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

Xinyan Velocity Yu[†] **Harsh Jhamtani** **Soham Dan**[†] **Benjamin Van Durme** **Patrick Xia**
University of Southern California Microsoft
xinyany@usc.edu patrickxia@microsoft.com
[†]Work done while at Microsoft

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 Transforms (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 generated based on a small sample of target-domain passages; notably, this induction is fully automated—no human inspection of the target documents and no knowledge of test-time queries—preserving data privacy in settings where manual review is restricted. Across three model scales and twelve diverse domains (BRIGHT), ASTrucT yields an average improvement of 3.77 percentage points (pp) nDCG@10 over direct finetuning, and 1.10 pp over the pretrained baseline, consistently mitigating negative transfer. Furthermore, we find that these improvements are driven by the structural inductive bias of the transformation rather than the density of query-conditioned content retention. These findings provide a practical strategy for practitioners to finetune an embedding model for retrieval without eyes-on access to the underlying documents or test-time queries.

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: how should available training data be used—as-is, preprocessed, or avoided 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 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 reframes the central question to "how should we prepare training data to maximize robustness against unknown distributional shifts?"

Our answer focuses on document preprocessing. We propose Adaptive Structured Transformation (AStrucT), a privacy-preserving preprocessing technique that reorganizes training documents into domain-focused structures prior to model finetuning. Unlike prior expansion work that augments documents with generated content, our approach restructures existing content through explicit sectioning, reordering, and contextualization. Across three model scales (Qwen3-Embedding 0.6B, 4B, 8B; Zhang et al. (2025b)) and twelve diverse domains from BRIGHT (mathematics, biology, programming, social sciences; Su et al. (2025)), we show that finetuning on structured transformation *mitigates negative transfer*, yielding an average improvement of 3.77 pp over direct finetuning, and often matching or exceeding pretrained baseline, even when synthetic supervision is misaligned.

Transformation gains are largest in domains that exhibit the strongest negative transfer under direct finetuning. We observe the largest improvements in technical domains, with +12.37 pp for LeetCode and +10.62 pp for Earth Science. Critically, these gains emerge from the transformation structure itself rather than from content retention, i.e., the amount of query-relevant information preserved in the transformed documents. We observe negligible correlation ($r = 0.15, p = 0.66$) between content retention and retrieval effectiveness. This suggests that explicit document reorganization provides generalizable robustness across distributional boundaries.

While document expansion techniques have been extensively studied for improving retrieval effectiveness (Nogueira et al., 2019; Bonifacio et al., 2022; Weller et al., 2024; Suzgun et al., 2025), these approaches primarily target inference-time document enhancement or assume access to in-distribution queries for training document expansion models. In contrast, we investigate whether structural transformation applied during training data preparation can proactively mitigate distribution shift without requiring test-set validation or auxiliary expansion models.

Our findings provide a practical strategy for practitioners: when training data provenance is uncertain or test-set validation is infeasible, structural transformation offers reliable insurance against distribution shift with minimal overhead. The approach requires no manual inspection of target documents, no test-set queries, no iterative evaluation, and introduces computational costs only during a one-time preprocessing step, making it particularly suitable for resource-constrained or privacy-sensitive deployments. [1]

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 reactive 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 into two stages: `GENERATE` and `TRANSFORM`.

2.1 STEP 1: SCHEMA GENERATION (`GENERATE`)

The first stage automatically generates a domain-specific schema (S_d) by prompting a strong LLM to analyze sample passages from the target domain. Given a domain d and a small set of k

[1] Code will be made available.

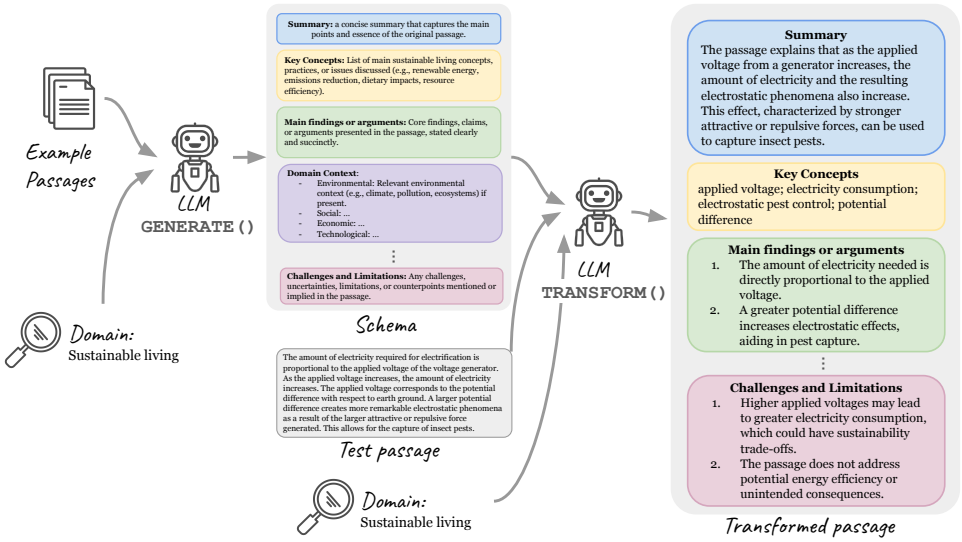

Figure 1: Adaptive Structured Transformation is a two-stage framework for generating structured representations of documents in arbitrary target domains. GENERATE uses sample passages to create a domain-specific *schema*. TRANSFORM then applies the schema over the full document collection.

sample passages $p_1, p_2, \ldots, p_k$ ($k = 5$ in our experiments), we use the following procedure: $\texttt{GENERATE}(d, p_1, \ldots, p_k) \rightarrow S_d$.

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.

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 maybe unique to that domain (e.g. "citations" for scientific articles).

To ensure that schema generation does not rely on full-corpus statistics or downstream evaluation signals, we include a single illustrative example from a disjoint legal-domain corpus (CLERC; Hou et al. (2025)) in the GENERATE prompt. This example is used solely to demonstrate the expected format of the schema itself and is independent of our training and evaluation data used in our experiments.

2.2 Step 2: Document Transformation (TRANSFORM)

The TRANSFORM stage applies the generated schema S_d to each document p in the training corpus: $\texttt{TRANSFORM}(d, S_d, p) \rightarrow \hat{p}$ where $\hat{p}$ is a structured JSON representation of p following schema S_d. Crucially, this transformation is query-independent: it extracts and reorganizes potentially relevant information into an abstracted format without knowledge of the downstream test-set tasks or query patterns.

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.

3 Experiments

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[2]. 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

The full prompts can be found in the Appendix B. `GENERATE` contains a seed prompt from a legal information retrieval dataset (CLERC; Hou et al. (2025)) which is independent of the BRIGHT document collection and the domains. We use CLERC to develop the `GENERATE` and `TRANSFORM` prompt without prior knowledge of BRIGHT. Our method leads to a +3% nDCG@10 performance increase on a subset of CLERC, therefore, we proceed to use the prompt for our main experiments.

3.3 MODELS

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

We use GPT-4.1 via OpenAI's API[3] 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 constant warmup ratio of 0.06. We train for 700 steps, evaluating every 10 and using early stopping with a patience of 5. In practice, most runs ended within a couple of hundred steps, ranging from 0.2 to 6 hours depending on the model size.

4 RESULTS

4.1 TRAINING WITH ADAPTIVE STRUCTURED TRANSFORMATIONS

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).

We observe a predominant but not 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 1.67, 2.94, and 3.11 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 uniform: direct finetuning yields modest gains in domains such as biology and robotics, and incurs substantially smaller regressions in StackOverflow and AoPS. We also observe weaker negative transfer for smaller models, suggesting an interaction between model capacity and susceptibility to misaligned training signals. Taken together, these results highlight a critical challenge for practitioners: 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 rapidly improve—particularly in domains such as code—naive finetuning on misaligned synthetic supervision can not only fail to help, but actively harm performance.

[2]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.

[3]`https://openai.com/api/`

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
Macro_Avg	14.69	**17.57**	16.36	18.01	**21.98**	20.95	17.80	**22.02**	20.91

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. "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).

AStrucT acts as a safeguard against 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 19.6%, 22%, 23.7% over direct finetuning for the 0.6B, 4B, and 8B scales, respectively, compared to direct finetuning, and by 7.4%, 4.9%, and 5.3% over the pretrained baseline. While no single method dominates on every domain, AStrucT provides the most reliable aggregated gains across diverse domains and model scales.

A central question in this analysis is whether finetuning is necessary if the pretrained model performs adequately. We argue that while general-purpose pretrained models provide a strong baseline, niche domains such as enterprise and personalization may still require optimization for specialized vocabularies and complex document structures that only domain-specific data can provide. AStrucT offers reliable insurance, allowing practitioners to realize the benefits of domain adaptation while mitigating the inherent risks of distribution shift from synthetic or uncertain data sources.

4.2 Test-time Retrieval with AStrucT

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
Macro-Avg	-0.91	-0.60	+3.94	+1.66	-0.08	+1.08

Table 2: 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.

We further evaluate the impact of AStrucT by testing models on transformed test data to isolate the effects of structural abstraction on retrieval quality. While a potential concern is that such transformations might degrade performance by obfuscating document structure, our results consistently demonstrate the opposite.

We observe that models finetuned on untransformed data exhibit domain-dependent gains when retrieving from transformed test documents (see Table 2 for full results). These improvements are particularly substantial in technical domains, such as LeetCode (+7.42 to +10.67 pp) and Robotics (+2.44 to +5.90 pp). This indicates that structural transformation creates a high-signal representation that retrievers can exploit more effectively than raw text, moving beyond simple distribution alignment.

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 2.07 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 ranking. During training, contrastive learning operates over the seed corpus filtered by a FineWeb-Edu classifier (Penedo et al., 2024), where transformed 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 unselective application of synthetic transformations at test time can introduce noise that outweighs their structural benefits.

5 Analysis

Our results demonstrate that Adaptive Structured Transformations yields consistent gains across diverse domains, but the magnitude of these improvements varies significantly. In this section, we analyze the structural properties of our transformations to understand why they provide robust insurance against distribution shift. We first evaluate different strategies for integrating transformed data into the retrieval pipeline, followed by a qualitative and corpus-level analysis of the generated schemas.

5.1 Ablations: Data Mixing

We compare *transform-only* (our default) against *prepend* (concatenating the transformation before the original text) and *mix* (indexing transformed and untransformed documents as separate items). More details can be found in Appendix C. *Transform-only* consistently achieves the highest macro-average nDCG@10 across all model scales. Gains from *transform-only* grow with model capacity, and procedural domains such as LeetCode and TheoremQA benefit most, validating our choice of ***transform-only*** for all main experiments.

5.2 Understanding Transformation Effectiveness via Query-Specific Filtering

Our structural transformations can improve retrieval performance for at least two distinct reasons. First, under a **content distillation** hypothesis, transformations act as an implicit filter that removes irrelevant or low-signal information, leaving behind a cleaner, more query-relevant representation. If this mechanism dominates, then domains in which a larger fraction of transformed content remains relevant to a given query should exhibit greater performance gains. Second, under a **structural inductive bias** hypothesis, the benefit arises from the imposed organization itself: by presenting information in a consistent, abstracted structure, the transformation biases the retriever toward more effective matching, even when much of the structure is sparsely populated. In this case, performance improvements should be largely independent of how much query-relevant content is retained.

To distinguish between these hypotheses, we introduce a query-specific `FILTER` stage that prunes transformed documents to retain only fields deemed relevant to a particular query while preserving

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 3: Domain-specific statistics and performance gains across three Qwen3 model sizes. We filter fields for relevance using GPT 4.1 and report the resulting improvement from training on these filtered structured documents. TQA refers to TheoremQA (Chen et al., 2023).

the original schema structure. This allows us to measure how much content survives aggressive query-conditioned distillation and to test whether performance gains correlate with content retention or instead persist despite substantial schema sparsity.

We evaluate the relationship between **query-conditioned content retention** and retrieval performance using two metrics across domains:

- **Field empty rate** measures the percentage of schema fields that the `FILTER` stage removes when conditioning on a specific query. This serves as a proxy for how much transformed content is retained after aggressive, query-specific distillation. We compute this by summing empty field instances across all fields and documents, then dividing by the total possible field slots per domain.
- **Average gain** measures the mean retrieval improvement across all model scales (0.6B, 4B, 8B). We report the average nDCG@10 difference between models trained on transformed data and those trained on the original untransformed data, evaluated on BRIGHT test queries and documents.

Table 3 summarizes these statistics across domains, reporting field empty rates alongside average retrieval gains. Not all documents are successfully transformed, due to cases where all fields are pruned or API failures occur. We report detailed error statistics and analyses in Appendix D. Nevertheless, each domain contains a sufficient number of documents to support robust corpus-level analysis and to draw conclusions about the relationship between content retention and downstream retrieval performance.

Field-level content retention. Our analysis of 160,824 transformed training documents reveals a consistent two-tier pattern across domains. Core semantic fields like `summary` and `key_concepts` exhibit high retention rates across all domains ($\geq 90\%$ non-empty), indicating that transformation reliably preserves high-level semantic information. In contrast, domain-specific auxiliary fields display substantial variability and are frequently pruned under query conditioning. Technical domains in particular exhibit pronounced sparsity: in LeetCode, the `examples` field remains empty in 90.7% of the documents, while in AoPS the `final_answer` is removed in 66.9% of cases. By comparison, conceptual fields such as `key_concepts`, `domain_concepts`, and `core_concepts` consistently exceed 95% retention in 9 out of 10 domains where they are present.

Structural robustness under sparse content. If performance gains were primarily driven by content distillation, we would expect retrieval improvements to correlate with the amount of query-relevant content retained after filtering. However, we observe no such relationship. Across domains (Table 3), the correlation between field empty rates and average retrieval gain is negligible ($r = 0.15, p = 0.66$). Notably, the domains with the largest improvements—LeetCode (+12.37 pp) and Earth Science (+10.62 pp)—retain only a moderate fraction of schema fields (34.7% and 24.2% empty, respectively). Conversely, domains with the highest content retention, such as Biology (12.7% empty) and Robotics (17.8% empty), exhibit substantially smaller gains (+3.06 pp and

+1.80 pp). These results indicate that retrieval effectiveness does not depend on dense or exhaustive content retention, but instead, remains robust even when the transformed representation is highly sparse.

Taken together, these findings make it unlikely that content distillation is the primary mechanism underlying AStrucT's effectiveness observed in Section 4, and instead support the structural inductive bias hypothesis: the imposed structural abstraction of document content, rather than the quantity of retained query-relevant information, is the dominant contributor to improved retrieval robustness. We note that query-specific filtering can also alter the difficulty of contrastive learning by simplifying hard negatives, which may offset potential benefits from content distillation under standard retriever pipelines. Consequently, while our analysis suggests that content distillation alone does not explain the observed gains in our experimental setup, a more exhaustive test would require constructing equally challenging hard negatives in the filtered representation space. Accordingly, we adopt the `TRANSFORM`-only setting—without post-hoc query-specific filtering—for all main results, as structural reorganization alone provides the desired insurance against distribution shift.

5.3 Negative Transfer in Embedding Space

Negative transfer from HQ FT is directly visible in embedding geometry: query-positive cosine similarity drops from 0.467 to 0.396, and hard negatives become more similar to queries than true positives—the model grows more confused after training. AStrucT partially reverses both effects, with per-domain recovery in query-positive cosine similarity correlating with nDCG@10 gain ($r = 0.72$). A similar pattern holds for the embedding dynamics: AStrucT reduces drift from pretrained positions by $\sim$40%, and drift reduction correlates the most with retrieval improvements ($r = 0.82$). Full analysis and per-domain breakdowns are in Appendix E.

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) leverage LLMs to create new training instances. However, these approaches focus on content generation, implicitly relying on LLM-generated queries that will align with the target distribution. In contrast, our Adaptive Transformation is a preventive safeguarding pre-processing step that reorganizes and summarizes existing document content into domain-specific schemas without generating new text.

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., QA-Expand (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) rely on generating synthetic queries, mining the negative documents from the target corpus, and scoring them with a cross-encoder to provide pseudo-labels. While this pipeline provides a high-quality training signal, it fundamentally requires "look-ahead" access to the target distribution and involves computational overhead for cross-encoder labeling. In contrast, pipelines like Shao et al. (2025) focus on generating reasoning-intensive "hard queries" directly from the document collection without an auxiliary labeling stage.

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.

6.3 Privacy-Preserving and Preventive Processing

Most prior work in retrieval adaptation and data augmentation focuses on improving performance under the assumption of access to downstream signals, through mechanisms such as synthetic supervision (Wang et al., 2024), corpus-level optimization (Wang et al., 2022), or iterative validation (Thakur et al., 2021). However, in practice, practitioners often face settings where such access is unavailable and additional training must be performed under uncertainty, with no opportunity to detect or correct negative transfer. More broadly, work on domain-adaptive training has shown that continued optimization can degrade performance when the adaptation signal is misaligned with downstream use (Gururangan et al., 2020), rather than improving robustness. Despite this risk, there has been little investigation into training-time pre-processing strategies that explicitly aim to prevent performance regression in dense retrieval models under blind or privacy-constrained deployments. Our work addresses this gap by studying fixed, query-independent structural transformations as a preventive safeguard, providing robustness against synthetic or misaligned training signals without relying on target-side manual access or test-time adaptation.

7 Conclusions

In this work, we investigate the challenge of negative transfer in dense retrieval, where finetuning on misaligned or synthetic data causes significant performance degradation. We introduced Adaptive Structured Transformations, a safeguarding preprocessing technique that restructures training documents into custom domain-focused schemas.

Our method is amenable to resource-constrained or privacy-sensitive settings, as it requires no manual inspection of target documents and no access to test-time queries, and incurs only a one-time preprocessing overhead. Our experiments across three model scales and twelve diverse reasoning domains demonstrate that: (1) structural transformation effectively mitigates distribution shift, recovering performance losses from negative transfer, and frequently exceeding the pretrained baseline across all model scales, achieving macro-average relative improvements of 19.6%, 22.0%, and 23.7% for the 0.6B, 4B, and 8B models, respectively, compared to direct finetuning; and (2) these gains are driven by the imposed structure itself, as we only observe a negligible correlation between content retention and retrieval effectiveness. We discuss limitations and avenues for future work in Appendix F.

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.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

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/.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL https://aclanthology.org/2024.acl-long.642/.

Zora Zhiruo Wang, Akari Asai, Xinyan Velocity Yu, Frank F. Xu, Yiqing Xie, Graham Neubig, and Daniel Fried. CodeRAG-bench: Can retrieval augment code generation? In *Findings of the Association for Computational Linguistics: NAACL 2025*, 2025. URL https://aclanthology.org/2025.findings-naacl.176/.

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

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 4: 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).

B Prompts

The `GENERATE` and the `TRANSFORM` prompts used in our experiment are in Figure 2 and Figure 3.

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.)

GENERATE

```
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

C ABLATION: DETAILS ABOUT DATA MIXING

Domain	Qwen-0.6B			Qwen-4B			Qwen-8B		
	∪	⊕	→	∪	⊕	→	∪	⊕	→
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
Sustain. living	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
Macro-Avg	16.06	16.38	**17.57**	20.56	20.83	**21.98**	19.35	20.49	**22.02**

Table 5: Comparison of data mixing strategies for retrieval from BRIGHT (nDCG@10). ∪=*mix*: union of transformed and original documents; ⊕=*prepend*: transformed documents concatenated with original documents; →=*transform-only*: structured transformation only. Transform-only consistently achieves the best macro-average performance across all model scales.

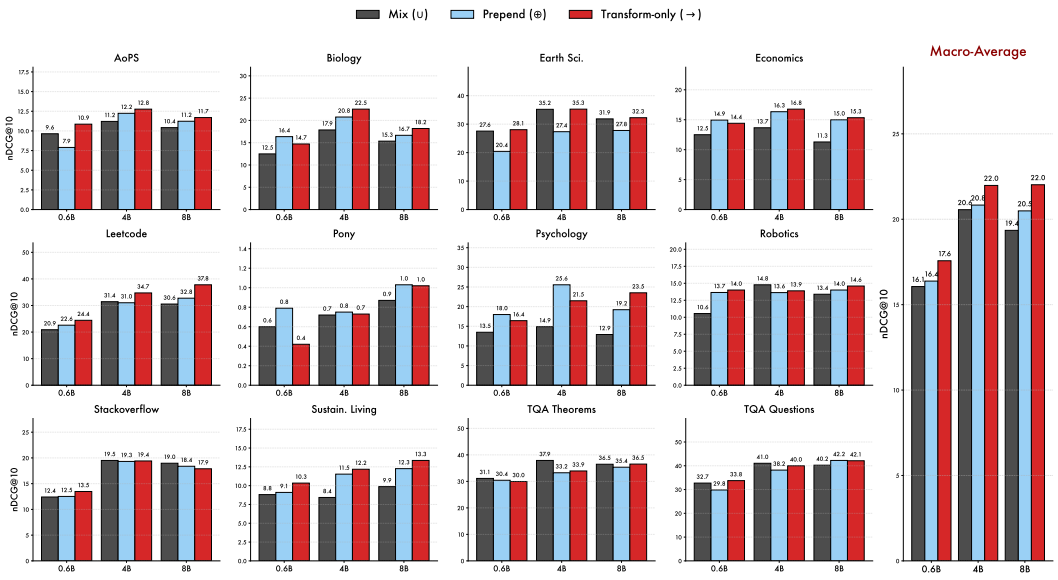

Figure 5: Comparison of data mixing strategies for retrieval from BRIGHT (nDCG@10). ∪=*mix*: union of transformed and original documents; ⊕=*prepend*: transformed documents concatenated with original documents; →=*transform-only*: structured transformation only. Transform-only consistently achieves the best macro-average performance across all model scales. The full numerical results is in Table 5.

As shown in Figure 5, *transform-only* consistently achieves the highest macro-average performance across all model scales. While *prepend* improves over *mix* in several domains, especially those requiring semantic abstraction, such as psychology and economics, it remains unstable and is frequently outperformed by *transform-only*. This suggests that reintroducing raw text can dilute the retrieval signal provided by the structured representation. Conversely, *mix* rarely yields the best results and consistently underperforms on average, indicating that allowing the retriever to rank heterogeneous document representations independently increases ranking noise rather than improving recall.

The gains from *transform-only* become more pronounced as model capacity increases, implying that larger retrievers are more capable of exploiting the structure induced by $\hat{p}$. Domain-level analysis further shows that procedural and solution-oriented domains such as LeetCode and TheoremQA benefit the most from *transform-only*, supporting the hypothesis that structured transformations capture reasoning-relevant information more effectively than raw or mixed text. These findings additionally validate our choice of ***transform-only*** for the experiments in our study.

D Query-specific Filtering Error Rate

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

Table 6: Transformation effectiveness: Domain-level 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 3 and Table 6 show that `FILTER` operations exhibit varying API success rates across domains, with error rates ranging from 0.03% to 28.19%. These errors reflect technical challenges in the filtering process: malformed JSON outputs, safety filtering, and other GPT-4 API failures, rather than fundamental limitations of the underlying schemas. Importantly, high error rates do not preclude retrieval gains: Sustainable living achieves +3.36 pp improvements despite 28.19% errors, while Economics gains +2.12 pp with 18.58% errors. This suggests that while `FILTER` stage can be analytically informative, it is not a prerequisite for effective document transformation. Accordingly, our main results (Table 1) rely solely on the `TRANSFORM` stage without query-specific filtering.

E Embedding Space Analysis

As our previous analysis in Section 5 establishes that the quantity of query-relevant content does not predict 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 distance 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 7 *(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 (0.132 vs 0.122), meaning that the model becomes more "confused" to 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

	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

Table 7: 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.

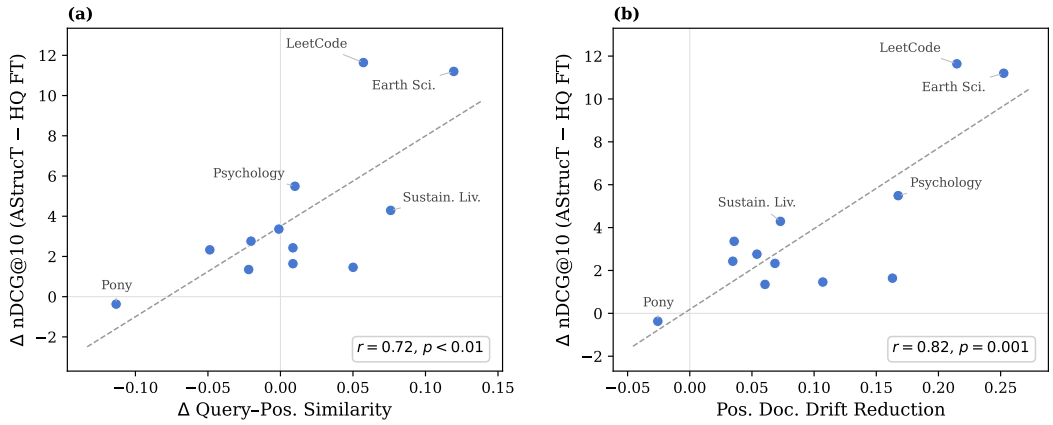

Figure 6: Relationship between embedding space changes and retrieval improvement (AStrucT vs. HQ FT) for Qwen3-4B across 12 BRIGHT domains. *Left*: change in query-positive cosine similarity vs. nDCG@10 gain. *Right*: reduction in positive document embedding drift vs. nDCG@10 gain. Each point is one domain. Drift reduction is the strongest predictor of retrieval improvement ($r = 0.82$, $p = 0.001$).

similarity (AStrucT minus HQ FT) correlates with nDCG@10 gain ($r = 0.72, p < 0.01$; Figure 6 *left*), indicating retrieval gains arise primarily from better promotion of the true positives rather than suppression of hard negatives.

Embedding drift as an implicit regularization mechanism. Table 7 *(bottom)* reveals the asymmetry in how the two finetuning approaches reshape the pretrained space. We measure drift as cosine distance between each item's embedding in the pretrained model and its counterpart in the finetuned model. 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 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 near-zero drift: in LeetCode, where No FT scores 37.11 and HQ FT incurs a 14.00 pp degradation, AStrucT limits query and positive movement to 0.086 and 0.098, preserving most of the pretrained performance (34.73).

Drift reduction highly correlates with retrieval improvement. Per-domain positive document drift reduction correlates with the nDCG@10 gain at $r = 0.82, p = 0.001$ (Figure 6 *right*), and query drift

reduction yields $r = 0.79, p = 0.002$. Furthermore, we found that AStrucT 's absolute movement does not predict the absolute nDCG@10 performance ($r = 0.19, p = 0.55$).

Combined with the content-retention analysis (Section 5.2), these findings provide two independent lines of evidence for AStrucT's mechanism: its effectiveness does not depend on the density of retained query-relevant content ($r = 0.15$, $p = 0.66$), but is instead driven by the degree to which structural transformation constrains embedding drift during finetuning ($r = 0.82$, $p = 0.001$).

F LIMITATIONS

Model and LLM dependencies. The benefits of AStrucT might vary across model architectures. Due to computational constraints, we only experiment on the Qwen3 model family. While our results demonstrate consistent patterns across all model scales, we cannot claim that these findings generalize to other embedding architectures, such as sentence transformers (Reimers & Gurevych, 2019), ColBERT-style models (Khattab & Zaharia, 2020), or proprietary embedding models. Our method also depends on GPT-4.1 JSON mode for both schema generation and transformation generation, and relies heavily on the instruction-following capabilities of the underlying LLM. We observed significant variability in the `FILTER` stage, with format error ranging from 0.03% to 28.19% across domains, and the majority of them are due to malformed JSON, safety filtering, and repetitive token generation. These failures highlight the vulnerability of our method: the specific "prompt-sensitivity" or the instruction-following capability of the model used to create the schema and conduct the transformation could potentially limit the reproducibility of the method. Changing the underlying models or the prompts could yield different results.

Computational and preprocessing overhead. Although Adaptive Transformation is query-independent and operates as a one-time cost, and our training corpus is, in general, way smaller than the test corpus, it still requires LLM-based processing for every document within the training corpus. For massive document collections, the cost of transforming the entire dataset into domain-specific schemas could be prohibitively high compared to naive finetuning or utilizing the vanilla pretrained model. Furthermore, prompt engineering for the `GENERATE` stage could also incur more cost.

Empirical Scope and Comparisons. Our experiments simulate distribution shift through training on LLM-generated synthetic queries and negative documents while testing on human-authored queries from BRIGHT, with partially shared document collections. While this represents a common distribution shift from synthetic data, we do not evaluate our method's performance under other common distribution shift scenarios, such as temporal drift, geographic variation, or domain transfer. Additionally, despite our evaluation domains being diverse, they are primarily from academic and technical sources within the BRIGHT benchmark. Real-world enterprise documents may exhibit different characteristics and challenges, such as proprietary jargon or multilingual content. Finally, our definition of distribution shift focuses on query distribution misalignment. We leave document-side shifts to future work. Additionally, while we position our method as complementary to existing expansion methods, which primarily target inference-time enhancement or assume access to in-distribution data, we do not empirically evaluate how AStrucT performs relative to these methods. Future work should conduct controlled comparisons to establish when training-time transformation is preferable to inference-time expansion or hybrid expansion strategies.

Limited evaluations. Our evaluation focuses exclusively on retrieval quality measured by nDCG@10, without assessing whether these gains translate to improvements on downstream tasks. In practical systems, retrieval is typically embedded within larger pipelines such as retrieval-augmented generation or knowledge-grounded dialogue systems, where end-task performance is the ultimate measure of success. While retrieval metrics offer a standardized and widely adopted basis for comparison, they may not fully reflect practical utility. Prior work, such as Wang et al. (2025), demonstrates that strong retrieval performance does not necessarily yield strong end-to-end results, as the end-task results also depend on the generator's capability. Future work should therefore evaluate whether the retrieval improvements induced by structured transformation translate into gains in complete task pipelines, and whether structured representations interact with downstream models differently than natural-language document representations.