LOGIC BUS: WHEN OPERATORS MATTER, NOT JUST VECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Two-tower models are trained to align the vectors their encoders produce, while the encoders themselves—the operators—are left unused as a choice. Beyond the default of passing each side through its own tower, a trained system holds two alternatives, at no training cost, that beat it on part of the input space: the second tower applied to both sides, and the pretrained backbone applied to both. We call a system that chooses among these pairings a Logic Bus. Contrastive training fits the two towers on different marginals of the training pairs, so they differ in training coverage. We show that, when fitting follows data density, each tower's bias grows as its coverage thins. It follows that no fixed pairing is least biased everywhere, provided each tower covers some inputs more densely than the other, and the backbone is better than the towers on some inputs and worse on others; when either condition fails, the choice shrinks or disappears. In an image–text catalogue, reducing only training coverage moves the trained towers from 20 category-level MRR points above the backbone on the tail to 52 below it. In text retrieval the best pairing changes from dataset to dataset: the three fixed pairings score within about one point of each other in aggregate, while choosing the pairing per dataset gains 8.4 nDCG@10 over the best of them.

1 INTRODUCTION

A trained two-tower model is used in one way: each side of a pair passes through its own tower, and the two vectors are compared (Huang et al., 2013; Karpukhin et al., 2020; Radford et al., 2021). Training aligns those vectors (van den Oord et al., 2018), and when the system fails on part of its inputs the remedies change them again, through more or synthetic data, re-weighting or distillation (Wang et al., 2022; Bonifacio et al., 2022; Menon et al., 2021; Reddi et al., 2021). The towers themselves—the mappings from inputs to vectors, which we call *operators*—are not treated as something to choose.

The question arose in a production recommendation system. For many infrequent queries it returned the same items whatever the query said; encoding those queries with the item tower instead of the query tower brought back relevant items (Appendix B.3).

A trained system can score a pair in three ways, its *configurations* (Figure 1): its own towers (OWN), the second tower on both sides (ROUTE), and the pretrained backbone on both sides (RAW). A two-tower system used this way is what we call a *Logic Bus*: its operators stay fixed, and a dispatcher decides which configuration scores an input, or how the three scores are weighted.

Our contributions are threefold.

- **The problem.** The default configuration is regularly beaten by configurations the system already contains, and averaging over datasets hides this (Section 4.2).
- **Why it arises.** If fitting follows data density, bias grows as coverage thins and, under two stated conditions, no configuration is least biased everywhere (Section 3); coverage alone reproduces the reordering over a frozen backbone (Section 4.3).
- **What a Logic Bus recovers.** Choosing a configuration per dataset, per query cluster or without labels beats every fixed configuration (Section 4.4); no label-free per-query rule was found (Section 4.5).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

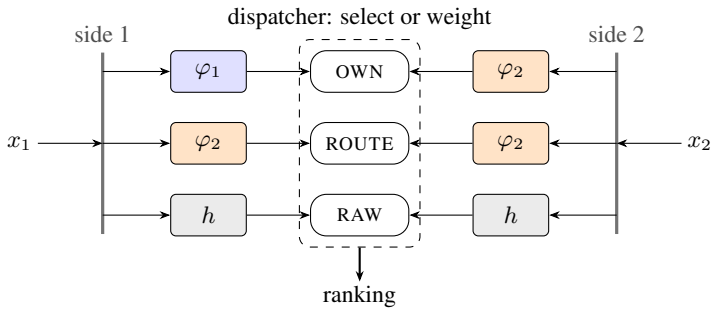


Figure 1: A Logic Bus. The operators of a trained two-tower system—the towers φ_1, φ_2 and the pretrained backbone h—stay fixed. Each side's input reaches all of them, and the three pairings give three scores: OWN (φ_1, φ_2), ROUTE (φ_2, φ_2) and RAW (h, h). A dispatcher selects one score or weights the three, per domain or per input; nothing is retrained.

2 Related Work

Two-tower models and the asymmetry of their sides. Two-tower models encode each side separately and are trained contrastively on paired data, in text retrieval (Huang et al., 2013; Karpukhin et al., 2020), image–text retrieval (Radford et al., 2021; Zhai et al., 2023) and recommendation (Covington et al., 2016; Yi et al., 2019). Whether the two sides should share parameters has been studied as an architectural choice made before training: siamese encoders outperform asymmetric ones for question answering unless the projection layer is shared (Dong et al., 2022), shared and specialised blocks can be interleaved in one encoder (Cheng et al., 2023), and the text side alone can be trained against a frozen image encoder (Zhai et al., 2022). The two sides are also known to differ after training: skewed item frequencies bias in-batch training in recommendation (Yi et al., 2019; Yang et al., 2020), and image and text embeddings remain systematically apart in contrastive models (Liang et al., 2022). We take this asymmetry as given and ask what a system already trained with it offers at inference.

Remedies act on the vectors. When a two-tower system fails on sparsely covered inputs, the usual remedies change the parameters that produce the vectors. Out-of-domain degradation in dense retrieval (Thakur et al., 2021) is addressed with synthetic training data (Bonifacio et al., 2022; Wang et al., 2022; Dai et al., 2023) or distillation (Hofstätter et al., 2020; Reddi et al., 2021); sparse long-tail feedback in recommendation with auxiliary self-supervised objectives (Yao et al., 2021) and corrected sampling (Yi et al., 2019; Yang et al., 2020); long-tail learning more generally with rebalanced losses (Menon et al., 2021). Closest in spirit, WiSE-FT observes that a zero-shot model is more robust under distribution shift than its fine-tuned version and interpolates their weights (Wortsman et al., 2022). It combines two versions of one encoder; we assign different operators to the two sides of a pairing, including the other side's tower.

Choosing among models at inference. Routing each query to one of several domain-specific expert embedding models improves on a single general-purpose model (Lee et al., 2025), and choosing a dense retriever for a new collection without labels has been posed as a problem in its own right, with existing unsupervised estimators found ineffective (Khramtsova et al., 2023)—consistent with our failure to find a label-free per-query criterion. These methods choose among separately trained models. The configurations we choose among are contained in one trained two-tower system and require no additional model.

3 Coverage and the Choice of Configuration

3.1 Setup

Let $h_1 : \mathcal{X}_1 \to \mathbb{R}^{d_0}$ and $h_2 : \mathcal{X}_2 \to \mathbb{R}^{d_0}$ be pretrained base encoders for the two sides (the same function when $\mathcal{X}_1 = \mathcal{X}_2$). Contrastive training from (h_1, h_2) on pairs $\{(x_1, x_2)\}$ yields towers

φ_1, φ_2. Let P_1, P_2 be the marginals of the training pairs. Each tower encodes only its own side, so φ_1 is fitted under P_1 and φ_2 under P_2 however symmetric the loss, and the two marginals generally differ: an item in many pairs is fitted many times, a query that occurs once is fitted once.

A configuration is a pair of encoders scoring (x_1, x_2) by the cosine of their outputs. The configurations available without further training are

$$\mathcal{C} \;=\; \big\{ \underbrace{(\varphi_1, \varphi_2)}_{\text{OWN}},\; \underbrace{(\varphi_2, \varphi_2)}_{\text{ROUTE}},\; \underbrace{(h_1, h_2)}_{\text{RAW}} \big\}. \tag{1}$$

The mirror (φ_1, φ_1) is omitted: it changes only the side-2 encoder, while the failure studied here is sparse coverage of side-1 inputs. Configurations are compared only through the rankings they induce.

3.2 Coverage governs bias

We treat every configuration as a linear map W applied to the base representation h: RAW applies the identity, OWN and ROUTE the maps induced by training. The bias of W on a region R of the representation space is its expected error there,

$$\mathcal{B}_R(W) := \mathbb{E}_{x \sim R} \, \|(W - W_R)\, x\|^2 = \operatorname{tr}\big[(W - W_R)\, \Sigma_R\, (W - W_R)^\top\big], \tag{2}$$

where W_R is the ideal map on R and Σ_R the second moment of h on R. The two trained configurations are modelled as sharing the optimum W_0 and second moment Σ_0 of the remaining, main region, and as differing only in the mass ε their fitting distributions place on R.

Assumption 1 (Density-governed fitting)**.** *The trained map is the minimiser of the density-weighted squared error under its fitting distribution.*

The assumption is exact for a linear head trained to convergence over a frozen backbone, approximate for shallow towers over a frozen backbone, and not expected to hold under end-to-end fine-tuning; Section 4.3 examines both regimes.

Lemma 1 (Coverage–degradation)**.** *Let the fitting distribution place mass ε on R, with $\Sigma_0, \Sigma_R \succ 0$. Under Assumption 1 the trained map $\widehat{W}$ satisfies*

$$\widehat{W} - W_R \;=\; (1 - \varepsilon)\, (W_0 - W_R)\, \Sigma_0 A^{-1}, \qquad A := (1 - \varepsilon)\Sigma_0 + \varepsilon \Sigma_R. \tag{3}$$

With S such that $S^\top \Sigma_0 S = I$ and $S^\top \Sigma_R S = \operatorname{diag}(\lambda_i)$ (a generalised eigendecomposition) and k_i the i-th column of $(W_0 - W_R) S^{-\top}$,

$$\mathcal{B}_R(\widehat{W}) = \sum_i \lambda_i \, c_i(\varepsilon)^2 \, \|k_i\|^2, \qquad c_i(\varepsilon) = \frac{1 - \varepsilon}{(1 - \varepsilon) + \varepsilon \lambda_i}, \tag{4}$$

which is non-increasing in ε, and strictly decreasing whenever $W_0 \neq W_R$.

On a region its fitting distribution covers poorly, a tower therefore follows the optimum of the main region. A difference between the marginals is not enough to make the choice matter: if P_2 covered every region at least as densely as P_1, ROUTE would be weakly better everywhere.

Proposition 1 (No dominant configuration)**.** *Write $\varepsilon_k(R)$ for the mass P_k places on R, and suppose*

- *(a) there are regions R and R', with $W_0 \neq W_R$ and $W_0 \neq W_{R'}$, such that $\varepsilon_1(R) < \varepsilon_2(R)$ and $\varepsilon_1(R') > \varepsilon_2(R')$;*
- *(b) there is a region on which $\mathcal{B}_R(I)$ is strictly below the smaller of the two trained configurations' biases, and a region on which it is strictly above it.*

Then no element of $\mathcal{C}$ is least biased on every region. If (a) fails, one of OWN *and* ROUTE *is weakly less biased than the other on every region; if (b) fails,* RAW *is either never strictly least biased or weakly least biased everywhere.*

The proposition concerns approximation bias. Which configuration wins on a given distribution, and whether choosing improves ranking, are measured in Section 4. Hypothesis (a) is weakened by a symmetric training distribution and (b) by a backbone already adapted to the task. Proofs and a numerical check are in Appendix A.

4 EXPERIMENTS

We ask whether the three configurations trade places in trained systems (§4.2), whether training coverage alone produces the reordering (§4.3), how much a Logic Bus recovers by choosing or combining without retraining (§4.4), and what it could not do (§4.5).

4.1 SETUP

Table 1: The two settings.

	Text retrieval	Image–text retrieval
Towers φ_1, φ_2	DPR question / passage encoders	two shallow MLPs over frozen CLIP
Backbone h (RAW)	BERT, mean-pooled	CLIP ViT-B/32
Data	8 BEIR datasets, 50k documents each	20k products, 136 categories
Groups compared	datasets (domains)	head / tail by category frequency
Metric	nDCG@10	category-level MRR

We use DPR (Karpukhin et al., 2020) trained on NQ with its BERT initialisation (Devlin et al., 2019), BEIR (Thakur et al., 2021), and CLIP (Radford et al., 2021) (Table 1). Each BEIR dataset is treated as a domain. Scores are cosine similarities; every choice is made on a validation half and evaluated on the other half. Details are in Appendix B.1.

4.2 THE CONFIGURATIONS TRADE PLACES

Table 2: The three configurations of DPR on eight BEIR datasets (nDCG@10, full query set). Best per row in bold.

Dataset	OWN	ROUTE	RAW	winner
NQ	**56.17**	25.90	15.70	OWN
SQuAD	55.65	50.56	**57.88**	RAW
HotpotQA	**28.43**	16.93	18.92	OWN
FIQA	**7.11**	3.61	2.41	OWN
NFCorpus	**14.84**	13.03	4.91	OWN
Quora	29.51	**71.61**	70.13	ROUTE
SciFact	20.73	**25.87**	16.70	ROUTE
ArguAna	3.81	7.01	**21.53**	RAW
Average	27.03	26.82	26.02	

In Table 2 four datasets prefer OWN, two ROUTE and two RAW, by margins of up to 42 points, yet the three averages lie within about one point. An evaluation on the aggregate finds no reason to choose. Quora and ArguAna pair inputs of the same form; the reordering also occurs on question-to-passage (SQuAD, RAW) and claim-to-abstract (SciFact, ROUTE) tasks.

The reordering does not depend on the checkpoint or the index. DPR-multiset, trained on four QA datasets, scores higher with OWN (NQ 56.74, SQuAD 61.84) but three datasets still prefer another configuration: Quora (70.13 for RAW against 35.92), ArguAna (21.53 against 8.95) and SciFact (30.30 for ROUTE against 24.00). Growing the index from 1k to 50k documents never changes whether OWN wins, and the margins are wider at 50k than at 1k (RAW minus OWN: Quora $+32.52 \rightarrow +40.62$, NQ $-34.84 \rightarrow -40.47$). Nor is it a general weakness of dense retrieval: in single-tower retrievers, which encode both sides with one set of weights so that ROUTE coincides with OWN, no reordering occurs—Contriever (Izacard et al., 2022) and multilingual-E5 (Wang et al., 2024) beat RAW on every dataset (47.36 and 62.68 against 26.02 on average). The reordering appears only where the two sides have separate weights.

4.3 COVERAGE ALONE PRODUCES THE REORDERING

Table 2 shows the reordering but not its cause: the BEIR datasets differ from NQ in domain as well as in coverage, and DPR is fine-tuned end to end. To isolate coverage, we keep the image–

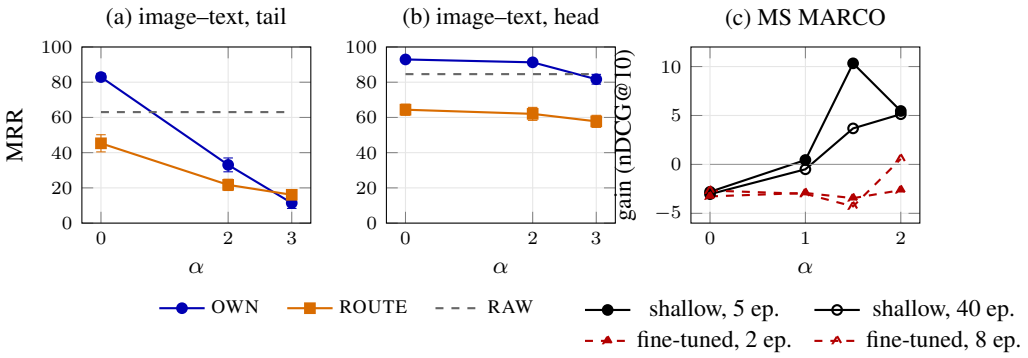


Figure 2: Training coverage and the configurations. (a, b) Image–text retrieval: only the concentration α of the training distribution varies; the share of products seen in training is 78%, 37.5% and 23% at $\alpha = 0, 2, 3$; mean and standard deviation over five seeds. (c) MS MARCO: gain of ROUTE over OWN in nDCG@10 for shallow towers over a frozen backbone and for full encoders fine-tuned end to end.

text architecture and data fixed and sample training pairs with probability proportional to category frequency raised to α, so that rare categories are seen less and less often (Figure 2a,b). The tail is the quarter of the held-out products whose categories are rarest in the catalogue. Over five seeds, on the tail the trained towers fall from 19.9 ± 3.2 MRR points above RAW to 51.6 ± 5.9 below it; on the head their margin stays positive until $\alpha = 3$, where it averages -2.9 ± 3.0. ROUTE, fitted on the denser marginal, follows a different curve: on the tail it trails OWN by 37.6 points at $\alpha = 0$ and passes it by 4.7 at $\alpha = 3$, in every seed, while on the head, where coverage stays high, it never catches up. A choice made on a validation half follows coverage identically in every seed: on the tail it takes OWN at $\alpha = 0$ and RAW at $\alpha = 2$ and 3, and on the head it keeps OWN until $\alpha = 3$.

The same manipulation on text separates the two regimes of Assumption 1 (Figure 2c). With shallow towers over frozen DistilBERT (Sanh et al., 2019) and MS MARCO (Bajaj et al., 2016) pairs drawn under a Zipf law over queries, concentrating training reduces the queries seen at least once from 8,327 to 1,313 and moves the gain of ROUTE over OWN from -2.80 to $+10.33$ nDCG@10. Across the eight runs, concentration correlates with a fixation index, the mean top-20 overlap between distinct queries (Spearman $+0.878$, $p = 0.004$), and fixation with the gain ($+0.857$, $p = 0.007$)—the behaviour first seen in the production system. Fine-tuning both full encoders on the same data leaves the gain negative in seven of eight runs and unrelated to fixation ($p = 0.53$). Coverage is thus a cause of the reordering for towers over a frozen backbone; whether it also contributes to the reordering of the fine-tuned DPR is not tested here.

4.4 USING THE LOGIC BUS: CHOOSING AND COMBINING

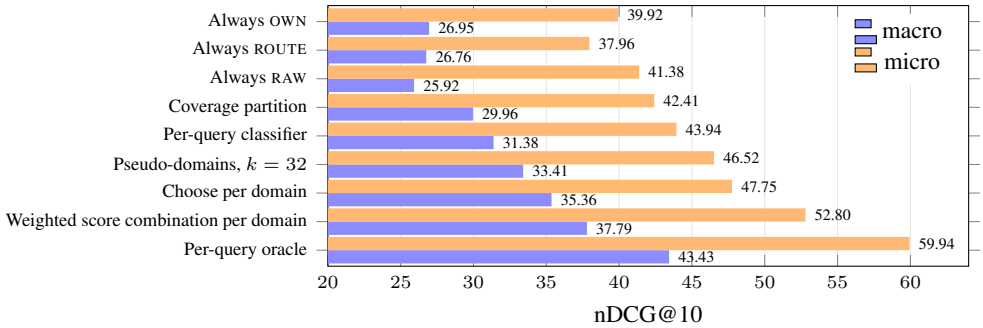


Figure 3: What a Logic Bus recovers on text retrieval without retraining, from a fixed configuration to the per-query oracle. Mean over 20 validation/test splits; macro averages datasets, micro averages queries. Full results with intervals in Appendix B.2.

A deployed Logic Bus has to decide without test labels. We split the 12,732 text queries into validation and test halves twenty times. SQuAD contributes 46.6% of the queries, so we report both the per-query (micro) average, which SQuAD dominates, and the per-dataset (macro) average. Figure 3 orders the strategies by the supervision they use.

Choosing per dataset. Taking the configuration that is best on each dataset's validation half gives 35.36 macro against 26.95 for the best fixed configuration, a gain of 8.4. Under 500 redraws that split each dataset separately, it wins every time (mean gain 8.31, middle 95% of redraws $[7.78, 8.79]$); a cluster bootstrap that also resamples the eight datasets gives 5.14, 95% interval $[0.50, 10.98]$. About 15 labelled queries per dataset already reach 34.00, 83% of the gain.

Without domain tags. Clustering the query stream by its backbone embeddings into 32 pseudo-domains and choosing per pseudo-domain recovers 77% of the gain macro (33.41) and 81% micro, although the groups barely match the datasets (adjusted Rand index 0.100). A coverage partition that uses no relevance labels—queries close to the training data go to OWN, the others to ROUTE or RAW depending on how far the passage tower moves them from the backbone—still beats every fixed configuration (29.96 macro, 42.41 micro). One global choice is worse than none (25.92 macro): pooled validation data favour RAW because SQuAD dominates them.

Combining. The three scores can also be combined instead of chosen between: each configuration's scores are z-scored per query and summed with weights fitted per dataset on the validation half. This gives 37.79 macro and 52.80 micro, above choosing. With weights fitted on each dataset's full query set, and so descriptive only, a mixture beats every single configuration on SQuAD and Quora ($57.88 \rightarrow 66.90$, $71.61 \rightarrow 76.59$), while on NQ and ArguAna the best weights reduce to one configuration. Equal weights (34.38 macro) and reciprocal rank fusion (Cormack et al., 2009) (32.16) fall below choosing. In image–text retrieval, averaged over the six segments and five seeds, choosing reaches 80.02 ± 1.92 and weighting 81.25 ± 1.72, against 73.76 ± 2.26 for the best fixed configuration; weighting is ahead of choosing in every seed, by 0.9 to 1.5 points.

4.5 LIMITS

Two results we sought do not hold. A ranking guarantee from the bias of Section 3 would need each query's embedding displacement δ to stay below half its margin γ between the positive and the strongest negative; the mean margin is negative on every dataset while δ averages 0.69–0.86, and the condition holds for 0 of 12,732 queries. Nor did we find a label-free rule per query. Agreement between the two towers predicts whether ROUTE beats OWN with AUCs from 0.27 to 0.64 within datasets, so even the direction of the relation changes from one dataset to another. A classifier on eight label-free query features, trained on oracle labels, is right for 63.7% of queries (chance 33.3%) yet scores 31.38 macro, below choosing per cluster. At the dataset level, none of twelve tests of six label-free signals survives correction for multiple comparisons (strongest Spearman $+0.83$, $p = 0.010$). The 8.1 points between choosing per dataset and the per-query oracle remain unexplained.

5 CONCLUSION

A trained two-tower system can score a pair with its own towers, with the second tower on both sides, or with the pretrained backbone, and the default is not reliably the best of the three. When fitting follows data density, a tower's bias grows where its training coverage thins, so no configuration is least biased everywhere. In text and image–text retrieval the configurations trade places, and a Logic Bus that chooses among them per dataset gains 8.4 nDCG@10 where the fixed configurations differ by one point; most of this remains without domain tags.

Limitations. The theory concerns approximation bias, not ranking quality, and its assumption describes towers trained over a frozen backbone better than encoders fine-tuned end to end. Choosing needs relevance labels on a validation split. The image–text setting judges relevance at the level of categories, which is coarser than product-level relevance. The industrial case study uses selected queries and a single annotator. We did not vary the asymmetry between the two training marginals directly, so its effect on hypothesis (a) is untested. We do not compare choosing configurations with remedies that change the vectors, such as adapting the towers to each target domain.

Future work. The largest open gap is per-query dispatch: 8.1 points separate choosing per dataset from the per-query oracle, and no label-free signal we tested closes it. The theory is written in terms of how densely training covered an input, which the owner of a deployed system can count from its training pairs. Estimating that coverage for each incoming query, and dispatching on it instead of on a validation split, is one route to closing it. A second direction asks how far the principle reaches beyond two towers. Any architecture whose components are fitted on different input distributions holds compositions of its trained parts other than the one it is deployed with. Whether networks can be designed so that such compositions are dispatched per input, instead of being fixed when training ends, is open.

AI USE STATEMENT

In this work we used generative AI tools for the following tasks with required disclosure: developing and refining the theoretical framework, formulating the mathematical claims and writing their proofs, generating the synthetic data used in the numerical check of Appendix A, designing experiments and giving feedback on them, implementing the experimental code, cleaning and reformatting datasets, interpreting results, and translating the author's notes into English. We did not use generative AI tools for qualitative or thematic data analysis, which is not applicable to this work. We also used generative AI tools for tasks with recommended disclosure: drafting and editing sections of the paper, suggesting experimental parameters, searching for and summarising related literature, formatting references, creating figures, editing code, and suggesting the structure of the paper and its keywords. The author reviewed all AI-assisted work, checking every reported number against the raw result files and every reference against its publisher's metadata. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All models and datasets used are public, with the exception of the industrial case study of Appendix B.3, which is illustrative and supports no claim. Appendix B.1 specifies every setting, including checkpoints, index construction, the validation protocol, the label-free partition and the tower training recipes. Proofs and the numerical check are in Appendix A; the check depends only on NumPy. Code for all experiments and the result tables behind every reported number are provided as supplementary material.

REFERENCES

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. MS MARCO: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. InPars: Unsupervised dataset generation for information retrieval. In *SIGIR*, 2022.

Hao Cheng, Hao Fang, Xiaodong Liu, and Jianfeng Gao. Task-aware specialization for efficient and robust dense retrieval for open-domain question answering. In *ACL*, 2023.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *SIGIR*, 2009.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *RecSys*, 2016.

Zhuyun Dai, Vincent Y 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. In *ICLR*, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Zhe Dong, Jianmo Ni, Dan Bikel, Enrique Alfonseca, Yuan Wang, Chen Qu, and Imed Zitouni. Exploring dual encoder architectures for question answering. In *EMNLP*, 2022.

Sebastian Hofstätter, Sophia Althammer, Michael Schröder, Mete Sertkan, and Allan Hanbury. Improving efficient neural ranking models with cross-architecture knowledge distillation. *arXiv preprint arXiv:2010.02666*, 2020.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2013.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*, 2022.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *EMNLP*, 2020.

Ekaterina Khramtsova, Shengyao Zhuang, Mahsa Baktashmotlagh, Xi Wang, and Guido Zuccon. Selecting which dense retriever to use for zero-shot search. In *SIGIR-AP*, 2023.

Hyunji Lee, Luca Soldaini, Arman Cohan, Minjoon Seo, and Kyle Lo. RouterRetriever: Routing over a mixture of expert embedding models. In *AAAI*, 2025.

Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. In *NeurIPS*, 2022.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Sashank Reddi, Rama Kumar Pasumarthi, Aditya Krishna Menon, Ankit Singh Rawat, Felix Yu, Seungyeon Kim, Andreas Veit, and Sanjiv Kumar. RankDistil: Knowledge distillation for ranking. In *AISTATS*, 2021.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *NeurIPS Datasets and Benchmarks*, 2021.

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

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *NAACL*, 2022.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multilingual E5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *CVPR*, 2022.

Ji Yang, Xinyang Yi, Derek Zhiyuan Cheng, Lichan Hong, Yang Li, Simon Xiaoming Wang, Taibai Xu, and Ed H. Chi. Mixed negative sampling for learning two-tower neural networks in recommendations. In *WWW Companion*, 2020.

Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H. Chi, Steve Tjoa, Jieqi Kang, and Evan Ettinger. Self-supervised learning for large-scale item recommendations. In *CIKM*, 2021.

Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chi. Sampling-bias-corrected neural modeling for large corpus item recommendations. In *RecSys*, 2019.

Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

A PROOFS

Lemma 1. The density-weighted error $(1-\varepsilon)\,\mathrm{tr}[(W-W_0)\Sigma_0(W-W_0)^\top] + \varepsilon\,\mathrm{tr}[(W-W_R)\Sigma_R(W-W_R)^\top]$ is minimised at $\widehat{W}A = (1-\varepsilon)W_0\Sigma_0 + \varepsilon W_R\Sigma_R$; subtracting W_RA gives (3). In the basis S, $A = S^{-\top}[(1-\varepsilon)I + \varepsilon\Lambda]S^{-1}$, so $\widehat{W} - W_R = KDS^\top$ with $K = (W_0 - W_R)S^{-\top}$ and $D = \mathrm{diag}(c_i)$, and $\mathcal{B}_R(\widehat{W}) = \mathrm{tr}(KD\Lambda DK^\top) = \sum_i \lambda_i c_i^2 \|k_i\|^2$. Each c_i decreases strictly in ε, and some $k_i \neq 0$ when $W_0 \neq W_R$. □

Proposition 1. By the lemma, on a region with $W_0 \neq W_R$ the trained configuration with larger coverage is strictly less biased. Under (a), ROUTE beats OWN on R and OWN beats ROUTE on R'; under (b), a trained configuration beats RAW on some region. If (a) fails, the coverage order of OWN and ROUTE, and hence their bias order, is the same on every region. If (b) fails, $\mathcal{B}_R(I)$ is either at least the smaller trained bias on every region or at most it on every region, so RAW is either never strictly least biased or weakly least biased everywhere. □

Numerical check. With non-commuting Σ_0, Σ_R, a least-squares fit on sampled points matches (3) to within 0.47%, and three regions make ROUTE, OWN and RAW respectively least biased. The unweighted norm $\|\widehat{W} - W_R\|_F$ grows with coverage in 334 of 2,000 random instances; $\mathcal{B}_R$ decreases in all of them.

B EXPERIMENTAL DETAILS AND INDUSTRIAL CASE STUDY

B.1 EXPERIMENTAL DETAILS

Text retrieval. OWN and ROUTE use the public DPR checkpoints trained on NQ (pooled output; DPR-multiset for the checkpoint comparison); RAW is `bert-base-uncased` with mean pooling, also in the single-tower comparison. Queries are truncated to 64 tokens and documents to 256. The index holds the first 50k documents of each corpus, and queries without a relevant document in it are dropped, leaving 12,732 (1,400 NQ, 5,928 SQuAD, 1,351 HotpotQA, 612 FIQA, 323 NFCorpus, 1,417 Quora, 300 SciFact, 1,401 ArguAna). Smaller indexes are subsampled with all relevant documents kept, five times per size.

Choosing and combining. Splits use seeds 42–61. *Pseudo-domains*: k-means on backbone embeddings of validation queries; clusters with fewer than 20 validation queries take the global choice; test queries go to the nearest centroid. *Per-query classifier*: logistic regression on eight label-free features (cosine of the query under the passage tower and the backbone, and under the two towers; top-1 score and top-1 minus top-2 margin of each configuration), trained on the per-query best configuration. *Coverage partition*: queries whose mean cosine to their ten nearest NQ queries exceeds the 25th percentile of validation NQ queries go to OWN, the rest to ROUTE if the passage tower moves them from the backbone by more than the validation median, and to RAW otherwise. The label budget (1%–50% of each dataset, 200 repeats) and the 500 significance redraws split each dataset separately; the cluster bootstrap resamples datasets and then test queries, 2,000 times. Combination z-scores each configuration's scores per query and searches weights on a simplex grid of step $1/4$; RRF uses $k = 60$.

Image–text retrieval. 20,000 products of *Fashion Product Images (Small)*[1], taken in dataset order with at most 800 per category (136 categories); category frequency is counted over the full dataset. 400 products are held out; the tail is the bottom quartile by category frequency. Towers are Linear(512,512)–ReLU–Linear(512,128), trained with InfoNCE (temperature 0.05, 128 positives and 256 random negatives per batch, Adam 10^{-3}, 30 epochs) on 30,000 pairs; retrieval is against the whole catalogue. Results are means over seeds 42–46, each redrawing the evaluation set, the training sample and the initialisation.

[1] `https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-small`

MS MARCO. Frozen `distilbert-base-uncased`, 8,000 passages, 40,000 training pairs, 800 evaluation queries, training pairs drawn with a Zipf law of exponent α over queries; towers Linear–ReLU–Linear (256 hidden, 128 output) trained as above with batches of 64 and 128 random negatives, 5 and 40 epochs. The fixation index is computed over 3,000 random pairs of evaluation queries. Fine-tuning trains both full encoders (AdamW at 2×10^{-5}, batches of 32 with 64 random negatives) for 2 and 8 epochs.

B.2 FULL RESULTS

Table 3: Text retrieval: choosing and combining (Figure 3). Mean over 20 validation/test splits with the range of the middle 95% of splits for the macro average.

Strategy	Supervision	micro	macro	macro range
Always OWN	none	39.92	26.95	[26.47, 27.50]
Always ROUTE	none	37.96	26.76	[26.30, 27.21]
Always RAW	none	41.38	25.92	[25.57, 26.48]
One global choice	labels	41.38	25.92	[25.57, 26.48]
Coverage partition	none	42.41	29.96	[29.39, 30.42]
Per-query classifier	labels	43.94	31.38	[30.79, 32.16]
Pseudo-domains, $k = 8$	labels per cluster	45.67	32.36	[31.30, 33.24]
Pseudo-domains, $k = 32$	labels per cluster	46.52	33.41	[32.75, 33.97]
Choose per domain	tags + labels	47.75	35.36	[34.89, 35.93]
Equal-weight score combination	none	49.74	34.38	[33.69, 34.91]
RRF	none	47.20	32.16	[31.63, 32.67]
Weighted RRF per domain	tags + labels	50.77	36.57	[36.00, 37.32]
Weighted score combination per domain	tags + labels	52.80	37.79	[37.29, 38.39]
Per-query oracle	per-query labels	59.94	43.43	[42.96, 43.93]

B.3 INDUSTRIAL CASE STUDY

Table 4 gives top-20 recall for eleven queries, selected for illustration and judged by one annotator, in a production recommendation system whose training positives concentrate on popular items. On nine infrequent queries that returned identical results under OWN, recall rises from $1/180$ to $135/180$ under ROUTE; for example, "Do children need to learn programming?" returned short-video production courses under OWN and children's programming courses under ROUTE.

Table 4: Top-20 recall in a production recommendation system (queries translated). Tail: infrequent query with identical top results across distinct queries under OWN; Head: frequent query.

Query	OWN	ROUTE	Type
Pastry training school	19/20	20/20	Head
English language training	18/20	20/20	Head
Baking training center	1/20	20/20	Tail
Breakfast cooking training	0/20	3/20	Tail
E-commerce training	0/20	19/20	Tail
Bookkeeping training	0/20	20/20	Tail
Hamburger culinary school	0/20	20/20	Tail
Maya 3D training	0/20	7/20	Tail
Chef training	0/20	19/20	Tail
Floral design training	0/20	18/20	Tail
Hazardous operations certificate	0/20	9/20	Tail