Semantically Similar, Logically Distinct: Diagnosing the Semantic-Answerability Gap in Table RAG

Anonymous ACL submission

Abstract

Tables are a critical knowledge source in retrieval-augmented generation (RAG), but a retrieved table may lack sufficient evidence to answer a query, a property we call **answerability**. While answerability broadly concerns whether a source or collection of sources contains sufficient evidence, retrieval models optimized for semantic relevance do not guarantee it even in the single-source case, creating a fundamental mismatch. To study this, we introduce **TCR-Bench**[1], a diagnostic benchmark for **T**able **C**ontent-level Answerability in **R**AG, built around **sibling tables**, i.e., tables with highly similar schemas but subtle content differences. TCR-Bench reveals a persistent **Semantic-Answerability Gap**: dense retrievers frequently retrieve the correct sibling group yet fail to identify the uniquely answerable table, dropping QA performance from 0.755 (oracle) to 0.330 (top-5 retrieved). This gap stems from semantic accumulation, schema-level cue dependence, and weak row-column binding. We further adopt **Answerability-Aware Reranking (AAR)**, a lightweight two-stage pipeline applying direct query-table answerability judgment, which improves top-1 target retrieval from 18.2% to 57.4%, suggesting that much of the failure reflects missing answerability verification rather than insufficient model capacity alone.

1 Introduction

Tables in enterprise databases, data lakes, and web resources contain rich structured information (Cafarella et al., 2008; Stonebraker et al., 2013; Nargesian et al., 2018). In RAG systems (Lewis et al., 2020; Huang and Huang, 2024), retrieved sources must provide not only topical relevance but sufficient evidence to answer a query, a property we term **answerability**. Semantic relevance does not guarantee this: two sources may appear equally relevant while differing critically in whether they contain the required evidence. Table retrieval offers a clean setting for studying this problem, as precise row-column grounding and numerical matching (Herzig et al., 2020; Deng et al., 2022) make answerability failures easier to isolate.

Query: Who won more than 29 gold medals at the 2016 Olympics?

Retrieved Table A			Retrieved Table B		
Country	Year	Gold	Country	Year	Gold
USA	2016	46	GBR	2012	29
GBR	2016	27	GBR	2016	27
…	…	…	…	…	…
Dense Retriever Score = 0.88			Dense Retriever Score = 0.89		
Answerable			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 slightly higher because it more directly matches salient query tokens such as 2016 and 29.

Figure 1 illustrates the core challenge: two sibling tables with nearly identical schemas receive similar retrieval signals, yet only one contains sufficient evidence to answer the query. Such siblings are common in real data lakes (Lou et al., 2024; Yang et al., 2021; Shraga and Miller, 2023), arising from filtering, truncation, or minor revisions. We term this failure mode the **Semantic-Answerability Gap**: retrievers capture coarse similarity but fail to distinguish answerability. On our benchmark, dense retrievers identify the answerable table only **18.2%** of the time, causing QA performance to fall from **0.755** to **0.330**.

Existing table QA benchmarks such as WikiTableQuestions and HybridQA (Pasupat and Liang, 2015; Chen et al., 2020) assume the relevant table is given, bypassing retrieval entirely. Recent Table RAG work (Pan et al., 2022; Balaka et al., 2025; Yu et al., 2025) improves retrieval and reasoning, yet evaluation still emphasizes semantic

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

relevance over evidence sufficiency.

To study this problem systematically, we introduce **TCR-Bench**, a diagnostic benchmark for **T**able **C**ontent-level Answerability Benchmark in **R**AG. Built around sibling tables, TCR-Bench isolates answerability from coarse semantic relevance and enables controlled analysis of retrieval behavior. Our experiments show that the Semantic-Answerability Gap stems from semantic accumulation, schema-level cue dependence, and weak row-column binding, rather than superficial artifacts such as query phrasing or serialization format.

We further adopt **Answerability-Aware Reranking (AAR)**, a lightweight two-stage pipeline that reintroduces explicit answerability judgment, improving top-1 retrieval from **18.2%** to **57.4%**. This result suggests that a substantial portion of the failure stems from missing answerability verification rather than insufficient model capacity alone.

Our contributions are threefold. **(1) We identify and formalize the Semantic-Answerability Gap:** to our knowledge, the first work to define answerability as distinct from semantic relevance, and to show that dense retrievers systematically confuse the two. Within our controlled sibling-table setting, we trace this gap to three mechanisms: semantic accumulation, schema-level cue dependence, and weak row-column binding. **(2) We introduce TCR-Bench:** a sibling-table benchmark purpose-built to measure content-level answerability, isolating it from coarse semantic relevance and enabling controlled analysis of retrieval behavior. **(3) We propose AAR:** showing that explicit answerability verification substantially closes the gap, pointing to content-aware retrieval as a necessary direction.

2 Related Work

2.1 Beyond Semantic Relevance in Retrieval

Recent work has identified intrinsic limitations of embedding-based retrieval beyond coarse semantic similarity. Theoretical analyses show that single vector representations have fundamental expressivity limits (Weller et al., 2025; Vangara and Gopinath, 2026), with diminishing returns from scaling (Killingback et al., 2026) and degeneration effects in practice (Guo et al., 2024).

Semantic similarity alone is insufficient in certain settings, with (Belfathi et al., 2026) empirically revealing performance gaps on difficult instances without explicitly attributing them to answerability. Other works incorporate robustness (Liu et al., 2026), QA accuracy (Nian et al., 2025), or multi-hop reasoning (Trivedi et al., 2023; Asai et al., 2024), yet none define answerability as a dimension separate from semantic relevance. Our work formalizes this as the **Semantic-Answerability Gap**: retrieved documents may be semantically similar to the query while only one contains the specific evidence needed.

2.2 Table RAG Methods and Benchmarks

Table retrieval has progressed from table-specific dense retrievers (Herzig et al., 2021) to generic ones like DPR (Karpukhin et al., 2020), which (Wang et al., 2022) show can match specialized architectures. Thus, structural bias alone is insufficient for fine-grained content distinction, and retrieval performance on existing benchmarks is largely driven by semantic relevance. Simpler schema-aware and title-aware representations (Kim et al., 2024; Khanna and Subedi, 2025) prove effective, confirming that coarse relevance signals dominate current settings. Hybrid and multi-table pipelines (Chen et al., 2024b; Zhang et al., 2025a; Zhang and Chen, 2025) similarly optimize for relevance matching.

Existing benchmarks (Herzig et al., 2021; Zhang and Balog, 2018; Cui et al., 2025; Ji et al., 2024; Strich et al., 2026; Xu et al., 2025; Zou et al., 2025) increase difficulty through multi-hop reasoning, mixed modalities, or heterogeneous corpora, yet consistently frame retrieval as semantic matching. None requires distinguishing the uniquely answerable table among semantically similar candidates. TCR-Bench targets this gap directly through **sibling tables**, isolating fine-grained answerability selection from coarse semantic retrieval.

3 TCR-Bench: A Diagnostic Benchmark for Table Content-Level Answerability in RAG

3.1 From Semantic Relevance to Answerability

Existing Table RAG retrievers optimize for semantic relevance, i.e., topical similarity between query and table, but retrieval for generation requires **answerability**. Broadly, answerability means a source or collection of sources contains sufficient evidence to answer a query. In table retrieval, we operationalize it in the single-source setting as: a table is answerable if it provides all required rows,

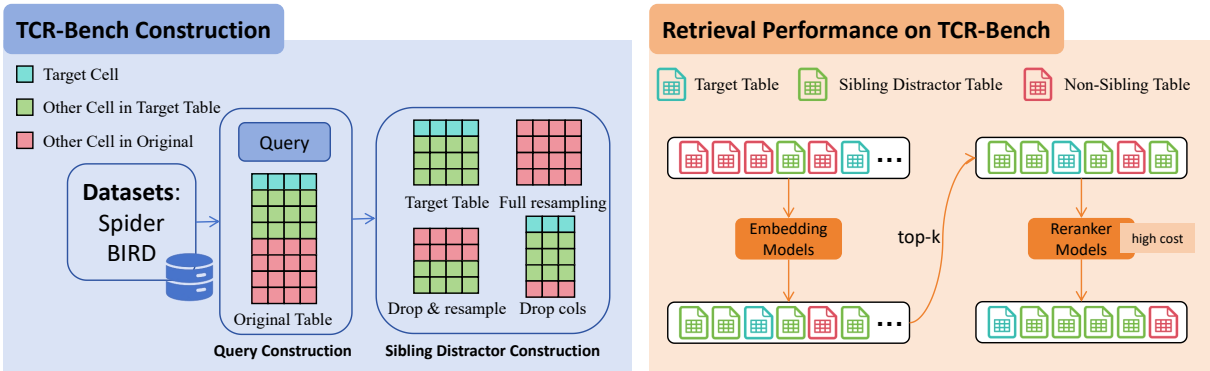

Figure 2: Overview of the TCR-Bench pipeline and retrieval performance. The left panel shows its construction, and the right panel shows that the embedding model can distinguish Sibling Distractor Tables from Non-Sibling Tables but confuses Target with Sibling Distractor Tables, while the reranker correctly separates all three at a higher computational cost.

values, and attribute bindings to produce a non-empty, constraint-consistent result.

We define three table types: **Target Tables**, which satisfy all answerability criteria; **Sibling Distractor Tables**, which are semantically similar but remove or violate answer-bearing evidence; and **Non-Sibling Tables**, which are unrelated in schema and semantics.

3.2 Operationalizing Answerability with Controlled Sibling Tables

TCR-Bench (**T**able **C**ontent-level Answerability Benchmark in **R**AG) is a **controlled diagnostic benchmark** designed to isolate answerability-aware retrieval from confounding factors such as multi-table reasoning and generation-stage effects. By design, it prioritizes diagnostic precision over corpus breadth, providing a focused setting for isolating whether retrievers can distinguish answer-bearing evidence from semantically similar but non-answerable content.

Each query is associated with one Target Table and multiple Sibling Distractor Tables derived from the same source table. As shown in Figure 2, Sibling Distractor Tables preserve strong semantic overlap while removing answerability under the query constraints, ensuring exactly one valid Target Table per query. While controlled, this design reflects realistic scenarios such as partial exports, schema projections, and filtered table versions.

Answerability-aware retrieval requires schema grounding, condition matching, value coverage, and row-column consistency. We use relatively large tables (up to ∼8K tokens) to expose the tension between semantic similarity and content-level answerability under compression pressure.

3.3 Benchmark Construction

3.3.1 Data Source and Task Design

TCR-Bench is built from two well-known Text-to-SQL benchmarks, Spider (Yu et al., 2018) and BIRD (Li et al., 2024), whose databases are derived from real-world relational tables across diverse domains. We consider TableQA with three query types: **Exact Match (EM)**, **Selective Filtering (SF)**, and **Selective Aggregation (SA)**, with 1-3 conjunctive conditions over target columns. Tables are rendered in **Markdown**, **HTML**, **CSV**, and **Mixed** formats.

Diagnostic subset. EM queries form a diagnostic subset where retrieval is restricted to sibling groups from the same source tables, isolating retrieval fidelity under minimal reasoning complexity and enabling controlled analysis of schema and query perturbations.

3.3.2 Sibling Distractor Construction

For each query, Sibling Distractor Tables are generated from the same source table by selectively removing query-condition or target columns and deleting or resampling rows violating query constraints. Query logic is then verified across the corpus to confirm that each query has exactly one answerable Target Table. To prevent shortcut matching via surface overlap, we subsequently apply paraphrased queries and schemas via GPT-OSS-120B (OpenAI, 2025), along with numerical and date perturbations. We further perform targeted manual inspection over the most likely semantically ambiguous candidates outside each sibling group, and remove cases that may introduce unintended alternative answer tables; full details are in

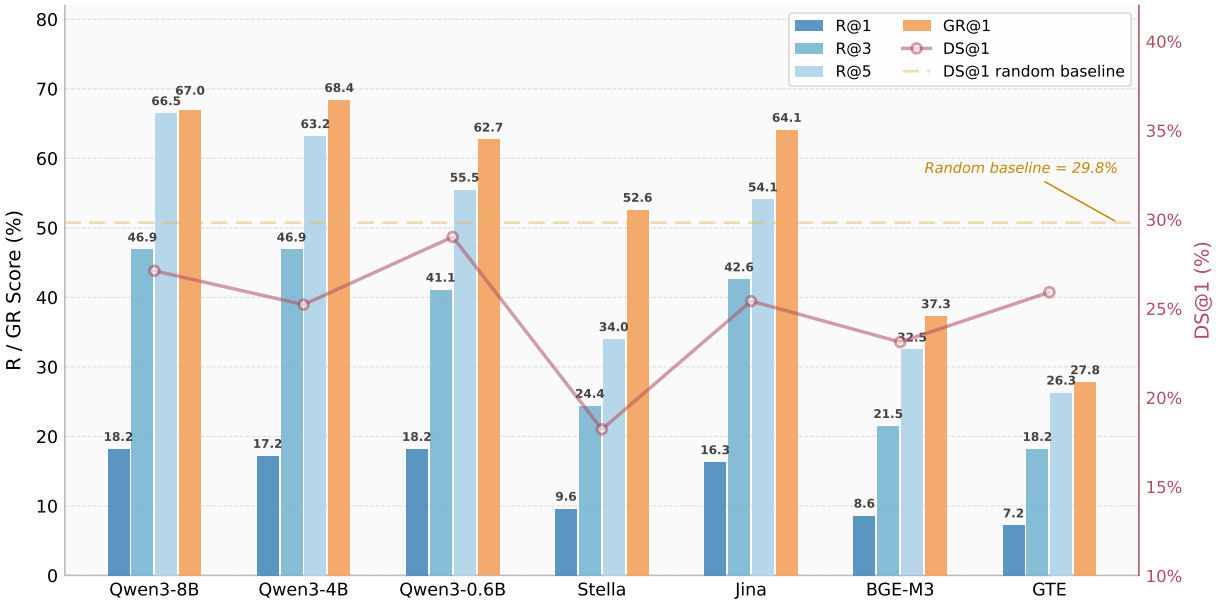

Figure 3: Retrieval performance on TCR-Bench using Mixed Format. Abbreviated model names are used.

Appendix A.

3.4 Benchmark Statistics

TCR-Bench contains 209 queries (98 EM, 55 SF, 56 SA) and 637 tables, with tables shared across query types. Despite its moderate scale, difficulty stems from dense, content-aware Sibling Distractor Tables and controlled perturbations rather than corpus size. Full statistics are in Appendix A. Future work will extend the framework to multi-table and heterogeneous-source settings, where answerability is compositional and evidence may span modalities beyond structured tables.

4 Primary Observations from Main Experiments

4.1 Retrieval Evaluation

4.1.1 Embedding Models

We evaluate representative embedding models spanning different architectural families and parameter scales, including Qwen3-Embedding-0.6B, Qwen3-Embedding-4B, Qwen3-Embedding-8B (Zhang et al., 2025c), stella_en_1.5B_v5 (Zhang et al., 2025b), jina-embeddings-v4 (Günther et al., 2025), bge-m3 (Chen et al., 2024a), and gte_Qwen2-7B-instruct (Li et al., 2023), as shown in Figure3; their abbreviations are used hereafter when unambiguous. These models cover both general-purpose and instruction-tuned embeddings, ranging from sub-billion to multi-billion parameter scales. Detailed 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., 2025; Strich et al., 2026) measures retrieval performance at a fixed granularity: $R@k = \frac{1}{N}\sum_{i=1}^{N}\mathbb{I}(y_i \in R_i^k)$. In our inter-table retrieval context, $R@k$ serves as the indicator of **answerability retrieval**, assessing whether the model pinpoints the exact evidence-bearing table.

Top-k Group Recall ($GR@k$) applies to inter-table retrieval. Let $\mathcal{G}_i$ denote the Sibling Table Group for query i. Inspired by $CR@k$ (Thakur et al., 2021), we define: $GR@k = \frac{1}{N}\sum_{i=1}^{N}\frac{|R_i^k \cap \mathcal{G}_i|}{\min(|\mathcal{G}_i|,k)}$. At the inter-table level, $GR@k$ maps to **coarse semantic relevance**, measuring whether the retriever lands in the correct topical neighborhood; we focus primarily on $GR@1$.

Top-1 Discriminative Score ($DS@1$): $DS@1 = R@1/GR@1$. This metric quantifies the gap between semantic relevance and answerability by measuring how well the model differentiates the target within the retrieved semantic group. Under random selection within a Sibling Table Group, the expected $DS@1$ equals the reciprocal of the average group size, which is 0.221 on the original dataset and 0.298 excluding column-deletion variants. A $DS@1$ near these bounds indicates the retriever struggles with content-aware answerability.

4.1.3 Main Results

Figure 3 reports retrieval performance under the **Mixed** setting. Full results are in Appendix D.1.

Across all models, a substantial **Semantic-Answerability Gap** emerges: semantic relevance is far easier to achieve than answerability. Even the strongest model reaches $GR@1 = 0.670$, yet its answerability retrieval remains at only $R@1 = 0.182$, revealing that successful semantic matching rarely translates into reliably identifying the truly answerable table. Scaling model size improves consistency but only partially addresses this bottleneck: increasing Qwen3 from 0.6B to 8B raises $R@5$ by 19.8%, yet top-1 improvements remain modest relative to group-level semantic matching.

Most models score near or below the random-selection baseline of 0.298 in $DS@1$ (with significance analysis in Appendix N confirming that these methods do not perform significantly better than chance), indicating that fine-grained discrimination among semantically matched sibling tables is not significantly better than chance. As an additional observation, in column-deletion cases that disrupt schema structure, top-1 rates drop to 0.0718 for Qwen3-4B and 0.0574 for Stella, indicating that models do remain sensitive to explicit structural mismatches. However, they fail to distinguish answerability differences within schema-consistent tables. This suggests that current dense retrieval representations capture topical and schema-level similarity effectively, but fundamentally lack the sensitivity required for fine-grained evidence verification such as precise row-column bindings, value constraints, and condition consistency.

This pattern is not limited to standard dense embeddings: alternative first-stage paradigms, including DTR-style, ColBERT-style, sparse, and hybrid retrieval, also fail to substantially close the answerability gap (Appendix D.2 and D.3)

4.2 Downstream QA Evaluation

To examine how retrieval quality propagates to generation, we evaluate downstream QA performance using retrieved tables.

4.2.1 Experimental Setup

We evaluate the QA pipeline of LongTableBench (Li et al., 2025) using Qwen3-30B-A3B-Thinking-2507 (Team, 2025) as the downstream model, reporting average F1 under four settings: (1) Oracle (Target Table only), (2) No-Table, and (3) Retrieved Top-k Tables ($k \in \{1, 3, 5\}$), where retrieved tables are concatenated in retrieval order.

To analyze how effectively downstream QA utilizes retrieved evidence, we define $\text{Eff}@k = (\text{QA}@k/\text{QA}@\text{Oracle})/\text{R}@k$, where $QA@k$ denotes QA F1 using Top-k retrieved tables. Intuitively, $Eff@k$ measures whether the QA model can effectively exploit the Target Table once it appears in the retrieved set, with values near 1 indicating minimal interference from additional tables.

4.2.2 Downstream QA Results

Table 1 summarizes downstream QA performance. The Oracle setting achieves $F1 = 0.755$, while No-Table drops to 0.014, confirming that successful answering depends heavily on retrieved tabular evidence rather than parametric memorization. This near-zero No-Table result also makes the efficiency analysis more reliable, since downstream performance is overwhelmingly determined by retrieval quality.

Method	QA@1	QA@3	QA@5	Eff@1	Eff@3	Eff@5
Qwen3-0.6B	**0.144**	0.244	0.276	1.046	0.462	0.659
Qwen3-4B	0.113	**0.269**	0.329	0.871	0.320	0.690
Qwen3-8B	0.139	0.266	**0.330**	1.014	0.393	0.657

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.

Although recall increases with larger k, downstream gains remain limited: even Top-5 retrieval reaches only $F1 = 0.330$. Meanwhile, $Eff@1$ remains close to 1, indicating near-optimal QA once the Target Table is ranked first. The substantial drops in $Eff@3$ and $Eff@5$ are consistent with prior observations that additional semantically related context can introduce interference in RAG systems (Yoran et al., 2024; Kim and Lee, 2024; Iratni et al., 2025; Ouyang et al., 2025).

4.3 Summary and Reflections on the Answerability Bottleneck

Our experiments reveal a systemic gap between coarse semantic relevance and fine-grained answerability in retrieval. While embedding models reliably locate the correct topical neighborhood, $DS@1$ scores near or below the random-selection baseline confirm that discriminating the exact evidence-bearing table within a sibling group remains highly unsolved. Downstream QA results reinforce this picture: $Eff@k$ degrades sharply

beyond k=1, underscoring that retrieval precision at the top rank is the decisive factor for generation quality. Together, these results establish the Semantic-Answerability Gap as a fundamental bottleneck in Table RAG, which we investigate further in subsequent experiments. We first ask whether surface-level factors can account for this failure, before probing deeper into the retrieval objective itself.

The Answerability Bottleneck in Table RAG

Dense retrievers achieve strong coarse semantic relevance but exhibit near-chance fine-grained answerability. This **Semantic-Answerability Gap** persists across all evaluated models and scales.

5 Investigation I: Surface Variations Do Not Explain Answerability Failures

The results in Section 4 reveal a large gap between group-level retrieval and exact target identification. A natural question is whether this failure reflects sensitivity to surface-level artifacts, such as serialization choices or query phrasing, or a more persistent mismatch between the semantic retrieval objective and answerability discrimination. We examine both factors on the **Diagnostic Subset** using Qwen3-4B and Stella.

5.1 Effect of Table Serialization Format

We evaluate six table formats: **Markdown**, **CSV**, **HTML**, **Mixed**, **Sen** (Sentence), and **SenS** (Sentence_Shuffle). **Sen** converts each row into a declarative sentence while preserving row order; **SenS** applies the same conversion but independently shuffles the attribute order within each sentence, motivated by prior evidence that models are sensitive to column order (Cong et al., 2023).

As shown in Table 2, performance remains relatively stable across all formats: for Qwen3-4B, $R@1$ ranges from 0.186 to 0.257 and $GR@1$ from 0.757 to 0.857. Structured formats and sentence-based representations yield comparable results, and the similar performance between Sen and SenS further indicates that attribute order within rows contributes little to retrieval decisions. Consistency analysis (Appendix E.2) confirms that models retrieve highly overlapping candidate sets across formats, though internal ranking may vary.

5.2 Effect of Query Paraphrasing

We evaluate three query variants: **Final Query (NL)**, the natural-language query used in the main benchmark; **Sentential Query**, which explicitly expresses the same condition in sentence form; and **Template Query**, a structured template emphasizing attribute-value constraints.

As shown in Table 3, retrieval performance shows only moderate variation across formulations. Consistency analysis (Appendix E.3) shows that candidate sets remain largely stable under paraphrasing even when the Target Table's ranking shifts, suggesting models capture logical intent but lack the granularity to discriminate among structurally similar tables.

Model	Format	R@k			GR@1	DS@1
		@1	@3	@5		
Qwen3-4B	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
	Sen	0.200	0.529	0.743	0.800	0.250
	SenS	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
	Sen	**0.314**	**0.500**	0.643	**0.786**	**0.400**
	SenS	0.286	0.471	**0.671**	0.771	0.370

Table 2: Retrieval performance under different table serialization formats, showing largely stable performance across formats.

Model	Query Type	R@k			GR@1	DS@1
		@1	@3	@5		
Qwen3-4B	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

Table 3: Retrieval performance under different query formulations, showing largely stable performance across query types.

Answerability Gap Transcends Surface Variations

Neither serialization format nor query phrasing explains the observed retrieval failures. The stability of candidate sets across perturbations confirms that the bottleneck lies in an objective-level mismatch: semantic retrieval handles topical filtering well but is poorly aligned with the fine-grained answerability discrimination required to identify a unique Target Table.

6 Investigation II: Semantic Retrieval Collapses Answerability Distinctions

Having ruled out surface artifacts, we investigate why dense semantic retrieval is poorly aligned with

answerability despite strong group-level semantic matching. We identify three patterns showing that the contrastive alignment objective causes retrievers to accumulate topical signal rather than verify the precise evidence required to answer a query.

6.1 Semantic Volume Bias: Evidence from Sentential Queries

We probe whether retrieval is driven by semantic coverage or logical sufficiency by converting table rows into natural language sentences under three transformations: **(1) Sen**: full row in original order; **(2) SenS**: all column-value pairs with randomized phrase order; **(3) SenSS**: a shuffled subset of column-value pairs sufficient to uniquely identify the row. We report inter-table (Target Table selection) and intra-table (row localization) retrieval using Qwen3-4B and Stella on the Diagnostic Subset under Mixed format.

Table 4 and Figure 4 reveal three patterns. First, retrieval scales with semantic volume rather than sufficiency: reducing queries to subsets degrades accuracy if the subset uniquely identifies the row (e.g., Qwen3-4B row-level $R@1$ drops 0.871 to 0.619). This suggests that contrastive training rewards broad topical coverage, not constraint completeness. Second, models are largely insensitive to phrase order (Sen vs. SenS ≤ 0.02), treating sentences as unordered token pools. Third, row-level retrieval outperforms 10-line chunk retrieval (0.871 vs. 0.387 for Qwen3-4B), consistent with length-induced embedding collapse (Zhou et al., 2025). Despite high row-level $R@1$ (>0.85), inter-table $R@1$ remains <0.26, indicating that local key-value associations are captured but do not translate into global answerability discrimination.

Model	Type	R@k			GR@1	DS@1
		@1	@3	@5		
Qwen3-4B	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

Table 4: Inter-table retrieval by query type: Sen/SenS consistently outperform SenSS while phrase order has minimal impact.

6.2 Failure of Row-Column Binding Under Column Shuffling

Using the **Diagnostic Subset**, we construct a **Minimal Contrastive Set** for each table, consisting of the Target Table and a column-shuffled distractor. This setup follows the column-wise shuffling protocol in (Wang et al., 2022). We evaluate three configurations (details in Appendix I): (1) Ori (original table), (2) RID (each cell prefixed with its row index), and (3) RID+Shuf (row-indexed with column shuffling applied).

As shown in Table 5, Ori achieves the strongest discrimination (Qwen3-4B Mixed: $R@1 = 0.443$), while RID+Shuf reduces accuracy to 0.314 and row-ID augmentation provides only a modest improvement (0.371), indicating that explicit positional signals do not restore binding between values and their column semantics. That column-shuffled tables remain sufficiently similar to confuse retrieval suggests that the semantic retrieval objective does not require row-column consistency to be preserved: aggregate topical overlap is sufficient to produce near-identical retrieval scores. Full results across all formats appear in Appendix H.

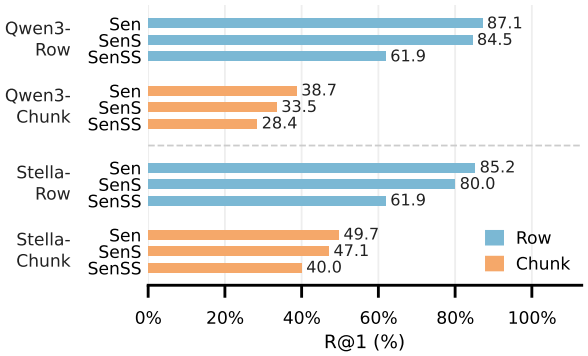

Figure 4: Intra-table retrieval by granularity and query type: row-level outperforms 10-line chunks; SenSS degrades accuracy despite logical sufficiency. Qwen3: Qwen3-4B. Full results in Appendix F.

Model	Config	Format	R@1	GR@1	DS@1
Qwen3-4B	Ori	Mixed	0.443	0.671	0.660
Qwen3-4B	Ori	Markdown	0.343	0.643	0.533
Stella	Ori	Mixed	0.229	0.471	0.485
Qwen3-4B	RID	Mixed	0.371	0.671	0.553
Stella	RID	Mixed	0.243	0.471	0.515
Qwen3-4B	RID+Shuf	Mixed	0.314	0.643	0.489
Stella	RID+Shuf	Mixed	0.200	0.429	0.467

Table 5: Retrieval on Minimal Contrastive Set. Column shuffling incurs only a modest penalty, indicating that the retrieval objective does not enforce row-column binding.

6.3 Conclusion: Systemic Failure Modes

Experiments reveal three core failure modes that expose fundamental limitations of current LLM-

based dense embeddings for tabular data.

Retrieval reward depends on token density Performance scales with token density rather than logical sufficiency, suggesting embeddings favor semantic volume over complete reasoning cues.

Dependence on schema-level cues Models overrely on high-level schema signals, limiting discrimination among structurally similar tables and reflecting absent relational supervision.

Weak row-column bindings Row-column associations are weak, as column-shuffled distractors incur little penalty, suggesting embeddings act as bag-of-phrases aggregators rather than structured relational representations.

Additional analyses of other factors (e.g., schema sensitivity and positional effects) are provided in Appendix J, K, and L.

7 Answerability-Aware Reranking for Table Retrieval

To examine the importance of explicit answerability modeling at the retrieval stage, we introduce **Answerability-Aware Reranking** (**AAR**), which evaluates each candidate jointly with the query. Unlike single-vector retrieval, cross-encoder interaction enables direct assessment of whether a table contains sufficient structured evidence to satisfy the query, making answerability an explicit scoring objective rather than an implicit byproduct of semantic similarity.

We instantiate AAR with two variants: (1) **AAR-CE**, a cross-encoder initialized from the same Qwen3 family as the embedding retriever (Qwen3-Embedding-8B with Qwen3-Reranker-8B), ensuring comparable model capacity and identical initialization lineage under different scoring paradigms; and (2) **AAR-Judge**, an LLM-based binary answerability judge (Balaka et al., 2025) using Qwen3-30B-A3B-Thinking-2507 ($Temperature$=0).

Method	R@1	R@5	GR@1	DS@1
AAR-CE	**0.574**	**0.766**	**0.818**	0.702
AAR-Judge	0.536	0.737	0.761	**0.704**

Table 6: Answerability-aware reranking on the Qwen3-Embedding-8B backbone (k=10). Adding query-conditioned interaction substantially improves exact table selection on TCR-Bench.

Both variants substantially improve retrieval over the dense baseline. AAR-CE raises R@1 from 0.182 to 0.574 and $GR@1$ from 0.670 to 0.818; AAR-Judge achieves $R@1 = 0.536$ and the best $DS@1 = 0.704$. Because the reranker and retriever share the same model family and comparable initialization, these gains suggest that explicit interaction-based answerability modeling rather than differences in model capacity. Full results and significance analysis are in Appendix M and N.

Answerability Modeling as a Core Retrieval Objective

Explicit answerability modeling substantially improves exact target identification, suggesting that a major bottleneck in Table RAG retrieval lies in the lack of interaction-based answerability assessment during the initial retrieval phase. While AAR provides a practical mitigation, natively incorporating answerability into the retrieval objective, rather than deferring it to a reranking stage, remains a promising direction for future work.

8 Conclusion

The **Semantic-Answerability Gap** identifies a fundamental retrieval failure mode: high semantic relevance does not guarantee that a retrieved source contains sufficient evidence to answer a query. We introduce **TCR-Bench**, a diagnostic benchmark built around sibling tables, to study this gap in a controlled setting where answerability can be precisely evaluated. Experiments show that dense retrievers reliably identify semantically relevant candidate groups yet fail to identify the uniquely answerable table within them, leading to a sharp drop from oracle QA performance of 0.755 to 0.330 with top-5 retrieved tables. Controlled analyses attribute this gap to three patterns (*Semantic Accumulation*, *Weak Row-Column Binding*, and *Schema-Level Cue Dependence*), reflecting the mismatch between contrastive alignment objectives and fine-grained content verification. **AAR** substantially closes this gap via explicit query-table interaction, confirming that the bottleneck lies in answerability modeling rather than model scale or serialization. This motivates a broader shift in retrieval system design toward treating answerability as a core objective, beyond coarse semantic matching.

Limitations

To the best of our knowledge, this work is the first to explicitly define and systematically study the Semantic-Answerability Gap as a distinct retrieval problem. The gap is defined under a controlled setting where answerability can be isolated from semantic relevance, but the concept extends beyond

tabular retrieval. In broader retrieval environments, answerability may become less tractable: passages may provide partial evidence, multi-source settings distribute supporting facts across documents, and verification tasks may admit no fully satisfying source. These settings call for richer notions of partial or compositional answerability, an open direction for the field.

At the benchmark level, TCR-Bench intentionally balances realism and controllability, preserving practically motivated properties such as schema overlap and incomplete exports while constraining the retrieval space for precise diagnostic analysis. Real-world table ecosystems remain substantially more complex, with noisier schemas, evolving tables, and heterogeneous evidence sources.

AAR mitigates rather than closes the Semantic-Answerability Gap. Reranking improves target identification via explicit query-table interaction, suggesting answerability can be recovered with better evidence sufficiency modeling. However, this adds cost and depends on initial retrieval quality. Overall, future systems should embed content- and structure-aware representations earlier, balancing efficiency and precise answerability.

Ethical Considerations

TCR-Bench is constructed entirely from publicly available datasets (Spider and BIRD) and synthetically generated data, containing no personally identifiable or sensitive information. Semantic perturbations and query templates were generated automatically and manually verified by the authors to ensure quality and appropriateness. Large language models were used solely for paraphrasing queries and schemas, minor code generation assistance, and minor manuscript polishing. To promote transparency and reproducibility, we plan to publicly release all data, code, and evaluation protocols upon publication. While TCR-Bench is intended to advance retrieval research in Table RAG, practitioners applying it to sensitive domains (e.g., healthcare, finance) should incorporate appropriate domain-specific safeguards.

References

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

Muhammad Imam Luthfi Balaka, David Alexander, Qiming Wang, Yue Gong, Adila Krisnadhi, and Raul Castro Fernandez. 2025. 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.

Anas Belfathi, Nicolas Hernandez, Laura Monceaux, Warren Bonnard, and Richard Dufour. 2026. Semantic reranking at inference time for hard examples in rhetorical role labeling. *arXiv preprint arXiv:2605.18007*.

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

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *Preprint*, arXiv:2402.03216.

Peter Baile Chen, Yi Zhang, and Dan Roth. 2024b. 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)*, pages 2687–2699.

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

Tianji Cong, Madelon Hulsebos, Zhenjie Sun, Paul Groth, and HV Jagadish. 2023. Observatory: Characterizing embeddings of relational tables. *Proceedings of the VLDB Endowment*, 17(4):849–862.

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

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.

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

Xingzhuo Guo, Junwei Pan, Ximei Wang, Baixu Chen, Jie Jiang, and Mingsheng Long. 2024. On the embedding collapse when scaling up recommendation models. In *International Conference on Machine Learning*, pages 16891–16909. PMLR.

Michael Günther, Saba Sturua, Mohammad Kalim Akram, Isabelle Mohr, Andrei Ungureanu, Sedigheh Eslami, Scott Martens, Bo Wang, Nan Wang, and Han Xiao. 2025. jina-embeddings-v4: Universal embeddings for multimodal multilingual retrieval. *Preprint*, arXiv:2506.18902.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. 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*, pages 512–519.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4320–4333.

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

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

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

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 6769–6781.

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

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

Julian Killingback, Mahta Rafiee, Madine Manas, and Hamed Zamani. 2026. Scaling laws for embedding dimension in information retrieval. *arXiv preprint arXiv:2602.05062*.

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

Kiseung Kim and Jay-Yoon Lee. 2024. 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*, pages 22149–22161.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, and 1 others. 2024. 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.

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

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

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

Peiyang Liu, Qiang Yan, Ziqiang Cui, Di Liang, Xi Wang, and Wei Ye. 2026. Beyond semantic relevance: Counterfactual risk minimization for robust retrieval-augmented generation. *arXiv preprint arXiv:2605.01302*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2024b. 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)*, pages 450–482.

Yuze Lou, Chuan Lei, Xiao Qin, Zichen Wang, Christos Faloutsos, Rishita Anubhai, and Huzefa Rangwala. 2024. Datalore: Can a large language model find all lost scrolls in a data repository? In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 5170–5176. IEEE.

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

Jinming Nian, Zhiyuan Peng, Qifan Wang, and Yi Fang. 2025. 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)*, pages 136–146.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Jie Ouyang, Tingyue Pan, Mingyue Cheng, Ruiran Yan, Yucong Luo, Jiaying Lin, and Qi Liu. 2025. 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)*, pages 6036–6063.

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

Panupong Pasupat and Percy Liang. 2015. 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)*, pages 1470–1480.

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

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*.

Roee Shraga and Renée J Miller. 2023. Explaining dataset changes for semantic data versioning with explain-da-v. *Proceedings of the VLDB Endowment*, 16(6).

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

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

Jan Strich, Enes Kutay Isgorur, Maximilian Trescher, Chris Biemann, and Martin Semmann. 2026. 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)*, pages 165–191.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. 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*, pages 645–654.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. 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)*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. 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)*, pages 10014–10037.

Anirudh Bharadwaj Vangara and Ashwin Gopinath. 2026. The geometry of consolidation. Submitted to NeurIPS 2026. arXiv version in this repository.

Blerta Veseli, Julian Chibane, Mariya Toneva, and Alexander Koller. 2025. Positional biases shift as inputs approach context window limits. *arXiv preprint arXiv:2508.07479*.

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

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

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

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

Junwen Yang, Yeye He, and Surajit Chaudhuri. 2021. Auto-pipeline: synthesizing complex data pipelines by-target using reinforcement learning and search. *Proceedings of the VLDB Endowment*, 14(11):2563–2575.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2024. Making retrieval-augmented language models robust to irrelevant context. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. 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*.

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

Yijiong Yu, Huiqiang Jiang, Xufang Luo, Qianhui Wu, Chin-Yew Lin, Dongsheng Li, Yuqing Yang, Yongfeng Huang, and Lili Qiu. 2024. Mitigate position bias in large language models via scaling a single dimension. *arXiv preprint arXiv:2406.02536*.

Chi Zhang and Qiyang Chen. 2025. Hd-rag: Retrieval-augmented generation for hybrid documents containing text and hierarchical tables. *arXiv e-prints*, pages arXiv–2504.

Chi Zhang, Qiyang Chen, and Mengqi Zhang. 2025a. Mixture-of-rag: Integrating text and tables with large language models. *arXiv preprint arXiv:2504.09554*.

Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025b. Jasper and stella: distillation of sota embedding models. *Preprint*, arXiv:2412.19048.

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

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. 2025c. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yuqi Zhou, Sunhao Dai, Zhanshuo Cao, Xiao Zhang, and Jun Xu. 2025. Length-induced embedding collapse in transformer-based models.

Jiaru Zou, Dongqi Fu, Sirui Chen, Xinrui He, Zihao Li, Yada Zhu, Jiawei Han, and Jingrui He. 2025. Rag over tables: Hierarchical memory index, multi-stage retrieval, and benchmarking. *arXiv preprint arXiv:2504.01346*.

A Benchmark Construction, Augmentation, and Representation

A.1 Distractor Sampling Parameters

For the generation of sibling distractors, we adhere to the following constraints:

- **Length Constraint:** All sub-tables are sampled to a length of 6,500-7,800 tokens (via `tiktoken`[2]).
- **Augmentation Strategies:** We employ three augmentation strategies to construct distractor tables:
  - **Removing Condition or Target Columns:** We remove either the condition column or the target column from the original table (only one column is removed at a time) to prevent direct retrieval of the answer while preserving the remaining table structure.
  - **Row Deletion:** We randomly delete 30%-70% of the rows from the original table and then resample additional rows from the remaining corpus until the resulting table falls within the predefined token length range.
  - **Row Resampling:** We completely resample rows from the table corpus to construct a new table of the required length.
- **Answer Filtering:** For both the row deletion and row resampling strategies, we explicitly filter out any rows that would allow the query to be answered correctly. Furthermore, we conduct a global verification step across all table groups to ensure that no distractor table, including those sampled from other groups, contains rows that could answer the current query.
- **Verification:** Human experts corrected approximately 5% of the GPT-generated paraphrases where semantic shifts occurred.

A.2 Representation Transformation Rules

To rigorously evaluate whether embedding models rely on surface-level token matching, we apply three distinct types of transformations:

A.2.1 Date Format Augmentation

Standard $YYYY-MM-DD$ dates are transformed into multiple representations:

- **English Abbreviations:** e.g., *15/Jan/2023*, *Jan. 15, 2023*.
- **Full English Spellings:** e.g., *January 15th, 2023*.
- **Roman Numerals:** e.g., *2023-I-15*, where year, month, and day are in Roman numerals.
- **Numeric Words:** e.g., *two thousand twenty-three. one. fifteen*.

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

A.2.2 Numeric and Scale Perturbations

Numeric columns are augmented to create alternative representations:

- **Scaling Large Numbers:** Numbers in the thousands are converted to compact forms with *k* or *K* suffix, e.g., $15000 \rightarrow 15k$.
- **Multiplicative and Additive Shifts:** Small numeric perturbations such as multiplying/-dividing by 10 or 100, or adding/subtracting small constants.
- **Numeral Systems Conversion:** Numbers are converted to Roman numerals or English words, e.g., $102 \rightarrow C$ or *One hundred and two*.

A.3 Choice of GPT-OSS-120B for Semantic Augmentation

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 was among the strongest locally deployable open-weight models, offering full reproducibility without reliance on proprietary API endpoints and strong instruction-following capability, which is well-suited for controlled paraphrase generation. To mitigate potential biases that this GPT-based model might introduce, all subsequent components, including dense retrievers, rerankers, and downstream QA models, are explicitly chosen from non-GPT families.

A.4 Manual Validation and False-Negative Reduction

To improve annotation quality and reduce false negatives, we perform a multi-stage manual validation procedure over both query answerability and schema transformations.

First, uniqueness is guaranteed within each sibling group by construction: only the Target Table satisfies all query constraints. We then apply the query logic globally to the full corpus to eliminate surface-level duplicate answerability.

To further reduce semantic false negatives, we conduct targeted manual inspection over the most likely ambiguous candidates. Specifically, for each query, we retrieve the top-k candidate tables outside the corresponding sibling group using multiple similarity measures. The candidate sets from different retrieval signals are merged, and the resulting tables are manually inspected for semantic equivalence. This process focuses on cases where different schemas or column names may express the same underlying concept, potentially leading to unintended alternative answer tables. Ambiguous cases are removed from the benchmark.

In addition, we manually verify the validity of all queries and schema transformations introduced during data construction. This includes checking that rewritten queries preserve the original intent and constraints, and that schema-level transformations remain semantically consistent with the source tables without introducing annotation artifacts or unintended shortcuts.

While it is impossible to completely eliminate every potential false negative in the benchmark, this validation pipeline substantially reduces the risk to a controllable level. More importantly, our main experimental conclusions remain robust to the residual noise: the performance gap between first-stage retrieval and answerability-oriented reranking is still large and statistically significant across evaluations (see Appendix N for detailed significance analysis).

A.5 Geometric Characteristics of the Embedding Space

We analyze the geometric properties of table-level embeddings produced by different models, following the diagnostic framework introduced in (Vangara and Gopinath, 2026). The analysis focuses on two key indicators of the embedding space: **effective dimensionality** (d_{eff}) and **mean pairwise cosine distance** ($\bar{d}$).

Effective dimensionality, defined as the participation ratio of the covariance spectrum, measures how many independent directions carry meaningful variance in the embedding space. A lower value indicates more concentrated information. The mean pairwise cosine distance captures the overall separation between table embeddings in the vector space, with higher values reflecting greater semantic distinctiveness.

We computed these two metrics for 637 data tables across six model/format combinations, with embedding dimensions ranging from 1024 to 2560. Results are shown in Table 7.

Two observations emerge from the results:

1. **The embedding space exhibits clear inter-table separation.** Mean pairwise cosine dis-

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 7: Geometric indicators of the embedding space across models and formats

tances range from 0.3958 to 0.5491 across all models, indicating that even structurally similar tables remain well-separated in the embedding space.

2. **Effective dimensionality is consistently high.** All models exhibit d_{eff} values between 43 and 59, substantially higher than the typical values reported for natural language text ($\approx$16) in (Vangara and Gopinath, 2026). This suggests that table-level embeddings encode semantically rich and diverse information that cannot be fully captured by a small number of principal directions.

Taken together, these two indicators characterize our embedding space as well-separated and information-diverse. This geometric profile provides a useful reference for interpreting retrieval behavior.

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., 2025c).** The **Qwen3** embedding series includes **Qwen3-0.6B**, **Qwen3-4B**, and **Qwen3-8B**. 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., 2025b). 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.

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

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]
```

Table 8: Instruction prepended to user queries for the retrieval 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:

Table 9: The structured prompt used for the downstream TableQA generation stage.

prepend a specific instruction to the user query to guide the embedding space towards TableQA relevance. The exact instruction used is detailed in Table 8. Note that traditional lexical methods, such as BM25 (Robertson and 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 9, the system prompt ensures that the model outputs are parsable and normalized across diverse table schemas.

D Comprehensive Retrieval Results

D.1 Dense Embedding Retrieval Results

Table 10 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 and Zaharia, 2020), sparse lexical retrieval (BM25) (Robertson and Zaragoza, 2009), and symbolic-neural hybrid retrieval.
```

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
	Sen	0.144	0.297	0.397	0.478	0.545	0.641	0.624	0.596	0.573	0.560	0.224
	SenS	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
	Sen	0.139	0.306	0.440	0.545	0.632	0.684	0.682	0.667	0.656	0.653	0.203
	SenS	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
	Sen	0.148	0.297	0.431	0.531	0.593	0.636	0.639	0.611	0.603	0.596	0.233
	SenS	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
	Sen	0.120	0.258	0.368	0.426	0.483	0.517	0.510	0.499	0.477	0.466	0.231
	SenS	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
	Sen	0.096	0.163	0.225	0.311	0.349	0.354	0.328	0.330	0.341	0.329	0.270
	SenS	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
	Sen	0.177	0.273	0.368	0.474	0.522	0.550	0.548	0.530	0.513	0.503	**0.322**
	SenS	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
	Sen	**0.206**	0.306	0.397	0.502	0.550	0.641	0.615	0.609	0.598	0.582	0.321
	SenS	0.163	0.292	0.397	0.493	0.560	0.632	0.632	0.617	0.602	0.588	0.258

Table 10: Comprehensive evaluation results across different input formats.

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
Qwen3-4B + BM25 hybrid	Hybrid	0.215	0.455	0.612	0.689	0.312

Table 11: Retrieval performance across alternative retrieval paradigms.

Model Descriptions For DTR, we evaluate two variants: `tapas_nq_retriever_large` and `tapas_nq_hn_retriever_large`. The "nq" suffix indicates 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 and Zaharia, 2020) is designed for passages and does not natively support 8K-token contexts. We therefore adopt three ColBERT-style models from HuggingFace that have been extended or adapted for longer sequences, namely `Reason-ModernColBERT`[3], `SauerkrautLM-Multi-ModernColBERT`[4], and `reason-colBERT-150M-GTE-ModernColBERT`[5], as available open-source checkpoints without further fine-tuning. 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.

Results Table 11 summarizes the retrieval performance across paradigms.

Analysis Across paradigms, DTR models exhibit extremely low retrieval accuracy, with R@1 remaining below 1%. ColBERT-style late-interaction retrievers substantially improve recall metrics but still fail to reliably distinguish answerable tables from structurally similar distractors. The symbolic-neural hybrid setup achieves marginal improvements over pure dense retrieval, but increasing the BM25 contribution consistently degrades performance.

These results suggest that augmenting dense retrieval with symbolic lexical signals provides limited complementary information in this setting. More importantly, the failure pattern is consistent across dense, late-interaction, sparse, and hybrid paradigms, indicating that the bottleneck is not architecture-specific. Instead, the core difficulty arises from the intrinsic ambiguity among structurally similar tables.

D.3 LLM-Augmented Dense Retrieval

We further evaluate several augmentation strategies commonly used in retrieval pipelines to determine whether the performance gap observed in the main experiments can be mitigated by adding auxiliary semantic signals.

Experimental Setup Inspired by prior work (Gao et al., 2023; Shen et al., 2023), 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 12 summarizes the results under the Qwen3-Embedding-8B retrieval backbone.

Analysis Both semantic augmentation methods provide only limited improvements over the baseline dense retrievers. Meta-information generation slightly improves recall metrics but does not

[3] https://huggingface.co/lightonai/Reason-ModernColBERT
[4] https://huggingface.co/VAGOsolutions/SauerkrautLM-Multi-ModernColBERT
[5] https://huggingface.co/fjmgAI/reason-colBERT-150M-GTE-ModernColBERT

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

Table 12: Retrieval performance of augmentation strategies across different embedding models.

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.

This observation suggest that simply enriching the semantic representation of tables is insufficient to resolve ambiguity among structurally similar tables. In contrast, the reranking approaches presented in the main text (AAR) are more effective because they enable direct query-table interaction during scoring.

E Consistency Analysis Across Surface Variations

To better understand retrieval stability under surface-level perturbations, we conduct a detailed consistency analysis across both table serialization formats and query formulations. The goal is to determine whether retrieval errors stem from sensitivity to superficial variations or from deeper representational limitations.

E.1 Consistency Metrics

Beyond absolute accuracy, we evaluate four complementary consistency metrics comparing retrieval outputs under different configurations:

- **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**: measured using 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.

These metrics allow us to distinguish between three possible scenarios: (1) identical retrieval behavior, (2) stable candidate sets with different rankings, and (3) completely divergent retrieval outputs.

E.2 Serialization Format Consistency

Table 13 reports consistency metrics across serialization formats using the **Mixed** format as the base configuration.

Several observations emerge. First, **Hit Consistency** and **Partial Consistency** remain high across most format pairs, indicating that models retrieve largely overlapping candidate sets regardless of serialization conventions. Second, the particularly strong consistency between **Sentence** and **Sentence_Shuffle** suggests that retrieval embeddings are largely insensitive to row ordering.

However, **Ranking Consistency** remains comparatively low (RBO around 20-30%), indicating that while candidate tables remain similar, their internal ranking positions frequently shift. This pattern implies that serialization changes do not disrupt coarse semantic matching but can influence fine-grained similarity scoring.

E.3 Query Formulation Consistency

We also analyze retrieval stability under different query formulations. Table 14 reports consistency metrics using the **Final Query** as the base configuration.

The results show that retrieval behavior remains highly stable under query paraphrasing. In particular, the high **Partial Consistency** indicates that candidate tables remain largely invariant across query formulations. Meanwhile, the moderate **Top-1 Identity** scores suggest that ranking differences arise primarily from subtle scoring shifts rather than changes in semantic understanding.

Model	Config 1 vs. 2	Hit Consis.	Ranking (RBO)	Partial Consis.	Top-1 Id.
Qwen3-4B	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. Sentence	75.71%	25.88%	92.86%	38.57%
	Mixed vs. Sentence_shuf	82.86%	25.06%	88.57%	41.43%
	Sent. vs. Sent_shuf	**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. Sentence	77.14%	20.43%	82.86%	34.29%
	Mixed vs. Sentence_shuf	74.29%	20.05%	78.57%	35.71%
	Sent. vs. Sent_shuf	**91.43%**	**34.43%**	**95.71%**	**77.14%**

Table 13: Consistency analysis across serialization formats and narrative structures.

Model	vs. Query	Hit Consis.	Ranking (RBO)	Partial Consis.	Top-1 Id.
Qwen3-4B	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%

Table 14: Consistency analysis across query variations (Base: Final Query).

Overall, these results confirm that query phrasing does not substantially alter the semantic interpretation captured by the retrievers.

F Detailed Intra-table Retrieval Results

For completeness, we report the full numerical results corresponding to Figure 4 in Table 15. The table provides exact R@k values for both row-level and chunk-level retrieval across different query types.

Model	Config	Type	R@1	R@3	R@5
Qwen3-4B	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 15: Intra-table retrieval by granularity and query type: row-level achieves higher accuracy than chunk-level ("Chunk" = 10-line segments); SenSS reduces accuracy despite logical sufficiency.

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

Table 16 summarizes performance differentials between 10-line and 20-line chunk settings. The results indicate that increasing chunk size does not yield consistent improvements. For Qwen3-4B, $R@1$ rises from 0.387 (10-line) to 0.432 (20-line) under *sentence_full*, yet shuffle variants show negligible or negative gains. Stella exhibits a similar pattern, with modest improvements under *sentence_full* but instability across other conditions. Empirical evidence further reinforces the granularity paradox: retrieval accuracy saturates at fine granularity and is weakened by contextual noise in larger chunks.

H Full Retrieval Results Across All Table Formats

Table 17 reports the complete retrieval results on the Minimal Contrastive Set across all table formats (CSV, HTML, Markdown, and Mixed) and configurations. These results extend the summarized comparison in Table 5 and provide a more detailed view of how format variations interact with row-identifier augmentation and shuffled distractors.

Model	**Type**	**10-line R@1**	**20-line R@1**	**Δ**	**10-line R@3**	**20-line R@3**	**Δ**
Qwen3-4B	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

Table 16: 10-line vs 20-line Chunk Retrieval Comparison.

Model	Config	Format	R@1	R@2	R@3	GR@1	DS@1
Qwen3-4B	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
Qwen3-4B	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
Qwen3-4B	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

Table 17: Full retrieval performance across all table formats and configurations.

I Row-Indexing and Column-Shuffling Configurations

For each table, we generate a column-shuffled distractor table by independently permuting each column's values. This preserves the topical semantics of each column while disrupting row-level entity alignment. We then evaluate three configurations:

1. **Ori (Original Table):**

$$\begin{bmatrix} A & B \\ 1 & 2 \\ 3 & 4 \end{bmatrix}$$

2. **RID (Row-Indexed Table):** each cell is prefixed with its row index

$$\begin{bmatrix} A & B \\ r0:1 & r0:2 \\ r1:3 & r1:4 \end{bmatrix}$$

3. **RID+Shuf (Row-Indexed with Column Shuffling):** column-shuffled on top of row-indexing

$$\begin{bmatrix} A & B \\ r0:1 & r1:4 \\ r1:3 & r0:2 \end{bmatrix}$$

This setup allows us to systematically test whether the model embeddings preserve relational information at the row level while controlling for column semantics.

J Sensitivity Analysis: Schema vs. Cell-Level Signals

This experiment provides a preliminary observation that retrieval models appear more sensitive to schema-level modifications than to cell-level changes. However, because column headers consistently appear at the beginning of serialized tables, this effect may partially reflect positional bias rather than genuine schema sensitivity.

J.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 schema and structure; (2) **Schema Perturbation (Header Rename)**, which renames a query-relevant column header to a generic token (e.g., "Price" → "Column_2") without modifying cell values; and (3) **Schema Perturbation (Column Deletion)**, removing the column referenced by the query to test logical dimension dependence. These interventions preserve minimal semantic alignment while altering either factual content or structural anchors.

J.2 Results and Analysis

Table 18 (Mixed format) reports retrieval performance under perturbation. The data reveal pronounced schema dominance: for Qwen3-4B, DS@1 rises from 0.578 under content perturbation to 0.810 when a header is renamed. Higher DS@1 indicates a stronger preference toward the original Target Table over its perturbed counterpart. Sensitivity to schema changes substantially exceeds sensitivity to cell-level modifications, indicating that retrieval similarity appears substantially more sensitive to schema-level modifications than to factual cell-level changes.

Cell-level perturbation demonstrates model blindness to factual changes. Although modified tables no longer satisfy query conditions, they are frequently ranked at top-1 because the unchanged schema preserves high semantic similarity. The logical breakdown is further evidenced by column deletion results (Stella DS@1=0.629), which exceed content-perturbation sensitivity (0.611) but remain dominated by schema effects.

J.3 Key Conclusion

Embedding representations exhibit a schema-over-content bias: tabular understanding is anchored by headers rather than relational cell structures. Consequently, models struggle to discriminate between tables sharing identical schemas but differing in factual values. Fine-grained cell-level discrimination remains beyond the capacity of current embedding-based retrieval mechanisms, explaining persistent errors in Table RAG scenarios that require structural and factual reasoning.

We acknowledge that positional bias may partially contribute to this observation, given that schema tokens appear at leading positions in most serialized formats. Appendix K confirms that front-loaded content does confer a measurable retrieval advantage. Nevertheless, positional effects alone are unlikely to fully account for the observed phenomenon, suggesting that schema dominance may also reflect deeper representational limitations. Importantly, both schema-over-content bias and po-

Configuration	Model	R@1	R@2	R@3	GR@1	DS@1
Content Perturb.	Qwen3-4B	0.371	0.671	0.757	0.643	0.578
	Stella	0.314	0.486	0.529	0.514	0.611
Rename Header	Qwen3-4B	0.486	0.671	0.814	0.600	0.810
	Stella	0.386	0.514	0.557	0.500	0.771
Delete Column	Qwen3-4B	0.443	0.714	0.800	0.600	0.738
	Stella	0.314	0.429	0.486	0.500	0.629

Table 18: Retrieval performance on Minimal Contrastive Sets under small-scale perturbations (Mixed format).

sitional sensitivity are themselves deficiencies of current embedding-based retrieval that merit attention.

K Position Bias and the "Header Preference"

This experiment investigates whether embedding models exhibit **position bias** when encoding tables. Prior work on long-context language models has shown that model performance can vary significantly depending on where relevant information appears in the input sequence, a phenomenon commonly referred to as the **lost-in-the-middle** effect (Liu et al., 2024a). Specifically, we test whether the physical placement of a relevant row affects retrieval similarity, even when the table content remains unchanged.

K.1 Experimental Setup

We construct pairs of tables that are **semantically identical and both target**. 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" Table, although in reality **both tables are valid**.

The metric **DS@1** is used to measure model preference between the two tables. 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 directly reflects positional preference in these pairwise comparisons.

K.2 Results

Setting	Model	R@1	GR@1	DS@1
Top vs Original	Qwen3-4B	0.557	0.686	0.813
Top vs Original	Stella	0.314	0.514	0.611
Top vs Middle	Qwen3-4B	0.571	0.686	0.833
Top vs Middle	Stella	0.314	0.514	0.611
Bottom vs Middle	Qwen3-4B	0.371	0.629	0.591
Bottom vs Middle	Stella	0.300	0.514	0.583
Top vs Bottom	Qwen3-4B	0.557	0.671	0.830
Top vs Bottom	Stella	0.314	0.500	0.629

Table 19: Position bias evaluation under different row placements. DS@1 > 0.5 indicates preference for the first table.

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

For **Qwen3-4B**, 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 reinforces this observation (DS@1 = **0.830**), suggesting that early placement significantly increases retrieval preference.

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**). This suggests that earlier placement within the table body can still influence similarity scores, though less strongly than header-adjacent placement.

For **Stella**, the pattern is more moderate but still consistent. Across all four settings, DS@1 remains between **0.58** and **0.63**, indicating a stable preference toward the first table in each comparison. This implies that the model also exhibits a positional bias favoring earlier occurrences of relevant rows, although the magnitude of the effect is smaller than that observed in Qwen3-4B.

K.3 Discussion

Our analysis indicates that embedding models 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. This suggests that positional signals play a meaningful role in similarity computation. Similar positional effects have been widely observed in transformer-based models, where attention patterns and positional encodings can introduce systematic biases toward certain regions of the input sequence (Yu et al., 2024).

At the same time, the effect remains weaker than structural signals such as column-name alignment observed in other perturbation experiments. In other words, while schema-level cues dominate retrieval similarity, row-level evidence is still influenced by its physical placement within the table.

K.4 Implication

The experiment highlights that embedding models implicitly prioritize information appearing near the beginning of a table. This behavior resembles the way models process textual documents, where early content often receives disproportionate influence in the final representation. Such primacy effects have been reported in long-context evaluation studies, where models tend to prioritize information appearing at the beginning of the context window (Liu et al., 2024a; Veseli et al., 2025).

Consequently, embeddings may rely on positional heuristics rather than explicitly reasoning about whether the table contains the information required to answer a query. This further suggests that current table embedding models do not fully capture the relational structure of tables and instead encode them in a manner closer to structured text.

L 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 some representative dense embedding models on the transposed-table corpus using the same retrieval protocol and evaluation metrics as the main experiments.

Results Table 20 reports the retrieval performance on transposed tables.

Model	R@1	R@3	R@5	GR@1	DS@1
Qwen3-0.6B	0.072	0.211	0.311	0.311	0.231
Qwen3-4B	0.081	0.273	0.392	0.450	0.181
Stella	0.038	0.163	0.258	0.301	0.127

Table 20: Retrieval performance on transposed tables under the Mixed format setting.

Analysis All evaluated models show consistent performance degradation after table transposition.

We attribute this decline to two factors. First, original tables are largely row-oriented, where attributes of the same entity remain locally grouped in serialized sequences. Since most benchmark queries are entity-centric, this structure helps embedding models capture entity-level semantics. After transposition, related attributes become dispersed, weakening local semantic coherence.

Second, transposition scatters schema information across the sequence, weakening local entity coherence and making retrieval substantially more difficult. This observation is consistent with the "Header Preference" phenomenon discussed in Appendix K.

Method	Retrieval Model	R@k					GR@k					DS@1
		@1	@2	@3	@4	@5	@1	@2	@3	@4	@5	
Qwen3-0.6B	ARR-CE (T5)	0.297	0.435	0.507	0.545	0.555	0.737	0.703	0.641	0.596	0.556	0.403
	ARR-CE (T10)	0.359	0.541	0.617	0.665	0.689	0.789	0.761	0.719	0.695	0.664	0.455
	ARR-CE (T∞)	0.507	0.694	0.804	0.866	0.900	0.895	0.861	0.818	0.794	0.771	0.567
	ARR-Judge (T5)	0.383	0.455	0.512	0.545	0.555	0.689	0.639	0.600	0.574	0.556	0.556
	ARR-Judge (T10)	0.469	0.589	0.641	0.675	0.684	0.751	0.694	0.633	0.603	0.589	0.624
Qwen3-4B	ARR-CE (T5)	0.426	0.522	0.589	0.617	0.632	0.789	0.749	0.711	0.682	0.662	0.539
	ARR-CE (T10)	0.498	0.598	0.651	0.694	0.722	0.828	0.801	0.777	0.757	0.735	0.601
	ARR-Judge (T5)	0.445	0.550	0.598	0.612	0.632	0.746	0.725	0.692	0.673	0.662	0.596
	ARR-Judge (T10)	0.517	0.622	0.689	0.703	0.718	0.804	0.766	0.721	0.688	0.676	0.643
Qwen3-8B	ARR-CE (T5)	0.507	0.608	0.636	0.660	0.665	0.770	0.730	0.684	0.653	0.622	0.658
	ARR-CE (T10)	0.574	0.675	0.722	0.761	0.766	0.818	0.787	0.754	0.732	0.709	0.702
	ARR-Judge (T5)	0.498	0.608	0.641	0.660	0.665	0.732	0.694	0.652	0.629	0.622	0.680
	ARR-Judge (T10)	0.536	0.651	0.684	0.722	0.737	0.761	0.720	0.681	0.650	0.637	0.704

Table 21: 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.

Overall, the results suggest that embedding-based table retrievers are sensitive not only to table semantics but also to structural layout and serialization order.

M Full Experimental Results for AAR

In the following, Tk denotes that reranking is applied to the top-k candidates returned by the embedding-based retrieval stage, while T∞ indicates reranking over all retrieved samples. T∞ serves as an empirical performance upper bound for the reranking pipeline; due to its higher computational cost, we report it only for the ARR-CE configuration with Qwen3-0.6B. Table 21 provides the complete experimental results across retrieval backbones.

N 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 further conduct paired bootstrap significance tests comparing reranking-based AAR variants against the embedding-only Qwen3-8B retrieval baseline.

Results Table 22 summarizes the statistical estimates for representative retrieval and reranking models, while Table 23 reports the paired bootstrap comparison between reranking-based AAR variants and the Qwen3-8B baseline.

Analysis The statistical analysis supports the main conclusions of the paper. Although several first-stage retrieval paradigms achieve moderate gains in recall-based metrics, their DS@1 improvements remain statistically indistinguishable from Random retrieval under paired bootstrap testing.

In contrast, both AAR reranking variants produce large and statistically significant improvements, with substantial effect sizes and highly significant p-values. Compared with the embedding-only Qwen3-8B baseline, reranking substantially improves both R@1 and DS@1, demonstrating that the primary limitation lies not in coarse retrieval recall itself, but in the inability of first-stage retrievers to distinguish answerable tables from structurally similar distractors. Query-aware reranking substantially alleviates this issue.

O Summary of Experimental Observations

Table 24 summarizes the major experimental observations regarding answerability-aware table retrieval and the Semantic-Answerability Gap.

P Benchmark Availability and Open-Source Plan

During the anonymous review period, the TCR-Bench benchmark can be accessed at: https://anonymous.4open.science/r/TCR-Bench-05F6.

Upon acceptance, we will open-source the benchmark under the CC BY-SA 4.0 license.

The current anonymous repository already includes the core resources required for reproducing the main experiments in this paper, including:

- table data files,

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
Qwen3-4B + 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	Reranker	0.574	0.702	[0.636, 0.767]	< 0.001
AAR-Judge	Reranker	0.536	0.704	[0.636, 0.774]	< 0.001

Table 22: Bootstrap confidence intervals and significance tests for representative retrieval methods.

Comparison	ΔR@1	95% CI	ΔDS@1	95% CI	p-value
AAR-CE(T10) $-$ Qwen3-8B	+0.394	[0.321, 0.474]	+0.431	[0.330, 0.529]	< 0.001
AAR-Judge(T10) $-$ Qwen3-8B	+0.353	[0.273, 0.435]	+0.434	[0.336, 0.539]	< 0.001

Table 23: Paired bootstrap significance tests comparing reranking-based AAR variants with the embedding-only Qwen3-8B retrieval baseline. Confidence intervals are computed from 1,000 bootstrap samples.

- complete query files,
- files for the Diagnostic subset,
- and the full implementation of the RAG pipeline used in this paper, including dense embedding retrieval, AAR reranking, and downstream TableQA.

After the review process, we will continue to refine the repository with additional resources, including:

- intermediate files used during benchmark construction,
- other benchmark variants used in the paper,
- and implementations based on additional retrieval architectures, such as DTR and ColBERT.

Observations	Implications for Answerability Retrieval	§
Best embedding retriever achieves only 18.2% Top-1 Target Table retrieval despite consistently retrieving tables from the correct sibling group	Dense retrieval captures coarse semantic neighborhoods but fails to reliably identify the uniquely answerable table	4.1
QA performance remains near oracle when the Target Table is ranked first, but degrades rapidly as additional retrieved tables are introduced	Answerability precision at top ranks is more important than broad semantic recall; semantically related distractors interfere with downstream reasoning	4.2
Retrieval performance remains relatively stable across Markdown, CSV, HTML, and sentence-based serializations	The Semantic-Answerability Gap is not caused by surface formatting or serialization artifacts	5.1
Query paraphrasing changes ranking but preserves highly overlapping candidate sets	Retrievers capture general query intent but lack the fine-grained discrimination required for answerability verification	5.2
Reducing a query to a logically sufficient subset substantially lowers retrieval accuracy	Retrieval rewards semantic volume and token coverage rather than minimal answer-bearing evidence	6.1
Row-level retrieval exceeds 0.87 R@1, while table-level Target Table selection remains below 0.26	Local key-value associations are captured, but do not scale to reliable table-level answerability discrimination	6.1
Column shuffling only moderately reduces retrieval accuracy despite breaking row-column semantics	Embeddings weakly encode relational structure and fail to robustly preserve row-column bindings required for answerability	6.2
Interaction-based reranking substantially improves exact Target Table identification	Explicit query-table interaction restores answerability verification that is difficult to capture through single-vector semantic retrieval alone	7

Table 24: Summary of experimental observations on answerability-aware table retrieval and the Semantic-Answerability Gap.