SEMANTICALLY SIMILAR, YET NOT ANSWERABLE: DIAGNOSING THE SEMANTIC-ANSWERABILITY GAP IN TABLE RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

In retrieval-augmented generation (RAG), **semantic relevance** asks whether a source matches a query in meaning, while **answerability** asks whether it contains sufficient information to answer the query. A **Semantic-Answerability Gap (SAG)** may arise in retrieval when a retriever can reach semantically relevant sources yet fail to identify those that are uniquely answerable. We uncover this gap using tables as a controlled setting, where shared schemas and entities provide strong semantic signals while localized content and row-column bindings distinguish answerable from non-answerable sources. Using **TCR-Bench**, a controlled sibling-table benchmark, we find that dense retrievers achieve only **18.2%** top-1 target retrieval, reducing QA F1 from **0.755** with the oracle table to **0.330** with retrieved top-5 tables. Controlled diagnostics show that retrievers favor semantic volume over sufficiency, respond weakly to row-column binding disruptions, and struggle to distinguish Targets from Siblings. Explicit answerability assessment substantially improves target identification, while fine-tuning shows that answerability is learnable but difficult to transfer without compromising broad semantic retrieval.

1 INTRODUCTION

In retrieval-augmented generation (RAG) systems (Lewis et al., 2020; Huang & Huang, 2024), retrieval must do more than find sources relevant to a query: the retrieved source should contain sufficient information to support a valid answer. We refer to this property as **answerability**, which is distinct from semantic relevance, or whether a source matches the meaning or topic of a query. A source may therefore be relevant without being sufficient to answer the query. This raises a natural question: can retrieval identify sources that are both relevant and answerable?

We study this question using tables as a controlled setting. Their structured, localized information allows us to construct sources that are highly similar in semantics and structure but differ in whether they contain the information needed to answer a query. This makes tables a useful setting for disentangling semantic relevance from answerability. Moreover, tables are widely used as information sources in enterprise databases, data lakes, and web resources (Cafarella et al., 2008; Stonebraker

Query: In Q3 2023, which supplier had deliveries overdue by more than two weeks?

Retrieved Table A——Suppliers, Q3 2023

Supplier	Days Overdue	Order ID	Category
S-1042	**18**	20239123	Electronics
S-0873	9	20238841	Mech.
...	...	...	...

Dense Retriever Score = 0.88
Answerable

Retrieved Table B——S-1042, by period

Period	Days Overdue	Order ID	Category
2023 Q1	9	20231204	Electronics
2022 Q4	14	20224502	Electronics
...	...	...	...

Dense Retriever Score = 0.89
Not Answerable

Figure 1: Two sibling tables with nearly identical schemas yield similar retrieval signals but differ in answerability: Table A contains the answer, while Table B scores higher because its value (14 days) is semantically close to the query's threshold (two weeks), despite belonging to the wrong period.

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

et al., 2013; Nargesian et al., 2018), where semantically similar tables may likewise differ in the information they contain.

Existing table QA benchmarks such as WikiTableQuestions and HybridQA (Pasupat & Liang, 2015; Chen et al., 2020) assume that the relevant table is already given, and thus do not evaluate this retrieval problem. Recent Table RAG work (Pan et al., 2022; Balaka et al., 2025; Yu et al., 2025a) improves retrieval and reasoning, yet existing evaluations largely emphasize semantic relevance rather than whether a retrieved table is uniquely sufficient to answer the query. This makes it difficult to isolate answerability from coarse semantic matching.

To study this distinction, we introduce **TCR-Bench**, a controlled diagnostic benchmark for **T**able **C**ontent-level Answerability in **R**AG. TCR-Bench pairs answerable Target Tables with highly similar but non-answerable Sibling Distractors, varying answerability while keeping much semantic and structural context fixed. This controlled design tests whether a retriever can identify the uniquely answerable target rather than merely reaching its semantic neighborhood.

Using TCR-Bench, we identify a consistent **Semantic-Answerability Gap (SAG)** in retrieval: semantically relevant sources can be reached without reliably identifying the uniquely answerable one. We primarily study this phenomenon in **dense retrieval**. The best-performing dense retriever reaches the relevant table group at top-1 for 67.0% of queries but retrieves the uniquely answerable target for only 18.2%. This gap also has a clear downstream impact: QA F1 drops from 0.755 with the oracle table to 0.330 with retrieved top-5 tables. Diagnostics show that surface form is not the main cause: retrievers favor semantic volume over sufficiency, respond weakly to column-value binding disruptions, and form compact Target–Sibling neighborhoods without global collapse.

We further use **Answerability-Aware Reranking (AAR)** to test whether explicit query-conditioned answerability assessment can alleviate this failure. AAR improves top-1 retrieval from **18.2%** to **57.4%**. However, answerability-oriented fine-tuning is strongly distribution-dependent, generic retrieval training does not induce comparable answerability discrimination, and simple joint training does not recover the separate gains of the two objectives. Thus, learning answerability and preserving broad, transferable semantic retrieval remain distinct challenges.

Figure 2 summarizes the overall study. Our contributions are threefold. **(1) We identify the Semantic-Answerability Gap:** we show that dense retrievers can reach the correct semantic neighborhood while failing to identify the uniquely answerable target, establishing answerability as a distinct retrieval dimension beyond semantic relevance. **(2) We diagnose dense retrieval behavior in a controlled setting:** using TCR-Bench, we show that retrievers favor semantic volume over sufficiency, remain weakly sensitive to disrupted row-column bindings, and place Targets and Siblings in compact local neighborhoods without evidence of global collapse. **(3) We investigate answerability-aware retrieval:** explicit answerability modeling substantially improves target identification, while answerability-oriented, generic retrieval, and joint training experiments show that transferable answerability discrimination and broad semantic retrieval do not necessarily align across settings.[1]

2 Related Work

2.1 Beyond Semantic Relevance: Answerability in Retrieval

Recent work has identified limitations of embedding-based retrieval beyond semantic similarity, including expressivity limits (Weller et al., 2026), false positives at scale (Reimers & Gurevych, 2021), diminishing returns from scaling (Killingback et al., 2026), and reliance on surface cues over factual evidence (Fayyaz et al., 2025; Wu et al., 2025).

More importantly, semantic similarity does not always imply answer support. Recent benchmarks show that similarity-based retrieval aligns more with topical relevance than decision usefulness (Schimanski et al., 2026), while answer-centric supervision improves retrieval when positives are defined by answer support (Nian et al., 2025; Zhou et al., 2026). Other studies address misleading context (Kim & Lee, 2024; Liu et al., 2026) and multi-hop retrieval (Trivedi et al., 2023;

[1]Anonymous code and data are available at `https://anonymous.4open.science/r/TCR-Bench-05F6`.

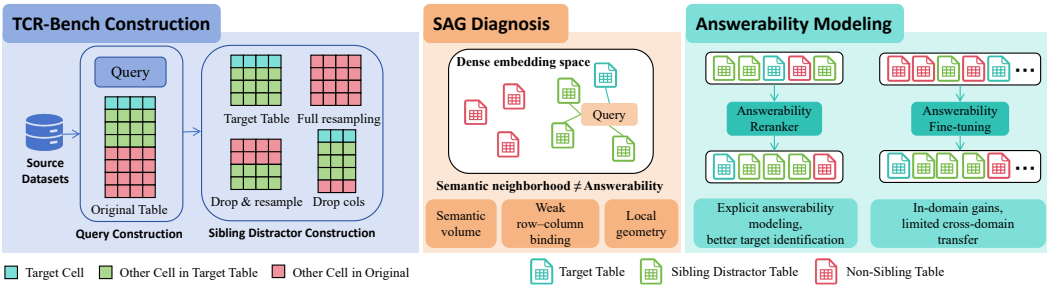


Figure 2: Overview. Left: TCR-Bench pairs each answerable Target Table with semantically similar but non-answerable Sibling Distractors. Center: dense retrieval exhibits the **Semantic-Answerability Gap (SAG)**, with controlled diagnostics probing factors associated with this behavior. Right: explicit answerability modeling improves target identification, while answerability-oriented training shows limited cross-domain transfer.

Asai et al., 2024). However, these works do not isolate answerability as a distinct dimension from semantic relevance or systematically disentangle the two under controlled conditions. We formalize this mismatch as the **SAG** and study it through controlled sibling-table diagnostics.

2.2 Table RAG Methods and Benchmarks

Table retrieval has explored both table-specific dense retrievers (Herzig et al., 2021) and generic retrievers such as DPR (Karpukhin et al., 2020), with evidence that generic architectures can perform comparably to specialized ones (Wang et al., 2022). These findings suggest that structural bias alone may not fully capture fine-grained distinctions within tables. Meanwhile, existing evaluations have largely focused on semantic relevance, often leaving finer-grained content distinctions less examined. Simpler schema-aware and title-aware representations (Kim et al., 2024; Khanna & Subedi, 2025) have also shown strong performance, while hybrid and multi-table pipelines (Chen et al., 2024b; Zhang et al., 2026) similarly focus on relevance matching. Taken together, these settings often rely on relatively coarse signals of relevance, leaving finer-grained content-level distinctions less directly tested.

Existing benchmarks (Herzig et al., 2021; Zhang & Balog, 2018; Cui et al., 2025; Ji et al., 2024; Strich et al., 2026; Xu et al., 2025; Zou et al., 2026) increase difficulty through multi-hop reasoning, mixed modalities, or heterogeneous corpora, yet consistently frame retrieval as semantic matching. To our knowledge, existing benchmarks do not explicitly require distinguishing a uniquely answerable table from semantically similar candidates. TCR-Bench targets this gap directly through **sibling tables**, isolating fine-grained answerability selection from coarse semantic retrieval.

3 A Controlled Setting for Studying Answerability

3.1 Answerability as a Distinct Retrieval Dimension

We define **answerability** as whether a table contains sufficient information to produce a valid answer to a given query. This is distinct from semantic relevance: a table may be highly relevant to a query while still being unable to answer it. We therefore distinguish three table types: **Target Tables**, which are answerable; **Sibling Distractor Tables**, which are highly similar to the target but non-answerable; and **Non-Sibling Tables**, which are less related in schema or semantics.

3.2 TCR-Bench: Controlled Sibling-Table Setting

TCR-Bench is designed to diagnose the discrepancy between semantic relevance and answerability. For each query, we construct one Target Table and multiple Sibling Distractor Tables from the same source table. Sibling Distractor Tables preserve substantial semantic and structural similarity with the target while being modified to become non-answerable. This creates a controlled setting where semantic similarity is largely preserved while answerability varies.

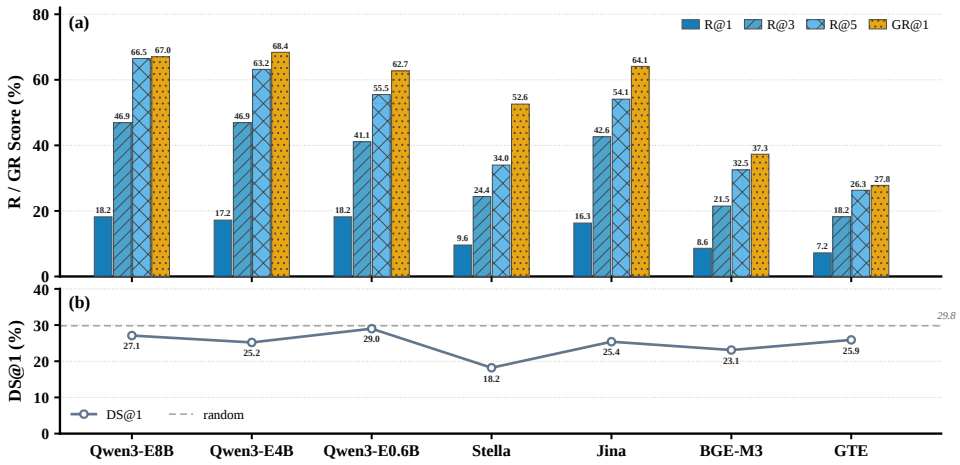


Figure 3: Retrieval performance on TCR-Bench using the Mixed format. Abbreviated model names are used for compact presentation.

We further vary queries, schemas, and table representations to reduce reliance on superficial matching. Tables are rendered in Markdown, HTML, CSV, and Mixed formats. Detailed construction and validation procedures are provided in Appendix A.

3.3 Benchmark Construction and Validation

TCR-Bench is constructed from databases originating from Spider (Yu et al., 2018) and BIRD (Li et al., 2024). We formulate three TableQA tasks with increasing complexity: **Exact Match (EM)**, **Selective Filtering (SF)**, and **Selective Aggregation (SA)**, involving 1–3 conjunctive conditions. Formal definitions are provided in Appendix A.2.

For each query, Sibling Distractor Tables are generated through controlled schema- and row-level modifications, followed by answerability verification and manual validation. We additionally define a diagnostic subset of EM queries with tables restricted to their sibling groups, allowing answerability to be examined under largely fixed semantic similarity. The final benchmark contains 209 queries (98 EM, 55 SF, and 56 SA) and 637 tables.

To test whether the observed phenomenon extends beyond controlled construction, we additionally introduce **Real-TCR**, a natural-negative collection derived from naturally occurring tables with real-world data partitions. Unlike TCR-Bench, its candidates are not generated through answerability-targeted transformations. Real-TCR contains 77 queries and 217 tables, while Real-TCR-small contains 38 queries and 117 tables. Detailed construction and results are provided in Appendix E.

4 Semantic Relevance vs. Answerability

4.1 Retrieval Evaluation

4.1.1 Embedding Models

We evaluate representative embedding models across architectural families and parameter scales, including Qwen3-Embedding-0.6B, Qwen3-Embedding-4B, Qwen3-Embedding-8B (Zhang et al., 2025b), stella_en_1.5B_v5 (Zhang et al., 2025a), jina-embeddings-v4 (Günther et al., 2025), bge-m3 (Chen et al., 2024a), and gte_Qwen2-7B-instruct (Li et al., 2023). Detailed model descriptions are provided in Appendix B.

4.1.2 Evaluation Metrics

We evaluate retrieval from two complementary perspectives: *semantic relevance* and *answerability*. Let y_i be the ground-truth target and R_i^k the top-k retrieved items for query i.

Top-k Recall ($R@k$) (Ji et al., 2024; Zou et al., 2026; Strich et al., 2026) is defined as $R@k = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(y_i \in R_i^k)$. Since each query in TCR-Bench has a unique answerable Target Table, $R@k$ measures whether the retriever successfully identifies the **answerable target**, rather than merely retrieving a semantically relevant table.

Top-k Group Recall ($GR@k$) measures coarse semantic relevance. Let $\mathcal{G}_i$ denote the Sibling Table Group for query i, and R_i^k the set of top-k retrieved tables for query i. Inspired by the formulation of $CR@k$ (Thakur et al., 2021), we define $GR@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|R_i^k \cap \mathcal{G}_i|}{\min(|\mathcal{G}_i|,k)}$. We primarily report $GR@1$ to measure whether retrieval reaches the correct semantic neighborhood.

Top-1 Discriminative Score ($DS@1$) is defined as $DS@1 = R@1/GR@1$, with $DS@1 = 0$ when $GR@1 = 0$. It measures how effectively the retriever identifies the answerable target within its semantically relevant group. Under random selection within a Sibling Table Group, the expected $DS@1$ is 0.221 on the original dataset and 0.298 when column-deletion variants are excluded.

4.1.3 Main Results

Figure 3 reports retrieval performance under the **Mixed** format, with full results in Appendix D.1.

The results reveal a consistent **Semantic-Answerability Gap (SAG)** across the evaluated embedding models. The strongest dense retriever achieves $GR@1 = 0.670$ but only $R@1 = 0.182$, showing that identifying the correct semantic neighborhood is substantially easier than identifying the answerable target. Scaling Qwen3 from 0.6B to 8B improves $R@5$ by 19.8%, but yields only modest gains in $DS@1$.

Most models remain near or below the random-selection baseline of $DS@1 = 0.298$, with significance analysis in Appendix M confirming that they do not significantly outperform chance. Column-deletion cases provide a useful contrast: their $R@1$ drops to 0.072 for Qwen3-E4B and 0.057 for Stella, indicating sensitivity to explicit structural mismatches, while discrimination among schema-consistent tables remains difficult. These results suggest that dense retrieval captures topical and schema-level similarity effectively, but is substantially less sensitive to fine-grained content relations such as row-column bindings, value constraints, and condition consistency.

A consistent qualitative pattern emerges on Real-TCR. Jina achieves $GR@1 = 0.948$ but only $R@1 = 0.143$ and $DS@1 = 0.151$, while Qwen3-E4B reaches $GR@1 = 0.909$ with $DS@1 = 0.043$. Across both Real-TCR and Real-TCR-small, the evaluated retrievers consistently achieve high group recall but substantially lower target recall, indicating that the difficulty of distinguishing the uniquely answerable target persists beyond the controlled setting of the main benchmark. Complete results are reported in Appendix E.

More broadly, supplementary experiments with alternative first-stage retrievers, including DTR-style, ColBERT-style, sparse, and hybrid methods, show that the observed gap is not substantially reduced by these alternatives (Appendix D.2 and D.3).

4.2 Downstream QA Evaluation

To examine how retrieval quality propagates to generation, we evaluate downstream QA using the LongTableBench (Li et al., 2025b) pipeline with Qwen3-30B-A3B-Thinking-2507 (Team, 2025). We report average F1 under five settings: Oracle (Target Table only), No-Table, and top-k retrieval with $k \in \{1, 3, 5\}$, where retrieved tables are concatenated in retrieval order.

We define $\text{Eff@}k = (\text{QA@}k/\text{QA@Oracle})/R@k$, where $QA@k$ denotes QA F1 using top-k retrieved tables. This measures how effectively the QA model exploits the Target Table once it is retrieved, with values near 1 indicating limited additional interference.

The Oracle setting achieves $F1 = 0.755$, whereas No-Table drops to 0.014, confirming that downstream answering depends heavily on retrieved tabular information. However, increasing k provides limited benefit: even top-5 retrieval reaches only $F1 = 0.330$. Meanwhile, $Eff@1$ remains close to 1 across all three model sizes, indicating that QA is largely effective once the Target Table is ranked first. In contrast, $Eff@3$ and $Eff@5$ consistently decline as more tables are introduced, consistent with prior observations that additional semantically related context can cause interference in RAG systems (Yoran et al., 2024; Kim & Lee, 2024; Iratni et al., 2025; Ouyang et al., 2025).

Table 1: $QA@k$ and $Eff@k$ using top-k retrieved tables from different embedding retrievers. The Oracle (Target Table only) and No-Table settings achieve F1 scores of 0.755 and 0.014, respectively.

Method	QA@1	QA@3	QA@5	Eff@1	Eff@3	Eff@5
Qwen3-E0.6B	**0.144**	0.244	0.276	1.046	0.786	0.659
Qwen3-E4B	0.113	**0.269**	0.329	0.871	0.760	0.690
Qwen3-E8B	0.139	0.266	**0.330**	1.014	0.751	0.657

4.3 The Answerability Bottleneck

The main experiments establish two observations. First, retrievers reliably reach the correct semantic group but rarely identify the answerable target. Second, downstream QA works well only when the target is ranked first, while deeper retrieval mostly adds semantically similar but non-answerable tables that interfere with generation.

The Answerability Bottleneck

Dense retrievers identify semantically relevant sources well but often perform close to the random-selection baseline when distinguishing the answerable target within that group. This **SAG** persists across the models and scales we evaluate, including different retrieval settings.

5 Diagnosing Dense Retrieval Behavior

The main experiments show that dense retrieval can often reach the semantic neighborhood containing the Target Table, but has difficulty distinguishing it from closely related Sibling Distractors. We conduct controlled diagnostics to identify what information the evaluated representations rely on and where this discrimination failure arises. We first examine *(1)* sensitivity to surface-level serialization and query wording, and then investigate *(2)* sensitivity to semantic content and fine-grained table structure, including the embedding geometry underlying the observed behavior.

5.1 Robustness to Surface Variations

We first test whether the observed behavior depends on how tables or queries are expressed. For tables, we evaluate six serialization formats on the Diagnostic Subset: CSV, HTML, Markdown, Mixed, and two sentence-based representations, **TSen** and **TSenS**. TSen converts each row into a declarative sentence while preserving attribute order, whereas TSenS randomly shuffles the attributes. For queries, we compare three semantically equivalent formulations: the original natural-language query, a sentential formulation, and a template-based formulation (Appendix F.1).

Across these variations, retrieval remains broadly stable for both Qwen3-E4B and Stella. Qwen3-E4B achieves $R@1$ between 0.186 and 0.257 across the six table formats, with $GR@1$ between 0.757 and 0.857. Query reformulation similarly produces limited changes. Thus, the Target–Sibling discrimination failure is not primarily an artifact of serialization or query wording. Complete results and consistency analyses are provided in Appendix F.

5.2 Sensitivity to Semantic and Structural Information

We next examine which aspects of table content affect dense retrieval, focusing on semantic information volume and structural bindings relevant to answerability.

Semantic volume. We first ask whether dense retrieval responds to information useful for semantic matching or to information sufficient for identifying the answer. For each table row, we construct three sentence-based representations. **Sen** preserves the complete row and original attribute order. **SenS** preserves the complete row but randomly shuffles attributes, isolating sensitivity to ordering. **SenSS** retains only a subset of column-value pairs sufficient to uniquely identify the relevant row.

Table 2 shows that Sen and SenS perform similarly, whereas both generally outperform SenSS. For Qwen3-E4B, reducing a row to the sufficient subset lowers $R@5$ from 0.800–0.806 to 0.703. The

Table 2: Inter-table retrieval under different sentence-based representations. Sen preserves the complete row, SenS additionally shuffles the attribute order, and SenSS retains only a sufficient subset of attributes for answering the query.

Model	Type	R@k			GR@1	DS@1
		@1	@3	@5		
Qwen3-E4B	Sen	0.232	0.529	0.800	**0.987**	0.235
	SenS	**0.252**	**0.548**	**0.806**	**0.987**	**0.255**
	SenSS	0.226	0.490	0.703	0.865	0.261
Stella	Sen	**0.213**	**0.561**	0.768	**0.994**	**0.214**
	SenS	0.206	0.555	**0.774**	0.987	0.209
	SenSS	0.148	0.439	0.626	0.794	0.187

small difference between Sen and SenS further indicates that attribute order is not the primary factor. Within these controlled representations, retrieval appears more responsive to the amount of semantic content than to whether the representation contains only information sufficient for identification.

Within the Target Table, row-level retrieval substantially outperforms retrieval over 10-line chunks (Figure 4), indicating that dense embeddings capture local row-level content, while aggregation can dilute query-relevant information. Complete results are reported in Appendix G.

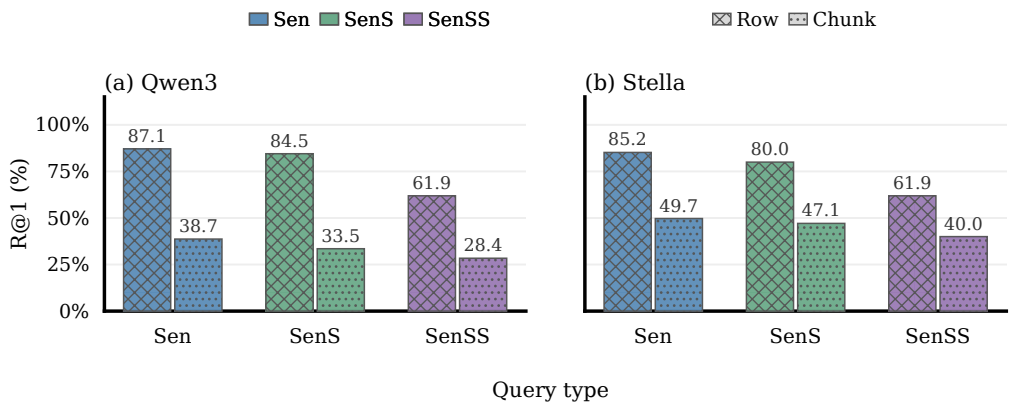


Figure 4: Intra-table retrieval under different granularities and sentence-based representations. Row-level retrieval substantially outperforms 10-line chunks, while reducing a row to a sufficient subset in SenSS degrades retrieval.

Row-column binding. Semantic content alone does not determine answerability: answers can depend on binding values to the correct columns and rows. On the Diagnostic Subset, we build a Minimal Contrastive Set with three configurations: **Ori** (original table), **RID** (each cell prefixed with its row index), and **RID+Shuf** (RID with values further shuffled within each column), which keeps explicit row indices and lexical content while breaking row–column bindings.

Table 3 shows that Ori and RID both outperform RID+Shuf (e.g., Qwen3-E4B Mixed: 0.443/0.371 vs. 0.314). Thus, even when row identity is made explicit, disrupting within-column value placements still degrades discrimination, indicating limited sensitivity to fine-grained row–column consistency. Full results are in Appendix G.

Embedding geometry. The perturbation results show that the evaluated dense retrievers are more sensitive to semantic content than to some fine-grained structural changes. We next ask whether the discrimination failure reflects global representation collapse or local ambiguity among semantically related tables. Following (Vangara & Gopinath, 2026), we report effective dimensionality d_{eff} (how many directions carry most of the variance) and mean pairwise cosine distance $\bar{d}$.

Table 3: Retrieval on the Minimal Contrastive Set. Ori is the original table; RID adds row indices; RID+Shuf further shuffles values within each column on top of RID.

Model	Config	Format	R@1	GR@1	DS@1
Qwen3-E4B	Ori	Mixed	0.443	0.671	0.660
	Ori	Markdown	0.343	0.643	0.533
	RID	Mixed	0.371	0.671	0.553
	RID+Shuf	Mixed	0.314	0.643	0.489
Stella	Ori	Mixed	0.229	0.471	0.485
	RID	Mixed	0.243	0.471	0.515
	RID+Shuf	Mixed	0.200	0.429	0.467

We find evidence for the latter. Across all 637 tables, embeddings remain broadly dispersed (d_{eff} in the tens; $\bar{d} \in [0.40, 0.55]$). In contrast, Target+Sibling groups form compact, low-dimensional neighborhoods (median $\bar{d}^{\text{loc}} = 0.08$–$0.11$; median $d_{\text{eff}}^{\text{loc}} \approx 2.4$–$2.9$; Table 24). Figure 5 makes this concrete: sibling groups appear as tight pockets, while within-pocket cosine ranking can prefer a Sibling over the Target.

To quantify this final discrimination step, we use the Target–Sibling margin $\Delta_q = \cos(q, t_q) - \max_{s \in S_q} \cos(q, s)$, where t_q is the Target and S_q is the corresponding Sibling set. A positive margin means that the Target is closer to the query than every Sibling. Mean Δ_q is negative across all evaluated models and correlates more strongly with $R@1$ than with $GR@1$, indicating that remaining errors concentrate on query-to-table discrimination after the semantic neighborhood has been reached. Full analyses are in Appendix G.3.

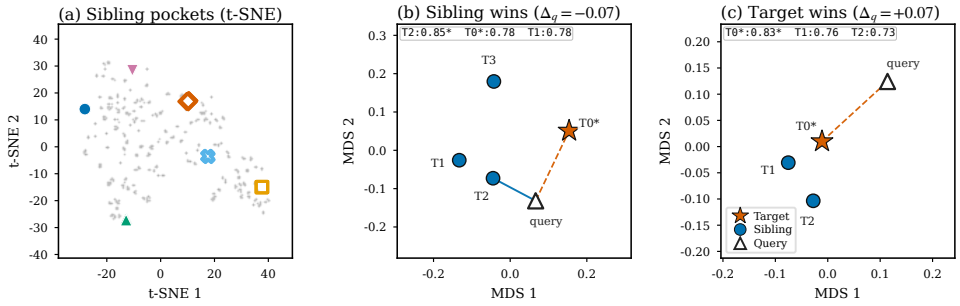


Figure 5: Embedding geometry (Qwen3-E4B, Mixed). (a) Sibling groups form tight t-SNE pockets: inn (filled circle), baseball (open square), WDI (filled triangle), music (open diamond), codebase (filled inverted triangle), games (cross). (b,c) Within a pocket, cosine may prefer a Sibling ($\Delta_q < 0$) or the Target ($\Delta_q > 0$).

Diagnostic Summary. Within our evaluation, serialization, query wording, and global embedding collapse do not appear to explain the observed gap. Instead, the tested dense retrievers tend to favor semantic volume over information sufficiency, show weak sensitivity to row-column bindings, and place Targets and Siblings in compact local neighborhoods that remain difficult to separate.

Similarity Is Not Sufficiency

The gap is not a surface artifact: dense retrievers favor semantic volume over sufficiency, respond weakly to row-column bindings, and leave Targets and Siblings hard to separate in compact local neighborhoods.

6 Modeling Answerability for Retrieval

To further examine whether explicitly modeling answerability can alleviate the SAG, we conduct two complementary experiments: answerability-aware reranking and answerability-oriented fine-tuning. The former asks whether query-conditioned interaction can distinguish the target from se-

mantically similar tables, while the latter asks whether this behavior can be learned by an embedding model and transferred beyond the training distribution.

Query-Conditioned Answerability Assessment. We use **Answerability-Aware Reranking** (**AAR**) as a diagnostic framework to explicitly assess answerability after dense retrieval. Unlike single-vector retrieval, query-conditioned interaction directly evaluates whether a candidate table is sufficient to satisfy the query. We consider three variants: **AAR-CE**, using Qwen3-Reranker-8B (Qwen3-R8B) paired with the Qwen3-E8B retriever; **AAR-LLM-PW**, using Qwen3-30B as a pointwise LLM-based answerability classifier (Balaka et al., 2025); and **AAR-LLM-LW**, using the same Qwen3-30B model to jointly rank candidates in a listwise setting.

Table 4: Answerability-aware reranking on the Qwen3-E8B backbone (k=10).

Method	R@1	R@5	GR@1	DS@1
AAR-CE	**0.574**	0.656	0.766	**0.750**
AAR-LLM-PW	0.536	**0.737**	0.761	0.704
AAR-LLM-LW	0.450	0.708	**0.770**	0.584

All three variants substantially improve target identification over dense retrieval, with R@1 increasing from 0.182 to 0.450–0.574. The gain persists under both pointwise and listwise LLM reranking, indicating that it is not specific to a particular reranking formulation. The matched Qwen3-E8B and Qwen3-R8B comparison further suggests that the improvement is associated with the change from dense embedding to query-conditioned reranking rather than model scale: both use the same pretrained initialization and 8B parameter scale, but differ in objective. Full results and significance analysis are in Appendix K and Appendix M.

Learning Answerability in Embeddings. We further fine-tune Qwen3-E0.6B with query–target positives and sibling-table hard negatives to test whether answerability can be learned directly by an embedding model. The best TCR-Bench model improves R@1 from 0.182 to 0.493 and DS@1 from 0.290 to 0.691, showing that answerability is learnable under matched supervision. However, the learned signal is strongly distribution-dependent: the TCR-Bench-trained model achieves only 0.065 R@1 on Real-TCR and substantially degrades retrieval on the Table RAG benchmark TARGET (Ji et al., 2024). The reverse transfer from Real-TCR to TCR-Bench is similarly weak, while generic training on TabFact and FeTaQA, two subsets of TARGET, improves semantic retrieval without inducing comparable answerability-sensitive discrimination. Moreover, a simple mixture of TCR-Bench and TARGET training data does not simultaneously recover the gains of the two objectives. Thus, learning answerability and preserving broad, transferable semantic retrieval remain distinct challenges. Detailed configurations and results are given in Appendix L.

Answerability Is Learnable, but Transfer Remains Difficult

Explicit answerability modeling improves target identification in our evaluation, but learning transferable answerability discrimination while preserving broad semantic retrieval remains nontrivial.

7 Conclusion

Our experiments reveal a consistent **Semantic-Answerability Gap (SAG)** in retrieval: semantically relevant sources can be reached without reliably identifying the uniquely answerable target. We primarily study this phenomenon in **dense retrieval**, using **TCR-Bench** to isolate the gap through sibling tables that are highly similar in semantics and structure but differ in answerability. Our diagnostics show that retrievers favor semantic volume over sufficiency, respond weakly to row-column bindings, and leave Targets and Siblings hard to separate locally. Explicit answerability modeling substantially improves target identification, while fine-tuning shows that answerability can be learned but remains strongly distribution-dependent. Generic retrieval training does not automatically induce comparable answerability-sensitive discrimination, and simple joint training does not simultaneously preserve gains in answerability and broad semantic retrieval. These findings motivate retrieval objectives and models that treat answerability as a distinct retrieval dimension while jointly preserving broad semantic relevance and transferable answerability discrimination.

AI USE STATEMENT

In this work, we used generative AI tools for minor code generation assistance, manuscript editing and polishing, and paraphrasing queries and table schemas during dataset construction. We have not used generative AI tools for developing theoretical models or conceptual frameworks, formulating mathematical claims, assisting in proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning or reformatting datasets, performing qualitative or thematic data analysis, or interpreting experimental results, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for retrieving and discovering information, such as finding related work. We have reviewed all AI-assisted work. LLM-paraphrased queries and table schemas were manually verified by the authors for semantic correctness and appropriateness, LLM-generated code was tested for correctness before use, and all AI-assisted manuscript edits were reviewed and revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

TCR-Bench is constructed from publicly available databases, including Spider and BIRD, together with synthetically generated table variants and query paraphrases. The benchmark contains no personally identifiable information. Dataset construction involves automatic generation followed by manual author verification to check the quality and validity of queries and table variants. The benchmark and accompanying resources are intended for research use under the stated license.

As a supplementary natural-negative validation, we additionally evaluate **Real-TCR**, a small collection derived from de-identified operational tables provided under a data-use agreement with an industry partner and annotated by personnel of that organization. Due to contractual and data-use constraints, Real-TCR is not included in the anonymous repository, including in de-identified form. Real-TCR is used only for external validation and cross-distribution diagnostics and is not required to reproduce the main findings or controlled analyses of TCR-Bench. We do not release any underlying operational data or identifying information from this collection. Practitioners applying Table RAG methods to sensitive or proprietary data should use appropriate access controls, data governance, and privacy safeguards beyond those provided by this benchmark.

REPRODUCIBILITY STATEMENT

Anonymous code and data for TCR-Bench are available at the anonymous repository: `https://anonymous.4open.science/r/TCR-Bench-05F6`. The repository provides the core resources needed to reproduce the main experiments and controlled diagnostics, including the table data, complete query files, the Diagnostic subset, the RAG pipeline implementation, and configuration files specifying the experimental settings and hyperparameters. The implementation covers dense retrieval, AAR reranking, and downstream TableQA used in the main experiments. The paper and appendix provide the benchmark construction procedure, evaluation metrics, experimental settings, and additional diagnostic analyses needed to interpret and reproduce the reported results. Dense retrieval components produce deterministic outputs given fixed model weights and inputs. LLM-based components are run with temperature set to 0, although minor nondeterminism may remain due to hardware, software, or library implementations. As described in the Ethics Statement, Real-TCR cannot be included in the anonymous repository because of data-use restrictions on the underlying industry data; consequently, results that directly depend on Real-TCR cannot be independently reproduced from the provided resources. These results serve as supplementary external validation, while the main TCR-Bench results and controlled analyses are reproducible using the resources provided in the anonymous repository.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Muhammad Imam Luthfi Balaka, David Alexander, Qiming Wang, Yue Gong, Adila Krisnadhi, and Raul Castro Fernandez. Pneuma: Leveraging llms for tabular data representation and retrieval in an end-to-end system. *Proceedings of the ACM on Management of Data*, 3(3):1–28, 2025.

Michael J Cafarella, Alon Y Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: Exploring the power of tables on the web. *Proc. VLDB Endow.*, 1(1):538–549, 2008.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

Peter Baile Chen, Yi Zhang, and Dan Roth. Is table retrieval a solved problem? exploring join-aware multi-table retrieval. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 2687–2699, 2024b.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1026–1036, 2020.

Lingxi Cui, Guanyu Jiang, Huan Li, Ke Chen, Lidan Shou, and Gang Chen. Tablecopilot: A table assistant empowered by natural language conditional table discovery. *Proceedings of the VLDB Endowment*, 18(12):5399–5402, 2025.

Mohsen Fayyaz, Ali Modarressi, Hinrich Schuetze, and Nanyun Peng. Collapse of dense retrievers: Short, early, and literal biases outranking factual evidence. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9136–9152, 2025.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021a.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2288–2292, 2021b.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 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)*, pp. 1762–1777, 2023.

Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Sedigheh Eslami, Scott Martens, Bo Wang, Nan Wang, and Han Xiao. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval, 2025. URL https://arxiv.org/abs/2506.18902.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 512–519, 2021.

Yizheng Huang and Jimmy Huang. A survey on retrieval-augmented text generation for large language models. *arXiv preprint arXiv:2404.10981*, 2024.

Malika Iratni, Mohand Boughanem, and Taoufiq Dkaki. Dynamic context selection for retrieval-augmented generation: Mitigating distractors and positional bias. *arXiv preprint arXiv:2512.14313*, 2025.

Xingyu Ji, Aditya Parameswaran, and Madelon Hulsebos. Target: Benchmarking table retrieval for generative tasks. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

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)*, pp. 6769–6781, 2020.

Sujit Khanna and Shishir Subedi. Tabular embedding model (tem): Finetuning embedding models for tabular rag applications. In *Intelligent Computing-Proceedings of the Computing Conference*, pp. 448–460. Springer, 2025.

Omar Khattab and Matei Zaharia. 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*, pp. 39–48, 2020.

Julian Killingback, Mahta Rafiee, Madine Manas, and Hamed Zamani. Scaling laws for embedding dimension in information retrieval. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 790–800, Melbourne, VIC, Australia, 2026. Association for Computing Machinery. doi: 10.1145/3805712.3809708. URL https://doi.org/10.1145/3805712.3809708.

Kihun Kim, Mintae Kim, Hokyung Lee, Seong Ik Park, Youngsub Han, and Byoung-Ki Jeon. Thorr: Complex table retrieval and refinement for rag. In *IR-RAG@ SIGIR*, pp. 50–55, 2024.

Kiseung Kim and Jay-Yoon Lee. Re-rag: Improving open-domain qa performance and interpretability with relevance estimator in retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22149–22161, 2024.

Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. Splade-v3: New baselines for splade. *arXiv preprint arXiv:2403.06789*, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Da Li, Keping Bi, Jiafeng Guo, and Xueqi Cheng. Tailoring table retrieval from a field-aware hybrid matching perspective. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 27681–27692, 2025a.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36, 2024.

Liyao Li, Jiaming Tian, Hao Chen, Wentao Ye, Chao Ye, Haobo Wang, Ningtao Wang, Xing Fu, Gang Chen, and Junbo Zhao. Longtablebench: benchmarking long-context table reasoning across real-world formats and domains. *Findings of the Association for Computational Linguistics: EMNLP*, 2025, 2025b.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024a.

Peiyang Liu, Qiang Yan, Ziqiang Cui, Di Liang, Xi Wang, and Wei Ye. Beyond semantic relevance: Counterfactual risk minimization for robust retrieval-augmented generation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1232–1243, Melbourne, VIC, Australia, 2026. Association for Computing Machinery. doi: 10.1145/3805712.3809631. URL https://doi.org/10.1145/3805712.3809631.

Tianyang Liu, Fei Wang, and Muhao Chen. Rethinking tabular data understanding with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 450–482, 2024b.

Fatemeh Nargesian, Erkang Zhu, Ken Q Pu, and Renée J Miller. Table union search on open data. *Proceedings of the VLDB Endowment*, 11(7):813–825, 2018.

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. W-rag: Weakly supervised dense retrieval in rag for open-domain question answering. In *Proceedings of the 2025 International ACM SIGIR conference on innovative concepts and theories in information retrieval (ICTIR)*, pp. 136–146, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruiran Yan, Yucong Luo, Jiaying Lin, and Qi Liu. Hoh: A dynamic benchmark for evaluating the impact of outdated information on retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6036–6063, 2025.

Feifei Pan, Mustafa Canim, Michael Glass, Alfio Gliozzo, and James Hendler. End-to-end table question answering via retrieval-augmented generation. *arXiv preprint arXiv:2203.16714*, 2022.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, 2015.

Nils Reimers and Iryna Gurevych. The curse of dense low-dimensional information retrieval for large index sizes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 605–611, 2021.

Stephen E. Robertson and Hugo Zaragoza. *The Probabilistic Relevance Framework: BM25 and Beyond*. Now Publishers Inc., 2009. ISBN 9781601981919. Foundations and Trends in Information Retrieval.

Tobias Schimanski, Stefanie Lewandowski, Christian Woerle, Nicola Reichenau, Yauheni Huryn, and Markus Leippold. Usefulbench: Towards decision-useful information as a target for information retrieval. *arXiv preprint arXiv:2604.15827*, 2026.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Yibin Lei, Tianyi Zhou, Michael Blumenstein, and Daxin Jiang. Retrieval-augmented retrieval: Large language models are strong zero-shot retriever. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15933–15946, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.943. URL https://aclanthology.org/2024.findings-acl.943/.

Adarsh Singh, Kushal Raj Bhandari, Jianxi Gao, Soham Dan, and Vivek Gupta. Craft: Training-free cascaded retrieval for tabular qa. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3284–3298, 2026.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

Michael Stonebraker, Daniel Bruckner, Ihab F Ilyas, George Beskales, Mitch Cherniack, Stanley B Zdonik, Alexander Pagan, and Shan Xu. Data curation at scale: the data tamer system. In *Cidr*, volume 2013, 2013.

Jan Strich, Enes Kutay Isgorur, Maximilian Trescher, Chris Biemann, and Martin Semmann. T2-ragbench: Text-and-table aware retrieval-augmented generation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 165–191, 2026.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 645–654, 2024.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

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.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 10014–10037, 2023.

Anirudh Bharadwaj Vangara and Ashwin Gopinath. The geometry of consolidation. GitHub repository, 2026. URL https://github.com/niashwin/geometry-of-consolidation. Submitted to NeurIPS 2026.

Blerta Veseli, Julian Chibane, Mariya Toneva, and Alexander Koller. Positional biases shift as inputs approach context window limits. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=vlUk8z8LaM.

Zhiruo Wang, Zhengbao Jiang, Eric Nyberg, and Graham Neubig. Table retrieval may not necessitate table-specific model design. In *Proceedings of the workshop on structured and unstructured knowledge integration (SUKI)*, pp. 36–46, 2022.

William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the theoretical limitations of embedding-based retrieval. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 11202–11216, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/130b10e31f5c3d730162323b2685bd19-Paper-Conference.pdf.

Haoyu Wu, Qingcheng Zeng, and Kaize Ding. Fact or facsimile? evaluating the factual robustness of modern retrievers. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 5351–5355, 2025.

Chuan Xu, Qiaosheng Chen, Yutong Feng, and Gong Cheng. mmrag: A modular benchmark for retrieval-augmented generation over text, tables, and knowledge graphs. In *International Semantic Web Conference*, pp. 3–21. Springer, 2025.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 29862–29883, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8011b23e1dc3f57e1b6211ccad498919-Paper-Conference.pdf.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Xiaohan Yu, Pu Jian, and Chong Chen. Tablerag: A retrieval augmented generation framework for heterogeneous document reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14074–14093, 2025a.

Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. Mitigate position bias in llms via scaling a single hidden states channel. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6092–6111, 2025b.

Chi Zhang, Qiyang Chen, and Mengqi Zhang. Mixture-of-rag: Integrating text and tables with large language models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1880–1891, 2026.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. Jasper and stella: distillation of sota embedding models, 2025a. URL https://arxiv.org/abs/2412.19048.

Shuo Zhang and Krisztian Balog. Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 world wide web conference*, pp. 1553–1562, 2018.

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. *arXiv preprint arXiv:2506.05176*, 2025b.

Jiawei Zhou, Hang Ding, and Haiyun Jiang. Ark: Answer-centric retriever tuning via kg-augmented curriculum learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8191–8208, 2026.

Jiaru Zou, Dongqi Fu, Sirui Chen, Xinrui He, Zihao Li, Yada Zhu, Jiawei Han, and Jingrui He. RAG over tables: Hierarchical memory index, multi-stage retrieval, and benchmarking. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 38133–38152, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1902. URL https://aclanthology.org/2026.findings-acl.1902/.

A Benchmark Construction Details

This section provides the detailed construction procedures and implementation constraints used to instantiate TCR-Bench. The goal is to isolate content-level answerability from coarse semantic relevance: for each query, an answerable Target Table is paired with highly similar but non-answerable Sibling Distractor Tables, so that answerability can vary while much of the semantic and structural context remains fixed. Here we describe how these tables are generated, how surface-level shortcuts are controlled, and how unique answerability and annotation quality are verified.

A.1 Source Tables and Query Generation

We construct TCR-Bench from databases originating from Spider (Yu et al., 2018) and BIRD (Li et al., 2024). These benchmarks provide relational tables from diverse real-world domains and naturally occurring schemas, allowing our controlled construction to retain realistic table structures while enabling precise manipulation of answer-bearing evidence.

For each source table, we generate TableQA queries based on its schema and cell values. We consider three task types: Exact Match (EM), Selective Filtering (SF), and Selective Aggregation (SA), corresponding to progressively more complex query semantics. Each query contains 1–3 conjunctive conditions over target columns. The formal definitions and examples of these tasks are provided in Appendix A.2.

For each query, the original source table serves as the basis for constructing a Target Table and its associated Sibling Distractor Tables. The Target Table remains uniquely answerable, whereas Sibling Distractors are produced by controlled schema- and row-level modifications that preserve

substantial semantic and structural similarity while removing the evidence needed to answer the query.

A.2 Task Definitions

We formalize three query-answer tasks of increasing semantic complexity: Exact Match (EM), Selective Filtering (SF), and Selective Aggregation (SA). Each task is defined over a relational table $\mathcal{T}$ with schema $\mathcal{S} = (A_1, \ldots, A_k)$, where each attribute A_i takes values from a discrete or continuous domain.

A query Q comprises a set of selection predicates $\mathcal{P}$ and, depending on the task type, may additionally specify a target attribute A_t and an aggregation function f. Let $R \subseteq \mathcal{T}$ denote the set of rows satisfying all predicates in $\mathcal{P}$:

$$R = \{\, r \in \mathcal{T} \mid \forall p \in \mathcal{P} : p(r) = \text{true} \,\}.$$

Exact Match (EM). In the EM setting, every predicate in $\mathcal{P}$ is a conjunction of equality conditions of the form $A_i = v_i$, where v_i is a literal value from the domain of A_i. The query requests the value(s) of a designated target column A_t from the rows in R. Under the key-based uniqueness assumption, R is either empty or a singleton. In practice, the answer is the unique value of A_t in the matching row.

Example. For the table shown in Table 5, the query "What is C when A = 5 and B = 3?" yields the unique answer 4.

Selective Filtering (SF). The SF task relaxes the equality restriction. At least one predicate employs a comparison operator from the set $<, >, \leq, \geq, \neq$. Predicates may still contain equality conditions, but the filter is no longer restricted to exact key lookups. The answer is the set of values of the target column A_t over the filtered rows R.

Example. The query "What is C when A = 5 and B ¡ 5?" filters the rows using an equality condition on A and an inequality condition on B, returning 4.

Selective Aggregation (SA). The SA task extends SF with a downstream aggregation operation. The query first applies the selection predicates $\mathcal{P}$ to obtain the filtered set R, and then computes an aggregate statistic over the values of a target column A_t. The aggregation function f is drawn from SUM, AVG, MIN, MAX.

The answer is the scalar result

$$f(\{\, r[A_t] \mid r \in R \,\}).$$

Example. The query "What is the average of C when A $\geq$ 5 and B $\geq$ 3?" filters the rows satisfying both inequalities and returns the mean of the corresponding C values, which is 5.

Table 5: Illustrative relational table for the task definitions.

A	B	C
5	3	4
6	8	6

These three tasks form a progression from exact symbolic lookup to conditional filtering and aggregation over a set of retrieved values. Together, they probe complementary reasoning requirements for determining whether a table contains sufficient evidence to answer a query.

A.3 Sibling Distractor Generation

The primary objective of Sibling Distractor construction is to preserve semantic and structural similarity while removing answer-bearing evidence. We therefore generate distractors from the same source table rather than independently sampling unrelated tables.

We employ three complementary generation strategies:

- **Removing Condition or Target Columns.** We remove either a query-condition column or the target column from the source table. Only one column is removed at a time, so that the resulting table retains most of the original schema while becoming insufficient for answering the query.
- **Row Deletion.** We randomly delete 30%–70% of the rows from the source table. Additional rows are subsequently sampled from the remaining corpus when necessary to satisfy the predefined table-length constraint. Rows that would provide the required answer evidence are explicitly excluded.
- **Row Resampling.** We construct a new table by resampling rows from the available table corpus while preserving the relevant schema characteristics. The resulting table remains structurally plausible but does not contain the evidence required by the query.

These strategies create distractors with different forms of missing evidence. Column-level modifications primarily remove schema or attribute grounding, whereas row-level modifications remove the specific records required to satisfy the query. Combining these mechanisms prevents the benchmark from reducing answerability to a single perturbation pattern.

A.4 Sampling Constraints and Answer Filtering

To ensure comparable retrieval conditions across candidate tables, all generated sub-tables are constrained to approximately 6,500–7,800 tokens, measured using `tiktoken`[2]. This relatively large context size is intended to embed the relevant evidence within a substantial amount of table content, so that retrieval cannot be trivially solved by inspecting a small table. As we further show in Appendix D.5, however, this challenge persists even for smaller tables.

For row deletion and row resampling, we explicitly filter out rows that would allow the query to be answered correctly. More importantly, answerability is verified globally across the complete benchmark corpus rather than only within the corresponding sibling group. This prevents an independently sampled table from accidentally becoming an alternative answerable table for the same query.

After this filtering procedure, each query is associated with exactly one valid Target Table. Sibling Distractor Tables remain semantically plausible but fail at least one requirement necessary for producing a valid answer.

A.5 Query and Schema Augmentation

We introduce controlled semantic and surface variations to reduce the possibility that retrieval can be solved through direct lexical matching. These transformations are applied while preserving the underlying query semantics and answerability relation. Tables are additionally rendered in Markdown, HTML, CSV, and Mixed formats. The Mixed format refers to a combination of the first three, where each table is randomly assigned one of them; this assignment is pre-determined to ensure experimental consistency.

A.5.1 Query and Schema Paraphrasing

Queries and schemas are paraphrased to introduce linguistic and naming variation. The generated paraphrases are required to preserve the original query intent, constraints, and target attributes. Manual validation is subsequently applied to cases where the generated text may introduce semantic shifts.

Schema paraphrasing is particularly important because Sibling Distractor Tables originate from the same source table and therefore naturally share substantial lexical overlap. Varying column names reduces the extent to which retrieval can rely on direct schema-token matching.

[2] `https://github.com/openai/tiktoken`

A.5.2 Date Format Augmentation

Date values are transformed into multiple surface representations while preserving their underlying temporal values. Specifically, standard $YYYY - MM - DD$ representations may be converted into:

- **English Abbreviations:** e.g., *15/Jan/2023* or *Jan. 15, 2023*.
- **Full English Spellings:** e.g., *January 15th, 2023*.
- **Roman Numerals:** e.g., *2023-I-15*.
- **Numeric Words:** e.g., *two thousand twenty-three. one. fifteen*.

These transformations preserve the semantic identity of dates while substantially changing their token-level representations.

A.5.3 Numeric and Scale Perturbations

We similarly transform numerical values to reduce reliance on exact lexical matching. The augmentation strategies include:

- **Scaling Large Numbers:** numbers in the thousands are represented using compact forms with *k* or *K*, e.g., $15000 \rightarrow 15k$.
- **Multiplicative and Additive Shifts:** numerical values are perturbed through controlled multiplication, division, or small additive shifts. The type of perturbation applied to each column is explicitly indicated in the column name and also communicated to the model in the relevant prompt. This type of perturbation is applied relatively infrequently.
- **Numeral-System Conversion:** numbers are represented using Roman numerals or English words, e.g., $102 \rightarrow$ *C* or *One hundred and two*.

All transformations are applied subject to the requirement that they do not alter the intended query semantics or introduce unintended answerable distractors.

A.6 Semantic Augmentation Model

We use GPT-OSS-120B (OpenAI, 2025), an open-weight model released by OpenAI under the Apache 2.0 license, for query and schema paraphrasing. At the time of benchmark construction, it provided a strong locally deployable open-weight option, allowing the augmentation process to be reproduced without reliance on proprietary API endpoints.

The model is used only for semantic augmentation and does not participate in retrieval or downstream question answering. To reduce the possibility that benchmark behavior is tied to the augmentation model, subsequent experimental components, including dense retrievers, rerankers, and QA models, are selected from non-GPT model families.

All generated paraphrases are subsequently validated to ensure that they preserve the original query intent and schema semantics.

A.7 Manual Validation and False-Negative Reduction

We employ a multi-stage validation procedure to improve annotation quality and reduce false-negative cases.

First, answerability is verified programmatically across all tables. Within each sibling group, only the Target Table is allowed to satisfy all query constraints. We additionally evaluate the query logic against the full candidate corpus to identify tables outside the sibling group that may inadvertently contain sufficient evidence to answer the query.

Second, we conduct targeted manual inspection of potentially ambiguous tables. For each query, we retrieve the top-k candidate tables outside its corresponding sibling group using multiple similarity signals. Candidate sets from different retrieval signals are merged, and the resulting tables are

manually examined for semantic equivalence with the Target Table. This procedure focuses on cases where different schemas or column names may express the same underlying concept and therefore create unintended alternative answer tables. Ambiguous cases are removed from the benchmark.

Third, all generated query and schema transformations are manually checked. Annotators verify that rewritten queries preserve the original intent and constraints, and that schema transformations remain semantically consistent with the source tables.

Approximately 5% of GPT-OSS-generated paraphrases required manual correction due to semantic shifts.

Although complete elimination of false negatives is difficult in an open-ended semantic space, the combination of global answerability checking, targeted retrieval-based inspection, and manual validation substantially reduces this risk.

A.8 Benchmark Statistics

The final benchmark contains 209 queries, including 98 EM, 55 SF, and 56 SA queries, together with 637 tables. Tables may be shared across different query instances. We additionally define a diagnostic subset restricted to sibling groups, allowing answerability to be examined under largely fixed semantic similarity.

The benchmark is intentionally moderate in corpus size because its primary purpose is controlled diagnosis rather than large-scale corpus retrieval. The difficulty is introduced through the construction of semantically similar Sibling Distractor Tables, unique answerability, large table contexts, and surface-level variations.

B Detailed Description of Embedding Models

This appendix summarizes the embedding model families evaluated in the main experiments. Since many table-specific embedding models, like DTR (Herzig et al., 2021), often use very limited context, and studies such as Wang et al. (Wang et al., 2022) have shown that table-specific embeddings do not necessarily outperform general text embeddings in the Table RAG domain, we focus only on commonly used, open-source, and relatively strong embedding models, all of which are capable of handling 8K context windows.

- **Qwen3 Embedding Family (Zhang et al., 2025b).** The **Qwen3** embedding series includes **Qwen3-E0.6B**, **Qwen3-E4B**, and **Qwen3-E8B**. These instruction-aware text embedding models support long contexts (up to 32k tokens), multilingual input (100+ languages), and Matryoshka Representation Learning (MRL) for flexible embedding dimensions. Larger variants generally provide stronger semantic representation quality.
- **Stella (Zhang et al., 2025a). stella_en_1.5B_v5** is a 1.5B-parameter English sentence embedding model optimized for semantic similarity and dense retrieval. It produces fixed-length embeddings and serves as a competitive mid-scale dense baseline.
- **Jina (Günther et al., 2025). jina-embeddings-v4** is a $\sim$3.8B-parameter multilingual and multimodal embedding model. It supports unified text-image representations, as well as both single-vector dense and multi-vector late-interaction retrieval.
- **BGE-M3 (Chen et al., 2024a). bge-m3** is a multilingual embedding model designed for dense, sparse (lexical), and multi-vector retrieval within a unified architecture. It supports long inputs (up to 8192 tokens) and enables hybrid retrieval pipelines.
- **GTE (Li et al., 2023). gte_Qwen2-7B-instruct** is a 7B-parameter instruction-tuned embedding model from the GTE series. It is optimized for semantic matching and large-scale dense retrieval.

Model Diversity. The selected models differ in parameter scale (0.6B–8B+), modality support (text-only vs. multimodal), and retrieval design (dense-only vs. hybrid/multi-vector); however, we utilize only text-only dense retrieval in this work.

Table 6: Instruction prepended to user queries for the retrieval stage.

Retrieval Instructions

```
Given a TableQA query, retrieve a relevant table that can answer
the query.  Note that the retrieved table should contain sufficient
information to provide an answer, rather than resulting in an empty
answer.  [Context-Specific Task Description]
```

C Prompts Used in Our Pipeline

C.1 Retrieval Prompt

Since we utilize the Qwen3-Embedding series, which is an instruction-aware retrieval model, we prepend a specific instruction to the user query to guide the embedding space towards TableQA relevance. The exact instruction used is detailed in Table 6. Note that traditional lexical methods, such as BM25 (Robertson & Zaragoza, 2009), do not utilize this instruction.

C.2 Downstream Question Answering Prompt

For the RAG-based generation stage, we adopt a structured prompt following the style of LongTableBench. This prompt enforces strict formatting constraints, including numerical normalization (Roman to Arabic), handling of derived columns (e.g., reversing mathematical operations), and date standardization.

As shown in Table 7, the system prompt ensures that the model outputs are parsable and normalized across diverse table schemas.

D Comprehensive Retrieval Results

This section collects the full dense-retrieval tables and related first-stage comparisons that support Section 4, including alternative paradigms, query-type breakdowns, and table-length scaling.

D.1 Dense Embedding Retrieval Results

Table 8 presents the complete benchmark results for all evaluated models across all supported formats.

D.2 Alternative Retrieval Paradigms

Beyond standard dense embedding retrieval, we further evaluate several alternative retrieval paradigms, including table-specific retrievers (DTR) (Herzig et al., 2021), ColBERT-style late-interaction retrieval (Khattab & Zaharia, 2020), sparse lexical retrieval (BM25 and SPLADE (Formal et al., 2021b;a; Lassance et al., 2024)), symbolic-neural hybrid retrieval, row-level multi-vector retrieval, cascaded retrieval (CRAFT) (Singh et al., 2026), and table-specific hybrid retrieval (THYME) (Li et al., 2025a).

Model Descriptions For DTR, we evaluate two variants: `tapas_nq_retriever_large` and `tapas_nq_hn_retriever_large` [3]. The "nq" suffix indicates that the model is fine-tuned on Natural Questions (Herzig et al., 2021), while "hn" denotes training with hard negative sampling. For ColBERT-style retrieval, we note that the original ColBERT architecture (Khattab & Zaharia, 2020) is designed for passages and does not natively support 8K-token contexts. We therefore adopt

[3] We use official DTR weights in a PyTorch reimplementation; other public adaptations (e.g., deepset/tapas-large-nq-reader) yielded inferior results.

Table 7: The structured prompt used for the downstream TableQA generation stage.

QA System Prompt

System Prompt:
Requirements:
Please read the following table and then answer the questions based on the table. Organize your answers into a list of strings, with each element being an answer item (The number of answer items is less than 11). If the question includes a special requirement, such as outputting a dictionary, please fulfill that specific request. Otherwise, always output a list of strings, even if there is only one answer item. Please place your answers between the ``` and ```.

Notes:
The table may include non-standard formats for numbers or dates. For numbers, formats may include Roman numerals, English words, or scientific notation. Whenever possible, please convert these into Arabic numerals in your response.
Additionally, some columns are derived through mathematical operations. For example, `total_add_10` indicates that the values in this column are obtained by adding 10 to the original values. You should return the original values (i.e., subtract 10 from the current values). Furthermore, for any values ending with "k" or "K", convert them into standard Arabic numerals. Please convert all dates to the format %Y-%m-%d ¡other time elements¿ (in Arabic numerals) in your response.

Additionally, even if there are duplicate answer items, you need to output all of them.

Example Output Format:
```['answer_item_1', 'answer_item_2', ..., 'answer_item_n']```

Please do not generate any text after outputting the final answer.

User Prompt:
table information:
{`table_infos`}

Question: {`query`}
Answer:

three ColBERT-style models from HuggingFace that have been extended or adapted for longer sequences, namely `Reason-ModernColBERT`[4], `SauerkrautLM-Multi-ModernColBERT`[5], and `reason-colBERT-150M-GTE-ModernColBERT`[6], as available open-source checkpoints without further fine-tuning. For sparse lexical retrieval, we consider both the classical term-matching approach BM25 and its learned counterpart `splade-v3` (Lassance et al., 2024)[7].

For row-level multi-vector retrieval, each table row (concatenated with its header) is independently embedded, and its similarity score with the query is computed separately; row-level scores are then aggregated into a single table-level score. We experimented with several aggregation strategies (mean, max, and top-k averaging) and found max aggregation to perform best; we therefore report only the max-aggregation results. For hybrid retrieval, we perform a grid search over dense-BM25 interpolation weights using Qwen3-Embedding-4B as the dense retriever; the best-performing configuration uses a dense:BM25 ratio of 9:1.

We further reproduce CRAFT (Singh et al., 2026), a multi-model cascaded retrieval framework that combines sparse and dense retrieval with LLM-based text augmentation. To adapt CRAFT to our setting, we replace its original retrieval backbones with available open-source checkpoints while preserving the overall cascading procedure. Specifically, the retrieval cascade consists of `splade-v3`, Qwen3-Embedding-0.6B, and Qwen3-Embedding-4B. We evaluate two variants. *CRAFT-Simplified*

[4] https://huggingface.co/lightonai/Reason-ModernColBERT
[5] https://huggingface.co/VAGOsolutions/SauerkrautLM-Multi-ModernColBERT
[6] https://huggingface.co/fjmgAI/reason-colBERT-150M-GTE-ModernColBERT
[7] https://huggingface.co/naver/splade-v3
```

Table 8: Comprehensive evaluation results across different input formats.

Model	Format	R@k					GR@k					DS@1
		@1	@2	@3	@4	@5	@1	@2	@3	@4	@5	
Qwen3-Embedding-0.6B	CSV	0.139	0.273	0.378	0.469	0.526	0.608	0.605	0.579	0.561	0.539	0.228
	Mixed	0.182	0.297	0.411	0.502	0.555	0.627	0.617	0.593	0.574	0.556	0.290
	HTML	0.182	0.282	0.402	0.478	0.574	0.651	0.636	0.616	0.590	0.583	0.279
	Markdown	0.153	0.306	0.383	0.455	0.555	0.603	0.596	0.573	0.548	0.532	0.254
	TSen	0.144	0.297	0.397	0.478	0.545	0.641	0.624	0.596	0.573	0.560	0.224
	TSenS	0.124	0.287	0.373	0.483	0.541	0.641	0.622	0.585	0.567	0.559	0.194
Qwen3-Embedding-4B	CSV	0.158	0.311	0.464	0.579	0.646	0.656	0.682	0.681	0.663	0.653	0.241
	Mixed	0.172	0.321	0.469	0.560	0.632	0.684	0.708	0.686	0.670	0.662	0.252
	HTML	0.134	0.287	0.474	0.584	0.679	**0.713**	**0.725**	**0.708**	**0.696**	**0.687**	0.188
	Markdown	0.187	0.340	**0.512**	0.569	0.656	0.703	0.706	0.691	0.682	0.666	0.265
	TSen	0.139	0.306	0.440	0.545	0.632	0.684	0.682	0.667	0.656	0.653	0.203
	TSenS	0.158	0.321	0.488	0.579	0.646	0.684	0.667	0.663	0.652	0.644	0.231
Qwen3-Embedding-8B	CSV	0.124	0.301	0.445	0.545	0.627	0.646	0.636	0.632	0.611	0.610	0.193
	Mixed	0.182	0.344	0.469	**0.608**	0.665	0.670	0.658	0.641	0.628	0.622	0.271
	HTML	0.153	0.335	0.488	0.598	**0.684**	0.699	0.689	0.683	0.679	0.674	0.219
	Markdown	0.177	0.349	0.493	0.565	0.632	0.660	0.648	0.646	0.634	0.629	0.268
	TSen	0.148	0.297	0.431	0.531	0.593	0.636	0.639	0.611	0.603	0.596	0.233
	TSenS	0.177	0.306	0.407	0.507	0.560	0.632	0.632	0.619	0.604	0.597	0.280
bge-m3	CSV	0.067	0.177	0.273	0.354	0.388	0.459	0.419	0.402	0.384	0.376	0.146
	Mixed	0.086	0.177	0.215	0.282	0.325	0.373	0.352	0.338	0.334	0.316	0.231
	HTML	0.043	0.139	0.225	0.263	0.292	0.349	0.347	0.346	0.341	0.330	0.123
	Markdown	0.081	0.148	0.244	0.297	0.344	0.364	0.356	0.365	0.349	0.347	0.224
	TSen	0.120	0.258	0.368	0.426	0.483	0.517	0.510	0.499	0.477	0.466	0.231
	TSenS	0.115	0.234	0.325	0.388	0.450	0.545	0.510	0.496	0.488	0.470	0.211
gte_Qwen2-7B-instruct	CSV	0.072	0.134	0.196	0.234	0.297	0.330	0.311	0.301	0.291	0.290	0.217
	Mixed	0.072	0.139	0.182	0.211	0.263	0.278	0.263	0.252	0.236	0.240	0.259
	HTML	0.038	0.086	0.124	0.172	0.211	0.244	0.239	0.231	0.227	0.228	0.157
	Markdown	0.077	0.134	0.196	0.249	0.278	0.282	0.280	0.279	0.282	0.282	0.271
	TSen	0.096	0.163	0.225	0.311	0.349	0.354	0.328	0.330	0.341	0.329	0.270
	TSenS	0.091	0.167	0.215	0.278	0.321	0.388	0.342	0.343	0.333	0.331	0.235
jina-embeddings-v4	CSV	0.129	0.282	0.368	0.416	0.498	0.612	0.603	0.581	0.560	0.541	0.211
	Mixed	0.163	0.306	0.426	0.483	0.541	0.641	0.634	0.609	0.573	0.547	0.254
	HTML	0.158	**0.368**	0.474	0.584	0.646	0.689	0.687	0.667	0.644	0.634	0.229
	Markdown	0.139	0.268	0.373	0.445	0.522	0.598	0.591	0.576	0.560	0.562	0.232
	TSen	0.177	0.273	0.368	0.474	0.522	0.550	0.548	0.530	0.513	0.503	**0.322**
	TSenS	0.158	0.239	0.364	0.459	0.522	0.560	0.560	0.544	0.533	0.521	0.282
stella_en_1.5B_v5	CSV	0.129	0.244	0.335	0.407	0.464	0.541	0.531	0.515	0.505	0.501	0.239
	Mixed	0.096	0.172	0.244	0.287	0.340	0.526	0.476	0.448	0.417	0.405	0.182
	HTML	0.048	0.105	0.172	0.220	0.258	0.349	0.330	0.309	0.297	0.292	0.137
	Markdown	0.139	0.244	0.325	0.407	0.455	0.517	0.514	0.494	0.493	0.479	0.269
	TSen	**0.206**	0.306	0.397	0.502	0.550	0.641	0.615	0.609	0.598	0.582	0.321
	TSenS	0.163	0.292	0.397	0.493	0.560	0.632	0.632	0.617	0.602	0.588	0.258

removes the LLM-based augmentation component and therefore consists solely of the retrieval cascade. The full CRAFT variants additionally apply LLM-based augmentation using either Qwen3-30B-A3B-2507-Thinking or Qwen3.6-27B. This setup allows us to separately assess the effectiveness of the cascaded retrieval architecture and the LLM-based augmentation component under our long-context table setting.

Finally, we reproduce THYME (Li et al., 2025a), a table-specific retrieval method that adopts a column-oriented representation and combines multiple retrieval signals over table fields. THYME is particularly relevant as a direct comparison with DTR: both are table-specific retrieval architectures trained specifically for table retrieval, rather than general-purpose retrievers applied to tables. We evaluate THYME using `bert-base-uncased`, `bert-large-uncased`, and `splade_cocondenser_selfdistil`, corresponding to the backbone configurations considered in the original work. We preserve its column-oriented retrieval paradigm as closely as possible without additional training on our benchmark.

Results Table 9 summarizes the retrieval performance across paradigms. Overall, most alternative retrieval paradigms underperform the strongest dense and row-level multi-vector baselines in our setting. DTR performs particularly poorly, with both variants achieving R@1 below 1%. THYME performs better, with its strongest configuration reaching 0.062 R@1 and 0.371 DS@1, but remains substantially behind the strongest general-purpose retrieval methods.

CRAFT provides a competitive cascaded retrieval paradigm. *CRAFT-Simplified*, which removes LLM-based augmentation, achieves 0.215 R@1 and 0.313 DS@1, essentially matching the dense-BM25 hybrid baseline. However, adding LLM-based augmentation does not provide additional gains: the full CRAFT variants using Qwen3-30B-A3B-2507-Thinking and Qwen3.6-27B achieve lower R@1 and DS@1 than the simplified variant. Notably, the Qwen3.6-27B configuration performs worse than the Qwen3-30B-A3B-2507-Thinking configuration, suggesting that using a later or more capable augmentation model does not necessarily improve retrieval under our setting. This is also consistent with the limited benefits of LLM-based augmentation observed across the augmentation strategies evaluated in Appendix D.3.

For THYME, we adopt three backbone model families as the base models for training, namely `bert-base-uncased`, `bert-large-uncased`, and `splade_cocondenser_selfdistil`. We follow the original training methodology as closely as possible to ensure a fair reproduction of the experimental setup. The resulting performance remains limited, with the `bert-large-uncased` configuration achieving the best R@1 of 0.062. Since THYME and DTR are both table-specific retrieval architectures trained specifically for table retrieval, their results provide a direct comparison with specialized table retrieval approaches. THYME substantially outperforms DTR but still falls behind general-purpose dense and row-level multi-vector retrieval. Under our long-context heterogeneous table setting, the specialized table retrieval architectures considered here therefore do not close the Target–Sibling gap relative to the stronger general-purpose baselines in this comparison.

Analysis Across the paradigms we evaluate, the results reveal a consistent Semantic-Answerability Gap between retrieving a relevant table group and identifying the uniquely answerable table. DTR and THYME, despite being table-specific architectures trained for table retrieval, achieve substantially lower R@1 than the strongest general-purpose retrievers in this comparison. ColBERT-style and sparse retrievers improve recall considerably, but their DS@1 remains limited. The dense-BM25 hybrid and CRAFT-Simplified provide competitive performance, yet neither substantially closes the gap between group-level retrieval and exact table retrieval.

Row-level multi-vector retrieval is the strongest first-stage paradigm in this comparison, achieving the highest R@1 and GR@1 among the evaluated approaches. Nevertheless, its DS@1 remains substantially below its GR@1, indicating that even strong retrievers in this set can reach the correct topical group without reliably selecting the table that contains sufficient evidence for answering the query. This pattern persists across the dense, late-interaction, sparse, hybrid, cascaded, and table-specific paradigms we test.

Within our evaluation, the observed limitation is therefore not explained by the choice of first-stage architecture alone. Distinguishing the uniquely answerable table appears to require fine-grained matching of schema, values, and query constraints that the semantic, lexical, and table-specific

Table 9: Retrieval performance across alternative retrieval paradigms.

Model	Type	R@1	R@3	R@5	GR@1	DS@1
tapas_nq_hn_retriever_large	DTR	0.010	0.014	0.019	0.038	0.250
tapas_nq_retriever_large	DTR	0.005	0.010	0.010	0.005	1.000
Reason-ModernColBERT	ColBERT-style	0.086	0.196	0.282	0.411	0.209
SauerkrautLM-Multi-ModernColBERT	ColBERT-style	0.172	0.474	0.632	0.766	0.225
reason-colBERT-150M-GTE-ModernColBERT	ColBERT-style	0.086	0.239	0.340	0.435	0.198
BM25	Sparse	0.005	0.005	0.014	0.005	1.000
splade-v3	Sparse	0.091	0.244	0.368	0.392	0.232
Qwen3-E4B + BM25 hybrid	Hybrid	0.215	0.455	0.612	0.689	0.312
Qwen3-E0.6B	Row-level multi-vector	0.263	0.488	0.627	0.699	0.377
Qwen3-E4B	Row-level multi-vector	0.249	0.507	0.679	0.732	0.340
Qwen3-E8B	Row-level multi-vector	0.282	0.512	0.660	0.694	0.407
CRAFT-Simplified	Cascade	0.215	0.464	0.646	0.689	0.313
CRAFT (Qwen3-30B-A3B-2507-Thinking)	Cascade + LLM Aug.	0.196	0.450	0.589	0.670	0.293
CRAFT (Qwen3.6-27B)	Cascade + LLM Aug.	0.187	0.455	0.593	0.684	0.273
bert-base-uncased	THYME	0.014	0.062	0.086	0.067	0.214
bert-large-uncased	THYME	0.062	0.096	0.166	0.168	0.371
splade_cocondenser_selfdistil	THYME	0.014	0.096	0.158	0.134	0.107

representations we evaluate do not capture adequately. This is consistent with answerability acting as a retrieval signal that is only partially reflected by relevance-oriented similarity.

D.3 LLM-Augmented Dense Retrieval

We further evaluate several augmentation strategies commonly used in retrieval pipelines to determine whether the observed Semantic-Answerability Gap can be mitigated by adding auxiliary semantic signals.

Experimental Setup Inspired by prior work (Gao et al., 2023; Shen et al., 2024), we evaluate two semantic augmentation strategies using Qwen3-30B-A3B-Thinking-2507:

- **Meta-Info Generation.** For each table, we generate descriptive metadata summarizing table themes, column semantics, and value distributions. The generated metadata is concatenated with the table content before embedding.
- **Query-to-Table (Q2T).** Given a query, the model first generates a hypothetical "answer table". Retrieval is then performed by matching this generated table against candidate tables in the corpus.

Results Table 10 summarizes the results under the Qwen3-Embedding-8B retrieval backbone.

Table 10: Retrieval performance of augmentation strategies across different embedding models.

Method	Retrieval Model	R@1	R@3	R@5	GR@1	GR@3	GR@5	DS@1
Meta-Augmentation	Qwen3-Embedding-0.6B	0.144	0.426	0.584	0.646	0.770	0.833	0.222
	Qwen3-Embedding-4B	0.196	0.512	0.670	0.746	0.833	0.880	0.263
	Qwen3-Embedding-8B	0.153	0.498	0.660	0.742	0.785	0.837	0.206
Query-to-Table	Qwen3-Embedding-0.6B	0.153	0.388	0.478	0.550	0.651	0.679	0.278
	Qwen3-Embedding-4B	0.144	0.368	0.545	0.541	0.660	0.713	0.265
	Qwen3-Embedding-8B	0.153	0.402	0.502	0.565	0.632	0.684	0.271

Analysis Both semantic augmentation methods provide only limited improvements over the baseline dense retrievers. Meta-information generation slightly improves recall metrics but does not substantially improve the ranking of Target Tables. The Query-to-Table strategy frequently introduces hallucinated structures in the generated tables, which inject additional noise into the retrieval process.

Table 11: Retrieval performance by query type using Qwen3-Embedding-8B.

Type	R@1	R@3	R@5	GR@1	DS@1
All	0.182	0.469	0.665	0.670	0.271
EM	0.224	0.531	0.724	0.755	0.297
SF	0.182	0.418	0.600	0.564	0.323
SA	0.107	0.411	0.625	0.625	0.171

Table 12: Downstream QA performance by query type using Qwen3-Embedding-8B.

Type	Oracle	QA@1	QA@3	QA@5	Eff@1	Eff@3	Eff@5
All	0.755	0.139	0.266	0.330	1.014	0.751	0.657
EM	0.884	0.194	0.415	0.486	0.977	0.885	0.759
SF	0.711	0.129	0.180	0.204	0.998	0.605	0.479
SA	0.571	0.054	0.089	0.179	0.876	0.381	0.500

These observations suggest that, among the strategies we evaluate, simply enriching the semantic representation of tables is not sufficient to resolve ambiguity among structurally similar tables on TCR-Bench. In contrast, Answerability-Aware Reranking (AAR) is more effective in our setting because it enables direct query-table interaction during scoring.

D.4 Retrieval and Downstream QA Performance by Query Type

To further characterize where retrieval and reasoning difficulties arise, we report a breakdown of retrieval and downstream QA performance by query type using Qwen3-Embedding-8B as the retriever.

The breakdown reveals three main patterns. First, EM is the easiest query type overall, achieving the highest group recall, exact retrieval, and downstream QA performance among the three types. Even so, its DS@1 remains near the random-selection baseline, indicating that a Semantic-Answerability Gap is present even for the easiest queries. Second, SA is the hardest type for exact answerability discrimination, obtaining the lowest R@1 and DS@1 of the three types. Third, coarse retrieval and within-group discrimination behave as distinct sources of difficulty: SF has the lowest GR@1 among the three types, yet its DS@1 is higher than that of both EM and SA, showing that difficulty in locating the correct semantic group is not identical to difficulty in identifying the target within that group once it has been located.

These results suggest that retrieval difficulty and reasoning difficulty are related but not interchangeable. Given SA's low retrieval performance and low Oracle score, the non-monotonic Eff@k pattern for this type should be interpreted with caution: SA has comparatively few queries, so the effective sample after retrieval is small and downstream QA estimates can be unstable. We therefore refrain from drawing strong conclusions from minor numeric variations in Eff@k.

D.5 Effect of Table Length on Retrieval Performance

Setup. On TCR-Bench, the evaluated dense retrievers have difficulty locating the uniquely answerable Target among structurally similar Sibling Distractor Tables. To test whether this difficulty stems from semantic dilution caused by excessively long tables, we apply row sampling to every table in the candidate pool, reducing each table to $1/n$ of its original number of rows, where $n \in \{2, 3, 4, 5, 8, 10, 16\}$. For the Target Table, all rows relevant to the answer are retained, so that answer-bearing evidence is not removed during sampling. We report R@1, GR@1, and DS@1 across sampling ratios on three embedding models (Qwen3-E4B, Stella, and Jina); results are shown in Figure 6.

Results and analysis. All three models exhibit a consistent trend: as the sampling ratio decreases (i.e., tables become smaller), R@1, GR@1, and DS@1 improve to varying degrees. For Qwen3-E4B, R@1 rises from 0.172 at full length to 0.254 in the $1/16$ regime, with GR@1 improving

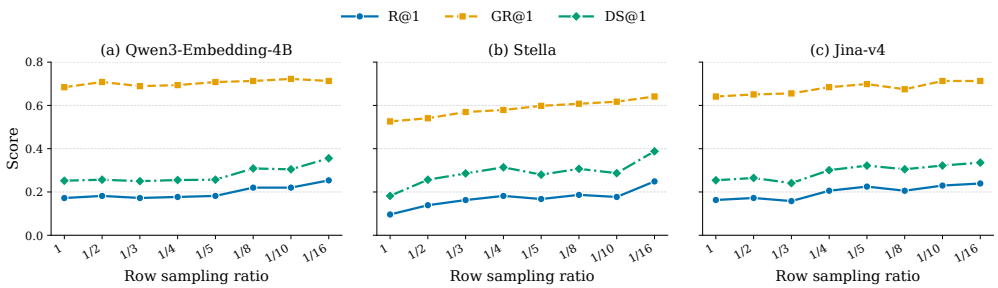


Figure 6: Effect of row-sampling ratio on retrieval performance. Each panel corresponds to one embedding model (Qwen3-E4B, Stella, Jina); curves show $R@1$, $GR@1$, and $DS@1$ against the retained-row ratio $1/n$.

Table 13: Statistics of the naturally occurring Real-TCR collections.

Benchmark	#Queries	#Tables	Table Length	Random DS@1
Real-TCR	77	217	3K–8K	0.047
Real-TCR-small	38	117	3K–8K	0.100

from 0.684 to approximately 0.713. The improvement is more pronounced for Stella and Jina; for instance, Stella's DS@1 rises from 0.182 at full length to 0.388 under $1/16$ sampling, while Jina's DS@1 rises from 0.254 to 0.336. This is consistent with the hypothesis that excessive table length dilutes the query-relevant signal for these dense embeddings.

However, even at the lowest sampling ratio (i.e., the smallest table size), the absolute values of R@1 and DS@1 remain low: none of the three models exceeds an R@1 of 0.26 or a DS@1 of 0.39. Shrinking tables therefore partially mitigates retrieval difficulty for these models, but does not eliminate Target–Sibling confusion. Even when sibling tables are compressed to a small scale, the evaluated models still do not consistently identify the uniquely answerable target. Within this length-scaling experiment, residual errors are thus unlikely to be explained by table length alone and are more consistent with limited sensitivity to fine-grained structural or value-level distinctions among near-duplicate tables.

E Real-TCR: Natural-Negative Validation

TCR-Bench is intentionally constructed as a controlled stress test in which sibling tables share coarse semantic and structural characteristics while differing in whether the evidence required by a query is present. While this controlled construction makes answerability measurable and enables systematic analysis, it also raises a natural concern: whether the observed Semantic-Answerability Gap is specific to the benchmark construction or also occurs in naturally organized table collections.

To address this concern, we construct **Real-TCR**, an additional evaluation collection from naturally occurring tables in a real-world environment. Unlike TCR-Bench, its candidate tables are not generated through answerability-targeted transformations. Instead, they follow natural partitions already present in the underlying data, such as by time window or organizational unit. As a result, the tables exhibit naturally occurring variation in schema, column sets, and length. The data are substantially different in both content and organization from the open-source data used in TCR-Bench, providing an additional evaluation in a naturally organized setting.

Real-TCR contains 77 queries and 217 tables, while Real-TCR-small contains 38 queries and 117 tables, with candidate groups capped at 10 tables. Table lengths range from 3K to 8K tokens. The corresponding random DS@1 baselines are 0.047 and 0.100, respectively.

The queries and target tables were manually annotated by personnel of an industry partner organization, followed by constraint verification, false-negative retrieval, and manual inspection. Additional information was available during annotation to perform global consistency checks and verify target

Table 14: Retrieval results on naturally occurring Real-TCR collections.

Benchmark	Model	R@1	GR@1	DS@1	Random DS@1
Real-TCR	Qwen3-E4B	0.039	0.909	0.043	0.047
	Jina	0.143	0.948	0.151	0.047
	Stella	0.104	0.870	0.119	0.047
Real-TCR-small	Qwen3-E4B	0.079	0.921	0.086	0.100
	Jina	0.184	0.947	0.194	0.100
	Stella	0.105	0.868	0.121	0.100

uniqueness. The underlying data consist of heterogeneous real-world operational records obtained under a data-use agreement. All data were de-identified prior to use and cannot be publicly released due to contractual constraints; Real-TCR therefore serves as a non-releasable external validation rather than a publicly reproducible benchmark.

The same qualitative pattern observed on TCR-Bench appears for the evaluated retrievers on this natural collection. For example, Jina achieves GR@1 = 0.948 on Real-TCR but only R@1 = 0.143 and DS@1 = 0.151. Qwen3-E4B achieves GR@1 = 0.909 but DS@1 = 0.043. Thus, these retrievers can identify the relevant semantic group with high accuracy while still struggling to select the exact table containing sufficient information to answer the query.

Real-TCR therefore provides a natural-negative validation of the phenomenon under a substantially different data organization. Unlike TCR-Bench, where distractors are constructed through controlled perturbations, the candidate tables in Real-TCR arise from naturally occurring partitions of real-world data. The persistence of the same qualitative pattern on this collection provides additional evidence that the observed gap is not an artifact unique to the construction procedure of TCR-Bench. We do not claim that every real-world data collection exhibits the same quantitative difficulty; rather, the shared qualitative pattern across the controlled benchmark and this naturally organized collection supports the diagnostic relevance of the finding beyond synthetic sibling construction alone.

F Complete Surface Variation Results

This section reports the query-variant definitions, complete retrieval numbers, and pairwise consistency statistics for the surface-variation analysis in Section 5.1. The results support the finding that Target–Sibling discrimination is not primarily an artifact of serialization format or query wording in our evaluation; they do not introduce a separate conclusion.

F.1 Query Variant Definitions

We evaluate three query variants, as summarized in Table 15.

Table 15: Summary of the three query variants.

Variant	**Format**	**Example**
Final NL	Natural-language question	"Which country won more than 29 gold medals in the 2016 Olympics?"
Sentential	Conditions expressed as separate sentences; target requested at the end	"The year is 2016. The number_of_gold_medals is greater than 29. Country is?"
Template	Structured templates with explicit placeholders for attributes, values, and conditions	"What is the Country when Year = 2016 AND Gold ¿ 29?"

For template queries, we use three sets of fixed patterns, each targeting a different query intention:

- **EM (Exact-Match Lookup):** Retrieve target values for rows satisfying conditions. *Patterns:* `"What is the {target} when {conds}?"`, `"Please find the {target} for rows where {conds}."`, `"List the {target} for the entries with {conds}."`, `"Which {target} values correspond to {conds}?"`
- **SF (Simple Filtering):** Similar to EM, with consistent emphasis on row-based filtering. *Patterns:* `"What is the {target} when {conds}?"`, `"List the {target} for rows where {conds}."`, `"Which {target} values correspond to rows where {conds}?"`, `"Please find the {target} for entries with {conds}."`
- **SA (Simple Aggregation):** Apply aggregation functions (e.g., COUNT, SUM, AVG) over the target. *Patterns:* `"What is the {agg} of {target} when {conds}?"`, `"Compute the {agg} of {target} for rows where {conds}."`, `"Find the {agg} value of {target} corresponding to rows where {conds}."`, `"Please calculate the {agg} of {target} for entries with {conds}."`

The condition placeholder `{conds}` is instantiated as conjunctions of attribute–value comparisons using symbolic operators (e.g., `Year = 2016 AND Gold > 29`).

Table 16: Complete retrieval results on the Diagnostic Subset under different table serialization formats.

Model	Format	R@k			GR@1	DS@1
		@1	@3	@5		
Qwen3-E4B	CSV	0.186	0.571	**0.771**	0.814	0.228
	Mixed	0.186	0.586	0.729	0.814	0.228
	HTML	0.200	0.543	0.757	**0.857**	0.233
	Markdown	**0.257**	0.557	0.729	0.814	0.316
	TSen	0.200	0.529	0.743	0.800	0.250
	TSenS	0.243	**0.614**	0.714	0.757	**0.321**
Stella	CSV	0.171	0.414	0.543	0.671	0.255
	Mixed	0.143	0.357	0.500	0.657	0.217
	HTML	0.086	0.257	0.414	0.629	0.136
	Markdown	0.171	0.386	0.514	0.629	0.273
	TSen	**0.314**	**0.500**	0.643	**0.786**	**0.400**
	TSenS	0.286	0.471	**0.671**	0.771	0.370

Table 17: Complete retrieval results under different query formulations.

Model	Query Type	R@k			GR@1	DS@1
		@1	@3	@5		
Qwen3-E4B	Final (NL)	0.186	**0.586**	**0.729**	**0.814**	0.228
	Sentential	0.229	0.471	**0.729**	0.800	0.286
	Template	**0.257**	0.571	0.714	0.786	**0.327**
Stella	Final (NL)	**0.143**	**0.357**	**0.500**	0.657	**0.217**
	Sentential	0.100	0.343	**0.500**	0.629	0.159
	Template	0.100	0.329	0.471	**0.714**	0.140

F.2 Consistency Analysis Across Surface Variations

We further compare retrieval outputs pairwise across serialization formats and query formulations, using the metrics below.

Table 18: Consistency analysis across serialization formats and narrative structures.

Model	Config 1 vs. 2	Hit Consis.	Ranking (RBO)	Partial Consis.	Top-1 Id.
Qwen3-E4B	Mixed vs. CSV	85.71%	29.22%	97.14%	58.57%
	Mixed vs. HTML	87.14%	27.76%	94.29%	45.71%
	Mixed vs. Markdown	81.43%	28.47%	97.14%	48.57%
	Mixed vs. TSen	75.71%	25.88%	92.86%	38.57%
	Mixed vs. TSenS	82.86%	25.06%	88.57%	41.43%
	TSen vs. TSenS	**84.29%**	**30.14%**	**94.29%**	**58.57%**
Stella	Mixed vs. CSV	88.57%	24.55%	88.57%	52.86%
	Mixed vs. HTML	82.86%	19.98%	71.43%	42.86%
	Mixed vs. Markdown	88.57%	25.87%	92.86%	55.71%
	Mixed vs. TSen	77.14%	20.43%	82.86%	34.29%
	Mixed vs. TSenS	74.29%	20.05%	78.57%	35.71%
	TSen vs. TSenS	**91.43%**	**34.43%**	**95.71%**	**77.14%**

Table 19: Consistency analysis across query variations (Base: Final Query).

Model	vs. Query	Hit Consis.	Ranking (RBO)	Partial Consis.	Top-1 Id.
Qwen3-E4B	Template	84.29%	31.67%	97.14%	58.57%
	Sentential	84.29%	27.65%	90.00%	48.57%
Stella	Template	95.71%	30.92%	94.29%	70.00%
	Sentential	90.00%	28.46%	90.00%	60.00%

- **Hit Consistency**: the proportion of instances where top-1 retrieval outcomes (hit vs. miss) are identical across two configurations.
- **Partial Consistency**: the proportion of cases where the top-1 table from the base configuration appears within the top-5 results of the comparison configuration.
- **Ranking Consistency**: Rank-Biased Overlap (RBO, $p = 0.9$) (Webber et al., 2010) between Top-5 ranking lists. RBO emphasizes higher-ranked items while accounting for list overlap.
- **Top-1 Identity**: the percentage of cases where the exact same table ID appears at rank 1 in both configurations.

High Hit or Partial Consistency with lower Ranking Consistency or Top-1 Identity corresponds to stable hit/miss behavior and overlapping candidates, but shifting order within the list.

F.3 Serialization Format Consistency

Table 18 reports consistency metrics across serialization formats using the Mixed format as the base configuration.

Hit Consistency and Partial Consistency are high for most format pairs, so target hit/miss and the base top-1's presence in the comparison top-5 are largely preserved. The TSen vs. TSenS comparison is also high, which is consistent with limited sensitivity to within-row attribute order. Ranking Consistency is lower (RBO around 20–30%): candidate sets remain similar, but order within the top-5 often changes. These statistics are consistent with coarse semantic matching remaining stable under serialization changes, while fine-grained scores still move.

F.4 Query Formulation Consistency

Table 19 reports the same metrics using the Final (NL) query as the base configuration.

Hit and Partial Consistency remain high under query reformulation, while Top-1 Identity is only moderate. As with serialization, this indicates that candidate membership is largely preserved and that remaining differences are mainly ranking shifts. Together with Table 17, the results are consis-

tent with the observation that query phrasing is not the primary driver of the Target–Sibling gap for the two evaluated models.

G Detailed Dense Retrieval Diagnostics

This section provides detailed perturbation settings, illustrative examples, and complete results for the diagnostic experiments in Section 5.2.

G.1 Semantic-volume perturbation

We investigate whether dense retrieval is sensitive to the amount of semantic content or to the information required for identification. For each table row, we construct three sentence-based representations. **Sen** converts the complete row into a declarative sentence while preserving the original attribute order. **SenS** applies the same conversion but randomly shuffles the attribute order within the sentence. **SenSS** retains only a subset of column-value pairs that is sufficient to uniquely identify the relevant row.

For illustration, given a row containing the attributes *Year*, *Country*, *Product*, and *Sales*, the representations can be written as follows:

> **Sen:** In 2020, the country was France, the product was Wine, and sales were 120.
>
> **SenS:** The product was Wine, sales were 120, in 2020, and the country was France.
>
> **SenSS:** In 2020, the product was Wine and sales were 120.

Sen and SenS preserve the complete row, whereas SenSS retains only a subset sufficient for identification in the corresponding diagnostic case. SenS isolates sensitivity to attribute order, while SenSS examines whether retrieval benefits from additional semantic content beyond what is logically sufficient for identification.

We evaluate these representations at both the inter-table and intra-table levels. At the inter-table level, the task is to identify the Target Table from the candidate tables. At the intra-table level, the task is to retrieve the relevant row or a chunk containing it from the Target Table.

The results show that Sen and SenS generally outperform SenSS despite SenSS retaining sufficient information for identification, while the difference between Sen and SenS remains small. This indicates that the evaluated embeddings are more responsive to the amount of semantic content than to attribute order or minimal information sufficiency.

At the intra-table level, row-level retrieval substantially outperforms retrieval over 10-line chunks. Figure 4 presents the corresponding results. This contrast indicates that the evaluated dense embeddings can capture local row-level content while suffering from information dilution at larger granularities.

G.1.1 Detailed Intra-table Retrieval Results

Table 20 reports the exact R@k values for row-level and chunk-level retrieval under the Sen, SenS, and SenSS representations.

G.1.2 Comparison of 10-line and 20-line Chunk Retrieval

Table 21 further compares 10-line and 20-line chunk settings. Increasing chunk size does not yield consistent improvements: for Qwen3-E4B, $R@1$ rises from 0.387 (10-line) to 0.432 (20-line) under Sen, and SenS improves by a similar margin, whereas SenSS declines; Stella shows modest $R@1$ gains, but SenSS $R@3$ drops from 0.645 to 0.561. Aggregating more rows therefore does not reliably help and can further dilute query-relevant information.

Table 20: Intra-table retrieval by granularity and representation: row-level retrieval outperforms 10-line chunks for both evaluated models; SenSS yields lower accuracy despite retaining a logically sufficient subset.

Model	Config	Type	R@1	R@3	R@5
Qwen3-E4B	Row	Sen	**0.871**	0.903	**0.929**
		SenS	0.845	**0.916**	**0.929**
		SenSS	0.619	0.735	0.800
	Chunk	Sen	**0.387**	**0.587**	**0.742**
		SenS	0.335	0.561	0.723
		SenSS	0.284	0.490	0.665
Stella	Row	Sen	**0.852**	**0.890**	**0.903**
		SenS	0.800	0.858	0.877
		SenSS	0.619	0.729	0.755
	Chunk	Sen	**0.497**	**0.710**	**0.832**
		SenS	0.471	0.677	0.819
		SenSS	0.400	0.645	0.768

Table 21: 10-line vs 20-line Chunk Retrieval Comparison.

Model	Type	10-line R@1	20-line R@1	Δ	10-line R@3	20-line R@3	Δ
Qwen3-E4B	Sen	0.387	0.432	+0.045	0.587	0.619	+0.032
	SenS	0.335	0.387	+0.052	0.561	0.619	+0.058
	SenSS	0.284	0.271	-0.013	0.490	0.445	-0.045
Stella	Sen	0.497	0.516	+0.019	0.710	0.742	+0.032
	SenS	0.471	0.516	+0.045	0.677	0.703	+0.026
	SenSS	0.400	0.413	+0.013	0.645	0.561	-0.084

G.2 Row-column binding perturbation

We next examine whether dense representations preserve **row-column bindings**, namely, whether values remain associated with the columns and rows in which they originally occur. Analogous to the three sentence-based representations above, for each table in the Diagnostic Subset we construct a Minimal Contrastive Set with three configurations: **Ori** is the original table; **RID** augments each cell with its row identifier; and **RID+Shuf** further shuffles values within each column while keeping the row identifiers attached to their original values. Thus Ori and RID preserve row-column bindings, with RID making row identity explicit, whereas RID+Shuf retains the same values, columns, row identifiers, and entity alignments, but breaks the original row-column bindings. In other words, all three variants preserve the alignment between entities and their associated values, allowing us to isolate the effect of row-level bindings.

For example, given an original table

A	B
1	2
3	4

the **RID** configuration augments each value with its original row identifier:

A	B
r0:1	r0:2
r1:3	r1:4

The **RID+Shuf** configuration then independently shuffles values within each column while retaining their row identifiers:

A	B
r0:1	r1:4
r1:3	r0:2

Full results across all formats appear in Table 22. This controlled construction tests whether the embedding is sensitive to fine-grained row–column associations rather than to shared lexical content alone: RID+Shuf keeps explicit row indices and the same cell values, yet rearranges their within-column placements.

Table 22: Full retrieval performance across all table formats and configurations.

Model	Config	Format	R@1	R@2	R@3	GR@1	DS@1
Qwen3-E4B	Ori	CSV	0.400	0.671	0.714	0.686	0.583
	Ori	Mixed	0.443	0.700	0.800	0.671	0.660
	Ori	HTML	0.386	0.686	0.800	0.671	0.574
	Ori	Markdown	0.343	0.671	0.729	0.643	0.533
Stella	Ori	CSV	0.229	0.529	0.571	0.557	0.410
	Ori	Mixed	0.229	0.443	0.529	0.471	0.485
	Ori	HTML	0.257	0.414	0.514	0.443	0.581
	Ori	Markdown	0.214	0.514	0.586	0.500	0.429
Qwen3-E4B	RID	CSV	0.357	0.729	0.757	0.629	0.568
	RID	Mixed	0.371	0.700	0.743	0.671	0.553
	RID	HTML	0.314	0.700	0.786	0.686	0.458
	RID	Markdown	0.400	0.657	0.743	0.657	0.609
Stella	RID	CSV	0.300	0.543	0.600	0.543	0.553
	RID	Mixed	0.243	0.429	0.500	0.471	0.515
	RID	HTML	0.200	0.329	0.343	0.329	0.609
	RID	Markdown	0.243	0.429	0.500	0.471	0.515
Qwen3-E4B	RID+Shuf	CSV	0.271	0.629	0.686	0.629	0.432
	RID+Shuf	Mixed	0.314	0.686	0.729	0.643	0.489
	RID+Shuf	HTML	0.371	0.657	0.743	0.671	0.553
	RID+Shuf	Markdown	0.343	0.686	0.743	0.714	0.480
Stella	RID+Shuf	CSV	0.200	0.471	0.529	0.500	0.400
	RID+Shuf	Mixed	0.200	0.400	0.457	0.429	0.467
	RID+Shuf	HTML	0.100	0.286	0.329	0.329	0.304
	RID+Shuf	Markdown	0.186	0.414	0.514	0.429	0.433

Across formats, Ori and RID both outperform RID+Shuf in most settings (e.g., Qwen3-E4B Mixed: $0.443/0.371$ vs. 0.314; Stella Mixed: $0.229/0.243$ vs. 0.200). Even when row identity is made explicit, disrupting within-column value placements still degrades discrimination. This suggests that the evaluated embeddings capture shared lexical and semantic content relatively well, but remain comparatively insensitive to fine-grained row–column binding.

G.3 Analysis of the Embedding Space

We analyze the geometry of table-level embeddings as a diagnostic of the dense retrieval setting underlying the observed discrepancy between $GR@1$ and $R@1$. The goal is to characterize three aspects of the representation space: whether the global embedding inventory exhibits an obvious collapse; whether local Target+Sibling groups form compact semantic neighborhoods; and whether query-to-table scoring distinguishes the answerable Target from nearby Siblings. The quantities below are standard spectral summaries of embedding sets (also used in related analyses such as (Vangara & Gopinath, 2026)); we use them only as diagnostics for the retrieval behavior in TCR-Bench, not as a geometric theory of the Semantic-Answerability Gap and not as a test of that work's consolidation theory.

Setup. We examine two scales.

- **Global (all 637 tables):** whether tables in the corpus are concentrated on a few shared directions or remain spread out.
- **Local (per query):** for each query, the answerable Target together with its Sibling Distractors; whether these tables form a compact group, and whether the query score prefers the Target inside that group.

The local group is the sibling set used by $GR@k$ in the main paper. We report:

- d_{eff} (*effective dimensionality*): how many independent directions carry most of the variance; smaller values indicate a thinner subspace.
- $\bar{d}$ (*mean pairwise cosine distance*): average $1 - \cos$ between pairs; larger values indicate greater separation.
- $\rho = \lambda_1 / \sum_i \lambda_i$ (*spectral concentration*): fraction of variance on the leading principal direction.
- $\Delta_q = \cos(q, t_q) - \max_{s \in S_q} \cos(q, s)$: the Target–Sibling cosine margin. Positive values mean the Target wins under cosine ranking; negative values mean a Sibling scores higher.

Global sanity check. Table 23 reports d_{eff} and $\bar{d}$ over all 637 tables. Mean pairwise distances are substantial ($\bar{d} \in [0.40, 0.55]$), and effective dimensionality remains in the tens (43–59). The corpus-level embedding space therefore does not look like a single collapsed blob. We treat these numbers only as a sanity check: they describe the full table inventory, not the small sibling groups associated with individual queries.

Table 23: Corpus-level geometry over all 637 tables (sanity check).

Model	Format	d_{eff}	$\bar{d}$
Qwen3-Embedding-4B	Mixed	55.947	0.5257
Qwen3-Embedding-4B	CSV	45.154	0.4818
Qwen3-Embedding-4B	HTML	58.698	0.5357
Qwen3-Embedding-4B	Markdown	56.038	0.5394
Qwen3-Embedding-0.6B	Mixed	50.795	0.4943
stella_en_1.5B_v5	Mixed	43.447	0.5491
jina-embeddings-v4	Mixed	49.654	0.3958

Table 24: Local embedding geometry on Target+Sibling groups. $d_{\text{eff}}^{\text{loc}}$ and ρ measure local dimensionality and spectral concentration, $\bar{d}^{\text{loc}}$ is mean pairwise cosine distance, and Δ_q measures the Target's margin over its nearest Sibling.

Model	Format	$d_{\text{eff}}^{\text{loc}}$ med.	ρ med.	$\bar{d}^{\text{loc}}$ med.	Δ_q mean	$R@1$	$GR@1$	$DS@1$
Qwen3-E4B	Mixed	2.69	0.50	0.106	−0.024	0.172	0.684	0.252
Qwen3-E0.6B	Mixed	2.79	0.47	0.113	−0.027	0.182	0.627	0.290
Qwen3-E0.6B	CSV	2.73	0.49	0.110	−0.026	0.139	0.608	0.228
Qwen3-E0.6B	HTML	2.70	0.52	0.084	−0.022	0.187	0.651	0.287
Qwen3-E0.6B	Markdown	2.72	0.51	0.092	−0.021	0.153	0.603	0.254
Stella	Mixed	2.39	0.57	0.104	−0.055	0.096	0.526	0.182
Jina	Mixed	2.94	0.46	0.088	−0.021	0.163	0.641	0.254

Local geometry and the $GR@1$ vs. $R@1$ gap. Table 24 summarizes the per-query local analysis over 209 queries. Sibling groups are compact: median local cosine distance is only about 0.08–0.11, and median local d_{eff} is about 2.4–2.9. Related tables therefore occupy a compact local region even though the full corpus remains spread out globally. This local structure is consistent with the retrieval behavior observed in the main results: models frequently retrieve a table from the correct sibling group without ranking the Target first. Concretely, for Mixed embeddings, 43%–51% of queries have $GR@1 = 1$ and $R@1 = 0$ (top-1 is a Sibling, not the Target). Figure 7 shows the same contrast on Qwen3-Embedding-4B Mixed: the corpus stays spread in the leading principal plane, a cosine t-SNE pulls each sibling group into a tight pocket, and the sign of Δ_q appears as a within-pocket ranking between the Target and a Sibling. A compact main-text view of panels (b)–(d) appears as Figure 5.

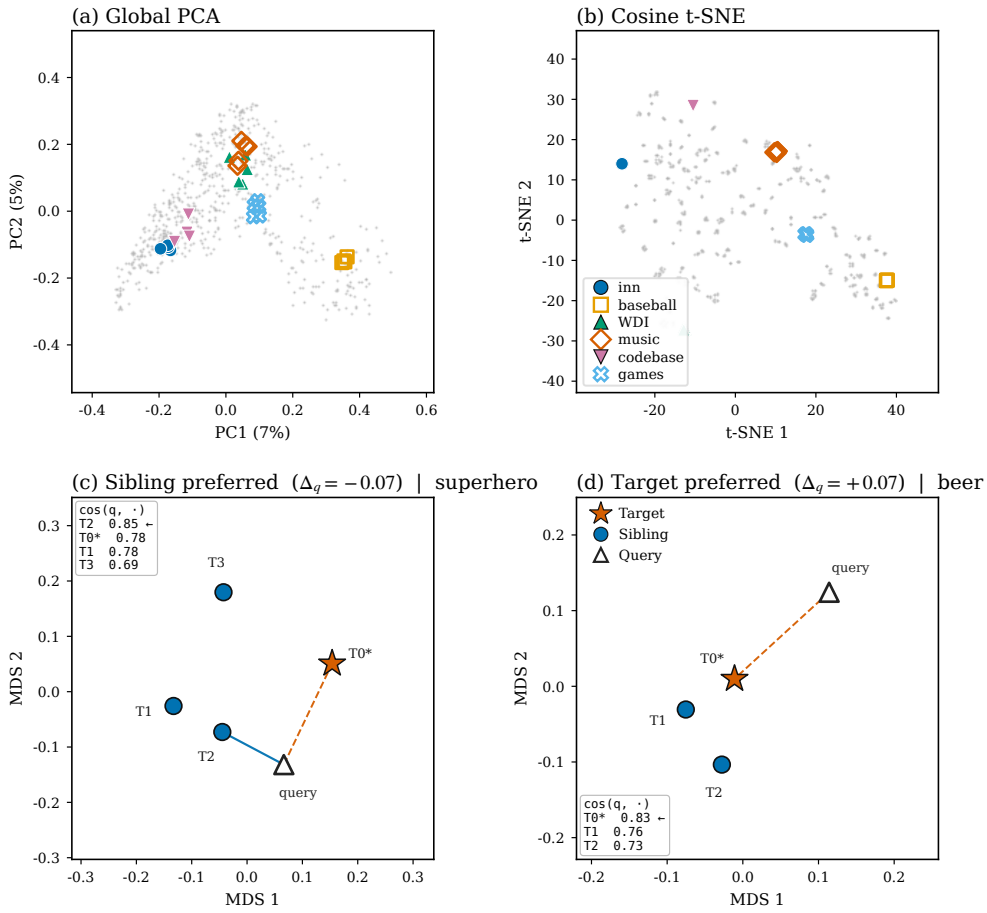


Figure 7: Embedding geometry of Qwen3-Embedding-4B (Mixed) on the 637-table corpus. Colored points are six Target+Sibling groups; gray points are the remaining tables. (a) The leading two principal components explain only 12% of the variance. Highlighted groups sit in different parts of a spread cloud rather than in one collapsed blob, consistent with $d_{\text{eff}} \approx 56$ in Table 23. (b) A cosine t-SNE (perplexity 30) of the same points. Each highlighted group forms a tight pocket. t-SNE coordinates are a visualization of local neighborhoods, not a metric embedding. (c, d) Cosine-distance MDS of one sibling group together with its query. Short tags mark each table; the inset lists $\cos(q, \cdot)$ in descending order (arrow on the top-1). The dashed segment joins the query to the Target; the solid segment, when drawn, joins it to the Sibling with the higher cosine. In (c) a Sibling outscores the Target ($\Delta_q = -0.07$); in (d) the Target wins ($\Delta_q = +0.07$).

Where the failure sits. Neighborhood shape alone does not strongly predict Target ranking: local d_{eff} and local $\bar{d}$ correlate only weakly with $R@1$ ($|r| \lesssim 0.25$). The query-conditioned margin Δ_q is more closely aligned with this within-neighborhood scoring step: it correlates with $R@1$ at about 0.32–0.54, but barely with $GR@1$. On average Δ_q is negative for every model we evaluate, and only 21%–29% of queries score the Target above the nearest Sibling. When the retriever reaches the neighborhood and also picks the Target, mean Δ_q is positive (e.g., $+0.025$ for Qwen3-Embedding-4B Mixed); when it reaches the neighborhood but picks a Sibling, mean Δ_q is negative (-0.041 on the same setting). We therefore treat Δ_q as a direct diagnostic of query-to-table scoring inside the sibling group, rather than as evidence for a new causal mechanism.

Scope of the local diagnosis. The local quantities above are defined on Target+Sibling Distractor groups. This controlled structure makes it possible to separate two retrieval stages that coincide under a singleton gold set: reaching a semantic neighborhood versus selecting the answerable Target within that neighborhood. In standard singleton-gold table-retrieval settings such as TARGET's

FeTaQA and TabFact splits, every query has $|\text{group}| = 1$. Then $GR@1$ coincides with $R@1$ by construction, $DS@1 = 1$, the rate of "neighborhood hit but Target miss" is identically zero, and Target–Sibling margins as well as local d_{eff} on sibling groups are undefined. Table 25 reports the same dense encoder (Qwen3-Embedding-4B) under Mixed-style encoding for illustration: on TARGET FeTaQA/TabFact the metrics collapse to exact table hit, whereas on TCR-Bench the same model yields $GR@1 = 0.684$ versus $R@1 = 0.172$, with 51.2% of queries satisfying $GR@1 = 1$ and $R@1 = 0$. Accordingly, the local quantities reported here should be understood as diagnostics enabled by the controlled retrieval structure, rather than as generic properties of table-retrieval benchmarks.

Table 25: Same dense encoder (Qwen3-Embedding-4B). Under singleton gold sets, group recall coincides with exact recall, so the local Target/Sibling diagnostics in this section are not defined.

Benchmark	$\|\text{group}\|$	$R@1$	$GR@1$	$DS@1$	$GR@1{=}1, R@1{=}0$
TARGET FeTaQA (valid)	1	0.505	0.505	1.000	0%
TARGET TabFact (valid)	1	0.493	0.493	1.000	0%
TCR-Bench	3–8	0.172	0.684	0.252	51.2%

Takeaway. Dense encoders do not simply collapse table representations into an undifferentiated space. Related tables instead form coherent local neighborhoods, as reflected by the relatively small local cosine distances together with substantial $GR@1$. The remaining difficulty is concentrated within those neighborhoods: distinguishing the answerable Target from Sibling Distractors. The Target–Sibling margin Δ_q is more directly related to this within-neighborhood scoring step than aggregate geometric statistics, providing a geometric characterization of the retrieval behavior underlying the observed $GR@1$–$R@1$ discrepancy.

Answerability-oriented fine-tuning on TCR-Bench (Appendix L) changes this picture in a way that is consistent with the same diagnostic. After tuning, mean Δ_q becomes positive and the neighborhood-hit/Target-miss rate drops, so the encoder does learn to score the Target above nearby Siblings on the training groups. At the same time local $\bar{d}$ rises sharply (median 0.113 to 0.481 for Qwen3-Embedding-0.6B Mixed): the compact sibling pockets of the base encoder are no longer preserved. The geometry therefore supports both sides of the training appendix—answerability is learnable inside matched sibling groups, but the update is not a drop-in repair of generic semantic retrieval. Figure 8 shows the same change for six sibling groups of Qwen3-Embedding-0.6B: the pockets that are compact before tuning spread apart afterward, and the mass of Δ_q moves from below zero to above zero.

H Sensitivity Analysis: Schema vs. Cell-Level Signals

This experiment provides an exploratory analysis of how the evaluated embeddings respond to schema-level and cell-level changes. We construct controlled minimal contrasts that isolate individual factors while keeping the remaining table content unchanged. The goal is to examine whether retrieval similarity responds differently to structural signals and factual cell content, providing a complementary perspective on the schema-level cue dependence observed on TCR-Bench.

H.1 Experimental Design

From the Diagnostic Subset, we generate one variant per table using three controlled perturbations: (1) **Content Perturbation (Cell-level)**, where answer-relevant cells are replaced by other valid entries from the same column while preserving the schema and table structure; (2) **Schema Perturbation (Header Rename)**, which renames a query-relevant column header to a generic token (e.g., "Price" $\rightarrow$ "Column_2") without modifying cell values; and (3) **Schema Perturbation (Column Deletion)**, which removes the column referenced by the query to examine the effect of removing a relevant structural dimension.

These perturbations are deliberately small and isolate individual factors as cleanly as possible. This design differs from the construction of TCR-Bench, which uses a broader set of transformations and substantially larger contrasts between target and sibling tables. In particular, the perturbations in

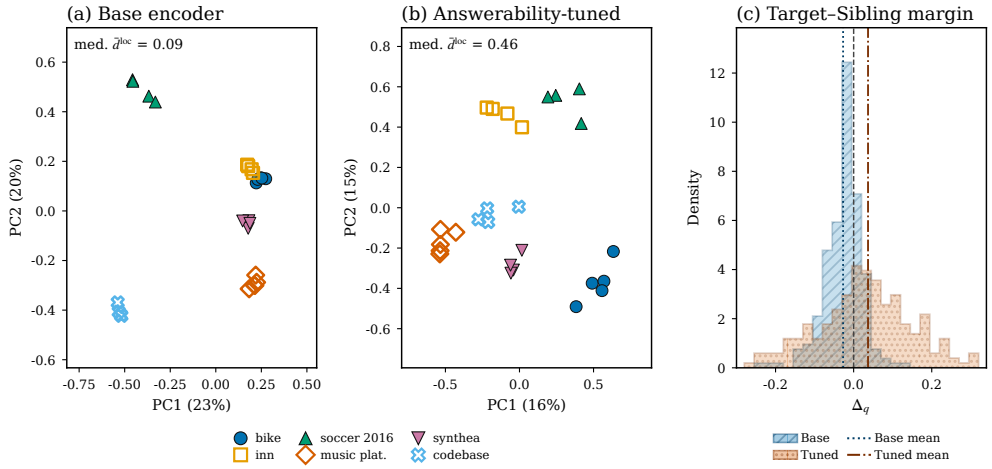


Figure 8: Answerability-oriented fine-tuning changes local geometry (Qwen3-Embedding-0.6B, Mixed). (a, b) PCA of the same six Target+Sibling groups before and after tuning. Each panel is fit separately. On these groups the median within-group cosine distance rises from 0.09 to 0.46; across all 209 queries the corresponding medians are 0.113 and 0.481. (c) Distribution of Δ_q on all 209 queries. The dashed line is zero; dotted and dash-dotted lines mark the Base and Tuned means (-0.027 and $+0.037$).

this appendix are minimal contrasts: only a small number of cells or a single schema component is modified while the remainder of the table is kept unchanged. The cell-level perturbation here directly replaces answer-relevant values with other entries from the same column, which can therefore change the factual identity of the affected entities. In contrast, the cell-level transformations used in TCR-Bench are designed to preserve the semantic identity of the underlying entities, for example through semantically similar substitutions, while introducing substantially larger changes in the table content. These differences make the two settings complementary: the present experiment isolates the effect of small, single-factor changes, whereas TCR-Bench evaluates retrieval under more substantial content and structural differences.

The difference is also reflected in the degree of overlap between targets and sibling tables. The minimal contrasts used here are intentionally much more similar than typical target-sibling pairs in TCR-Bench. Even when rows are shared in TCR-Bench, the maximum row-level overlap is approximately 70%, and such highly overlapping cases constitute only a small fraction of the benchmark. Moreover, TCR-Bench incorporates multiple transformation strategies rather than selecting a single dimension according to its effect on retrieval. The present experiment therefore serves as a controlled diagnostic analysis rather than a reproduction of the benchmark construction process.

H.2 Results and Analysis

Table 26 reports retrieval performance under these perturbations on the Mixed format. The results suggest an asymmetry in how the evaluated retrievers respond to schema-level and cell-level modifications. For example, for Qwen3-E4B, DS@1 is 0.578 under content perturbation, compared with 0.810 when a query-relevant header is renamed. Since a higher DS@1 indicates a stronger preference for the original target over its perturbed counterpart, the retriever is substantially more responsive to the header modification than to the corresponding cell-level modification in this controlled setting. A similar pattern is observed for Stella, where DS@1 increases from 0.611 under content perturbation to 0.771 under header renaming.

The cell-level perturbation results indicate that changing factual values alone can have a relatively limited effect on retrieval similarity when the surrounding schema remains unchanged. Although the perturbed tables no longer satisfy the original query conditions, their unchanged headers and structural organization can preserve substantial similarity to the target. This behavior is consistent with the failure mode studied in TCR-Bench, where semantically similar tables can remain difficult

to distinguish when answerability depends on fine-grained factual content. However, because the perturbations here are intentionally much smaller than those in TCR-Bench, these results should not be interpreted as estimating the prevalence or severity of this failure mode in realistic table collections.

Column deletion provides an additional structural contrast. For Qwen3-E4B, deleting the query-relevant column produces a DS@1 of 0.738, while Stella obtains 0.629. These values exceed the corresponding sensitivity to content perturbation for both models, although the magnitude of the difference varies across models. Together with the header-renaming results, these observations suggest that the evaluated embedding models can be substantially affected by changes to structural or schema-level signals, even when factual cell values remain unchanged.

H.3 Interpretation and Limitations

The results provide preliminary evidence that the evaluated embedding representations may rely more strongly on schema-level signals than on fine-grained factual cell-level signals in minimal contrastive settings. This may explain why tables with highly similar schemas but different answer-bearing values can remain difficult to distinguish. However, we treat this observation as exploratory rather than as a definitive characterization of the underlying representations.

A key consideration is that schema tokens typically occur near the beginning of serialized tables. Consequently, modifying a header also changes information at a highly salient positional region, making positional sensitivity a potential confounding factor. Section I holds table content and answerability fixed and varies only the position of the relevant row; that analysis shows that front-loaded content receives a measurable retrieval advantage. The larger response to header perturbations therefore cannot be attributed solely to schema semantics.

This positional confound primarily affects the interpretation of *why* the retrieval behavior occurs, rather than the empirical observation itself. Regardless of whether the underlying cause is schema semantics, positional salience, or their interaction, modifying relevant schema information can substantially alter retrieval behavior, whereas small factual changes can sometimes leave retrieval similarity largely unchanged. The present experiment therefore identifies a retrieval asymmetry without attempting to isolate its underlying causal mechanism.

Overall, this analysis serves as a controlled manipulation check that complements the broader benchmark results. It uses minimal single-factor contrasts to characterize model sensitivity, whereas TCR-Bench evaluates retrieval under more diverse and substantially larger content and structural transformations. The two settings consequently answer different questions: this experiment examines how retrieval responds to isolated changes, while TCR-Bench evaluates whether the tested retrievers can distinguish an answerable Target from semantically similar but non-answerable Sibling Distractors. The observed asymmetry is one possible explanation for that failure mode. The following section tests the positional confound; the precise contributions of schema information, positional effects, and cell-level representations remain open.

Table 26: Retrieval performance on Minimal Contrastive Sets under small-scale perturbations (Mixed format).

Configuration	Model	R@1	R@2	R@3	GR@1	DS@1
Content Perturb.	Qwen3-E4B	0.371	0.671	0.757	0.643	0.578
	Stella	0.314	0.486	0.529	0.514	0.611
Rename Header	Qwen3-E4B	0.486	0.671	0.814	0.600	0.810
	Stella	0.386	0.514	0.557	0.500	0.771
Delete Column	Qwen3-E4B	0.443	0.714	0.800	0.600	0.738
	Stella	0.314	0.429	0.486	0.500	0.629

I Position Bias and the "Header Preference"

The header-renaming contrast in Section H cannot separate schema semantics from positional salience, because headers occupy the beginning of a serialized table. We therefore test whether the evaluated embeddings exhibit **position bias** when table content and answerability are held fixed. Prior work on long-context language models has shown that performance can vary with the location of relevant information, a phenomenon commonly referred to as the **lost-in-the-middle** effect (Liu et al., 2024a). Here we ask a narrower question: whether the physical placement of a relevant row affects retrieval similarity even when both tables remain fully answerable.

I.1 Experimental Setup

We construct pairs of tables that are semantically identical and **both answerable**. The only difference is the **position of the relevant row**. The row is placed in one of three positions:

- **Top**: immediately after the header
- **Middle**: within the body of the table
- **Bottom**: near the end of the table

We evaluate four comparison settings:

- Top vs Original (original row order in the dataset)
- Top vs Middle
- Bottom vs Middle
- Top vs Bottom

For each comparison, the table listed first is treated as the formal Target for scoring, although **both tables are answerable**.

We reuse **DS@1** as a pairwise preference score. A value **greater than 0.5** indicates that the model tends to rank the *first table* higher, while a value **below 0.5** indicates preference for the *second table*. Thus, DS@1 here reflects positional preference rather than answerability discrimination.

I.2 Results

Table 27: Position bias evaluation under different row placements. DS@1 > 0.5 indicates preference for the first table.

Setting	Model	R@1	GR@1	DS@1
Top vs Original	Qwen3-E4B	0.557	0.686	0.813
Top vs Original	Stella	0.314	0.514	0.611
Top vs Middle	Qwen3-E4B	0.571	0.686	0.833
Top vs Middle	Stella	0.314	0.514	0.611
Bottom vs Middle	Qwen3-E4B	0.371	0.629	0.591
Bottom vs Middle	Stella	0.300	0.514	0.583
Top vs Bottom	Qwen3-E4B	0.557	0.671	0.830
Top vs Bottom	Stella	0.314	0.500	0.629

The results reveal a clear **header/top preference** for the two evaluated models. When the relevant row is moved to the top of the table, both tend to assign higher similarity scores to that table.

For **Qwen3-E4B**, the effect is particularly strong. In the *Top vs Original* and *Top vs Middle* comparisons, DS@1 reaches **0.813** and **0.833**, indicating a strong preference for tables where relevant evidence appears immediately after the header. The *Top vs Bottom* setting further supports this observation (DS@1 = **0.830**): early placement increases retrieval preference under these controlled contrasts.

The *Bottom vs Middle* comparison provides additional insight. When neither table places the relevant row near the header, the preference becomes weaker but still present (DS@1 = **0.591**). Earlier placement within the table body can therefore still influence similarity scores, though less strongly than header-adjacent placement.

For **Stella**, the pattern is more moderate but directionally consistent. Across all four settings, DS@1 remains between **0.58** and **0.63**, indicating a preference toward the first table in each comparison. This suggests a positional bias favoring earlier occurrences of relevant rows, although the magnitude of the effect is smaller than that observed in Qwen3-E4B.

I.3 Discussion

Our analysis indicates that the embedding models we evaluate exhibit a noticeable **position bias** when encoding tables. This observation is consistent with prior studies on long-context language models, which show that models often favor information appearing near the beginning or end of the input sequence while struggling to utilize information located in the middle (Liu et al., 2024a). Rows placed closer to the header are more likely to influence the embedding representation and thus receive higher similarity scores during retrieval.

Notably, the preference for top-positioned rows persists even when the alternative placement is only moved to the middle of the table, indicating that positional cues can influence similarity scores in this setting. Similar positional effects have been reported in transformer-based models, where attention patterns and positional encodings can introduce systematic biases toward certain regions of the input sequence (Yu et al., 2025b).

The position effect is weaker than the header-renaming contrast in Section H, so positional salience does not by itself account for the full schema-versus-cell asymmetry. It does, however, show that front-loaded content receives an independent retrieval advantage. Combined with the previous section, the header-renaming result should therefore be read as a mixture of schema-level and positional effects, rather than as a pure test of schema semantics.

I.4 Implication

Because both tables in this experiment are answerable, the result does not measure answerability discrimination. It shows that, even with answerability held fixed, the evaluated embeddings remain sensitive to where relevant evidence appears. This positional preference is one reason header-level modifications can move retrieval more than small factual edits in our contrasts, and it is consistent with encoding tables more like linearized text than like an explicit record of row-column bindings. Primacy effects of this kind have also been reported in long-context evaluation (Liu et al., 2024a; Veseli et al., 2025).

J Retrieval under Table Transposition

Prior studies in TableQA and table representation learning have shown that table orientation can substantially affect model behavior, especially for architectures relying on serialized table representations and positional encoding schemes (Liu et al., 2024b; Sui et al., 2024; Singha et al., 2023). Motivated by these observations, we investigate whether table transposition influences retrieval performance in our setting.

Specifically, we transpose each table by swapping rows and columns while preserving all original cell contents. The experiment is conducted exclusively under the Mixed format setting.

Experimental Setup We evaluate representative dense embeddings on the transposed-table corpus using the same retrieval protocol and evaluation metrics as in Section 4. To test whether the observed sensitivity is specific to single-vector retrieval, we additionally evaluate the row-level multi-vector approach from Appendix D.2 under the same transposed-table setting. This method always embeds each row of the table *as presented*: after transposition, a "row" is a row of the transposed table (i.e., an original column laid out as a row). We do not switch to column-wise indexing in order to recover the original entity rows.

Results Table 28 reports the retrieval performance on transposed tables.

Table 28: Retrieval performance on transposed tables under the Mixed format setting.

Model	R@1	R@3	R@5	GR@1	DS@1
Qwen3-E0.6B	0.072	0.211	0.311	0.311	0.231
Qwen3-E4B	0.081	0.273	0.392	0.450	0.181
Stella	0.038	0.163	0.258	0.301	0.127
Multi-vector (Qwen3-E0.6B)	0.153	0.388	0.550	0.632	0.242
Multi-vector (Qwen3-E4B)	0.201	0.455	0.603	0.689	0.292

Analysis All three evaluated single-vector embeddings show substantial degradation after table transposition.

A plausible account has two parts. First, the original tables are largely row-oriented: attributes of the same entity remain locally grouped after serialization. Many queries in TCR-Bench depend on such entity-level rows, so this layout can help the evaluated embeddings capture local row-level content. After transposition, related attributes become dispersed, weakening that local grouping. This is a stronger structural change than the format-level serialization variants considered in Section 5.1, which left Target–Sibling discrimination largely stable.

Second, transposition scatters schema information across the sequence and disrupts the alignment between values and their surrounding context. This is consistent with the header/top preference in Section I, and with limited sensitivity to row-column bindings when those associations are disrupted.

Although the evaluated row-level multi-vector retriever was comparatively strong in Appendix D.2, its performance also drops after transposition. Because the method still indexes the current table rows rather than the original entity rows (now columns), each vector after transposition aggregates values that no longer belong to the same entity. Its DS@1 falls to 0.242–0.292, near the random-selection baseline. This suggests that the gains of this multi-vector approach depend, at least in part, on the row-oriented organization of the original tables.

Overall, this orientation stress test indicates that, for the methods we evaluate, retrieval is influenced not only by table semantics but also by structural layout. Multi-vector retrieval remains more robust than single-vector retrieval under transposition, but the drop suggests that orientation sensitivity is not limited to single-vector embeddings and can also affect row-level retrieval in this setting.

K Answerability-Aware Reranking

K.1 Reranking Methods

We evaluate three answerability-aware reranking (AAR) configurations as a diagnostic complement to dense retrieval. AAR-CE uses the Qwen3-Reranker series as a dedicated pointwise cross-encoder, with Qwen3-R0.6B, Qwen3-R4B, and Qwen3-R8B paired with Qwen3-E0.6B, Qwen3-E4B, and Qwen3-E8B, respectively. Each pair has the same parameter scale and is initialized from the same pretrained model; the two members would therefore be expected to have comparable potential, and they differ primarily in how they are used: single-vector embedding retrieval versus query-conditioned reranking. This pairing, including the 8B comparison in Section 6, tests whether a change in operating mode is sufficient to alter Target–Sibling discrimination. AAR-LLM-PW and AAR-LLM-LW both use Qwen3-30B. AAR-LLM-PW follows the pointwise paradigm of Balaka et al. (2025), formulating reranking as binary classification: within the reranked pool, candidates predicted as "yes" are placed before those predicted as "no," while the relative order within each group is inherited from the dense retriever. AAR-LLM-LW instead adopts a listwise approach, feeding the query together with all candidate tables into the LLM and asking it to produce a full ordering. In all cases, reranking is applied to the top k candidates from the dense retriever, denoted Tk; T∞ reranks the full retrieved list.

Table 29: Complete experimental results across retrieval configurations. Here, Ti indicates that reranking is applied to the top i results returned by the embedding-based retrieval model.

Method	Retrieval Model	R@k					GR@k					DS@1
		@1	@2	@3	@4	@5	@1	@2	@3	@4	@5	
Qwen3-E0.6B	AAR-CE (T3)	0.273	0.383	0.411	-	-	0.694	0.665	0.593	-	-	0.393
	AAR-CE (T5)	0.297	0.435	0.507	0.545	0.555	0.737	0.703	0.641	0.596	0.556	0.403
	AAR-CE (T10)	0.359	0.541	0.617	0.665	0.689	0.789	0.761	0.719	0.695	0.664	0.455
	AAR-CE (T20)	0.416	0.589	0.684	0.732	0.770	0.823	0.797	0.758	0.744	0.736	0.506
	AAR-CE (T50)	0.459	0.632	0.742	0.799	0.833	0.876	0.835	0.797	0.782	0.772	0.525
	AAR-CE (T∞)	0.507	0.694	0.804	0.866	0.900	0.895	0.861	0.818	0.794	0.771	0.567
	AAR-LLM-PW (T5)	0.383	0.455	0.512	0.545	0.555	0.689	0.639	0.600	0.574	0.556	0.556
	AAR-LLM-PW (T10)	0.469	0.589	0.641	0.675	0.684	0.751	0.694	0.633	0.603	0.589	0.624
	AAR-LLM-LW (T5)	0.383	0.445	0.488	0.522	0.555	0.737	0.689	0.630	0.600	0.584	0.519
	AAR-LLM-LW (T10)	0.426	0.493	0.550	0.603	0.646	0.751	0.730	0.689	0.666	0.658	0.567
Qwen3-E4B	AAR-CE (T5)	0.426	0.522	0.589	0.617	0.632	0.789	0.749	0.711	0.682	0.662	0.539
	AAR-CE (T10)	0.498	0.598	0.651	0.694	0.722	0.828	0.801	0.777	0.757	0.735	0.601
	AAR-LLM-PW (T5)	0.445	0.550	0.598	0.612	0.632	0.746	0.725	0.692	0.673	0.662	0.596
	AAR-LLM-PW (T10)	0.517	0.622	0.689	0.703	0.718	0.804	0.766	0.721	0.688	0.676	0.643
	AAR-LLM-LW (T5)	0.474	0.522	0.550	0.603	0.632	0.770	0.737	0.713	0.692	0.679	0.615
	AAR-LLM-LW (T10)	0.435	0.526	0.593	0.632	0.684	0.804	0.792	0.767	0.743	0.733	0.542
Qwen3-E8B	AAR-CE (T5)	0.507	0.608	0.636	0.660	0.665	0.770	0.730	0.684	0.653	0.622	0.658
	AAR-CE (T10)	0.574	0.632	0.651	0.656	0.656	0.766	0.732	0.708	0.683	0.664	0.750
	AAR-LLM-PW (T5)	0.498	0.608	0.641	0.660	0.665	0.732	0.694	0.652	0.629	0.622	0.680
	AAR-LLM-PW (T10)	0.536	0.651	0.684	0.722	0.737	0.761	0.720	0.681	0.650	0.637	0.704
	AAR-LLM-LW (T5)	0.455	0.555	0.593	0.646	0.665	0.756	0.718	0.691	0.657	0.642	0.601
	AAR-LLM-LW (T10)	0.450	0.555	0.593	0.670	0.708	0.770	0.742	0.718	0.700	0.698	0.584

K.2 Results and Analysis

Table 29 reports the complete results across retrieval models and reranking configurations. Across the evaluated settings, AAR improves target identification over the corresponding dense ranking. The improvement is particularly clear for R@1. For the Qwen3-E0.6B and Qwen3-R0.6B pair, Mixed-format dense R@1 is 0.182; AAR-CE raises it to 0.297 at T5, 0.359 at T10, and 0.507 when all retrieved candidates are reranked. The same-scale pattern also holds at 4B and 8B: AAR-CE (T10) reaches 0.498 and 0.574, respectively, well above the dense Mixed-format R@1 of 0.172 and 0.182. Because each embedding/reranker pair shares scale and initialization, this gap is difficult to attribute to extra capacity. It indicates that changing the operating mode—from single-vector similarity to query-conditioned answerability assessment—substantially improves Target ranking once the Target is included in the reranked pool.

For AAR-CE on this pair, larger reranking budgets continue to help. As k increases from 5 to 50, R@5 rises from 0.555 to 0.833, while R@1 increases from 0.297 to 0.459. GR@1 also increases from 0.737 to 0.876, and DS@1 rises monotonically from 0.393 at T3 to 0.567 at T∞. The joint increase in R@1, GR@1, and DS@1 suggests that exposing more candidates to the reranker improves both neighborhood retrieval and within-group target identification, rather than trading one for the other. Gains remain substantial beyond T10, suggesting that restricting AAR-CE to a very small pool leaves answerable candidates unused. Listwise LLM reranking does not follow the same monotonic pattern: for Qwen3-E4B and Qwen3-E8B, AAR-LLM-LW at T10 yields lower R@1 than at T5.

Figure 9 visualizes this scaling for the Qwen3-E0.6B and Qwen3-R0.6B pair. Retrieval improves as more candidates are reranked, but the gains diminish at larger k. This gives an empirical trade-off between reranking budget and retrieval quality for this configuration.

Despite these gains, all evaluated reranking configurations are substantially more expensive than embedding retrieval. AAR-CE requires an additional cross-encoder forward pass for each reranked candidate, AAR-LLM-PW requires one LLM call per candidate, and AAR-LLM-LW jointly processes all candidates in a long context. We therefore use Tk as the default setting for controlled comparisons and report T∞ only for AAR-CE with the Qwen3-E0.6B and Qwen3-R0.6B pair. This setting reranks all retrieved candidates and serves as an empirical upper bound on the benefit obtainable from AAR-CE at this scale, rather than as an exhaustive evaluation across all model scales and reranking configurations.

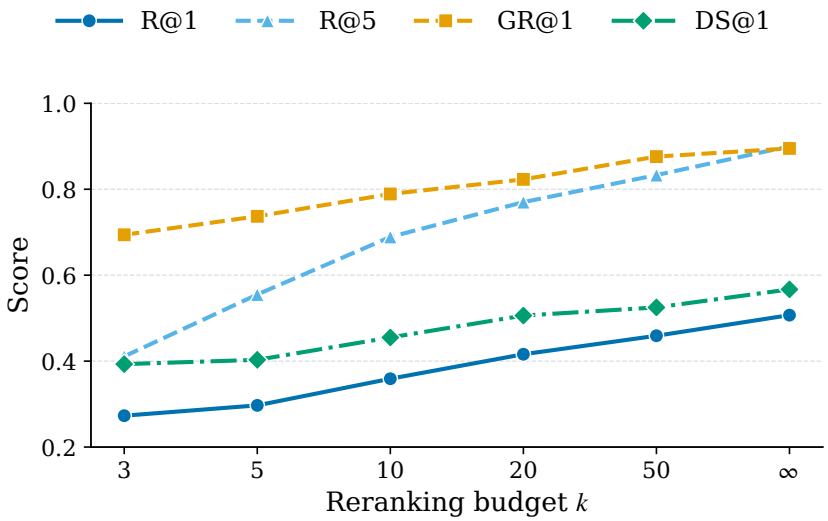


Figure 9: Scaling behavior of answerability-aware reranking with different reranking budgets k, using Qwen3-E0.6B for initial retrieval and Qwen3-R0.6B for reranking. For T3, R@5 is effectively R@3 because only the top-3 candidates are reranked.

L Answerability-Oriented Training

The reranking experiments show that explicit query-conditioned interaction can substantially improve target identification. We next investigate whether an embedding model can learn a similar answerability signal through direct supervision. These experiments are diagnostic and exploratory rather than conventional held-out evaluations, since neither TCR-Bench nor Real-TCR provides a dedicated training split. Our goal is to examine whether answerability can be learned and whether the resulting behavior transfers across table distributions while preserving general retrieval capability. We emphasize that the training setup here serves this exploratory purpose rather than aiming to establish a standardized or optimized training procedure; the experiments are intended as a diagnostic probe, not as a benchmark-oriented training recipe.

L.1 Experimental Setup

We fine-tune Qwen3-Embedding-0.6B using query–target positives and sibling-table hard negatives. The positive pair consists of the query and its target table, while sibling tables provide semantically related but non-target negatives. We use a maximum of 16 negatives per query. We first conduct preliminary experiments to determine the remaining training settings (including temperature) and then perform a learning-rate grid search. Based on these preliminary runs, the temperature is fixed to 0.1 throughout the search. The learning-rate candidates are 1×10^{-5}, 3×10^{-5}, 1×10^{-4}, 2×10^{-4}, 3×10^{-4}, 5×10^{-4}, 1×10^{-3}, 2×10^{-3}, and 5×10^{-3}. Other hyperparameters such as maximum sequence length and batch size are chosen under computational constraints. Unless noted otherwise, we report R@1, and on TCR-Bench / Real-TCR also GR@1 and DS@1 (DS@1 $=$ R@1/GR@1 at the aggregate level, following the main evaluation).

L.2 Training on TCR-Bench

The best TCR-Bench configuration uses a learning rate of 1×10^{-4}. We train for up to five epochs, with the best checkpoint obtained at epoch 4.

The best checkpoint reaches training R@1 of 0.493 with an average epoch loss of 0.0669. On TCR-Bench, it achieves R@1 of 0.493, GR@1 of 0.713, and DS@1 of 0.691.

The large improvement in R@1 and DS@1 shows that direct answerability-oriented supervision can substantially alter retrieval behavior under a matched distribution. This provides evidence that

Table 30: Training configuration for answerability-oriented fine-tuning on TCR-Bench.

Configuration	Value
Base model	Qwen3-Embedding-0.6B
Learning rate	1×10^{-4}
Temperature	0.1
Epochs	5 (best: epoch 4)
Batch size	8
Gradient accumulation	2
Maximum length (query / passage)	512 / 512
Maximum negatives	16
Warmup ratio	0.05
Weight decay	0.01
Mixed precision	bf16
Seed	42

Table 31: Answerability-oriented fine-tuning on TCR-Bench.

Model	R@1	GR@1	DS@1
Qwen3-Embedding-0.6B	0.182	0.627	0.290
Answerability-tuned	**0.493**	**0.713**	**0.691**

answerability is learnable for the embedding model we fine-tune, rather than inaccessible under sibling hard-negative supervision.

L.2.1 Geometry of the tuned encoder

We repeat the Target–Sibling diagnostics of Appendix G.3 on this checkpoint (Mixed encodings of the same 637 tables and 209 queries). Table 32 compares the unmatched Qwen3-Embedding-0.6B encoder with the TCR-Bench-tuned model. The Target–Sibling margin Δ_q changes sign (mean -0.027 to $+0.037$), the fraction of queries with $\Delta_q > 0$ rises from 27% to 63%, and the neighborhood-hit/Target-miss rate falls from 44.5% to 22.0%. When the tuned retriever reaches the sibling group and ranks the Target first, mean Δ_q is $+0.122$; when it reaches the group but ranks a Sibling first, mean Δ_q is -0.094. Query-to-table scoring inside the training sibling groups therefore does become answerability-sensitive, which is the geometric counterpart of the DS@1 gain.

The same update does not preserve the compact local neighborhoods of the base encoder. Median local pairwise cosine distance rises from 0.113 to 0.481, and corpus-level $\bar{d}$ rises from 0.494 to 0.952. Sibling tables that previously occupied a tight pocket are pushed far apart. The encoder learns an answerability-sensitive scoring rule on the groups it was trained on, but it does so by disrupting the semantic neighborhood structure that generic retrieval relies on.

Table 32: Target–Sibling geometry on TCR-Bench before and after answerability-oriented fine-tuning (Mixed). $\bar{d}^{\text{loc}}$ is the median local pairwise cosine distance; Δ_q is the Target–Sibling cosine margin.

Model	$\bar{d}^{\text{loc}}$ med.	Δ_q mean	$\Delta_q{>}0$	$R@1$	$GR@1$	$DS@1$
Qwen3-Embedding-0.6B	0.113	-0.027	27%	0.182	0.627	0.290
Answerability-tuned	0.481	$+0.037$	63%	0.493	0.713	0.691

L.2.2 Cross-Distribution Evaluation

We next evaluate whether this in-domain answerability signal transfers. We consider transfer between TCR-Bench and Real-TCR, as well as Real-TCR-Small. Since neither benchmark provides a dedicated training split, these experiments are controlled transfer diagnostics rather than standard held-out evaluations.

Table 33: Cross-distribution evaluation of answerability-oriented fine-tuning. The third row selects the best TCR-Bench-trained checkpoint by Real-TCR R@1 (full-parameter grid; LoRA ties at the same R@1). GR@1 and DS@1 are omitted for that row because the selection pass reported only R@1.

Training	Evaluation	Selection	R@1	GR@1	DS@1
TCR-Bench	Real-TCR	TCR-Bench R@1	0.065	0.831	0.078
TCR-Bench	Real-TCR-Small	TCR-Bench R@1	0.132	0.868	0.152
TCR-Bench	Real-TCR	Real-TCR R@1	0.156	—	—
Real-TCR	Real-TCR	Real-TCR R@1	**0.442**	**0.922**	**0.479**
Real-TCR	TCR-Bench	Real-TCR R@1	0.043	0.311	0.138

The TCR-Bench in-domain checkpoint transfers poorly: 0.065 R@1 and 0.078 DS@1 on Real-TCR, and 0.132 R@1 and 0.152 DS@1 on Real-TCR-Small. Because that evaluation uses a checkpoint selected on TCR-Bench, we additionally sweep the same training run (full-parameter and LoRA learning-rate grids) and retain the checkpoint with the highest Real-TCR R@1. The best transferred configuration reaches only 0.156 R@1 on Real-TCR—above the unmatched 0.6B baseline of 0.078, but far from matched Real-TCR training (0.442). LoRA matches this same Real-TCR R@1 while remaining close to the base encoder on TCR-Bench (0.191 R@1, 0.270 DS@1) and on TARGET, so the bump does not require a stable transferred answerability rule. Target-set selection therefore yields only a modest, brittle gain rather than a ready-made transferable recipe under this setup. In the reverse direction, Real-TCR training produces strong within-distribution performance (0.442 R@1, 0.479 DS@1) but drops to 0.043 R@1 and 0.138 DS@1 on TCR-Bench. The learned answerability behavior remains substantially distribution-dependent.

L.2.3 Effect on General Retrieval

We further evaluate the TCR-Bench-trained model on independently sampled subsets of TabFact and FeTaQA. These benchmarks do not define candidate groups, so only R@1 is reported.

Table 34: Retrieval performance of the TCR-Bench-trained model on independent TARGET subsets.

Benchmark	R@1
TabFact-valid sample	0.184
TabFact-test sample	0.179
FeTaQA-valid sample	0.152
FeTaQA-test sample	0.104

The TCR-Bench-trained model achieves R@1 of 0.184 and 0.179 on the TabFact validation and test samples, respectively, and 0.152 and 0.104 on the corresponding FeTaQA samples. Compared with the base-model results of 0.414 on TabFact-valid and 0.325 on FeTaQA-valid, these results indicate a substantial loss in broad semantic retrieval capability after answerability-oriented fine-tuning. The Real-TCR-selected checkpoint from the same run is no better on these TARGET subsets (0.147 / 0.152 on TabFact valid/test and 0.133 / 0.096 on FeTaQA valid/test). This is consistent with the geometry above: spreading sibling neighborhoods on TCR-Bench does not yield a retrieval function that remains useful on unrelated table collections.

L.3 Training on General Retrieval Data

The previous results show that answerability-oriented fine-tuning on TCR-Bench substantially degrades retrieval on TabFact and FeTaQA. We now consider the reverse direction: whether strengthening generic table retrieval on these TARGET collections improves answerability-sensitive ranking on TCR-Bench and Real-TCR.

We construct a 1:1 mixture from the TARGET TabFact and FeTaQA training splits, using 7,326 examples from each source (14,652 in total), with one TabFact query sampled per table. Table serializations follow the evaluation default (HTML for TabFact, JSON for FeTaQA), and each example uses the corresponding evaluation query prefix (fact verification vs. TableQA). Validation queries

are excluded from training. Unlike TCR-Bench and Real-TCR, these TARGET collections do not define sibling groups, so the objective is query–table relevance with in-batch negatives rather than target-versus-sibling discrimination.

We fine-tune Qwen3-Embedding-0.6B with LoRA ($r = 16$, $\alpha = 32$) using the same contrastive recipe and temperature 0.1. The learning-rate grid is 5×10^{-5}, 1×10^{-4}, 2×10^{-4}, 3×10^{-4}, and 5×10^{-4}. Checkpoints are selected independently on the TabFact-valid and FeTaQA-valid samples used in the main evaluation.

Table 35: Training configuration for general-retrieval fine-tuning on the TARGET TabFact+FeTaQA mix. The two learning rates correspond to the TabFact-valid and FeTaQA-valid selections, respectively.

Configuration	Value
Base model	Qwen3-Embedding-0.6B
Training data	TARGET TabFact+FeTaQA 1:1 mix (14,652)
Adaptation	LoRA ($r = 16$, $\alpha = 32$, dropout 0.05)
Learning rate	3×10^{-4} / 2×10^{-4} (see text)
Temperature	0.1
Epochs	5 (best: epoch 3 for both selections)
Batch size	4
Gradient accumulation	4
Maximum length (query / passage)	512 / 2048
Maximum negatives	16
Warmup ratio	0.05
Weight decay	0.01
Mixed precision	bf16
Seed	42

The TabFact-selected checkpoint uses a learning rate of 3×10^{-4} and reaches R@1 of 0.459 on TabFact-valid. The FeTaQA-selected checkpoint uses 2×10^{-4} and reaches R@1 of 0.408 on FeTaQA-valid. Both improve over the corresponding base-model results of 0.414 and 0.325. As above, TabFact and FeTaQA do not define sibling groups, so GR@1 and DS@1 are undefined and only R@1 is reported.

Table 36: In-domain R@1 of TARGET-mix LoRA checkpoints on the TabFact and FeTaQA validation samples.

Model	TabFact-valid	FeTaQA-valid
Qwen3-Embedding-0.6B	0.414	0.325
TARGET mix (TabFact-best)	**0.459**	0.370
TARGET mix (FeTaQA-best)	0.453	**0.408**

We then evaluate both checkpoints on TCR-Bench and Real-TCR, where sibling groups are defined and R@1, GR@1, and DS@1 are all reported.

Table 37: Transfer of TARGET-mix general-retrieval training to answerability benchmarks, compared with the unmatched base encoder and matched answerability-oriented fine-tuning.

Training	Evaluation	Model	R@1	GR@1	DS@1
—	TCR-Bench	Qwen3-Embedding-0.6B	0.182	0.627	0.290
TARGET mix	TCR-Bench	TabFact-best	0.100	0.450	0.223
TARGET mix	TCR-Bench	FeTaQA-best	0.163	0.555	0.293
TCR-Bench	TCR-Bench	Answerability-tuned	**0.493**	**0.713**	**0.691**
—	Real-TCR	Qwen3-Embedding-0.6B	0.078	0.870	0.090
TARGET mix	Real-TCR	TabFact-best	0.039	0.857	0.045
TARGET mix	Real-TCR	FeTaQA-best	0.091	0.896	0.101
Real-TCR	Real-TCR	Answerability-tuned	**0.442**	**0.922**	**0.479**

Neither TARGET-mix checkpoint recovers the answerability-oriented in-domain gains. On TCR-Bench, the FeTaQA-selected model remains closest to the base encoder: R@1 of 0.163 versus 0.182, GR@1 of 0.555 versus 0.627, and DS@1 of 0.293 versus 0.290. Neighborhood retrieval declines, while within-neighborhood discrimination is essentially unchanged. The TabFact-selected model degrades on all three metrics (0.100 / 0.450 / 0.223). On Real-TCR, both mix checkpoints retain high GR@1 (0.857 and 0.896 versus the base 0.870), but DS@1 stays near the base (0.045 and 0.101 versus 0.090). Matched answerability-oriented training instead reaches 0.493 / 0.713 / 0.691 on TCR-Bench and 0.442 / 0.922 / 0.479 on Real-TCR, with the largest relative gain in DS@1. Under this contrastive setup, generic query–table relevance training does not by itself induce the target-versus-sibling discrimination that answerability benchmarks require: models can still enter the correct semantic neighborhood while remaining unable to identify the answerable table. Together with the drop on TabFact and FeTaQA after TCR-Bench training, the two objectives are not interchangeable in a simple way.

L.4 Joint Training on TCR-Bench and TARGET

The preceding experiments examine the two directions separately. We next ask whether a single mixed training run can support both answerability-oriented and generic retrieval signals at once. This is again a diagnostic probe rather than an attempt to optimize a production recipe: if a straightforward mixture already aligned the two objectives, the Semantic-Answerability Gap would be easier to close through data mixing alone.

We concatenate the TARGET TabFact+FeTaQA mix above with TCR-Bench hard-negative examples. Because TCR-Bench contributes only 209 queries, concatenating it as-is would leave answerability supervision at about 1.4% of the mix; we therefore upsample TCR-Bench (with replacement, keeping sibling hard negatives) to 7,326 examples so that each epoch is approximately 1:1:1 TabFact / FeTaQA / TCR-Bench (21,978 examples total). Query prefixes and table serializations follow the same eval-aligned settings as Section L.3. We fine-tune with LoRA ($r = 16$, $\alpha = 32$) and a reduced learning-rate grid of 1×10^{-4} and 2×10^{-4} (rates that previously collapsed TARGET under full-parameter updates, or moved little under LoRA, are omitted). After each epoch we record R@1 on Real-TCR (sibling metrics GR@1/DS@1 are not used for selection here) and on the TabFact-valid and FeTaQA-valid samples (TARGET has no sibling groups, so only R@1 applies), and keep a separate best checkpoint per bench.

Table 38: Training configuration for joint TCR-Bench + TARGET-mix LoRA fine-tuning.

Configuration	**Value**
Base model	Qwen3-Embedding-0.6B
Training data	TCR-Bench + TabFact+FeTaQA (1:1:1, 21,978)
Adaptation	LoRA ($r = 16$, $\alpha = 32$, dropout 0.05)
Learning rate	1×10^{-4} / 2×10^{-4} (see text)
Temperature	0.1
Epochs	5
Batch size	4
Gradient accumulation	4
Maximum length (query / passage)	512 / 2048
Maximum negatives	16
Query format	eval_prefix

The mixture does not remove the tradeoff. Selecting on Real-TCR yields only 0.117 R@1—above the 0.078 baseline and the TARGET-mix-only transfer of 0.091, but far from matched Real-TCR training (0.442), and at the cost of TabFact-valid dropping to 0.374 (below the 0.414 base). Selecting on TabFact or FeTaQA recovers moderate TARGET gains (0.428 / 0.393), yet these remain below the TARGET-mix-only peaks (0.459 / 0.408), and Real-TCR stays near the base (0.065 / 0.078). No single checkpoint in the completed LoRA grid jointly exceeds the separate single-objective peaks on both sides. Under this exploratory mixture, therefore, simply pooling answerability hard negatives with generic TARGET relevance data does not yield an easy joint optimum.

Table 39: R@1 under joint TCR-Bench+TARGET LoRA training, compared with the base encoder, TARGET-mix-only LoRA, and matched Real-TCR training. Joint-mix rows use the checkpoint best on the named selection bench; TARGET-mix-only Real-TCR uses the FeTaQA-best transfer from Table 37.

Model	Real-TCR	TabFact-valid	FeTaQA-valid
Qwen3-Embedding-0.6B	0.078	0.414	0.325
TARGET mix only (best per bench)	0.091	**0.459**	**0.408**
Joint mix (Real-TCR-best)	**0.117**	0.374	0.367
Joint mix (TabFact-best)	0.065	0.428	0.382
Joint mix (FeTaQA-best)	0.078	0.422	0.393
Real-TCR matched training	0.442	—	—

L.5 Training on Real-TCR

We additionally fine-tune Qwen3-Embedding-0.6B directly on Real-TCR using the available hard-negative training data. The best configuration uses a learning rate of 1×10^{-4} and temperature 0.1, with the best checkpoint obtained at epoch 2 of 5.

Table 40: Training configuration for Real-TCR answerability-oriented fine-tuning.

Configuration	Value
Base model	Qwen3-Embedding-0.6B
Learning rate	1×10^{-4}
Temperature	0.1
Epochs	5 (best: epoch 2)
Batch size	8
Gradient accumulation	2
Query / passage maximum length	512 / 512
Maximum negatives	16
Warmup ratio	0.05
Weight decay	0.01
Mixed precision	bf16
Seed	42

The best checkpoint achieves training R@1 of 0.442 on the full Real-TCR training collection, with an average epoch loss of 0.905.

Table 41: Evaluation of the Real-TCR-trained model across table-retrieval distributions.

Benchmark	R@1	GR@1	DS@1
Real-TCR	0.442	0.922	0.479
Real-TCR-Small	0.711	0.947	0.750
TCR-Bench	0.043	0.311	0.138

Matched Real-TCR training remains the only setting in which Real-TCR R@1 rises substantially (0.078 to 0.442); the joint TCR-Bench+TARGET mixture above reaches at most 0.117. The matched gain does not transfer to TCR-Bench (0.043 R@1, 0.138 DS@1), consistent with the reverse of the modest Real-TCR-selected transfer above.

L.6 Discussion

Taken together, these experiments provide a nuanced interpretation of the Semantic-Answerability Gap under the diagnostic setups above. Direct supervision substantially improves answerability-sensitive retrieval on the training distribution, and the geometry of the tuned encoder confirms that this is not a bookkeeping artifact: Δ_q becomes positive on a majority of TCR-Bench queries, and DS@1 rises with it. Answerability is therefore learnable under matched sibling supervision.

That learned behavior, however, does not by itself resolve the gap. Transfer across table distributions remains weak even when we select the TCR-Bench-trained checkpoint by Real-TCR R@1 (0.156 versus a 0.078 baseline and 0.442 matched training). The same update spreads local sibling neighborhoods and degrades generic retrieval on TabFact and FeTaQA. The reverse is also true: strengthening generic table retrieval on a TARGET TabFact+FeTaQA mixture improves in-domain R@1 on those collections, yet leaves DS@1 near the base encoder on TCR-Bench and Real-TCR, even when GR@1 remains high. Mixing the two data sources in one LoRA run likewise fails to deliver an easy joint win: Real-TCR R@1 rises only modestly (to 0.117), while TARGET peaks fall short of TARGET-only training, and the best checkpoint on one side is not best on the other.

These results do not claim that no joint objective can succeed. They show that, under the straightforward contrastive recipes explored here, fitting an answerability signal, transferring it, and preserving broad semantic relevance do not come for free together. A hard-negative objective can improve discrimination among the alternatives encountered during training—geometrically, by enlarging Target–Sibling margins—but in our setting it spreads the compact neighborhoods that generic retrieval uses, and it does not yield a robust answerability capability on a shifted table distribution. Likewise, a general query–table relevance objective does not substitute for target-versus-sibling answerability supervision, and a simple data mixture does not automatically reconcile the two. These findings refine the scope of our main claim within the setups above: *among the generic semantic retrievers and contrastive recipes we evaluate, an answerability bottleneck persists, and straightforward answerability-oriented fine-tuning—including a simple TCR-Bench+TARGET mixture—does not provide an easy resolution, because answerability discrimination and broad semantic relevance are not automatically aligned*. Developing training objectives that jointly support both properties remains an open direction.

M Statistical Significance Analysis

Unless otherwise stated, all reported 95% confidence intervals for DS@1 and p-values are estimated via bootstrap resampling over queries ($N = 209$, 1,000 samples, percentile method). To evaluate the statistical reliability of the observed improvements, we conduct paired bootstrap significance tests comparing answerability-aware reranking variants against the embedding-only Qwen3-E8B retrieval baseline.

Results Table 42 summarizes the statistical estimates for representative retrieval and reranking models. Table 43 reports paired bootstrap comparisons between AAR variants and the Qwen3-E8B baseline; note that AAR-CE is based on Qwen3-R8B, whereas the other AAR variants are based on Qwen3-30B.

Table 42: Bootstrap confidence intervals and significance tests for representative retrieval methods.

Model	Type	R@1	DS@1	DS@1 (95% CI)	p-value vs. Random
tapas_nq_retriever_large	DTR	0.005	1.000	[0.000, 1.000]	0.350
tapas_nq_hn_retriever_large	DTR	0.010	0.250	[0.000, 0.600]	0.628
SauerkrautLM-Multi-ModernColBERT	ColBERT-style	0.173	0.225	[0.164, 0.291]	0.985
reason-colBERT-150M-GTE-ModernColBERT	ColBERT-style	0.086	0.198	[0.116, 0.286]	0.982
Qwen3-E4B + BM25 hybrid	Hybrid	0.215	0.312	[0.239, 0.389]	0.369
Qwen3-Embedding-8B	Dense	0.182	0.271	[0.200, 0.346]	0.748
Qwen3-Embedding-4B	Dense	0.172	0.252	[0.185, 0.322]	0.904
stella_en_1.5B_v5	Dense	0.096	0.182	[0.117, 0.257]	0.999
AAR-CE(T10)	Reranker	0.574	0.750	[0.682, 0.817]	< 0.001
AAR-LLM-PW(T10)	Reranker	0.536	0.704	[0.636, 0.774]	< 0.001
AAR-LLM-LW(T10)	Reranker	0.450	0.584	[0.503, 0.660]	< 0.001

Analysis Table 42 tests DS@1 against a random-selection baseline. For the dense retrievers and the alternative first-stage methods listed above, the 95% CIs include or lie below that baseline, and none of the p-values indicate a significant improvement over chance. For DTR, the DS@1 estimates are additionally unstable because group recall is near zero, which is reflected in the wide intervals. The three AAR variants are the exception: all have $p < 0.001$ against random selection, and their DS@1 CIs lie well above the baseline.

Table 43: Paired bootstrap significance tests comparing AAR variants with the embedding-only Qwen3-E8B retrieval baseline. Confidence intervals are computed from 1,000 bootstrap samples.

Comparison	ΔR@1	**95% CI**	ΔDS@1	**95% CI**	p**-value**
AAR-CE(T10) − Qwen3-E8B	+0.393	[0.316, 0.469]	+0.480	[0.377, 0.582]	< 0.001
AAR-LLM-PW(T10) − Qwen3-E8B	+0.353	[0.273, 0.435]	+0.434	[0.336, 0.539]	< 0.001
AAR-LLM-LW(T10) − Qwen3-E8B	+0.268	[0.206, 0.330]	+0.313	[0.232, 0.392]	< 0.001

Table 43 compares each AAR variant with Qwen3-E8B on the same queries. The paired differences in R@1 and DS@1 are positive, and all three tests yield $p < 0.001$. As noted above, AAR-CE uses Qwen3-R8B (same scale and initialization as Qwen3-E8B), whereas AAR-LLM-PW and AAR-LLM-LW use Qwen3-30B. These tests corroborate the improvements already reported.

N Summary of Experimental Observations

Table 44 summarizes the major experimental observations regarding answerability-aware table retrieval and the Semantic-Answerability Gap, as observed for the models evaluated in this study.

Table 44: Summary of experimental observations on answerability-aware table retrieval and the Semantic-Answerability Gap.

Observations	**Implications for Answerability Retrieval**	**§**
The strongest dense retriever reaches only 18.2% Top-1 Target Table retrieval despite consistently reaching the correct Sibling Table Group	The evaluated dense retrievers capture coarse semantic neighborhoods well, but struggle with the fine-grained discrimination needed to identify the uniquely answerable source	4.1
QA performance remains close to the oracle when the Target Table is ranked first, while adding more retrieved tables provides limited gains and can introduce interference	Retrieving the answerable source at high ranks is critical for downstream QA; additional semantically related but non-answerable sources can interfere with reasoning	4.2
Retrieval remains broadly stable across table serializations and semantically equivalent query formulations	The observed Semantic-Answerability Gap is not primarily an artifact of surface formatting or query wording in our evaluation	5.1
Complete row representations generally outperform representations retaining only a sufficient subset of attributes	The evaluated dense retrievers are more responsive to the amount of semantic content than to whether the representation contains only the information logically sufficient for identification	5.2
Row-level retrieval substantially outperforms retrieval over multi-row chunks	The evaluated dense embeddings can capture local row-level content, but aggregating multiple rows can dilute the information most relevant to the query	5.2
Ori and RID both outperform RID+Shuf, even though RID+Shuf retains explicit row indices	The evaluated dense embeddings remain relatively insensitive to fine-grained row–column bindings that can determine answerability	5.2
Target+Sibling groups form compact, low-dimensional neighborhoods, while the global embedding space remains broadly dispersed	For the models we evaluate, the discrimination failure is localized within semantically coherent neighborhoods rather than caused by global embedding collapse	5
Target–Sibling margins are consistently negative and correlate more strongly with $R@1$ than with $GR@1$	Remaining retrieval error concentrates on query-to-table discrimination after the correct semantic neighborhood has been reached	5
Query-conditioned answerability-aware reranking substantially improves $R@1$, from 0.182 to 0.450–0.574	Explicit answerability assessment can distinguish the Target from semantically similar Sibling Tables more effectively than single-vector retrieval in our setting	6
Answerability-oriented fine-tuning substantially improves in-distribution retrieval but degrades across independent datasets and can hurt broader semantic retrieval	Answerability can be learned by the embedding model we fine-tune, but jointly preserving answerability sensitivity and broad semantic retrieval remains challenging under the recipes we explore	6