 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. Also Table 20 show that using a better base model improves performance (RoBERTa-large vs. BERT-large).

E.2 Pretraining with contrastive loss

Table 14 shows the result of BERT-base and BERT-base with contrastive pretraining, which we use Contriever (Izacard et al., 2021). Please note that for the result, as Contriever was trained in the architecture of symmetric dual encoder, the results are from the architecture.

F Results with NQ as training dataset

F.1 LoRA vs. FT

Table 15 shows that parameter-efficient training (PEFT) consistently achieves higher performance in out-of-domain compared to the traditional approach of training full parameters (FT).

F.2 Batch Design

Adding Hard Negatives Table 16 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 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[11].

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.

[11] $R_{BM25+DPR}$ 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	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 7: "Ours" is the performance of Colbert we replicate for fair comparison and "Beir" is the performance provided from Table 21, which is widely used. The performance tends to be similar.

		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 8: 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.

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.

Increasing Batch Size Table 17 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.

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

(1) Increasing model size Table 18 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 19 shows the result of Colbert (Khattab and Zaharia, 2020a), 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.

G Comparing results from MSMARCO with results from 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 5a 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 §4, using mined hard negative when training on NQ seems to negatively

Neg Source	denoise		Out-Of-Domain (OOD)														In-Domain
			FI	TR	NF	QU	SD	SF	TO	AR	HO	DB	FE	CL	NQ	*Avg*	MS
w/o Neg	x	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
MSMARCO	x	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
BM25	x	FT	10.8	48.5	21.7	64.4	3.5	29.1	13.5	19.9	27.9	22.3	50.1	10.9	24.1	26.7	21.4
		LoRA	11.2	45.0	22.1	64.7	5.3	31.8	16.7	21.6	35.0	28.1	53.7	11.4	26.0	28.6	26.6
Self-Model	x	FT	9.5	35.1	20.1	52.7	3.2	26.5	17.9	17.6	14.8	25.7	33.2	9.9	21.6	22.1	16.8
		LoRA	10.2	44.6	23.3	56.6	3.5	26.4	21.6	12.0	14.1	26.5	36.8	15.2	20.9	24.0	21.9
Random	x	FT	19.4	62.1	26.1	72.5	7.1	44.0	24.7	32.1	42.8	43.9	54.8	13.7	32.9	36.6	30.8
		LoRA	20.2	62.6	26.7	74.0	7.6	49.7	30.4	32.8	44.1	43.3	57.6	14.6	33.7	38.3	30.4
BM25	o	FT	19.4	63.8	23.9	54.9	6.0	40.5	23.0	23.0	35.4	34.7	57.7	14.9	38.6	33.5	32.2
		LoRA	18.9	66.1	24.5	61.0	6.6	40.6	28.3	24.3	36.7	33.5	58.9	15.4	36.8	34.7	30.6
Model (Self)	o	FT	19.7	70.7	25.8	65.9	6.4	42.7	26.7	24.1	42.4	41.7	61.3	14.9	42.7	37.3	35.2
		LoRA	18.9	68.9	24.3	69.1	7.6	42.5	29.0	26.0	44.1	42.4	58.1	15.3	36.6	37.1	33.7
Other Model	o	FT	20.3	71.1	23.7	68.3	6.6	41.7	31.0	27.2	42.7	40.2	58.2	16.7	45.1	37.9	36.9
		LoRA	20.1	75.8	26.4	74.7	6.9	45.1	30.1	24.5	43.6	42.6	62.2	17.7	45.5	39.6	36.4

Table 9: Full result of when trained with hard negatives mined from various sources and when adding denoising step.

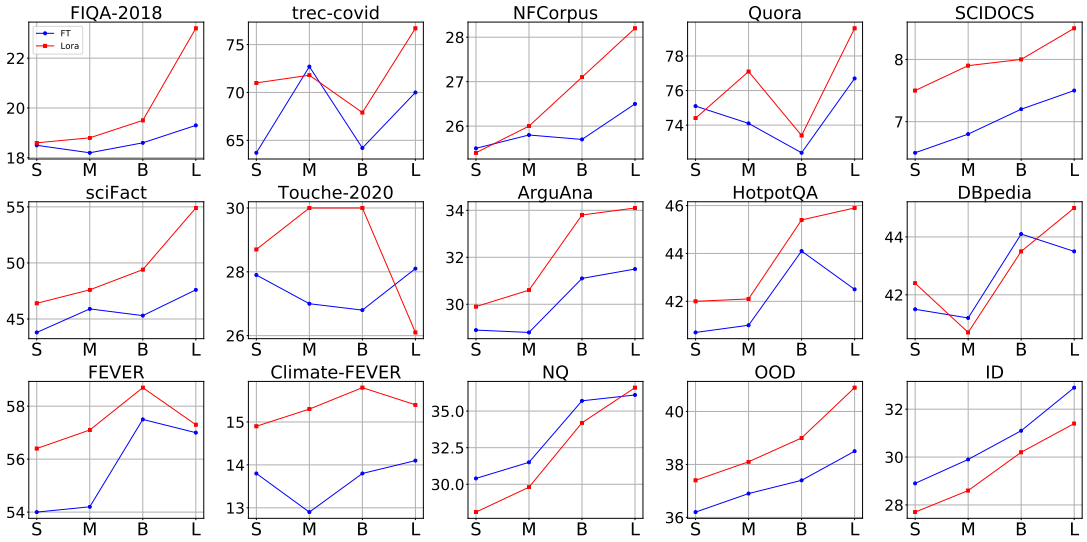

Figure 4: Overall BEIR performance of different base model sizes of asymmetric dual encoder trained with MSMARCO (without hard negatives). Performance consistently increases with larger encoder models for both in-domain and out-of-domain. Notation in x-axis indicates **S**: Bert Small, **M**: Bert Medium, **B**: Bert Base, **L**: Bert Large

affect out-of-domain performance. Figure 5b compares the effect of negatives on in-domain and out-of-domain performance for FT and LoRA trained asymmetric dual encoders. Compared to §4, we experiment with two different rankers to select hard negatives: BM25 and model-based negatives (DPR[12]). We also evaluate using a combination of both ("BM25+DPR" in fig. 5b). 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. 5b). 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

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

		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 10: 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.

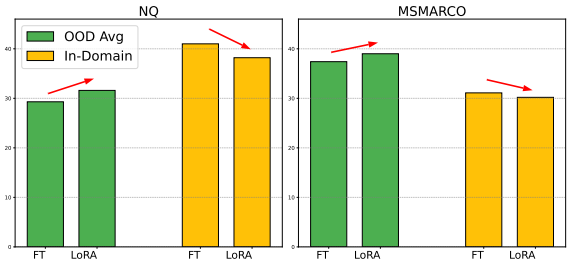

Figure 5(a): Performance of Asymmetric Dual Encoder when trained with NQ (left). Results tend to be similar to that of MSMARCO (right): LoRA tends to show higher performance in out-of-domain (OOD) but lower performance in in-domain.

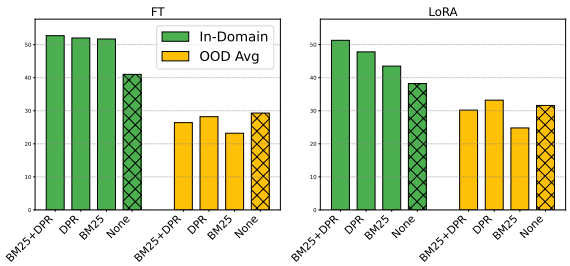

Figure 5(b): Performance of dual encoders trained on the NQ dataset with different negative sampling strategies. Adding hard negatives improves in-domain performance but degrades out-of-domain (OOD) performance for both FT and LoRA.

case, incorporating negative examples is consistently beneficial.

Advantages of LoRA on out-of-domain generalization is consistent across model sizes. Figure 6 presents the in-domain and out-of-domain performance across various model sizes. We can see that the trend when trained with MSMARCO persists; a superior base model consistently yields

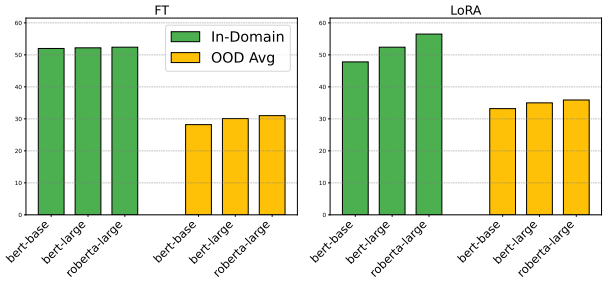

Figure 6: Impact of model size on the performance of dual encoders trained on NQ. Larger models consistently lead to higher performance in both in-domain and out-of-domain (OOD). LoRA shows larger gains when moving to larger models.

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 15 in Appendix F.1), confirming trends observed by Thakur et al. (2021b) 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 and Stickland, 2022; Litschko et al., 2022).

		Out-Of-Domain (OOD)														In-Domain (ID)
	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 11: Increasing batch size (increasing the number of in-batch negatives) consistently helps both in-domain and out-of-domain performance.

	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 12: 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	NQ	*Avg*	MS
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	**32.9**
	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
RoBERTa Large	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

Table 13: 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 (§3) and the possible detriment of hard negatives (§4) hold here as well.

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

	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 15: 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**.

		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 16: 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 17: 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.

		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 18: 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 19: Performance of Colbert trained with hard negatives sampled from DPR model and using NQ as the training dataset.

		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 20: 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 (§3) and the possible detriment of hard negatives (§4) hold here as well.

	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 21: Overall BEIR performance of a token-level late-interaction dual encoder following Khattab and Zaharia (2020a) 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.