ADAPTIVE STRUCTURED TRANSFORMATION: MITIGATING DISTRIBUTION SHIFT IN DENSE RETRIEVAL THROUGH TRAINING-TIME PREPROCESSING

Anonymous authors
Paper under double-blind review

ABSTRACT

Dense retrieval models are trained assuming that finetuning on task-relevant queries improves performance, yet this assumption can break down when training data contains synthetic components or originates from a misaligned distribution with target tasks. We find that in such scenarios, naively finetuning on seemingly relevant data can result in *negative transfer*, causing significant degradations over not finetuning at all. We propose Adaptive Structured Transformation (AStrucT), an automatic preprocessing technique that leverages off-the-shelf Large Language Models (LLMs) to organize training documents into domain-specific structures prior to finetuning. These domain-specific schemas are induced automatically from a small sample of target-domain passages, without manual schema design or access to test-time queries. Across three model scales and twelve diverse domains (BRIGHT), AStrucT yields an average improvement of 3.75 percentage points (pp) nDCG@10 over direct finetuning, and 1.25 pp over the pretrained baseline, mitigating negative transfer. For the 4B model, domain-matched templates outperform general rewriting and templates from other domains on average. These findings provide a practical strategy for adapting embedding models using automatically prepared training documents, without requiring LLM calls at inference.

1 INTRO

Dense retrieval models (Karpukhin et al., 2020; Izacard et al., 2021) represent queries and documents as vectors in a shared embedding space, enabling semantic search via nearest-neighbor retrieval. In practice, dense retrievers are often finetuned to new domains without access to real user queries, and sometimes without the ability to manually inspect the underlying documents due to privacy or access constraints. A common workaround in such settings is to generate synthetic queries conditioned on individual documents and treat each document as a gold-labeled positive for its generated queries (Shao et al., 2025). This approach implicitly assumes that the synthetic query generation process is aligned with the distribution of the downstream test queries. When this alignment assumption fails, naive finetuning can degrade retrieval performance, in some cases underperforming models that have never seen the additional training data.

This failure mode is closely related to *negative transfer*, a well-known risk in transfer learning where adaptation signals harm downstream performance when task similarity is assumed rather than verified (Rosenstein et al., 2005; Zhang et al., 2023). In practice, this leaves practitioners with a difficult decision: whether to use available training data as-is, preprocess it, or avoid it entirely. This dilemma is especially acute when data aligned with the downstream test distribution is unavailable during training, a common constraint in proprietary enterprise settings, rapidly evolving domains, or resource-constrained deployments where preliminary evaluation runs are prohibitively expensive. Detecting distribution shift typically requires evaluating performance on target test queries, allowing practitioners to observe degradation and adjust their adaptation strategy accordingly; without access to such target queries, practitioners cannot rely on this reactive approach.

In this work, we investigate this challenge in the context of dense retrieval model training, where LM-generated queries and negative passages are standard practices (Ma et al., 2021; Cho et al., 2022; Wu & Cao, 2024; Kachuee et al., 2025). While effective in many cases, these synthetic

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.

supervision signals can introduce opaque distribution shifts. In our experiments, such misalignment frequently results in substantial negative transfer, where naive finetuning degrades performance relative to models that receive no task-specific training at all. For example, on LeetCode with the Qwen3-Embedding 4B model, naive finetuning underperforms the pretrained model by 14.0 pp in nDCG@10. This negative transfer raises a central question of "how should we prepare training data to maximize robustness against unknown distributional shifts?"

Our contribution is a document-side training intervention: Adaptive Structured Transformation (AStrucT) develops a domain-specific transformation template, then uses it to rewrite training documents while keeping the supervision fixed. From five example passages, the `GENERATE` stage induces an output schema and transformation guidelines. The `TRANSFORM` stage reuses this template to generate structured text for *both positive and negative documents*, independently of their associated queries, while the original queries and positive/negative relevance assignments remain unchanged.

The motivation is to make domain-relevant concepts, relations, and examples explicit in the documents that supply the training signal. At inference, the trained retriever still indexes and retrieves *original raw documents*: neither an LLM call nor a transformed retrieval corpus is required. We evaluate this intervention across three Qwen3-Embedding scales (0.6B, 4B, 8B; Zhang et al. (2025b)) and twelve BRIGHT domains (Su et al., 2025). Across these scales, AStrucT improves average BRIGHT retrieval over direct HQ finetuning. Subsequent analysis finds that domain-matched transformation outperforms, on average, both general rewriting with the same LLM and templates induced for other domains.

This intervention differs from generating queries or new relevance labels (Bonifacio et al., 2022; Dai et al., 2023; Wang et al., 2022), and from rewriting the indexed corpus or maintaining generated context at inference (Nogueira et al., 2019; Weller et al., 2024; Suzgun et al., 2025). AStrucT does generate new text, but uses it as a replacement document view for training existing query-document assignments, not as an inference representation. This offers both substantial cost benefits for large datasets and an alternate path for addressing negative transfer.

AStrucT provides a strategy for practitioners when training data provenance is uncertain or test-set validation is infeasible. In our experiments, it improves average retrieval over both direct finetuning and the pretrained model at all three model scales. The approach requires no manual inspection of target documents and no test-set queries. Its LLM preprocessing cost is incurred once: the transformed training documents can be reused across training runs (Section 3.4), while inference uses raw documents without additional LLM calls.

2 Method: Adaptive Transformation

We propose Adaptive Structured Transformation (AStrucT, Figure 1), a technique that leverages off-the-shelf LLMs to automatically create domain-specific transformations. Our approach eliminates the need for extensive domain-specific knowledge or iterative test-set validation, requiring only a few document examples to generate a robust structural schema. Unlike adaptation frameworks like GPL (Wang et al., 2022) that depend on a full unlabeled target corpus for task generation, our method operates as a preventive preprocessing step during training data preparation. We summarize our method in two stages: `GENERATE` and `TRANSFORM`.

2.1 Step 1: Template Generation (`GENERATE`)

The first stage generates a domain-specific schema (S_d) and usage guidelines (G_d) by prompting a strong LLM to analyze sample passages from the target domain. Given a domain d and a small set of k sample passages $p_1, p_2, \ldots, p_k$ ($k = 5$ in our experiments), we use the following procedure: $\texttt{GENERATE}(d, p_1, \ldots, p_k) \rightarrow (S_d, G_d)$.

The resulting structured schema S_d is tailored to the domain and consists of: (1) **field definitions**: a set of attribute names and their expected types (e.g., strings, lists, nested objects), and (2) **field descriptions**: natural language descriptions of what information should be captured in each field.

Reusable transformation template. The prompt requests a schema with three to five top-level fields, including a summary, followed by an applied example and brief guidelines for consistent

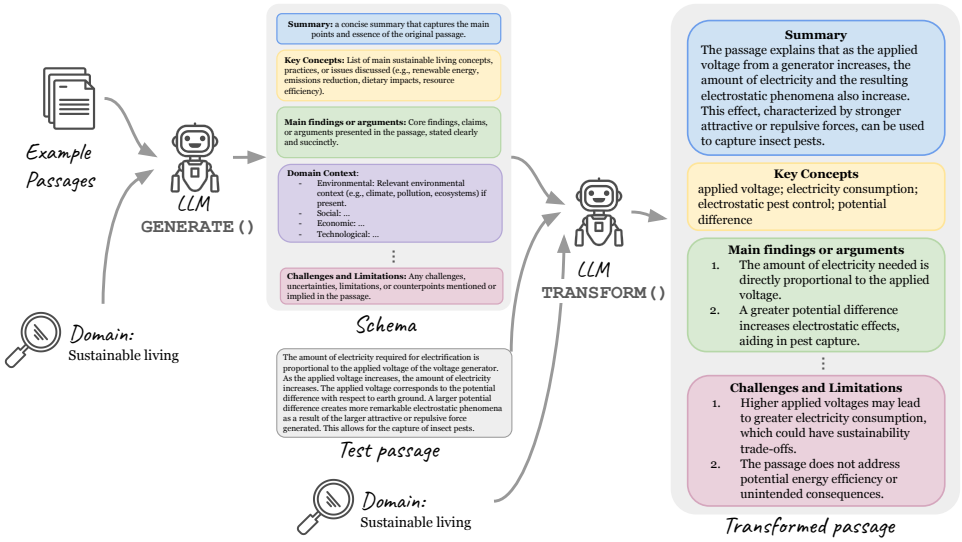


Figure 1: Adaptive Structured Transformation is a two-stage framework for generating structured representations of training documents. `GENERATE` uses sample passages to create a domain-specific schema and usage guidelines. `TRANSFORM` then applies this template to each training document.

use and edge cases. We combine the schema S_d and guidelines G_d with the instruction "You are a {domain} expert" to form the domain-specific transformation template. The same template is reused across documents in that domain. Full prompts are provided in Appendix B.

By utilizing only a handful of examples, this stage avoids the corpus-level dependency, which is common in adaptation methods that require processing entire document collections to align distributions (Ganin et al., 2016; Gururangan et al., 2020). The generation prompt instructs the LLM to act as a domain expert and identify common information types, key concepts, and structural patterns that may be unique to that domain (e.g. "citations" for scientific articles).

2.2 Step 2: Document Transformation (`TRANSFORM`)

The `TRANSFORM` stage applies the domain's template to each document p in the training corpus: $\texttt{TRANSFORM}(d, S_d, G_d, p) \rightarrow \hat{p}$, where $\hat{p}$ is a structured JSON representation of p following schema S_d and guidelines G_d. Both positive and negative documents are transformed independently, without supplying their associated queries or relevance assignments to the LLM.

We aggregate each document p_i in domain d to create a new document set $\hat{P} = \{\hat{p_1}, ..., \hat{p_n}\}$ as the training documents in our experiments. These representations are serialized as text for retriever training, with queries and relevance assignments unchanged. At inference, the retriever encodes raw queries and documents without further LLM calls.

3 Experiments

In our experiments, we both 1) study the extent of *negative transfer* via finetuning over an in-domain but misaligned dataset; and 2) study the effectiveness of AStrucT as a mitigation strategy. We use a previously-released dataset for the former and AStrucT for our *training-time* mitigation. For our main experiments, at test-time, we use the original raw text to build the dense index.

3.1 Datasets

We train our models using the Hard Query (HQ) subset of the ReasonIR dataset (Shao et al., 2025), which utilizes documents from the BRIGHT (Su et al., 2025) collection but relies on LLM-generated

synthetic queries and negative documents.[1] This allows us to simulate an applied workflow where the practitioner attempts to optimize models for their own document corpora using synthetic data generation, inadvertently introducing a synthetic-to-real distribution shift. In BRIGHT, TheoremQA Questions and AoPS share an identical document collection; consequently, the ReasonIR HQ contains a single unified query set for both domains. To avoid data duplication, we conduct training exclusively on these queries using the shared document collection. For evaluation, we employ the provided test queries and documents from BRIGHT for each domain. Dataset statistics are in Appendix A.

3.2 Prompts

We develop both prompts (Appendix B) using a subset of the legal retrieval dataset CLERC (Hou et al., 2025). The `GENERATE` prompt retains a legal passage/transformation example from this development source. Although CLERC is separate from BRIGHT, this use is for prompt development rather than an independent benchmark evaluation.

3.3 Models

We finetune Qwen3-Embedding (Zhang et al., 2025b) dense retriever models (0.6B, 4B, and 8B). All finetuning experiments use 8xA100 GPUs. In our work, "No FT" is the baseline representing the off-the-shelf Qwen3-Embedding model.

We use GPT-4.1 via OpenAI's API[2] with structured outputs to ensure JSON outputs. The models are trained following the GritLM (Muennighoff et al., 2025) code, using the default Qwen-3 Embedding model configurations. We use an effective global batch size of 192 for all model sizes, with one hard negative per query and treating remaining (cross-device) positive documents as in-batch negatives. For the larger 8B model only, we use a batch size of 96 with a gradient accumulation step size of 2 to achieve the same effective batch, although the in-batch negatives are impacted.

During training, documents and queries are truncated to 2048 tokens, with full documents available at test time. We use a learning rate of 6e-6 with a warmup ratio of 0.06. We train for up to 700 steps, evaluating every 10 steps and using early stopping with a patience of 5. We partition the ReasonIR-HQ examples into 80% training, 10% development, and 10% held-out splits. For AStrucT, we apply document transformations across all three splits, preserving the queries, relevance assignments, and split membership. The development split is used for checkpoint selection; the held-out split is not used in the experiments; and no BRIGHT evaluation query is used for training or checkpoint selection. In practice, most runs ended within a couple of hundred steps, ranging from 0.2 to 6 hours depending on the model size.

3.4 Preprocessing Cost

AStrucT transforms the positive and negative documents associated with the training examples, rather than the entire BRIGHT retrieval corpus. The transformed documents are prepared once and reused across retriever training runs and model scales. Inference continues to use raw queries and documents without additional LLM calls.

The preprocessing pool contains approximately 196.3k distinct passages across these three ReasonIR-HQ splits. Processing these passages with GPT-4.1 uses about 236.5M input tokens and 116.5M output tokens. At batch rates of $1 per million input tokens and $4 per million output tokens, this corresponds to an estimated document-transformation cost of $702.50, or about $0.0036 per document. The test-time experiments in Section 5.5 additionally transform the retrieval corpus. That preparation is separate from the cost reported above and is not required by our method.

4 Results

In Table 1, we show the retrieval quality of the models measured by nDCG@10 following prior work, especially on BRIGHT (Thakur et al., 2021; Shao et al., 2025; Su et al., 2025).

[1] BRIGHT is a test-only benchmark with no official training split; Shao et al. (2025) generated synthetic training data to enable model fine-tuning on BRIGHT's document collection.

[2] https://openai.com/api/

Table 1: The nDCG@10 performance comparison across models and domains. "HQ FT" denotes the pretrained model fine-tuned on the original ReasonIR HQ data. "No FT" denotes the pretrained model without finetuning. Macro averages give equal weight to all twelve evaluations; averages and derived comparisons are computed from the per-domain scores before final rounding. "Earth sci." refers to the Earth Science domain, and "Sustainable liv." refers to the Sustainable living domain. TQA refers to TheoremQA (Chen et al., 2023).

Domain	Qwen3-0.6B			Qwen3-4B			Qwen3-8B		
	HQ FT	AStrucT	No FT	HQ FT	AStrucT	No FT	HQ FT	AStrucT	No FT
Aops	8.83	**10.86**	10.48	10.34	**12.77**	11.93	9.46	**11.70**	9.06
Biology	12.63	**14.72**	11.50	20.20	**22.53**	16.38	13.42	**18.19**	16.40
Earth sci.	16.74	**28.05**	27.89	24.07	35.27	**35.31**	22.91	**32.26**	32.24
Economics	13.21	14.41	**15.70**	14.01	**16.77**	14.39	12.93	15.34	**16.05**
Leetcode	12.76	**24.43**	24.27	23.09	34.73	**37.11**	24.00	**37.79**	33.49
Pony	**1.26**	0.42	0.80	1.10	0.73	**1.41**	0.63	**1.02**	0.82
Psychology	14.01	**16.43**	13.19	16.00	**21.49**	20.29	18.47	**23.52**	22.62
Robotics	11.82	**14.01**	10.52	12.43	**13.89**	11.26	12.84	**14.60**	12.61
Stackoverflow	**13.71**	13.48	11.56	17.77	**19.41**	14.95	**18.17**	17.89	17.54
Sustainable liv.	8.60	**10.34**	9.98	7.89	**12.18**	10.85	9.30	**13.35**	12.43
TQA theorems	28.03	**29.96**	25.82	32.58	33.93	**40.45**	34.74	**36.53**	36.27
TQA questions	32.60	**33.75**	29.77	36.63	**39.99**	37.03	36.75	**42.11**	41.33
Average	14.52	**17.57**	15.96	18.01	**21.97**	20.95	17.80	**22.03**	20.91

We observe a predominant but non-universal pattern of *negative transfer* when directly finetuning on the original ReasonIR HQ dataset. On average, the HQ-finetuned models underperform the pretrained baseline by 1.44, 2.94, and 3.10 pp for the 0.6B, 4B, and 8B models, respectively, despite access to in-domain documents. This degradation is most pronounced in technical domains, with LeetCode exhibiting drops of 9.49 to 14.02 pp across model scales. However, this effect is not universal: HQ FT outperforms No FT in 13 of 36 model–domain settings, including StackOverflow and Robotics at all three scales. So, while synthetic domain-specific finetuning can occasionally benefit retrieval, across multiple models and domains, it harms performance on average.

Taken together, these results provide the motivation for our central question. While synthetic domain-specific data can be beneficial under some conditions, its effect is highly model- and domain-dependent, and difficult to anticipate without knowledge of the model's pretraining distribution. As pretrained embedding models further improve—particularly in domains such as code—naive finetuning on misaligned synthetic supervision can not only fail to help, but actively harm performance.

AStrucT helps mitigate this risk. By restructuring training documents into domain-focused schemas prior to finetuning, our approach achieves the best macro-average performance at all three model scales, with relative improvements of 21.0%, 22.0%, and 23.7% over direct finetuning for the 0.6B, 4B, and 8B scales, respectively, and 10.1%, 4.9%, and 5.4% over the pretrained baseline. While no single method dominates on every domain and model, AStrucT achieves higher average retrieval performance in the evaluated settings compared to both no finetuning and HQ-finetuning.

5 Analysis

We examine alternate methods for transforming documents at training time to study the necessity of domain-guided templates and whether there are still benefits to training on original (misaligned) documents. We also analyze the importance of fields within the rewrite schema and the geometry of the embeddings. Finally, we explore the use of AStrucT as a *test-time* application.

5.1 GPT-4.1 Document-Rewriting Controls

We compare AStrucT with two GPT-4.1 document-rewriting baselines for Qwen3-Embedding-4B. *General rewriting* prompts GPT-4.1 to rewrite each raw passage faithfully as plain text, preserving facts, qualifications, formulas, and code at approximately the source length. A single prompt is used across domains, without queries or domain-specific templates. *Wrong-domain transformation* applies

Table 2: GPT-4.1 document-rewriting controls for Qwen3-Embedding-4B, nDCG@10 on the percentage scale. Average is the unweighted mean across the twelve evaluations. The AStrucT average is recomputed from the rounded per-domain scores in Table 1.

Domain	AStrucT	General rewrite	Wrong-domain template
AoPS	12.77	11.24	10.66
Biology	22.53	13.81	16.61
Earth Science	35.27	22.79	27.90
Economics	16.77	12.95	15.37
LeetCode	34.73	28.40	28.15
Pony	0.73	0.95	0.44
Psychology	21.49	16.41	21.67
Robotics	13.89	12.02	12.98
StackOverflow	19.41	16.65	18.76
Sustainable Living	12.18	8.37	11.16
TQA Theorems	33.93	35.91	36.17
TQA Questions	39.99	38.92	40.14
Average	21.97	18.20	20.00

a template induced for another domain, changing the domain-expert instruction, usage guidelines, and schema together while keeping the source passage fixed.[3]

Both baselines use the same 196,348 passages and 100,504 query–document assignment records.[4] We retain the raw passage when generation fails, affecting 155 documents for general rewriting and 115 for wrong-domain transformation.

To keep the corpus size comparable for the general rewrite, the prompt for general rewriting requests approximately 80–120% of the source length, although all valid outputs are retained. This results in a corpus containing 89.6% as many tokens as the source corpus.

Across all twelve evaluations, AStrucT achieves an average of 21.97 nDCG@10, compared with 18.20 for general rewriting and 20.00 for wrong-domain transformation, yielding advantages of 3.77 and 1.97 pp, respectively. Both general rewriting and wrong-domain transformation also underperform the pretrained model on average (20.95), and general rewriting recovers only 0.19 pp of the 2.94 pp lost by HQ FT. Using the same GPT-4.1 model for general rewriting or applying a structured template from another domain does not reproduce the average retrieval gains of AStrucT.

The advantage of domain matching is not uniform, however. Wrong-domain transformation outperforms AStrucT on Psychology, TQA Questions and TQA Theorems, showing that a template induced for another domain can still be effective. This could reflect overlap between domain-specific schemas: shared fields such as `core_concepts` may capture useful information across domains. However, wrong-domain transformation performs substantially worse than AStrucT in several other domains, suggesting that simply organizing information as JSON is insufficient to reproduce the gains of domain-matched transformation.

5.2 Data Mixing Ablation Study

We examine whether retaining the original document alongside its transformation provides additional retrieval benefits. Table 3 compares using the transformed text alone with concatenating the two representations (*prepend*) or retaining them separately (*mix*). *Transform-only* achieves the highest average nDCG@10 at all three model scales, suggesting that combining representations offers no consistent advantage over the transformed view alone. Nevertheless, the domain-level results reveal complementary benefits: at 4B, *prepend* improves Psychology from 21.49 to 25.56, while *mix*

[3]Each domain borrows the next domain's template in one of two fixed cycles: (Biology, Economics, LeetCode, TQA Theorems) and (Earth Science, Psychology, Robotics, StackOverflow, Pony, Sustainable Living, AoPS). The last domain in each cycle borrows from the first.

[4]We retain 100,504 of the 100,521 ReasonIR-HQ examples after excluding 17 whose positive or negative documents lack valid transformation outputs.

improves TheoremQA Theorems from 33.93 to 37.90. These results suggest that the usefulness of retaining the original text depends on the domain.

Table 3: Data-mixing ablation (nDCG@10, percentage scale). $\cup$ denotes *mix*, $\oplus$ denotes *prepend*, and $\rightarrow$ denotes *transform-only*. Average gives equal weight to the twelve evaluations. Bold marks the best score within each model scale.

Domain	Qwen3-0.6B			Qwen3-4B			Qwen3-8B		
	$\cup$	$\oplus$	$\rightarrow$	$\cup$	$\oplus$	$\rightarrow$	$\cup$	$\oplus$	$\rightarrow$
AoPS	9.64	7.90	**10.86**	11.21	12.24	**12.77**	10.42	11.23	**11.70**
Biology	12.49	**16.36**	14.72	17.86	20.76	**22.53**	15.35	16.66	**18.19**
Earth sci.	27.56	20.44	**28.05**	35.21	27.35	**35.27**	31.85	27.78	**32.26**
Economics	12.50	**14.94**	14.41	13.66	16.34	**16.77**	11.29	14.99	**15.34**
LeetCode	20.88	22.59	**24.43**	31.43	31.03	**34.73**	30.55	32.75	**37.79**
Pony	0.60	**0.79**	0.42	0.72	**0.75**	0.73	0.87	**1.03**	1.02
Psychology	13.48	**17.99**	16.43	14.90	**25.56**	21.49	12.91	19.22	**23.52**
Robotics	10.55	13.65	**14.01**	**14.78**	13.63	13.89	13.39	14.01	**14.60**
StackOverflow	12.41	12.52	**13.48**	**19.51**	19.32	19.41	**18.97**	18.38	17.89
Sustainable liv.	8.81	9.10	**10.34**	8.43	11.53	**12.18**	9.87	12.26	**13.35**
TQA theorems	**31.12**	30.42	29.96	**37.90**	33.25	33.93	36.47	35.35	**36.53**
TQA questions	32.72	29.82	**33.75**	**41.05**	38.18	39.99	40.25	**42.24**	42.11
Average	16.06	16.38	**17.57**	20.56	20.83	**21.97**	19.35	20.49	**22.03**

5.3 Query-Conditioned Field Retention

To examine the content of the generated representations, we use GPT-4.1 to perform a query-conditioned filtering (`FILTER`) to retain fields judged relevant to each document's associated training query. This stage preserves the schema while removing irrelevant field content. We then measure which fields remain populated. Appendix D reports the full domain-level table alongside retrieval gains and filtering error rates.

Across 160,824 query-associated entries, overview fields such as `summary` and `key_concepts` have high retention rates ($\geq 90\%$ non-empty). Domain-specific auxiliary fields are more frequently pruned: in LeetCode, `examples` is empty in 90.7% of entries, while in AoPS, `final_answer` is removed in 66.9% of cases. Conceptual fields such as `key_concepts`, `domain_concepts`, and `core_concepts` exceed 95% retention in 9 out of 10 domains where they are present.

These findings suggest summaries and general concepts are more consistently deemed relevant than query-specific details. This matches intuition that, for a reasoning-oriented dataset like BRIGHT, general reasoning principles are worth extracting from a reference rather than surface-level similarities or solutions to different problems.

5.4 Embedding-Space Changes

We examine query–document alignment and movement from pretrained representations over Qwen3-4B in Appendix E. On average, HQ finetuning decreases query–positive cosine similarity by 0.071 and widens the negative–positive similarity margin by 0.010. With AStrucT, the corresponding changes are smaller: a 0.060 decrease in positive similarity and a 0.007 increase in the margin.

We find that across domains, larger reductions in movement are associated with greater retrieval gains. Furthermore, we find less embedding movement in AStrucT than in HQ FT, which suggests that one mechanism for the *negative transfer* in HQ FT could be due to overfitting to misaligned synthetic supervision. These analyses are observational; we are not proposing a general causal mechanism, especially since fine-tuning should cause some shifts in embedding space.

Table 4: Macro-average change in nDCG@10 (percentage points) when retrieving transformed rather than raw test documents, with equal weight across twelve BRIGHT evaluations. HQ FT denotes finetuning on the original HQ data; No FT denotes the pretrained retriever. Positive values indicate improvement.

Model	HQ FT	No FT
Qwen3-0.6B	-0.86	-0.49
Qwen3-4B	+3.94	+1.67
Qwen3-8B	-0.08	+1.09

5.5 Test-time Transformation with AStrucT

Following prior work around document rewriting (Nogueira et al., 2019; Weller et al., 2024) and test-time memory (Suzgun et al., 2025), we investigate whether AStrucT offers benefits as a test-time transformation strategy. Note that from a cost perspective, test-time AStrucT is expensive and often impractical, as it requires transforming the full retrieval corpus.

Transformation at test time only. Table 4 summarizes the macro-average effects (full table in Appendix C, Table 6). While test-time transformation benefits both 4B models, its effects at the other scales are mixed. Relatedly, Weller et al. (2024) find that generative query and document expansion tends to benefit weaker retrievers while harming stronger ones. Our results likewise show model-dependent benefits, although the gains do not increase with parameter count. In contrast, training-time AStrucT improves macro-average retrieval over HQ FT at all three scales while retaining raw documents at inference, showing that its training benefits do not require replacing the retrieval corpus with transformed representations.

Transformation at both training and test time. Training and testing simultaneously on transformed data yields a lower performance of 19.91 nDCG@10 for the 4B model, underperforming our main approach by 2.06 pp. We attribute this effect in part to properties of the BRIGHT document collection, where a substantial fraction of documents are extremely short (often fewer than 100 tokens), consisting primarily of titles, URLs, or section headers. When the entire corpus is transformed at inference time, these information-poor documents are synthetically expanded into structured representations, making previously ignorable items appear semantically plausible and increasing their competitiveness in the final ranked list.

During training, positives are BRIGHT seed documents filtered by a FineWeb-Edu classifier (Penedo et al., 2024; Shao et al., 2025) and hard negatives are LLM-generated passages, so transformed training documents are predominantly informative, allowing the model to learn discriminative structural signals. At inference time, this mismatch reduces the effective contrast between truly relevant and irrelevant documents, ultimately degrading retrieval performance. These results suggest that AStrucT is most effective as a training-time inductive bias, and that arbitrary applications of synthetic transformations at test time can introduce noise that outweighs their structural benefits. This may also explain the earlier observation of the mixed effects of test-time transformation with the original (HQ FT) data or no finetuning at all.

6 Related Work

6.1 Dense Retrieval and Enhancements for Retrieval

Dense retrieval models (e.g., Karpukhin et al. (2020); Khattab & Zaharia (2020)) encode queries and documents into shared representation spaces, enabling semantic search through either nearest neighbor search or approximate nearest neighbor search (Xiong et al., 2021). Training these models requires high-quality query-document pairs, typically relying on contrastive learning objectives (Izacard et al., 2021).

Traditional document expansion methods, such as Doc2Query (Nogueira et al., 2019), enhance retrieval by appending predicted queries to documents at index-time. More recent strategies (Bonifacio et al., 2022; Dai et al., 2023; Shao et al., 2025) use LLMs to generate training queries or pairs. However, these approaches focus on content generation, implicitly relying on LLM-generated

queries that will align with the target distribution. In contrast, AStrucT generates domain-specific structured views of existing training documents before finetuning, while leaving queries and relevance assignments unchanged. This preprocessing is designed to mitigate negative transfer without requiring document transformation at inference time.

Beyond document expansion, query expansion is a complementary line of techniques to enhance information retrieval quality, from early techniques such as expanding query terms using general dictionaries (Voorhees, 1993; Richardson & Smeaton, 1995; Smeaton et al., 1995; Liu et al., 2004) to modern model-driven strategies such as HyDE (Gao et al., 2022). More recent frameworks introduce multi-stage decomposition (e.g., AMD (Seo et al., 2025)) to produce diverse expansions. In parallel, inference-time augmentation methods such as Dynamic Cheatsheet (Suzgun et al., 2025) and Agentic Context Engineering (Zhang et al., 2025a) maintain evolving memories, while adaptive retrieval frameworks like FLARE (Jiang et al., 2023) and Self-RAG (Asai et al., 2023) dynamically decide whether, when, and how to retrieve external evidence during generation. These approaches operate orthogonally to our training-time document transformation focus.

6.2 Distribution Shift and Domain Adaptation

Dense retrieval models optimized for large-scale datasets often exhibit severe performance degradation when applied to out-of-domain tasks (Thakur et al., 2021). Standard adaptation techniques like GPL (Wang et al., 2022) generate synthetic queries, mine negatives from the target corpus, and score pairs with a cross-encoder to provide pseudo-labels. Beyond retrieval, continued pretraining on domain- and task-specific text has been shown to improve downstream language understanding performance (Gururangan et al., 2020). Pipelines like Shao et al. (2025) instead synthesize reasoning-intensive "hard queries" directly from documents to train retrievers for reasoning-based search. AStrucT complements query generation by transforming the documents used for finetuning while retaining the existing queries and relevance assignments.

Our empirical results reveal that even with access to in-distribution documents—as seen in the ReasonIR HQ dataset derived from BRIGHT documents—synthetic signals can still introduce opaque distribution shifts. This results in negative transfer, where direct fine-tuning causes the model to underperform vanilla pretrained model baselines. We demonstrate that document-level access is insufficient if the synthetic task distribution is misaligned with the target reasoning requirements.

7 Conclusions

We find that *negative transfer* can occur in misaligned in-domain finetuning data for retrieval across three Qwen3-Embedding model sizes and multiple domains in BRIGHT, a retrieval benchmark for reasoning.

We introduce Adaptive Structured Transformation (AStrucT), a document-side method for retrieval adaptation. A domain-specific transformation template derived from sample passages guides query-independent generation of both positive and negative training documents while preserving queries and relevance assignments. AStrucT addresses the issue of negative transfer by improving, on average, BRIGHT retrieval over both no finetuning and finetuning on misaligned documents. Further analysis supports the need for a domain-specific structured rewrite rather than a general rephrasing. This suggests that AStrucT is a viable approach for improving the robustness of retriever finetuning when training data may be misaligned with the target distribution.

Meanwhile, we view the exploration of further combinations of train-time transformations like AStrucT and test-time strategies like cheatsheets or agentic memory as a promising future direction, as we find only limited improvements applying AStrucT naively at test-time. Future studies to further mitigate negative transfer can consider a combination of retriever families and sizes, benchmarks, query-side rewriting, and training algorithms, like distillation.

AI use statement

We used GPT-4.1 to generate domain-specific transformation templates, synthetic training-document representations, and query-conditioned filtering described in this paper. We also used generative AI

tools to assist with experimental design and result interpretation, to draft and revise the manuscript, and to create and edit research software. The authors take responsibility for the final content of this work, including AI-assisted text, code, data, analyses, and claims.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*, 2023. URL https://arxiv.org/abs/2310.11511.

Luiz Bonifacio, Hugo Abonizio, Marzieh Fadaee, and Rodrigo Nogueira. Inpars: Data augmentation for information retrieval using large language models, 2022. URL https://arxiv.org/abs/2202.05144.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset, 2023. URL https://arxiv.org/abs/2305.12524.

Sukmin Cho, Soyeong Jeong, Wonsuk Yang, and Jong Park. Query generation with external knowledge for dense retrieval. In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2022. URL https://aclanthology.org/2022.deelio-1.3/.

Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=gmL46YMpu2J.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. URL http://jmlr.org/papers/v17/15-239.html.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022. URL https://arxiv.org/abs/2212.10496.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://aclanthology.org/2020.acl-main.740/.

Abe Bohan Hou, Orion Weller, Guanghui Qin, Eugene Yang, Dawn Lawrie, Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. CLERC: A dataset for U. S. legal case retrieval and retrieval-augmented analysis generation. In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. URL https://aclanthology.org/2025.findings-naacl.441/.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2021. URL https://arxiv.org/abs/2112.09118.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.495/.

Mohammad Kachuee, Sarthak Ahuja, Vaibhav Kumar, Puyang Xu, and Xiaohu Liu. Improving tool retrieval by leveraging large language models for query generation. In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, 2025. URL https://aclanthology.org/2025.coling-industry.3/.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020. URL https://aclanthology.org/2020.emnlp-main.550/.

Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert, 2020. URL https://arxiv.org/abs/2004.12832.

Shuang Liu, Fang Liu, Clement Yu, and Weiyi Meng. An effective approach to document retrieval via utilizing wordnet and recognizing phrases. SIGIR '04, 2004. URL https://doi.org/10.1145/1008992.1009039.

Ji Ma, Ivan Korotkov, Yinfei Yang, Keith Hall, and Ryan McDonald. Zero-shot neural passage retrieval via domain-targeted synthetic question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021. URL https://aclanthology.org/2021.eacl-main.92/.

Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=BC4lIvfSzv.

Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction, 2019. URL https://arxiv.org/abs/1904.08375.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849. Curran Associates, Inc., 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/370df50ccfdf8bde18f8f9c2d9151bda-Paper-Datasets_and_Benchmarks_Track.pdf.

Raymond Richardson and Alan F. Smeaton. Using wordnet in a knowledge-based approach to information retrieval. 1995. URL https://api.semanticscholar.org/CorpusID:2904992.

Michael T. Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G. Dietterich. To transfer or not to transfer. In *Neural Information Processing Systems*, 2005. URL https://api.semanticscholar.org/CorpusID:597779.

Wonduk Seo, Hyunjin An, and Seunghyun Lee. A new query expansion approach via agent-mediated dialogic inquiry, 2025. URL https://arxiv.org/abs/2502.08557.

Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2504.20595.

Alan F. Smeaton, Fergus Kelledy, and Ruairi O'Donnell. TREC-4 experiments at dublin city university: Thresholding posting lists, query expansion with wordnet and POS tagging of spanish. In *Proceedings of The Fourth Text REtrieval Conference, TREC 1995, Gaithersburg, Maryland, USA, November 1-3, 1995*, 1995. URL http://trec.nist.gov/pubs/trec4/papers/dublin.ps.gz.

Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Liu Haisu, Quan Shi, Zachary S Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O Arik, Danqi Chen, and Tao Yu. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ykuc5q381b.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. 2025. URL https://arxiv.org/abs/2504.07952.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

Ellen M. Voorhees. Using wordnet to disambiguate word senses for text retrieval. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '93, 1993. URL https://doi.org/10.1145/160688.160715.

Kexin Wang, Nandan Thakur, Nils Reimers, and Iryna Gurevych. GPL: Generative pseudo labeling for unsupervised domain adaptation of dense retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022. URL https://aclanthology.org/2022.naacl-main.168/.

Orion Weller, Kyle Lo, David Wadden, Dawn Lawrie, Benjamin Van Durme, Arman Cohan, and Luca Soldaini. When do generative query and document expansions fail? a comprehensive study across methods, retrievers, and datasets. In *Findings of the Association for Computational Linguistics: EACL 2024*, 2024. URL https://aclanthology.org/2024.findings-eacl.134/.

Mingrui Wu and Sheng Cao. Llm-augmented retrieval: Enhancing retrieval models through language models and doc-level embedding, 2024. URL https://arxiv.org/abs/2404.05825.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models, 2025a. URL https://arxiv.org/abs/2510.04618.

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10, 2023. URL http://dx.doi.org/10.1109/JAS.2022.106004.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025b. URL https://arxiv.org/abs/2506.05176.

A Dataset Statistics

B Prompts

The `GENERATE` prompt is in Figures 2 and 3; the `TRANSFORM` prompt is in Figure 4.

C Full Test-time Retrieval Results

The effect of test-time transformation across model scales is reported in Table 6.

D Query-Specific Filtering

Table 7 reports the full domain-level table alongside retrieval gains, and Table 8 reports filtering errors.

GENERATE

```
# Universal Domain Transformation Template Creator

You are an expert in the **{domain}** domain with deep
understanding of its key concepts, terminology, and analytical
requirements. You are tasked with creating a **concise, focused**
transformation template for the **{domain}** domain.

## Your Task
Analyze the provided sample passages and create a JSON
transformation template that can be universally applied to ANY
passage in the {domain} domain. **Focus on simplicity and essential
information only.**

## Template Requirements

### Core Functionality
- **Universal Applicability**: Must work for any passage type
within the specified domain
- **Consistent Structure**: Same field names and organization
across all transformations
- **Required Summary Field**: Every transformation must include a
concise summary that captures the essence of the original passage
- **Information Preservation**: Capture essential information while
being more concise than the original
- **Analysis-Ready**: Structure data to facilitate comparison,
reasoning, and future analysis tasks

### Design Principles
- **Maximum 3-5 top-level fields**: Keep the schema simple and
focused
- **Combine related information**: Group similar concepts together
rather than creating separate fields
- **Avoid redundancy**: Don't repeat information across multiple
fields
- **Prioritize core domain elements**: Focus only on what's most
critical for this domain
- **Enable future reasoning**: Structure information in a way that
supports downstream analysis, comparison, and reasoning tasks
across multiple passages

## Analysis Process
1. **Identify Core Patterns**: Look for the 2-3 most important
information types that appear across all passages
2. **Determine Essential Elements**: What are the absolute minimum
fields needed to capture the domain's essence?
3. **Design Structure**: Create a simple JSON schema with minimal
but comprehensive fields

## Output Format
Provide your transformation template in two parts:
### Part 1: Template Schema (Maximum 5 fields)
```json
{
  "summary": "Required: A concise summary that captures the main
  points and essence of the original passage",
  "field_name": "description of what goes here",
  "another_field": ["array if multiple items expected"],
  "nested_object": {
    "subfield": "description"
  }
}
```
```

Figure 2: The GENERATE prompt

GENERATE (cont.)

```
### Part 2: Applied Example
Show your template applied to one of the sample passages.

### Part 3: Brief Usage Guidelines
In 2-3 sentences, explain how to handle edge cases and ensure
consistent application.

### Legal Domain Example
Here's a **simplified** legal domain template:

Original Legal Passage:
{legal passage}
{legal transformation}

## Sample Passages for Analysis
{sample passages}

### Success Criteria
Your template succeeds if it:
- Has **3-5 fields maximum**
- Can be applied universally within the domain
- Captures essential information without redundancy
- Uses clear, descriptive field names
- Produces consistent, analyzable outputs

**Remember: Simplicity and focus are key. Avoid creating exhaustive
schemas with many bullet points or granular breakdowns.**
```

Figure 3: The GENERATE prompt (cont.)

TRANSFORM

```
You are a {domain} expert. Your task is to analyze the following
{domain} passage and return your answer in **strictly valid JSON**,
using the schema below:

```json
{the json schema from the seed output}
```

Strict guidelines:
{guidelines}

Respond **only** with the JSON object.

Here is the passage:
{passage}
```

Figure 4: The TRANSFORM prompt

Table 5: The test queries and documents are from BRIGHT (Su et al., 2025), the training queries are the Hard Query (HQ) subset of the synthetically-generated ReasonIR dataset (Shao et al., 2025).

Domain	# Train Queries (k)	# Test Queries	# Test Documents (k)
Biology	8.3	103	57.4
Earth Science	8.1	116	121
Economics	8.3	103	50.2
Psychology	8.5	101	52.8
Robotics	5.5	101	62
StackOverflow	8.4	117	107
Sustainable Living	8.5	108	60.8
Pony	1.9	112	7.89
LeetCode	8.4	142	414
AoPS	7.7	111	188
TheoremQA Theorems	6.8	76	23.8
TheoremQA Questions	7.7	194	188

Table 6: Performance impact of data transformation on test set (change in nDCG@10). Δ = (Transformed test) - (Non-transformed test) for each method. Positive values / Negative values indicate improvements / degradation when test data is transformed.

Domain	**Qwen3-0.6B**		**Qwen3-4B**		**Qwen3-8B**	
	Δ HQ FT	Δ No FT	Δ HQ FT	Δ No FT	Δ HQ FT	Δ No FT
AoPS	+0.45	-3.36	-2.63	-3.70	+0.88	-1.07
Biology	-0.32	+0.75	+4.07	+2.98	-0.84	+2.41
Earth Sci.	-2.49	-7.31	+12.34	-3.58	-4.54	-5.74
Economics	-1.00	+0.97	+3.97	+6.52	-2.88	+3.15
Leetcode	+7.42	-1.78	+10.67	-1.02	+8.58	+2.03
Pony	+0.32	+1.04	+0.17	+1.52	-0.23	+0.36
Psychology	-2.26	+0.16	+3.71	+3.07	-1.34	+2.79
Robotics	+2.44	+6.05	+5.90	+6.53	+2.58	+3.21
Stackoverflow	-2.51	+4.44	+0.35	+0.60	-0.48	+1.71
Sustain. living	+5.04	+1.17	+7.31	+5.54	+4.16	+4.22
TQA theorems	-13.06	-6.22	+0.97	-2.48	-6.96	-0.52
TQA questions	-4.33	-1.79	+0.40	+4.02	+0.09	+0.50
Average	-0.86	-0.49	+3.94	+1.67	-0.08	+1.09

D.1 Field Retention

For each domain, the *field empty rate* is the fraction of schema field instances left empty after filtering.

Across the eleven domains with filtering results, field empty rate is not significantly correlated with average retrieval gain ($r = 0.15, p = 0.66$). With eleven domains this test has limited power, so we treat it as a lack of evidence for a content-density effect rather than evidence against one.

D.2 Error Rates

Error rate measures the percentage of documents where the `FILTER` API failed to produce valid, parse-able JSON output. These failures include malformed outputs, syntax errors, repetitive token generation, and safety filtering. Some other errors are caused by the max token limit.

API-level analysis challenges. Table 8 reports query-conditioned filtering error rates ranging from 0.03% to 28.19% across domains. Failures include malformed JSON, safety filtering, repetitive generation, and output truncation. These failures limit the coverage of the field-retention analysis in Table 7. We use `FILTER` as a diagnostic tool to examine query-relevant content; the main retrieval

Table 7: Field retention after query-conditioned GPT-4.1 filtering, alongside retrieval gains of *unfiltered* AStrucT over HQ FT (Table 1). Gains are nDCG@10 percentage points; Avg. Gain averages the three model scales. Entries are query–document pairs. TQA refers to TheoremQA (Chen et al., 2023).

Domain	Entries	Empty Rate (%)	0.6B Gain	4B Gain	8B Gain	Avg. Gain
AoPS	15,384	36.20	2.03	2.43	2.24	2.23
Biology	16,600	12.74	2.09	2.33	4.77	3.06
Earth Science	16,222	24.16	11.31	11.20	9.35	10.62
Economics	16,504	26.89	1.20	2.76	2.41	2.12
Leetcode	16,848	34.68	11.67	11.64	13.79	12.37
Pony	3,824	34.65	-0.84	-0.37	0.39	-0.27
Psychology	16,957	22.39	2.42	5.49	5.05	4.32
Robotics	11,035	17.78	2.19	1.46	1.76	1.80
Stackoverflow	16,734	19.37	-0.23	1.64	-0.28	0.38
Sustainable Living	17,050	31.55	1.74	4.29	4.05	3.36
TQA Theorems	13,666	29.34	1.93	1.35	1.79	1.69
TQA Questions	-	-	1.15	3.36	5.36	3.29

Table 8: Query-specific filtering: Domain-level error rates.

Domain	Entries	Error Rate (%)
AoPS	15,384	1.22
Biology	16,600	3.55
Earth science	16,222	1.84
Economics	16,504	18.58
Leetcode	16,848	0.62
Pony	3,824	0.05
Psychology	16,957	4.66
Robotics	11,035	1.62
Stackoverflow	16,734	4.66
Sustainable living	17,050	28.19
TQA theorems	13,666	0.03

experiments train on the structured documents produced by `TRANSFORM`, without query-conditioned filtering.

E Embedding Space Analysis

Our previous analysis in Appendix D.1 reports no statistically significant association between field empty rate and average retrieval gain. Next, we consider the geometric properties of the embedding space under each training condition. We analyze the Qwen3-4B model from a static and a dynamic perspective. For the static geometric analysis, we calculate the alignment between the queries and documents in the embedding space, as measured by the average cosine similarity between the query embeddings and document embeddings. Then, we compute the drift of the queries and documents in the embedding space, as measured by the distance between the embedding under the No FT model and that of the final model. We use the original (untransformed) BRIGHT test documents for evaluation.

Query-document alignment. Table 9 *(top)* reports the macro-averaged cosine similarities between queries and their associated documents across all domains. All three models show a positive margin: the hardest negatives outscore true positives in similarity, as expected for a benchmark where retrieval requires non-trivial reasoning rather than surface-level overlap. HQ finetuning degrades this problem: positive similarity drops sharply (0.396 vs 0.467) and the neg-pos margin widens(0.132 vs 0.122), meaning that the model becomes more "confused" by hard negatives after training on the original ReasonIR synthetic data. AStrucT partially reverses both effects, recovering positive alignment (0.407) and narrowing the margin (0.129). The per-domain recovery in query-positive similarity

Table 9: Embedding space analysis for Qwen3-4B (macro-averaged over 12 domains). *Top*: cosine similarity between queries and their positive/top-10 negative documents. Margin = Neg−Pos; all values are positive because BRIGHT's hard negatives outscore positives in surface similarity—larger margin means the model is more confused by hard negatives. *Bottom*: embedding drift, measured as cosine distance between pretrained and finetuned representations (Movement rows), and ranking displacement (Rank row). †Average rank of the pretrained model's top-10 hardest negatives after finetuning (initial rank ≈ 6.0); higher values indicate greater ranking disruption.

	No FT	HQ FT	AStrucT
Query–Document Similarity			
Query-Positive	0.467	0.396	0.407
Top-10 Negative	0.589	0.529	0.536
Margin (Neg−Pos)	0.122	0.132	0.129
Embedding Drift from Pretrained			
Query Movement	—	0.239	0.154
Positive Doc Movement	—	0.234	0.134
Vanilla Negative Rank†	6.0	141.4	90.7

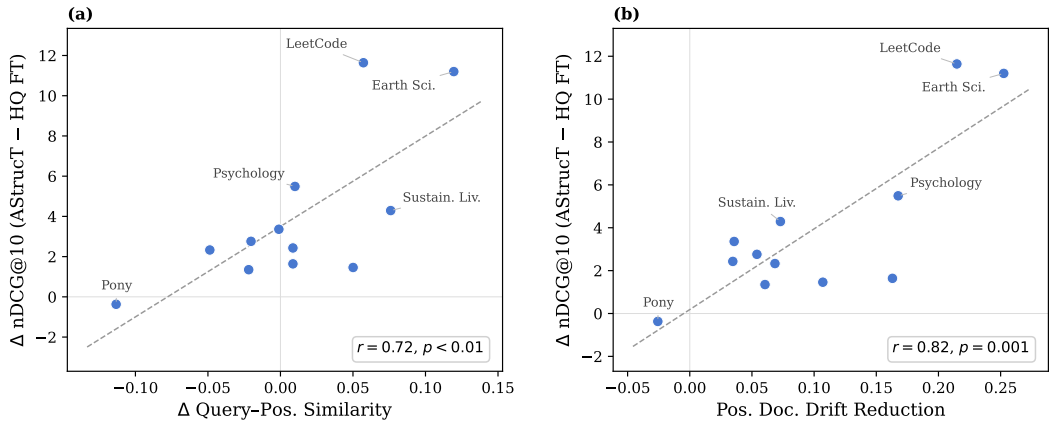


Figure 5: Relationship between embedding space changes and retrieval improvement (AStrucT vs. HQ FT) for Qwen3-4B across 12 BRIGHT domains. *Left*: change in query-positive cosine similarity vs. nDCG@10 gain. *Right*: reduction in positive document embedding drift vs. nDCG@10 gain. Each point is one domain. Drift reduction is the largest reported correlation with retrieval improvement ($r = 0.82$, $p = 0.001$). For comparison, across the same twelve domains, HQ FT's own embedding drift correlates with its signed retrieval degradation (No FT minus HQ FT nDCG@10) at $r = 0.27$, $p = 0.40$ for queries and $r = 0.46$, $p = 0.13$ for positive documents.

(AStrucT minus HQ FT) correlates with nDCG@10 gain ($r = 0.72, p < 0.01$; Figure 5 *left*), linking retrieval improvement to the recovery of query-positive alignment.

Embedding drift and retrieval improvement. Complementing the alignment results, Table 9 *(bottom)* reports drift as the cosine distance between each item's pretrained and finetuned embeddings. On average, HQ FT moves query embeddings by 0.239 and positive documents by 0.234; AStrucT reduces this drift by 35.7% and 42.9%, respectively. This reduced drift extends to the ranking structure: the pretrained model's top-10 hardest negatives are displaced to an average rank of 141 after HQ FT but only to rank 91 after AStrucT, indicating that AStrucT preserves the pretrained ranking landscape better than HQ FT. In domains where the pretrained model is already strong, this manifests as small drift: in LeetCode, where No FT scores 37.11 and HQ FT incurs a 14.02 pp degradation, AStrucT limits query and positive movement to 0.086 and 0.098, preserving most of the pretrained performance (34.73).

Across the twelve BRIGHT domains, larger reductions in embedding drift relative to HQ FT are associated with larger nDCG@10 gains: $r = 0.82, p = 0.001$ for positive-document drift reduction (Figure 5, *right*) and $r = 0.79, p = 0.002$ for query drift reduction. By contrast, AStrucT's absolute

embedding movement is only weakly correlated with its final nDCG@10 score ($r = 0.19, p = 0.55$). The content-retention analysis likewise shows little association between the density of retained query-relevant content and retrieval gain ($r = 0.15, p = 0.66$; Appendix D.1). Thus, the observed gains track changes in embedding geometry more closely than the measured density of query-relevant content.

F LIMITATIONS

Empirical scope. We evaluate GPT-4.1 transformations and Qwen3-Embedding retrievers on BRIGHT, with partially shared training and retrieval document collections. Other retriever families, locally hosted LLMs, benchmarks, and temporal or document-side shifts are not evaluated. We have not compared against alternative query-generation methods or measured downstream task quality beyond nDCG@10. In particular, newer models for transforming may be substantially cheaper than the costs reported in our study.

Cost and data access. Preprocessing cost grows with dataset size and output length (Section 3.4). Experiments using more negative samples would also require more transformations. Finally, our experiments send passages to an external LLM provider; settings where documents cannot be shared externally would require a self-hosted LLM.