MULTI^{3}IR: A Benchmark for Multi-perspective Multi-domain Multi-modal Information Retrieval

Seokwon Song **Sohyeon Kim** **Gunhee Kim**
Seoul National University
{seokwon.song, sohyeon.kim}@vision.snu.ac.kr, gunhee@snu.ac.kr

Abstract

Information retrieval (IR) increasingly targets open-ended queries that admit diverse perspectives. Existing IR benchmarks, however, focus primarily on closed-ended queries, while even open-ended benchmarks largely consist of queries whose supporting documents span a single subject domain and modality. We introduce MULTI^{3}IR, a benchmark that evaluates how well retrievers cover the multifaceted perspectives of open-ended queries across diverse domains and modalities. It comprises 104.9K Stack Exchange queries, each annotated with perspective descriptions that capture the query's implicit viewpoints. We further propose SPIN, a parameter- and label-efficient method that learns noise vectors to steer embeddings toward diverse yet meaningful semantic directions. Experiments show that existing multimodal retrievers suffer from single-perspective bias, while SPIN substantially improves perspective coverage on MULTI^{3}IR and generalizes well to unseen open-ended IR benchmarks. The dataset and experimental code are available at https://github.com/seokwon99/Multi3IR.

1 Introduction

Information retrieval (IR) has expanded from factoid queries to open-ended queries that decompose into **perspectives**, complementary sub-queries that each capture a distinct facet. For example, as illustrated in Figure 1, the query "Why are so few foods blue?" implicitly raises perspectives such as "Why are blue pigments so rare in nature?" and "How are blue foods generally perceived by people?" These perspectives provide information from diverse subject domains (e.g., biology, psychology) and modalities (e.g., text, image), collectively contributing to a more comprehensive answer.

Recent efforts have advanced open-ended IR in several directions. On the evaluation side, some benchmarks (Min et al., 2020; Zhao et al., 2024; Chen and Choi, 2025) incorporate open-ended queries that admit multiple answers grounded in different documents. On the method side, retrieval diversification, including query expansion (Gao et al., 2023; Wang et al., 2023; Zhang et al., 2024a) and multi-vector retrieval (Khattab and Zaharia, 2020; Humeau et al., 2019), seek to cover diverse aspects of a query rather than relying on a single dominant interpretation.

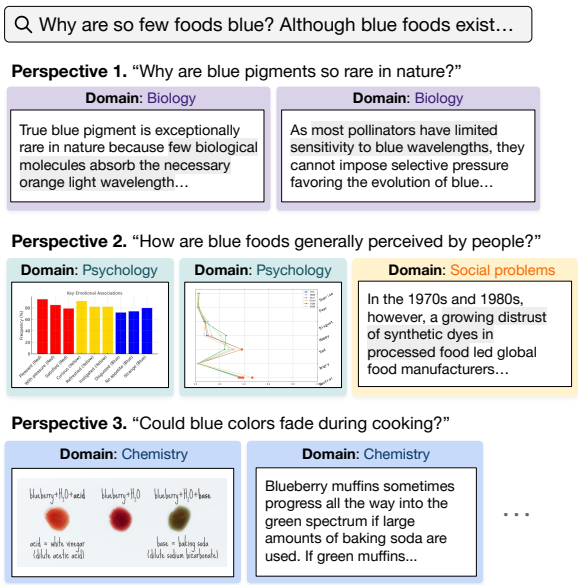


Figure 1: A datapoint example of MULTI^{3}IR. An open-ended query includes multiple implicit perspectives (three in this example), each paired with supporting documents from different domains and modalities.

However, current approaches remain limited in two key respects. First, as shown in Table 1, most IR benchmarks focus on closed-ended queries, overlooking information needs where quality depends on comprehensiveness or diversity. Even open-ended benchmarks largely consist of queries grounded in single-domain, text-only documents, leaving it unclear whether retrievers can provide comprehensive information across heterogeneous knowledge sources. Second, existing retrieval di-

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 59459–59478
October 24-29, 2026 ©2026 Association for Computational Linguistics

	#Q	Diversity of each Query		
		#SubQ	#Domain	#Modality
Closed-ended IR				
HotpotQA (2018)	113.0K	–	1.38	1.00
2WikiMHQA (2020)	192.6K	2.47	1.26	1.00
MultimodalQA (2021)	29.9K	1.83	1.20	1.23
FanOutQA (2024)	1.0K	6.92	1.67	1.00
WebQA (2022)	41.7K	–	1.28	1.00
InfoSeek (2023)	1.3M	–	1.00	1.00
M-BEIR (2024)	1.2M	–	1.44	1.00
MRMR (2025)	1.2K	–	1.03	1.05
Open-ended IR				
AmbigQA (2020)	12.0K	1.99	1.13	1.00
PIR (2024)	4.2K	2.39	1.63	1.00
BeRDS (2025)	3.2K	2.21	1.30	1.00
Multi^{3}IR (Ours)	104.9K	4.98	3.34	1.91

Table 1: Comparison of IR benchmarks. *Diversity per query* reports the average number of sub-queries (perspectives) per query, along with the number of distinct subject domains and modalities covered by their supporting documents. See Appendix A for details on subject domain labeling.

versification methods incur substantial overhead: query expansion requires external LLMs at inference time, while multi-vector retrieval requires costly document-relevance labels for fine-tuning.

In this paper, we introduce MULTI^{3}IR, an open-ended IR benchmark comprising 104.9K queries collected from Stack Exchange. For each query, we annotate a set of perspectives, each represented as a sub-query paired with the documents that support it. As shown in Table 1, each query in our benchmark requires comprehensive information spanning 3.34 domains and 1.91 modalities on average. We further propose SPIN, a parameter- and label-efficient IR training method that learns noise vectors to diversify the output of a frozen retriever into multiple perspective-aware embeddings, without requiring costly document-level annotations.

Our main contributions are as follows:

1. We propose MULTI^{3}IR, a large-scale benchmark of 104.9K open-ended queries that require comprehensive information across multiple subject domains and modalities.

2. We identify *single-perspective bias* in current multimodal retrievers, where retrieval for an open-ended query concentrates on a few dominant perspectives while neglecting the rest.

3. We propose SPIN, a parameter- and label-efficient training method that achieves higher perspective coverage than existing diversification methods.

2 Related Work

2.1 Open-ended Information Retrieval

Open-ended IR benchmarks examine whether a retriever can gather documents covering diverse information needs within a question. Early work such as AmbigQA (Min et al., 2020) introduced questions with multiple valid interpretations, with each disambiguation grounded in Wikipedia documents, but evaluates question answering rather than retrieval. PIR (Zhao et al., 2024) extends this setting to retrieval, evaluating whether a retriever can return documents relevant to an explicitly specified perspective within a query. Since users rarely state their perspectives, BeRDS (Chen and Choi, 2025) instead gives the retriever only the original question and evaluates whether the retrieved documents cover all annotated perspectives. However, these benchmarks largely consist of queries grounded in single-domain, text-only documents, limiting their ability to evaluate comprehensive retrieval across diverse knowledge sources. MULTI^{3}IR addresses this gap by collecting supporting documents across multiple domains in both text and image modalities.

2.2 Retrieval Diversification

Dense retrievers rank documents by similarity to a single query vector, which cannot be close to all relevant documents when they are scattered across the embedding space (Chen et al., 2025; Weller et al., 2025). Prior work introduces diversity at three stages. *Post-retrieval* re-ranks an initial candidate pool to promote diverse results (Carbonell and Goldstein, 1998; Kulesza and Taskar, 2012); since they only reorder the first-stage pool, documents missing from it remain unretrievable. *Query expansion* rewrites the query before retrieval, generating hypothetical documents (Gao et al., 2023) or pseudo-answers (Wang et al., 2023; Zhang et al., 2024a), at the cost of an LLM call for every query. *Multi-vector retrieval* represents a query with multiple embeddings (Khattab and Zaharia, 2020; Humeau et al., 2019), expanding the regions it can cover, but training them relies on costly document-level relevance annotations, especially for open-ended queries spanning multiple perspectives. SPIN addresses this by training a multi-vector retriever using only perspective descriptions, improving perspective coverage without document-level annotations or query-time LLM inference.

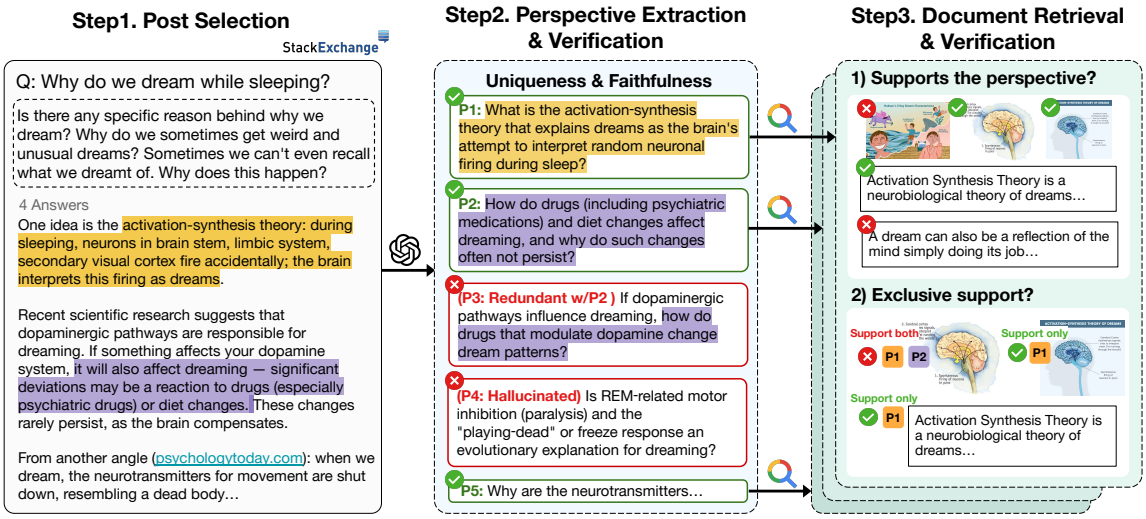


Figure 2: Overview of our data construction pipeline (§ 3.2).

3 The MULTI^{3}IR Benchmark

3.1 Task Formulation

Open-ended query Q involves multiple implicit perspectives (Figure 1). We define a set of perspectives P, where each $p_i \in P$ is a textual description grounded in documents D_i. Given Q and a corpus C, a retriever $f(\cdot)$ that maps inputs to a representation space returns the top-k documents as

$$D^*(k) = \underset{d \in C}{\text{Top-}k}\, f(Q)^\top f(d), \qquad (1)$$

The retrieval objective is to cover every perspective $p_i \in P$ by retrieving at least one document from its corresponding supporting set D_i.

3.2 Dataset Construction

As illustrated in Figure 2, we construct the dataset using an automated three-stage pipeline. After that, we conduct human annotation to construct test set. Overall dataset statistics are presented in Figure 3.

Selecting posts. We collect posts from Stack Exchange, a rich source of multi-answer threads with diverse perspectives. We use 77 sites across five categories, retain posts with more than 3 answers, and sample uniformly across categories. For each post, Q is the concatenated title and body, and A is the set of answers. See Appendix B.1 for details.

Perspective extraction and verification. We extract perspectives $\hat{P} = \{\hat{p}_1, \hat{p}_2, \ldots\}$ from A using GPT-5-mini, then verify them as follows:

1. **Uniqueness.** We encode all perspectives with all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) and discard any $\hat{p}_j$ whose maximum cosine similarity to $\hat{P} \setminus \{\hat{p}_j\}$ exceeds 0.8.

2. **Faithfulness.** We use Bespoke-MiniCheck-7B (Tang et al., 2024) to produce a binary entailment judgment between each $\hat{p}_j$ and A, and discard any $\hat{p}_j$ judged as not entailed.

We remove questions with fewer than four perspectives, yielding 104.9K questions and 521.7K perspectives. See Appendix B.2 for details.

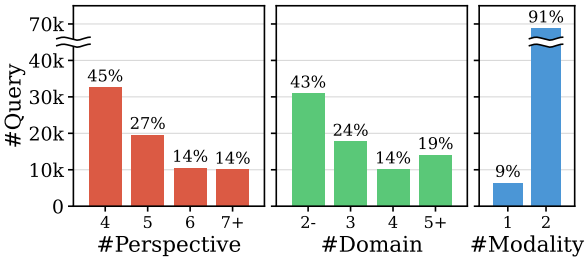
Document retrieval and verification. We collect documents supporting each perspective from two sources: Google Image Search for images and the Colossal Clean Crawled Corpus (C4) (Raffel et al., 2020) for text. Using each $\hat{p}_i$ as a query, we retrieve the top-10 documents from each and define $\hat{D}_i$ as their union. Since retrieved documents may support multiple perspectives, we enforce *exclusive support*: using Qwen3-VL-30B-A3B-Instruct (Team, 2025), we verify each $\hat{d} \in \hat{D}_i$ against $\hat{p}_i$ and the other perspectives, and define D_i as those supporting $\hat{p}_i$ alone. Finally, we label each document in D_i with a subject domain using EAI-Distill-0.5b (AI et al., 2025), yielding 1.01M multimodal documents with 3.34 domains and 1.91 modalities per query on average. See Appendix B.3 for retrieval and verification details, and Appendix A for domain labeling.

3.3 Human Verification

We verify each sample in two stages. Annotators first judge (Q1a) whether the perspective is relevant to the question, (Q1b) which other perspective

Statistic	Number
Query	
- Total	104,916
- Category type	5
- Avg. tokens	155.8
Perspective	
- Total	521,739
- Avg. tokens	27.1
Supporting Document	
- Total	1,012,061
- Modality type	2
- Domain type	94 (of 100)
Category of Queries	
- Culture & Recreation	34,776 (33.5%)
- Science	20,344 (19.6%)
- Business	19,132 (18.4%)
- Technology	15,756 (15.2%)
- Life & Arts	13,869 (13.4%)
Domain of Documents	
- Computer science & systems	197,451 (19.5%)
- Management & public relations	94,731 (9.4%)
- Law	69,191 (6.8%)
- Engineering	67,122 (6.6%)
- Others (90 domains)	583,449 (57.7%)
Modality of Documents	
- Text	631,377 (62.4%)
- Image	380,684 (37.6%)

(a) Overall dataset statistics. Gray sections report ratios sorted in descending order.



(b) Train query distribution by the number of perspectives, document domains, and document modalities.

Figure 3: Overall statistics and distribution analysis.

is closest to it, and (Q1c) whether the two are semantically redundant. They then rate whether the document fully, partially, or does not support (Q2a) the target perspective and (Q2b) the closest perspective from Q1b. Nearly all perspectives are relevant (99.1%) and unique (96.2%), with 90.2% of documents exclusively supporting their target perspective. We retain perspectives passing Q1 with their supporting documents, yielding a test split of 1.0K queries, 4.8K perspectives, and 12.5K supporting documents. See Appendix B.4 for details.

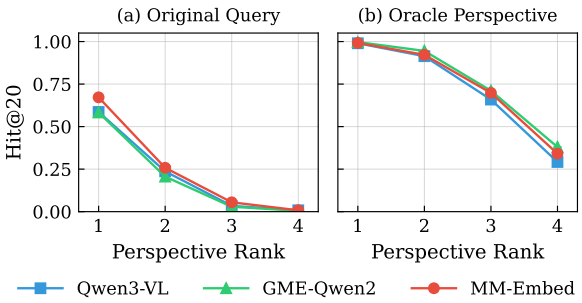


Figure 4: Preliminary analysis. Hit@20 is measured for each perspective and perspectives are ranked accordingly. (a) Naive retrieval using the original query. (b) Multi-query retrieval using ground-truth perspective descriptions, with results aggregated via Round Robin.

4 Approach

4.1 Preliminary Analysis

Open-ended queries often admit multiple perspectives, yet whether existing retrievers adequately capture such diversity remains unclear. We investigate this on our test set using three multimodal retrievers: MM-Embed-8B (Lin et al., 2024), GME-Qwen2-VL-7B-Instruct (Zhang et al., 2024b), and Qwen3-VL-Embedding-8B (Li et al., 2026).

Do retrievers cover multiple perspectives? We first examine perspective coverage (Figure 4(a)). Given the original query, we retrieve the top 20 documents and check whether at least one supporting document is retrieved for each perspective. We find that retrieval is heavily concentrated on a single dominant perspective, while the remaining ones receive substantially lower scores, revealing a *single-perspective bias*.

Where does the bottleneck originate? We next ask whether the bias stems from the query embedding failing to capture other perspectives or from poor alignment between the query and the document embeddings of their supporting documents (Figure 4(b)). When we instead use perspective descriptions as queries and aggregate the results via Round Robin, the previously missed documents become retrievable, yielding substantially more balanced performance across perspective ranks. This indicates that the bottleneck lies in the query embedding rather than the document space, motivating perspective-guided learning with multiple query vectors aligned to explicit perspective descriptions.

Algorithm 1 SPIN Optimization

Require: Frozen encoder M with L layers, training set $\mathcal{X}$, learnable noise vectors $\mathcal{E} = \{\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_m\}$, injection layer l, sigmoid temperature τ, bias b

for batch $\mathcal{B} = \{(q_n, \mathcal{P}_n)\}_{n=1}^{N} \subset \mathcal{X}$ **do** ▷ q_n: n-th query, $\mathcal{P}_n$: corresponding perspective descriptions
  for $n = 1, \ldots, N$ **do**
    $\mathbf{h}_l^n \leftarrow M_{0\to l}(q_n)$ ▷ $\mathbf{h}_l^n$: internal representation of q_n at layer l
    $\mathbf{S}_n \leftarrow \{M_{l\to L}(\mathbf{h}_l^n + \boldsymbol{\varepsilon}) \mid \boldsymbol{\varepsilon} \in \mathcal{E}\}$ ▷ $\mathbf{S}_n$: steered embeddings for q_n
    $\mathbf{T}_n \leftarrow \{M(p) \mid p \in \mathcal{P}_n\}$ ▷ $\mathbf{T}_n$: target embeddings for q_n
  end for
  $p_{\mathbf{s}}(\mathbf{t}) \leftarrow \sigma((\cos(\mathbf{s}, \mathbf{t}) - b)/\tau)$ ▷ $p_{\mathbf{s}}(\mathbf{t})$: probability that steered $\mathbf{s}$ covers target $\mathbf{t}$
  $\mathcal{L}_{\text{pos}} \leftarrow -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{|\mathbf{T}_n|}\sum_{\mathbf{t}\in\mathbf{T}_n}\log\Big(1 - \prod_{\mathbf{s}\in\mathbf{S}_n}\big(1 - p_{\mathbf{s}}(\mathbf{t})\big)\Big)$
  $\mathcal{L}_{\text{neg}} \leftarrow -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{|\mathbf{T}_{\neq n}|}\sum_{\mathbf{t}\in\mathbf{T}_{\neq n}}\sum_{\mathbf{s}\in\mathbf{S}_n}\log\big(1 - p_{\mathbf{s}}(\mathbf{t})\big)$ ▷ $\mathbf{T}_{\neq n}$: in-batch negative targets
  Minimize $\mathcal{L} = \mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{neg}}$
end for

4.2 Perspective-Guided Learning

To enable the retriever to interpret queries in diverse ways, we propose SPIN (Steering Perspectives by Injecting Noise), which requires only perspective descriptions and no document relevance annotations. Prior work (Skean et al., 2024, 2025) shows that intermediate transformer layers preserve richer semantic information, whereas the final layers converge toward a single fixed interpretation. Motivated by this, SPIN injects learnable noise vectors at an intermediate layer of a frozen retriever to steer its embeddings toward diverse perspectives.

Optimization process. We describe the optimization procedure in Algorithm 1. For each query q_n, we inject each learnable noise vector $\boldsymbol{\varepsilon} \in \mathcal{E}$ into the hidden representation of q_n at layer l and forward it through the remaining layers, yielding the steered embeddings $\mathbf{S}_n$. We then encode the perspective descriptions $\mathcal{P}_n$ with the same frozen encoder M into target embeddings $\mathbf{T}_n$, and optimize $\mathcal{E}$ so that $\mathbf{S}_n$ aligns with $\mathbf{T}_n$.

Since both $\mathbf{S}_n$ and $\mathbf{T}_n$ consist of multiple vectors, aligning the two sets requires a many-to-many optimization. Rather than imposing a fixed one-to-one assignment (e.g., Hungarian matching (Kuhn, 1955)), we formulate alignment as a probabilistic coverage problem. Specifically, each $\mathbf{s} \in \mathbf{S}_n$ independently produces a coverage probability for a target $\mathbf{t} \in \mathbf{T}_n$, and these probabilities are combined via a noisy-OR, so that $\mathbf{t}$ is considered covered if at least one $\mathbf{s}$ aligns with it. The positive loss $\mathcal{L}_{\text{pos}}$ encourages every $\mathbf{t} \in \mathbf{T}_n$ to be covered by *any* $\mathbf{s} \in \mathbf{S}_n$, whereas the negative loss $\mathcal{L}_{\text{neg}}$ requires *all* $\mathbf{s} \in \mathbf{S}_n$ to reject in-batch negatives from other queries, $\mathbf{T}_{\neq n}$.

Inference. After optimization, given a query q, we obtain m embeddings by injecting each optimized noise vector at layer l:

$$\mathbf{E}^q = \{R_{l\to L}\big(R_{0\to l}(q) + \varepsilon\big) \mid \varepsilon \in \mathcal{E}\}. \quad (2)$$

For each embedding $\mathbf{e} \in \mathbf{E}^q$, we independently retrieve a ranked list $\mathcal{R}_{\mathbf{e}}$ from the document corpus $\mathcal{C}$ via Maximum Inner Product Search:

$$\mathcal{R}_{\mathbf{e}} = \text{argsort}_{d\in\mathcal{C}}\ \mathbf{e}\cdot M(d). \quad (3)$$

Then, the m ranked lists are aggregated via Round Robin:

$$\mathcal{R}^q = \text{RoundRobin}(\mathcal{R}_{\mathbf{e}_1}, \mathcal{R}_{\mathbf{e}_2}, \ldots, \mathcal{R}_{\mathbf{e}_m}). \quad (4)$$

At each round $r = 1, 2, \ldots$, we iterate through $(\mathcal{R}_{\mathbf{e}_1}, \ldots, \mathcal{R}_{\mathbf{e}_m})$ in order and append the rank-r document of each list to $\mathcal{R}^q$, skipping duplicates. The procedure terminates once $|\mathcal{R}^q| = k$.

5 Experiments

5.1 Evaluation Metrics

We evaluate perspective coverage using two metrics. Hard coverage checks whether the retrieved documents include any annotated supporting document for each perspective, offering reproducibility and scalability. Since our annotations may miss relevant documents, soft coverage uses GPT-5-mini to assess whether each retrieved document supports a given perspective, recovering evidence that hard coverage would treat as a miss. These hard and soft coverage metrics are defined by

$$\text{HC@k} = \frac{1}{|P|}\sum_{i=1}^{|P|}\mathbf{1}[D^*(k) \cap D_i \neq \emptyset], \quad (5)$$

$$\text{SC@k} = \frac{1}{|P|}\sum_{i=1}^{|P|}\max_{d\in D^*(k)}\text{Judge}(p_i, d), \quad (6)$$

where $\text{Judge}(p_i, d) \in \{0, 1\}$ indicates whether p_i is fully supported by d. Specifically, we report HC at $k \in \{5, 10, 20, 100\}$ and SC at $k \in \{5, 10, 15, 20\}$, with SC limited to smaller k due to the API cost. See Appendix C.1 for evaluation details, including human agreement with the GPT-5-mini judge.

5.2 Baselines

We experiment with several retrieval diversification methods on three state-of-the-art multimodal retrievers: GME-Qwen2-VL-7B-Instruct (Zhang et al., 2024b), MM-Embed-8B (Lin et al., 2024), and Qwen3-VL-Embedding-8B (Li et al., 2026). For multi-vector retrieval methods ($m > 1$), results are aggregated via round robin. We use 72K instances as the training set and 32K as the validation set, and document embeddings remain frozen throughout. See Appendix C.3 for implementation details.

Naive. We evaluate the retrievers in both zero-shot and fine-tuned ways. For fine-tuning, we concatenate positive documents across all perspectives per query into a single positive set, and train the retriever with in-batch negatives using InfoNCE (Oord et al., 2018).

LLM-Expansion. We diversify the query space by prompting an LLM with the instruction "Generate {m} search queries related to: {user_query}" to obtain m distinct query reformulations. To adapt the model to our setting, we fine-tune Qwen3-4B (Team, 2025) to generate the annotated perspective descriptions corresponding to each query, where m is set to the number of annotated perspectives during training. At inference time, we fix $m = 5$ for consistency across queries.

Auto-Regressive Embedding (ARE). We implement the ARE method (Chen et al., 2025; Huo et al., 2026) in our setting, which generates multi-vector embeddings auto-regressively. During training, the retriever takes the query tokens followed by supporting document embeddings and produces one query embedding per position. The query embeddings are aligned with gold documents via Hungarian matching, and InfoNCE is applied with in-batch negatives. At inference, we fix $m = 5$ and perform m sequential forward passes. ARE cannot be applied to the encoder-based retriever MM-Embed, so we exclude it from the experiments.

SPIN. We inject m learnable noise vectors at an intermediate layer to steer final-layer embeddings, requiring only perspective descriptions without document relevance annotations. To account for different backbone depths, we set the injection point at $l = 0.5L$, corresponding to layers 16, 14, and 18 for MM-Embed ($L = 32$), GME-Qwen2 ($L = 28$), and Qwen3-VL ($L = 36$), respectively. See § 4.2 for details.

Oracle. We use oracle perspective descriptions as retrieval queries, serving as an upper bound for perspective-guided retrieval.

5.3 Results on the MULTI^{3}IR

We report the main results on MULTI^{3}IR in Table 2, where all baselines retrieve candidates from the full 1.01M gold document pool. We adopt HardCoverage@k (HC@k) and SoftCoverage@k (SC@k) as our primary evaluation metrics.

Current retrievers capture information from limited perspectives. We examine the zero-shot performance of naive retrievers on MULTI^{3}IR. They reach only up to 28.83 HC@10 and 39.52 SC@10, leaving a substantial gap to the oracle setting where the perspective descriptions are explicitly provided. Providing these perspectives boosts performance by 15.58 to 35.31 points on HC@10 and by 31.28 to 36.72 points on SC@10, suggesting that current retrievers struggle to identify perspectives that are only implicit in a query. The gap persists even at $k = 100$, with a difference of 19.45 to 35.60 points on HC@100, indicating that the failure is not one of retrieval depth but of query understanding.

How the retriever learns matters more than what it learns from. We compare methods within each level of supervision. Under document-level supervision, ARE improves over the fine-tuned naive retriever by 5.30 to 6.20 points on HC@10. Under perspective-level supervision, the gap widens: SPIN outperforms LLM-EXPANSION by 8.18 to 13.13 points on HC@10 and by 17.99 to 21.92 points on SC@10, despite using the same annotations and the same number of query embeddings. Notably, multiple query embeddings alone provide no benefit, as LLM-EXPANSION falls below the naive baseline on HC@10 while also using $m = 5$. Multi-vector retrieval helps only when the embeddings are explicitly trained to align with distinct perspectives, indicating that the architecture

Method	m	FT-Doc	FT-Persp	Hard Coverage (HC)				Soft Coverage (SC)			
				@5	@10	@20	@100	@5	@10	@15	@20
MM-Embed											
Naive	1	✗	✗	21.07	28.83	35.50	52.26	29.43	39.52	45.85	49.53
Naive	1	✓	✗	20.95	28.38	35.94	53.96	31.60	41.55	47.81	51.80
LLM-Expansion	5	✗	✓	15.63	22.11	30.11	47.37	27.16	37.13	43.25	47.32
SPIN	5	✗	✓	**25.75**	**35.24**	**41.24**	**59.17**	**49.22**	**59.05**	**64.31**	**67.54**
Oracle	4.6	✗	✗	34.00	44.41	53.79	71.71	57.55	70.80	76.31	79.29
GME-Qwen2											
Naive	1	✗	✗	16.18	21.81	27.92	42.08	26.59	35.40	40.58	44.86
Naive	1	✓	✗	17.69	24.20	31.68	50.84	26.28	36.19	41.66	46.38
LLM-Expansion	5	✗	✓	18.96	25.84	33.60	51.04	27.07	38.01	44.13	47.99
ARE	5	✓	✗	22.22	30.40	39.23	**59.35**	30.69	41.53	48.72	53.55
SPIN	5	✗	✓	**24.95**	**34.02**	**41.39**	58.62	**45.64**	**56.00**	**61.20**	**64.81**
Oracle	4.6	✗	✗	45.44	57.12	65.07	77.68	59.06	72.12	77.46	79.95
Qwen3-VL											
Naive	1	✗	✗	18.43	25.55	34.16	48.28	26.75	36.22	41.74	45.63
Naive	1	✓	✗	20.71	28.06	37.47	59.37	30.76	41.41	47.52	52.44
LLM-Expansion	5	✗	✓	18.44	25.33	33.11	50.95	27.06	37.75	43.80	48.31
ARE	5	✓	✗	24.75	33.36	42.71	61.60	33.50	44.73	51.37	56.01
SPIN	5	✗	✓	**26.44**	**37.83**	**45.55**	**63.44**	**49.62**	**59.35**	**64.47**	**67.47**
Oracle	4.6	✗	✗	42.96	52.31	60.83	73.76	58.18	70.04	75.31	78.14

Table 2: Results of perspective coverage (%) on the test split of Multi^{3}IR-Bench (1.01M candidates). *m* denotes the number of query embeddings. *FT-Doc* denotes fine-tuning with positive documents per query, and *FT-Persp* denotes fine-tuning with perspective descriptions per query. *Oracle* in gray background represents upper-bound performance for each retriever and is not included in the rankings.

and training objective, rather than the supervision signal alone, drive the improvement.

Perspective-guided learning boosts retrieval with lighter annotation effort. We compare ARE and SPIN, two multi-vector retrievers trained with document-level and perspective-level supervision, respectively. As shown in Table 2, SPIN outperforms ARE by 3.62 to 4.47 points on HC@10 and by 14.47 to 14.62 points on SC@10, demonstrating that perspective-level alignment is more effective than document-level supervision for training multi-vector retrievers. Perspective-level supervision is also substantially cheaper to obtain, as document relevance annotation dominates the overall synthesis cost (Appendix B.5). Perspective-guided learning is thus both more effective and lighter to annotate, offering a more scalable path toward training perspective-aware retrievers.

5.4 Results on Other Benchmarks

We report zero-shot retrieval performance on two unseen open-ended IR benchmarks, PIR (Zhao et al., 2024) and BeRDS (Chen and Choi, 2025), in Table 3. Since both provide perspective-level gold documents, we measure HC@k by strict document match, without a judge model. Each query is encoded from the original question alone, and the perspectives are never shown to the retriever. Further details on the benchmark subsets and corpora are in Appendix C.2.

SPIN improves over the naive retriever across all three backbones on both benchmarks. On PIR, the gains range from 9.37 to 11.73 points on HC@10 and remain substantial at k=100 (3.81 to 6.09 points). On BeRDS, where the naive retrievers already achieve high coverage, the gains are smaller for MM-EMBED and QWEN3-VL but reach 20.10 points on HC@10 for GME-QWEN2, which has the lowest naive coverage. The only exception is MM-EMBED at HC@100 on BeRDS, where both methods exceed 98%, leaving little room for improvement. Overall, these results show that perspective-aware steering learned on MULTI^{3}IR transfers to unseen benchmarks with different domains and corpora, without adaptation to the target distribution.

Method	m	FT-Doc	FT-Persp	PIR @5	PIR @10	PIR @20	PIR @100	BeRDS @5	BeRDS @10	BeRDS @20	BeRDS @100
MM-Embed											
Naive	1	✗	✗	51.12	59.98	69.29	87.41	79.10	88.85	94.95	**99.15**
SPIN	5	✗	✓	**63.33**	**70.25**	**80.55**	**93.50**	**83.45**	**90.20**	**96.80**	98.10
GME-Qwen2											
Naive	1	✗	✗	49.58	58.02	66.50	83.26	53.75	64.95	74.25	91.75
SPIN	5	✗	✓	**59.11**	**69.75**	**76.63**	**89.08**	**59.75**	**85.05**	**88.30**	**92.30**
Qwen3-VL											
Naive	1	✗	✗	51.06	60.10	69.96	87.16	75.05	84.30	89.90	96.15
SPIN	5	✗	✓	**55.21**	**69.47**	**78.36**	**90.97**	**76.20**	**86.90**	**92.40**	**96.80**

Table 3: Perspective coverage on the unseen benchmarks, PIR (Zhao et al., 2024) and BeRDS (Chen and Choi, 2025), measured by hard coverage (%). SPIN is fine-tuned on the training split of Multi^{3}IR.

	Number of Perspectives (m)				
Layer	1	3	5	7	9
Early ($l = 6$)	56.7	58.5	58.1	56.8	57.2
Mid ($l = 18$)	55.3	59.9	63.4	63.9	64.2
Late ($l = 30$)	62.1	62.8	61.0	60.5	60.7

Table 4: HC@100 score (%) across the number of noise vectors (m) for different noise injection layers (L).

Method	Layer	# Params	HC@100
Perturbation-based			
SPIN (m=5)	L0	20.5K	34.3
SPIN (m=5)	L18	20.5K	**63.4**
Prefix-tuning (k=10)	L0	204.8K	48.1
Prefix-tuning (k=10)	L18	204.8K	54.1
Reparameterization-based			
LoRA (r=16)	L0–36	209.7M	54.2
LoRA (r=16)	L18–36	104.9M	55.9
Adapter (r=64)	L0–36	83.9M	53.9
Adapter (r=64)	L18–36	41.9M	55.9

Table 5: Perspective coverage (%) comparison across steering methods and injection layers.

6 Analysis

All experiments in this section are conducted with Qwen3-VL-Embedding-8B (Li et al., 2026), which has 36 layers in total. Implementation details, including the training configuration and the retrieval instructions, follow Section 5.3.

6.1 Effect of Injection Layer

A central mechanism of SPIN is to inject noise into an intermediate layer of the retriever. As shown in Table 4, injecting at the mid layer (L=18) yields consistent improvements as m increases. In contrast, late-layer injection (L=30) plateaus as m increases, likely because the representations at deeper layers have already collapsed toward a single direction. These results suggest that intermediate-layer injection is essential for SPIN to fully benefit from increasing m.

6.2 Comparison with Existing Adaptation Methods

We isolate the methodological contribution of SPIN by comparing it with representative adaptation methods: perturbation-based methods such as prefix-tuning (Li and Liang, 2021), and weight-based methods such as LoRA (Hu et al., 2021) and adapter (Houlsby et al., 2019).

Adaptation method. At the intermediate layer, SPIN achieves the best HC@100 (63.4) with only 20.5K parameters, outperforming LoRA and adapter by 7.5 points while using over three orders of magnitude fewer parameters. This indicates that learned additive vectors steer the query representation more effectively than reparameterizing the backbone weights.

Injection layer. We next examine how the injection layer affects each method. As shown in Table 5, perturbation-based methods benefit substantially from intermediate-layer injection (SPIN +29.1, soft prompt +6.0), whereas reparameterization-based methods gain only marginally (+1.7 and +2.0) despite halving their parameters, as the frozen lower layers already provide strong representations.

Method	@5	@10	@20	@100
Modality coverage				
Naive	26.9	32.4	38.9	48.9
SPIN	40.2	53.0	59.7	78.2
Oracle	50.5	56.3	64.6	81.1
Domain coverage				
Naive	18.7	27.3	37.7	53.4
SPIN	30.6	42.7	50.2	68.3
Oracle	46.0	61.7	73.1	84.8

Table 6: Knowledge source coverage (%). For each query, we measure the fraction of distinct modalities or domains in its gold documents that are covered by the top-k retrieved results.

	Cosine Distance Quartile			
Method	**<25%**	**25–50%**	**50–75%**	**>75%**
Naive	69.6	50.0	44.4	29.2
SPIN, m=1	73.2 (+3.6)	60.0 (+10.0)	49.7 (+5.3)	38.0 (+8.8)
SPIN, m=3	74.6 (+5.0)	63.8 (+13.8)	54.9 (+10.5)	46.7 (+17.5)
SPIN, m=5	78.3 (+8.7)	67.4 (+17.4)	58.8 (+14.4)	53.6 (+24.4)

Table 7: HC@100 score (%) across four quartiles of queries, grouped by how dispersed their positive documents are in the embedding space (average pairwise cosine distance). Parenthesized values denote gains over the Naive baseline.

6.3 Knowledge Source Coverage

To quantify how well SPIN retrieves information across diverse knowledge sources spanning modalities and domains, we compare it with a zero-shot naive baseline and an oracle upper bound. As shown in Table 6, SPIN nearly saturates the oracle on modality coverage, reaching 78.2 versus 81.1 at k=100 and closing 91% of the naive-to-oracle gap. In contrast, SPIN closes only 35–47% of the naive-to-oracle gap for domain coverage, and the shortfall persists as k grows (68.3 versus 84.8 at k=100). Learning domain-aware steering vectors is a promising direction for closing this gap.

6.4 Effectiveness of SPIN on Clustered Positives

Prior work identifies a limitation of dense retrievers: a single query vector may not be close to all relevant documents when they are scattered across the embedding space (Chen et al., 2025; Weller et al., 2025). To examine whether SPIN is effective for such dispersed queries, we analyze performance by document dispersion. As shown in Table 7, SPIN yields its largest gains on dispersed queries (+24.4 HC@100 at m=5 for the top quartile), while also improving substantially on tightly clustered queries (+8.7).

7 Conclusion

In this work, we introduce MULTI^{3}IR, an open-ended IR benchmark of 104.9K queries from Stack Exchange, annotated with perspective descriptions and supporting documents. We further propose SPIN, a parameter- and label-efficient method that learns lightweight steering vectors to diversify a frozen retriever's output into multiple perspective-aware embeddings, using only perspective descriptions. Our experiments show that strong multi-modal retrievers suffer from a single-perspective bias, collapsing onto one dominant viewpoint. SPIN substantially mitigates this bias, outperforms document-supervised baselines with lighter annotation, and generalizes to external benchmarks. Further analysis reveals that the bias persists even when relevant documents are tightly clustered in the embedding space, and that injecting steering vectors at an intermediate layer is critical to SPIN's effectiveness.

Limitations

We acknowledge several potential limitations of our work. (1) The number of perspectives can vary substantially across queries, while our method uses a fixed number of perspective vectors m. Developing an adaptive mechanism that selects m based on each query is a promising direction for future work. (2) While SPIN reduces the need for document-level annotations, jointly leveraging perspective descriptions and their supporting documents during training could further improve perspective-specific retrieval. (3) Finally, MULTI^{3}IR is constructed through an automatic pipeline involving perspective extraction, document retrieval, and exclusive-support verification. Although we validate this pipeline both quantitatively and qualitatively, LLM-induced biases may not be fully eliminated and could affect the absolute scores reported in our evaluations.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grants funded by the Korea government (MSIT) (No. RS-2025-02263841, Development of a Real-time Multi-

modal Framework for Comprehensive Deepfake Detection Incorporating Common Sense Error Analysis; No. RS-2019-II191082, SW StarLab; No. RS-2026-25524173, Ultra-Long-Term Hierarchical Memory and Reasoning Architecture for Next-Generation Omnimodal Agents; No. RS-2025-25442338, SNU AI Star Fellowship Support Program; and No. RS-2021-II211343, SNU Artificial Intelligence Graduate School Program). Gunhee Kim is the corresponding author.

References

Essential AI, :, Andrew Hojel, Michael Pust, Tim Romanski, Yash Vanjani, Ritvik Kapila, Mohit Parmar, Adarsh Chaluvaraju, Alok Tripathy, Anil Thomas, Ashish Tanwer, Darsh J Shah, Ishaan Shah, Karl Stratos, Khoi Nguyen, Kurt Smith, Michael Callahan, Peter Rushton, Philip Monk, Platon Mazarakis, Saad Jamal, Saurabh Srivastava, Somanshu Singla, and Ashish Vaswani. 2025. Essential-web v1.0: 24t tokens of organized web data.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Yingshan Chang, Mridu Narang, Hisami Suzuki, Guihong Cao, Jianfeng Gao, and Yonatan Bisk. 2022. Webqa: Multihop and multimodal qa. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16495–16504.

Hung-Ting Chen and Eunsol Choi. 2025. Open-world evaluation for retrieving diverse perspectives. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8508–8528.

Hung-Ting Chen, Xiang Liu, Shauli Ravfogel, and Eunsol Choi. 2025. Beyond single embeddings: Capturing diverse targets with multi-query retrieval. *arXiv preprint arXiv:2511.02770*.

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. 2023. Can pre-trained vision and language models answer visual information-seeking questions? *arXiv preprint arXiv:2302.11713*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

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

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Transformer architectures and pre-training strategies for fast and accurate multi-sentence scoring. *arXiv preprint arXiv:1905.01969*.

Jiahao Huo, Yu Huang, Yibo Yan, Ye Pan, Kening Zheng, Wei-Chieh Huang, Yi Cao, Mingdong Ou, Philip S Yu, and Xuming Hu. 2026. Causalembed: Auto-regressive multi-vector generation in latent space for visual document embedding. *arXiv preprint arXiv:2601.21262*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Harold W. Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends® in Machine Learning*, 5(2-3):123–286.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Chen Keqin, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2026. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: Long papers)*, pages 4582–4597.

Sheng-Chieh Lin, Chankyu Lee, Mohammad Shoeybi, Jimmy Lin, Bryan Catanzaro, and Wei Ping. 2024. Mm-embed: Universal multimodal retrieval with multimodal llms. *arXiv preprint arXiv:2411.02571*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 5783–5797.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 3982–3992.

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. 2024. Does representation matter? exploring intermediate layers in large language models. *arXiv preprint arXiv:2412.09563*.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039*.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. 2024. Uniir: Training and benchmarking universal multimodal information retrievers. In *European Conference on Computer Vision*, pages 387–404. Springer.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. 2025. On the theoretical limitations of embedding-based retrieval. *arXiv preprint arXiv:2508.21038*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Le Zhang, Yihong Wu, Qian Yang, and Jian-Yun Nie. 2024a. Exploring the best practices of query expansion with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1872–1883.

Siyue Zhang, Yuan Gao, Xiao Zhou, Yilun Zhao, Tingyu Song, Arman Cohan, Anh Tuan Luu, and Chen Zhao. 2025. Mrmr: A realistic and expert-level multidisciplinary benchmark for reasoning-intensive multimodal retrieval. *arXiv preprint arXiv:2510.09510*.

Xin Zhang, Yanzhao Zhang, Wen Xie, Mingxin Li, Ziqi Dai, Dingkun Long, Pengjun Xie, Meishan Zhang, Wenjie Li, and Min Zhang. 2024b. Gme: Improving universal multimodal retrieval by multimodal llms. *arXiv preprint arXiv:2412.16855*.

Xinran Zhao, Tong Chen, Sihao Chen, Hongming Zhang, and Tongshuang Wu. 2024. Beyond relevance: Evaluate and improve retrievers on perspective awareness. *arXiv preprint arXiv:2405.02714*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37.

A Subject Domain Classification

We annotate each document's subject domain using the domain classifier[1], a text classification model that assigns a label from a two-level taxonomy of 10 top-level classes, each divided into 10 subclasses, for 100 domains in total (Table 8). Since the classifier accepts only text input, image documents require a textual surrogate. For WebQA, we use the caption paired with each image; for MultimodalQA, the title of the Wikipedia page from which the image is drawn; and for M-BEIR and MRMR, a caption generated by an image captioning model[2]. Since every document in Multi^{3}IR originates from a web source, we crawl the source page for each image and use its textual content. Each input is truncated to the model's 512-token context window, and the domain with the highest predicted probability is assigned.

B Data Collection Details

B.1 Raw Data Collection

We use the official StackExchange data dumps hosted on archive.org[3]. Although the dataset is publicly available, it may contain user-generated content such as usernames, posts, and comments that could include personally identifying or offensive information. The authors checked all examples and found no personally identifying information (PII).

We select this source for three reasons: (1) the multi-answer structure provides natural multi-perspective supervision, (2) community voting and accept-tagging serve as built-in quality control, and (3) external links cited in answers serve as a natural source of perspective-grounded evidence.

We collect data from 77 sites organized into 5 broad categories: *technology*, *culture & recreation*, *science*, *business*, and *life & arts*. Some sites appear in multiple categories under the multi-category specification (e.g., `graphic_design` belongs to both *technology* and *life & arts*). For each site, we retain a question only if it has more than three answers and receives votes from more than three distinct users, ensuring both quantity and diversity of viewpoints. For each retained post, the question Q is formed by concatenating the title and body, and the answer set A is the aggregation of all user responses. To avoid over-representation of dominant categories such as *technology*, we then apply category-level uniform sampling with a per-category cap. Per-category counts at each stage with representative sites are reported in Table 9.

B.2 Perspective Extraction and Verification

We extract perspectives using GPT-5-mini (gpt-5-mini-2025-08-07) via the OpenAI Batch API; the prompt and an example are shown in Figure 5. For uniqueness, we encode perspective summaries with the `sentence-transformers` library[4] and iteratively remove a perspective whose cosine similarity to any previously kept perspective exceeds the threshold; the threshold of 0.8 was chosen through manual inspection to best separate near-duplicates from genuinely distinct perspectives. For faithfulness, we concatenate the entire answer set into a single document and use it as the supporting document, and run the entailment check via the `minicheck` library[5] on a vLLM backend[6] with prefix caching. The per-category distribution after both filters is reported in Table 10.

B.3 Document Retrieval and Verification

We use Qwen3-Embedding-4B[7] served via vLLM to encode both the query and the C4[8] English documents with a 4,096-token context window. Queries are prefixed with "`Instruct: Given a web search query, retrieve relevant passages that answer the query\nQuery:{query}`", while documents are encoded without any prefix. Image candidates are obtained from the Google Images endpoint of the Serper API[9].

To verify whether each document exclusively supports its assigned perspective, we prompt a VLM using the template in Figure 6. We evaluate three models under two criteria: (i) *fully-only*, which accepts only *fully support* predictions, and (ii) *fully+partially*, which also accepts *partially support*. The former shows higher agreement with human judgments, while Qwen3-VL-30B-A3B-Instruct[10] achieves the best precision and recall.

[1] https://huggingface.co/EssentialAI/eai-distill-0.5b
[2] https://huggingface.co/Salesforce/blip-image-captioning-large
[3] https://archive.org/details/stackexchange
[4] https://www.sbert.net/
[5] https://github.com/Liyan06/MiniCheck
[6] https://github.com/vllm-project/vllm
[7] https://huggingface.co/Qwen/Qwen3-Embedding-4B
[8] https://huggingface.co/datasets/allenai/c4
[9] https://serper.dev/
[10] https://huggingface.co/Qwen/Qwen3-VL-30B-A3B-Instruct

Perspective Extraction Prompt

Given a question and a list of answers, extract the distinct perspectives (facets) covered in the answers as independent sub-questions. Each sub-question should be:

1. **Self-contained**: Understandable without the original question.
2. **Search-friendly**: Suitable as a query to a search engine to find supporting evidence.
3. **Specific**: Targeting exactly one perspective or facet from the answers.
4. **Non-redundant**: Each sub-question should cover a distinct aspect.

Output format (one <perspective> block per distinct viewpoint)

```
<perspective>
{Search-friendly description of the perspective}
  - {Supporting fact or quote}
  - ...
</perspective>
```

Example input

```
<question>
Why does my character walk forward automatically? ...
</question>

A1: Minecraft lost the "key W was released" event, so you're still
    going forwards. ...
A2: One cause is pressing a movement key + a GUI key simultaneously
    -- a "poor-man's autorun" that keeps you walking after closing
    the inventory. ...
A3: A known bug when moving and clicking at the same time.
    Workaround: update the LWJGL library. ...
```

Example output

```
<perspective>
Could Minecraft lose a "key released" event so the game keeps
    thinking W is held down?
  - Minecraft lost the "key W was released" event.
</perspective>
<perspective>
Can pressing a movement key and a GUI key simultaneously cause
    persistent forward walking?
  - A "poor-man's autorun" that keeps you walking after closing the
      inventory.
</perspective>
<perspective>
Is there a known LWJGL-related bug, and is updating LWJGL a
    documented workaround?
  - Known bug; workaround: update the LWJGL library.
</perspective>
```

Figure 5: Prompt template for perspective extraction. Blue text denotes instance-specific input, and violet text denotes the model's expected output.

Top-level class	Sub-class
Computer science	Computer science, knowledge & systems; Bibliography; Library & information sciences; Encyclopedias & books of facts; Magazines, journals & serials; Associations, organizations & museums; News media, journalism & publishing; Quotations; Manuscripts & rare books
Philosophy & psychology	Philosophy; Metaphysics; Epistemology; Parapsychology & occultism; Philosophical schools of thought; Psychology; Logic; Ethics; Ancient, medieval & eastern philosophy; Modern western philosophy
Religion	Religion; Philosophy & theory of religion; The Bible; Christianity & Christian theology; Christian practice & observance; Christian pastoral practice & religious orders; Christian organization, social work & worship; History of Christianity; Christian denominations; Other religions
Social sciences	Social sciences, sociology & anthropology; Statistics; Political science; Economics; Law; Public administration & military science; Social problems & social services; Education; Commerce, communications & transportation; Customs, etiquette & folklore
Language	Language; Linguistics; English & Old English languages; German & related languages; French & related languages; Italian, Romanian & related languages; Spanish & Portuguese languages; Latin & Italic languages; Classical & modern Greek languages; Other languages
Science	Science; Mathematics; Astronomy; Physics; Chemistry; Earth sciences & geology; Fossils & prehistoric life; Life sciences; biology; Plants (botany); Animals (zoology)
Technology	Technology; Medicine & health; Engineering; Agriculture; Home & family management; Management & public relations; Chemical engineering; Manufacturing; Manufacture for specific uses; Building & construction
Arts & recreation	Arts; Landscaping & area planning; Architecture; Sculpture, ceramics & metalwork; Drawing & decorative arts; Painting; Graphic arts; Photography & computer art; Music; Sports, games & entertainment
Literature	Literature, rhetoric & criticism; American literature in English; English & Old English literatures; German & related literatures; French & related literatures; Italian, Romanian & related literatures; Spanish & Portuguese literatures; Latin & Italic literatures; Classical & modern Greek literatures; Other literatures
History & geography	History; Geography & travel; Biography & genealogy; History of the ancient world; History of Europe; History of Asia; History of Africa; History of North America; History of South America; History of other areas

Table 8: Two-level subject domain taxonomy: 10 top-level classes, each divided into 10 sub-classes. Domains are labeled at the sub-class level.

Category	Example Sites	# Sites	Raw	After Quality Filter	After Sampling
technology	Stack Overflow, Super User, Unix & Linux, Database Administrators	28	26,777,541	3,545,885	95,121
culture & recreation	English Language, Travel, Movies & TV, Board Games	21	626,683	149,296	95,121
science	Mathematics, Physics, Chemistry, Biology	12	2,061,630	247,023	95,121
business	Personal Finance & Money, Workplace, Project Management	8	151,066	36,831	36,831
life & arts	Home Improvement, Gardening, Fitness, Parenting	8	168,113	31,707	31,707
Total	—	77	29,785,033	4,010,742	353,901

Table 9: Per-category counts of StackExchange questions at each stage of raw data collection: raw question count, after quality filtering (more than three answers and more than three distinct voters), and after category-level uniform sampling.

B.4 Human Annotation Procedure

We hire skilled raters on Amazon Mechanical Turk (MTurk) to verify the quality of extracted perspectives and the exclusivity of their paired documents. The verification proceeds in two stages: a *perspective evaluation* stage (Q1a/Q1b/Q1c) and a *document–perspective support* stage.

Annotators were selected based on their suc-

Category	Questions	Perspectives	Avg. per Q
culture & recreation	35,039	179,539	5.12
science	20,465	95,937	4.69
business	19,165	99,408	5.19
technology	16,085	74,213	4.61
life & arts	14,064	72,189	5.13
professional	98	453	4.62
Total	104,916	521,739	4.97

Table 10: Per-category distribution of retained questions and perspectives after extraction and filtering (train and evaluation splits combined).

Model	Criterion	Agr.	Prec.	Rec.	Pass
Qwen3-VL-Instruct	fully only	0.89	0.93	0.62	0.14
	fully + partially	0.77	0.78	0.79	0.35
Qwen3-VL-Thinking	fully only	0.89	0.95	0.48	0.10
	fully + partially	0.79	0.85	0.76	0.33
GPT-5-mini	fully only	0.90	0.92	0.38	0.08
	fully + partially	0.54	0.70	0.93	0.71

Table 11: Document verifier performance across models and criteria, evaluated on 300 documents (150 text and 150 image) with human annotations as ground truth. Agr., Prec., and Rec. denote agreement, precision, and recall, respectively.

cess in qualification HITs that assessed their ability to judge whether a perspective is meaningful and whether two perspectives express the same underlying idea. This qualification task consisted of 3 verified examples with 5 sub-questions each (15 sub-questions in total), with a payment of $2.00. We required annotators to be from English-speaking countries, have completed more than 10,000 HITS, and maintain a HIT approval rate greater than 95%.

After qualification, annotators received detailed instructions and proceeded to the perspective evaluation HIT. For each perspective p_i in a question with $4 \leq N \leq 6$ extracted perspectives, they answered three sub-questions: (Q1a) whether p_i is a meaningful perspective on the question (yes/no); (Q1b) which of the remaining $N-1$ perspectives is closest in meaning to p_i; and (Q1c) whether the perspective selected in Q1b is redundant with p_i, i.e., a paraphrase or duplicate (yes/no). Full annotator instructions appear in Figure 7. On an overlap subset of 30 questions per category, perspectives received unanimous *yes* for Q1a (meaningful, 99.1%) and *no* for Q1c (unique, 96.2%) across all domains. Q1b, an $(N-1)$-way classification harder than binary judgment, still reaches substantive agreement: $\kappa = 0.692$ (science) and 0.712 (tech) for two-annotator domains, and $\kappa = 0.739$ (business), 0.740 (culture), and 0.734 (life & arts) for three-annotator domains. The remaining 960 questions per category were annotated by a single annotator.

Stage	# Samples	Drop rate (%)
Document candidate	8,254,298	–
(1) Own-perspective support	1,503,151	81.8
(2) Sibling exclusivity	1,012,061	32.7

Table 12: Document verification result with Qwen3-VL-Instruct. Drop rates are relative to the previous stage. Stage 1 requires *full* support of the pair's own perspective; Stage 2 excludes documents also fully supporting a sibling perspective.

Metric	N	Result	Fleiss' κ
Perspective relevance	5.0K	0.99	0.72
Document exclusive-support	9.1K	0.90	0.59 / 0.72
Answer coverage	100	0.99	0.81
Perspective faithfulness	100	0.98	0.88

Table 13: Human verification of the automated pipeline. For document exclusive-support, Fleiss' κ is reported as 3-class / binary.

Next, we verify document–perspective support. For each retained perspective p_i, we identify its closest counterpart p_j from Q1b, and each retrieved document originally paired with p_i is judged on a 3-point scale (*fully supports*, *partially supports*, *does not support*) against both p_i and p_j. A document is retained only if it fully supports p_i but does not fully support p_j, ensuring exclusivity; the perspective is discarded if no supporting document remains. On an overlap subset of 30 questions per category, Fleiss' $\kappa = 0.588$.

B.5 Annotation Cost

We compare the annotation cost incurred at each stage of data synthesis. As shown in Tables 14 and 15, document relevance annotation dominates the overall cost, requiring 13.6× more local compute and 2.5× higher API cost than perspective-only supervision. Since SPIN requires only perspective-level supervision, it avoids this dominant cost component; our label-efficiency claim is therefore relative to methods that require document-level supervision.

C Experimental Details

C.1 Evaluation Setup

For the model-based judge, we use GPT-5-mini (gpt-5-mini-2025-08-07) via the OpenAI Chat

Stage	Component	Compute (PFLOPs)	API Cost ($)
Persp.	Extraction	—	0.238
	Verification	12.24	—
	Subtotal	**12.24**	**0.238**
Doc.	Retrieval	104.64	0.364
	Verification	49.73	—
	Subtotal	**154.37**	**0.364**

Table 14: Stage-wise breakdown of supervision cost.

Level	Included stage	Compute (PFLOPs)	API Cost ($)
Persp.	Persp.	12.24	0.238
Doc.	Persp. + Doc.	166.61	0.602
Ratio		**13.6**×	**2.5**×

Table 15: Overall cost by supervision level.

Completions API[11] with reasoning_effort=medium and max_completion_tokens=2000. The judge is given a single perspective and one retrieved document at a time, and produces a chain-of-thought followed by a verdict of *fully*, *partially*, or *not supported*; only *fully* is counted as positive. The judge prompt is shown in Figure 6.

To assess the reliability of the judge, we further measure its agreement with human annotators on a held-out sample of 100 perspective-document pairs, balanced across domains and modalities. We binarize the ratings into *fully supported* and *not supported*, where the latter includes both *partially* and *not supported*. Human annotators achieve substantial inter-annotator agreement (Fleiss' $\kappa = 0.743$), while the GPT-5-mini judge shows even stronger agreement with the human majority vote (Fleiss' $\kappa = 0.840$). These results indicate that the judge's decisions closely align with human judgments.

C.2 Details on Other Benchmarks

PIR. We use all six sub-datasets of PIR (Zhao et al., 2024), comprising 4,187 questions and 10,019 perspectives. Each question is retrieved against its own corpus, and every perspective carries its own relevance judgments, so no subset selection is needed.

BeRDS. BeRDS (Chen and Choi, 2025) is designed to be scored by a judge model over retrieved passages rather than against gold documents, and only its ArguAna subset provides perspective-level document annotations. The Kialo and OpinionQA subsets define perspectives but leave them unannotated, so we restrict our evaluation to ArguAna (1,000 questions, 2,000 perspectives). Because the ArguAna document pool contains only 1,891 unique documents, we add the 355,123-passage Wikipedia corpus released with BeRDS as distractors.

C.3 Implementation Details

C.3.1 Base retriever.

All methods for fine-tuning retrievers share the same instruction prompt, "Retrieve images or text relevant to the user's query." All resulting embeddings are L2-normalized. Document embeddings are precomputed once and remain frozen. The main results are reported from a single run.

For GME-Qwen2-VL[12] and Qwen3-VL-Embedding[13], the instruction and the query are wrapped with the Qwen2-VL chat template[14], producing

```
<|im_start|>system\n{instruction}<|im_end|>
<|im_start|>user\n{query}<|im_end|>
<|im_start|>assistant\n<|endoftext|>
```

Both are causal decoders, and we take the hidden state at the last token position as the query embedding.

For MM-Embed[15], which uses bidirectional attention, we instead concatenate the instruction and the query as "`Instruct: {instruction}\nQuery: {query}`" following its original convention, and obtain the query embedding via the model's latent-attention pooling head, which aggregates the full sequence of hidden states through a learned attention layer.

C.3.2 Fine-tuning

All fine-tuned methods are trained on the same 72K queries sampled from MULTI^{3}IR with AdamW (Loshchilov and Hutter, 2017) using a cosine schedule with 3% warmup. Method-specific details are described below.

Naive (fine-tuned). We fully fine-tune the retriever using InfoNCE with in-batch negatives (temperature τ=0.05), with a learning rate of 1e$-$5 and a batch size of 32 for 3 epochs. The maximum query length is set to 512 tokens.

[11] https://platform.openai.com/docs/api-reference/chat

[12] https://huggingface.co/Alibaba-NLP/gme-Qwen2-VL-7B-Instruct

[13] https://huggingface.co/Qwen/Qwen3-Embedding-8B

[14] https://huggingface.co/Qwen/Qwen2-VL-7B-Instruct/blob/main/chat_template.json

[15] https://huggingface.co/nvidia/MM-Embed

LLM-Expansion. We fully fine-tune Qwen3-4B[16] on (query, perspective description) pairs with a learning rate of $2\mathrm{e}{-5}$ for 2 epochs using a cosine schedule with 3% warmup. The effective batch size is 16 (batch size 4 with 4 gradient accumulation steps). At inference, we generate $m{=}5$ query reformulations per query using HuggingFace `transformers`[17] with `AutoModelForCausalLM.generate`, with *temperature*$= 0.7$, *top-p* $= 0.8$, *top-k* $= 20$, and a maximum of 384 tokens.

ARE. We fine-tune the retriever with LoRA ($r{=}16$) and an additional input projection layer $\mathbf{W}_{\text{in}} \in \mathbb{R}^{H\times H}$ ($H{=}3584$ for GME-Qwen2-VL, $H{=}4096$ for Qwen3-VL-Embedding; Xavier-initialized, without bias) that maps document embeddings into the language-model input space, using a learning rate of $3\mathrm{e}{-5}$, an effective batch size of 32. Predicted query embeddings are aligned to gold document embeddings via Hungarian matching, and the matched pairs are trained with InfoNCE over in-batch negatives with temperature $\tau{=}0.01$ for Qwen3-VL-Embedding and $\tau{=}0.03$ for GME-Qwen2-VL.

SPIN. We optimize m noise vectors $\boldsymbol{\epsilon} \in \mathbb{R}^{m\times H}$ injected into the mid layer (layer 18 of 36 for MM-Embed and Qwen3-VL-Embedding, layer 14 of 28 for GME-Qwen2-VL), with all encoder weights frozen. We set $m{=}5$, yielding 20,480 trainable parameters for MM-Embed and Qwen3-VL-Embedding ($H{=}4096$) and 17,920 for GME-Qwen2-VL ($H{=}3584$). The noise vectors are initialized from $\mathcal{N}(0, 0.01^2)$. Learning rates are tuned per backbone: $3\mathrm{e}{-4}$ for MM-Embed, $1\mathrm{e}{-2}$ for Qwen3-VL-Embedding, and $1\mathrm{e}{-3}$ for GME-Qwen2-VL. We use a batch size of 32. The training objective is the noisy-OR loss described in § 4.2, where each similarity is converted into a probability via $p(d) = \sigma((\mathrm{sim} - b)/\tau)$ with $\tau{=}0.1$ and $b{=}0.5$ fixed across all backbones.

C.3.3 Computational Resources

All experiments are run on NVIDIA A6000 48GB GPUs (Intel Xeon Silver 4210R CPU). Naive (fine-tuned) is trained with 4 GPUs using PyTorch Fully Sharded Data Parallel (FSDP)[18], and LLM-Expansion is trained with 4 GPUs using the TRL SFTTrainer[19] (which wraps HuggingFace Trainer's Distributed Data Parallel). ARE, SPIN, and zero-shot evaluation are run on a single GPU.

D Usage of GenAI

AI-assisted coding tools were used to support data analysis and visualization, including drafting and debugging scripts for figure/table generation. All code and results were subsequently reviewed and validated by the authors.

[16] https://huggingface.co/Qwen/Qwen3-4B
[17] https://github.com/huggingface/transformers
[18] https://pytorch.org/docs/stable/fsdp.html
[19] https://huggingface.co/docs/trl/sft_trainer

Perspective–Document Supportedness Judge Prompt

You are judging whether a document supports answering a specific perspective.
Follow this reasoning process:

1. Identify what the perspective specifically asks for.
2. Describe what the document actually contains — only what is literally visible. Do NOT assume or infer content.
3. Compare the two and give a verdict.

Verdicts:

- **fully**: The document explicitly contains the specific answer the perspective asks for.
- **partially**: The document has related information but does not fully answer the specific question.
- **not**: The document is only topically related or unrelated, with no specific answer.

Output constraints:

- Combine steps 1–3 into one paragraph inside `<reason>` tags.
- The verdict must be exactly one of: `fully`, `partially`, or `not`, inside `<verdict>` tags.

In-context examples

```
Example 1.
<reason>The perspective asks for a method to spatially join point
    data to polygon attributes in QGIS. The document shows the QGIS
    "Join Attributes by Location" dialog [...]. This is the exact
    tool needed to perform the spatial join.</reason>
<verdict>fully</verdict>

Example 2. ...

Example 3. ...

Example 4. ...
```

Input format

```
<document>{document}</document>
<perspective>{perspective}</perspective>
```

Output format

```
<reason>your reasoning combining steps 1-3 in one paragraph</reason>
<verdict>fully|partially|not</verdict>
```

Figure 6: Prompt and in-context examples used by the perspective–document supportedness judge. Examples 2–4 are abbreviated.

INSTRUCTIONS

Welcome. In this HIT, you will evaluate the quality of perspectives extracted for each question. Each item consists of:

- **A question:** The user's query.
- **Multiple perspectives (P1, P2, ..., PN):** Different viewpoints or angles on the question. N is typically between 4 and 6.

Your task: For **each** perspective P_i, answer three sub-questions:

- **Q1a:** Is P_i a well-grounded, meaningful perspective? (Yes/No)
- **Q1b:** Among the *other* perspectives, which one is closest in meaning to P_i?
- **Q1c:** Is the perspective you picked in Q1b actually redundant with P_i (i.e., a paraphrase or duplicate)? (Yes/No)

Note: You will repeat these three sub-questions for every perspective in the item. If the item has N perspectives, you will answer 3N sub-questions in total.

TASK

Question

Question: ${QUESTION_TEXT}

${QUESTION_CONTEXT}

Perspectives (${NUM_PERSPECTIVES})

${PERSPECTIVES_LIST}

Evaluate each perspective

${Q1_BLOCKS}

Figure 7: Instructions provided to Amazon Mechanical Turk raters for human annotation of perspective quality and redundancy.

INSTRUCTIONS

Welcome. In this HIT, you will evaluate how well a document **supports** two perspectives that were extracted for an open-ended question. Each item shows:

- **A question** (the original user post).
- **Two perspectives**, $\mathbf{P_i}$ and $\mathbf{P_j}$, that are about the question. P_i and P_j are close in meaning but not the same.
- **One document** retrieved from the web. The document may be a text snippet or an image.

Rate how well the document supports each perspective on a 3-point scale:

- **Fully supports**: the document clearly provides evidence, explanation, or argumentation directly relevant to the perspective.
- **Partially supports**: the document is relevant but only addresses the perspective in passing or only partly.
- **Does not support**: the document is off-topic or does not actually address the perspective.

You will give **2 ratings per item** in total — rate the document against P_i and against P_j.

TASK

Question

Question: ${QUESTION_TEXT}

${QUESTION_CONTEXT}

Perspectives

P_i ${PERSPECTIVE_I_TEXT}

P_j ${PERSPECTIVE_J_TEXT}

Document

${DOC_CARD}

P_i How well does the **document** support $\mathbf{P_i}$?

- Fully supports
- Partially supports
- Does not support

P_j How well does the **document** support $\mathbf{P_j}$?

- Fully supports
- Partially supports
- Does not support

Figure 8: Instructions provided to Amazon Mechanical Turk raters for human annotation of perspective–document supportedness.