TEmBed-T: A Multi-Dimensional Benchmark for Table-Level Embeddings

Ayeen Poostforoushan*
Independent Researcher

Liane Vogel
Technical University of Darmstadt

Carsten Binnig
Technical University of Darmstadt & DFKI & hessian.AI

ABSTRACT

Tabular data is the dominant structured-data modality, and learning table representations has become a core research direction. Table-level embeddings in particular underpin a wide range of applications, including table retrieval, data lake discovery, and table classification. Despite their importance, there is still limited understanding of how different embedding approaches behave across tasks, making systematic evaluation and analysis essential. In this work, we introduce a systematic evaluation of table-level embeddings that captures several complementary properties required for downstream effectiveness. We realize this evaluation by extending TEmBed, a recently proposed testbed for tabular embeddings, whose table-level coverage is currently limited to a single retrieval task. An empirical study over the TEmBed model pool confirms that no single model excels across all tasks, demonstrating that table-level embedding quality cannot be reduced to retrieval alone.

VLDB Workshop Reference Format:
Ayeen Poostforoushan, Liane Vogel, and Carsten Binnig. TEmBed-T: A Multi-Dimensional Benchmark for Table-Level Embeddings. VLDB 2026 Workshop: Tabular Data Analysis (TaDA).

VLDB Workshop Artifact Availability:
The source code, data, and other artifacts have been made available at https://github.com/IBM/table-representation-evals.

1 INTRODUCTION

The Importance of Table-Level Embeddings. Tabular data is the dominant modality in databases, enterprise systems, and the open web. Learning embeddings—that is, vector representations—of tables has therefore become a central challenge at the intersection of databases and machine learning. Although a growing number of table encoders have been proposed, systematic methods for evaluating and comparing the representations they produce remain underdeveloped [2]. Table-level embeddings are particularly important because they support retrieval, data discovery, schema understanding, and downstream prediction at the scale of entire tables. Yet, despite their relevance to these applications, relatively few models are explicitly designed to produce general-purpose table-level representations. This gap makes it difficult to determine which models are best suited to different table-centric workloads.

*Correspondence goes to ayeen.pf@gmail.com

Proceedings of the VLDB Endowment. ISSN 2150-8097.

Limitations of Existing Benchmarks. Existing benchmarks either measure end-to-end task performance, such as retrieval [8], question answering [8], and joinability detection [15], or probe specific invariances of tabular structures [7]. Consequently, they often conflate representation quality with downstream task success and provide limited insight into the capabilities that drive performance. We therefore argue that meaningful evaluation of table-level embeddings requires disentangling these capabilities and assessing them independently. Different downstream applications rely on different embedding properties, and models that perform well along one dimension may perform poorly along another. Property-level evaluation thus enables more meaningful comparisons and supports more informed model selection. TEmBed [16] takes an important step toward unified evaluation [11] across four embedding granularities: cell, row, column, and table. At the table level, however, its evaluation remains limited and does not isolate the properties that embeddings actually capture.

Our Contribution: TEmBed-T. To address these limitations, we extend TEmBed's table-level evaluation with TEmBed-T. We introduce three additional tasks, each designed to assess a distinct and practically relevant capability. First, we extend retrieval to the query-to-table setting and evaluate it across seven heterogeneous corpora [8]. This task measures cross-domain semantic alignment and complements TEmBed's existing table-to-table setup, thereby covering both query-to-table and table-to-table retrieval scenarios encountered in practice. Second, we introduce table shuffling, a controlled setting that disrupts relational structure while preserving surface content. This task assesses whether embeddings capture structural relationships rather than relying primarily on semantic or lexical cues. Third, we add table type detection [3] on header-stripped tables, isolating the ability to infer table semantics from cell values alone while removing reliance on schema information.

Key Findings. Across five representative approaches, we find that model rankings diverge substantially across these tasks. Importantly, as we show in our initial evaluation, no single embedding approach performs best across all evaluated properties, and strong performance in one setting does not reliably transfer to others. These findings underscore the need for property-driven evaluation and position TEmBed-T as a diagnostic complement to TEmBed's existing benchmark.

2 TEMBED-T: BENCHMARK EXTENSIONS

We identify three relevant properties based on which we select the tasks to extend TEmBed. We first discuss these properties, before we present our extensions in Section 2.1 to 2.3:

Cross-domain Robustness. is important as many applications like NL2SQL or table QA need to be able to work with data from various domains. An embedder that silos search to one domain

is not usable at scale. We probe this property with table retrieval across seven corpora spanning diverse origins, schemas, and sizes (Section 2.1).

Structural Fidelity. defines whether the encoder preserves the structure of a table. Consider a table of CEOs and companies:

CEO	Company
Elon Musk	Tesla
Andy Jassy	Amazon

Anchor

CEO	Company
Elon Musk	Amazon
Andy Jassy	Tesla

Value-shuffled (negative)

Permuting rows and columns changes only surface order while preserving row-wise integrity; a structure-aware embedder should produce a near-identical vector.[1] Shuffling values independently within columns, however, scrambles those associations: the value-shuffled table destroys row integrity while preserving the exact multiset of tokens. A bag-of-words model or an encoder whose attention ignores table structure cannot distinguish these two transformations. We operationalize this property with the table shuffling triplet protocol (Section 2.2).

Header-independent Semantic Preservation. captures whether the embedding retains the table's high-level identity from cell content alone. An embedder that lacks this property can be misled by a surface-level header: a table whose header reads `Event` but whose cells describe a restaurant may be embedded closer to event tables simply because the header token dominates the representation. Stripping headers forces the embedding to recover type from cell values, isolating whether the model abstracts over data rather than memorizing header vocabulary. We evaluate this property with the task of table type detection (Section 2.3).

2.1 Table Retrieval

Retrieving relevant tables e.g. out of a data lake given a query is usually performed by embedding each table and ranking the candidate tables in the corpus by relevance to the query embedding. The task is a reasonable choice to probe cross-domain robustness of table embeddings, as a robust embedding model must hold its ranking quality across heterogeneous corpora out of multiple domains.

Design Dimensions. Our framework lets users inspect embeddings by changing different dimensions of the experiments:

Row count. For fine-grained analysis, the row count included in table embeddings can be set from zero (table headers only), to using full table sizes. For this paper we report results with headers only (zero rows) and with 100 rows per table.

Serialization. We test markdown and CSV serialization to isolate sensitivity to surface formatting independently of content. At headers-only the two formats collapse to near-identical strings, so this axis matters only with rows present. Serialization applies only to text-based encoders.

Dataset. We evaluate over the five corpora packaged by TARGET [8]. FeTaQA [12], TabFact [6], and OTT-QA[5] are Wikipedia-derived corpora that pair tables with free-form QA, fact verification, and multi-hop QA queries respectively, covering general open-domain table content. Spider[2][19], and BIRD [9] are NL2SQL corpora with relational databases covering various domains. Together the corpora span individual tables and relational schemas, and varying corpus sizes, exercising heterogeneity that a single-corpus benchmark cannot.

2.2 Table Shuffling

We test structural fidelity with a triplet protocol. For an anchor table T, we construct a structurally faithful positive T^+ and a value-shuffled negative T^-, and evaluate their embedding distances to the anchor: $d_{\text{pos}} = d_{\cos}(f(T), f(T^+))$, $d_{\text{neg}} = d_{\cos}(f(T), f(T^-))$.

T^+ is obtained by permuting rows and columns, which preserves row-wise associations and thus relational structure. T^- shuffles values independently within each column, breaking row-wise associations while preserving the exact token multiset. Since both variants share identical token distributions, any separation must arise from structural encoding rather than lexical cues.

Design Dimensions. The triplet generation is controlled by four parameters that together determine discrimination difficulty:

Permutation type. Reordering rows and/or columns determines which structural axis is disrupted. Comparing row-against column-only accuracy reveals encoder-specific directional sensitivity.

Positive perturbation magnitude. $d_+ \in [0, 1]$ is the fraction of rows/columns randomly pairwise-swapped to generate T^+ while preserving row-wise associations.

Negative perturbation magnitude. $f_- \in [0, 1]$ is the fraction of columns selected for intra-column shuffling, and $d_- \in [0, 1]$ is the fraction of rows swapped within each selected column.

Table size ablation. Tables are filtered by row and column bounds before triplet generation. Three windows define the size axis: default ($\leq 100 \times 30$), BIG ($\leq 500 \times 80$), and SMALL ($\leq 20 \times 8$), all at fixed $d_+ = 0.75$, $f_- = 0.7$, $d_- = 0.8$. Performance differences isolate whether structural perception depends on table dimensions.

Datasets. We generate triplets from four TARGET datasets [8] containing semantically understandable cell values and from CKAN and ECB [15] with mostly statistical and numerical data.

2.3 Table Type Detection

Table type detection (TTD) evaluates header-independent semantic preservation [3]. Given a header-stripped table, we train a probe classifier on frozen embeddings to predict the table's Schema.org type from cell content alone. Unlike HyTrel [3], which fine-tunes the encoder end-to-end, we fix the encoder and treat classification performance as a measure of retained task-information information, analogous to row-level probing in TEmBed [16]. Removing headers prevents trivial reliance on header tokens, forcing the model to capture schema-level semantics from cell values.

Design Dimensions. Two parameters control the protocol:

Classifier. Three probe classifiers (XGBoost[4], MLP, and KNN) are trained independently on the same frozen embeddings. Consistent rankings across classifiers indicate that recoverability is driven by the embedding rather than the classifier.

Serialization. We evaluate both markdown and CSV serializations (Section 2.1).

Dataset. We use the TTD dataset from HyTrel [3], which is based on the WDC Schema.org corpus [14]. For efficiency and balanced macro-F1 evaluation, we subsample 800 training and 100 test tables for each of the 10 classes (8,000/1,000 total).

[1] The permutation-based positive assumes that row and column order carry no intrinsic semantics. For tables where ordering is meaningful (e.g., temporally sorted tables), this assumption may not hold.

[2] Spider comes separated into train, validation and test splits, we use all three of them for evaluation

3 INITIAL EVALUATION

3.1 Setup

Models. We evaluate on the four approaches included in TEmBed that produce table level embeddings. In addition, we add a term-frequency baseline as the extreme case of structural blindness in the table shuffling task and as a naive baseline across all tasks. The model pool spans three families: text-serialization transformers (MiniLM [17], Granite-R2 [1], GritLM [10]), a structure-aware table encoder (HyTrel [3]), and the term-frequency baseline (Hashing). For the Hashing approach, we use scikit-learn's [13] `HashingVectorizer`, which maps tokens to a fixed-size sparse hashed term-frequency vector. Token counts are accumulated in hash buckets and L2-normalized, producing a lexical-only embedding. We use embedding dimension of 32,768 in this benchmark.

Metrics. *Table Retrieval.* We report MRR and Recall on each corpus. *Table Shuffling.* We quantify the triplet condition $d_{\text{pos}} < d_{\text{neg}}$ (Section 2.2) with two complementary metrics. Triplet accuracy is the probability that the positive is closer to the anchor than the negative; scores below 0.50 indicate systematic inversion. The silhouette score measures the strength of that classification margin.

$$\text{Silhouette} = \mathbb{E}\left[\frac{d_{\text{neg}} - d_{\text{pos}}}{\max(d_{\text{pos}},\ d_{\text{neg}})}\right] \in [-1, 1] \tag{1}$$

Table Type Detection. We report F1-macro across all three classifiers.

Statistical Significance. For Table Retrieval and Table Shuffling, we report bootstrapped 95% confidence intervals obtained by resampling per-instance scores with replacement (per-query for retrieval, per-triplet for shuffling; 10,000 resamples).

3.2 Table Retrieval

We evaluate table retrieval with headers-only and the first 100 rows, as well as markdown and CSV serialization, as described in Section 2.1. Figure 1 shows per-dataset MRR@10 at `row_limit` = 100. Dense transformer encoders lead retrieval across most corpora, reflecting pretraining on natural-language text that transfers to table cell and header vocabulary. Domain-specific reversals emerge within this group: GritLM performs best on Wikipedia-derived corpora (FeTaQA, TabFact, OTT-QA), while Granite-R2 surpasses the other transformers on Spider. This reversal reflects Granite-R2's training emphasis on relational table data [1]. The cross-domain divergence confirms that no single encoder generalizes uniformly, and that evaluating over heterogeneous corpora is necessary to expose training-distribution bias.

Figure 2 compares retrieval performance under schema-only and first-100-rows configurations, averaged over datasets. All approaches improve when row content is provided, confirming that cell values carry signal beyond headers. All transformer encoders gain substantially from row content, while MiniLM shows a markedly smaller gain. This asymmetry indicates that MiniLM's lower-capacity representation already saturates on schema-level signal, leaving little room to leverage additional cell content. The comparison of serialization formats is reported in Table 1 in the Appendix.

3.3 Table Shuffling

We evaluate across 15 variations crossing permutation type, magnitude regime, and table window size; the full grid and ablations

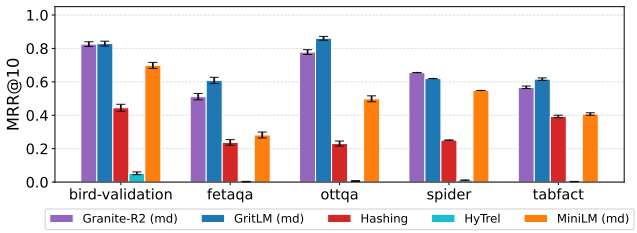

Figure 1: Table Retrieval: MRR@10 per dataset at `row_limit` = 100. Spider: aggregated over its three splits. Dataset heterogeneity drives domain-specific ranking reversals between leading approaches. Error bars show bootstrapped 95% CIs.

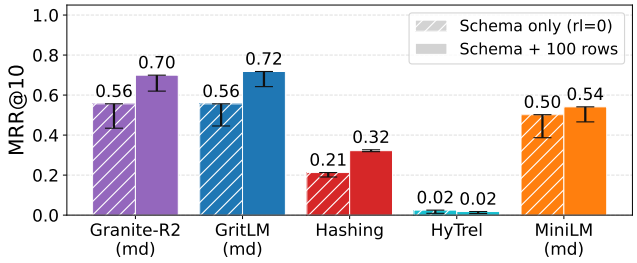

Figure 2: Table Retrieval: MRR@10 at `row_limit` = 0 (headers only) vs. `row_limit` = 100, averaged over datasets. The gap between bars reveals cell content leverage per approach. Error bars show bootstrapped 95% CIs.

are detailed in Appendix A.2. Figure 3 (middle) shows that all models except Hytrel perform poorly on the task. The full per-dataset accuracy breakdown at the canonical v0 variation (Table 5, Appendix) confirms that the structure-aware encoder consistently distinguishes structurally faithful from value-shuffled tables across all datasets. Transformer-based approaches and the bag-of-words baseline both score at or below random chance on lexically rich datasets, but the silhouette score (Figure 6, Appendix A.2) exposes two distinct failure modes underneath: Hashing ties exactly at zero, since its permutation-invariant nature cannot represent the anchor, positive, and negative as different vectors in the first place; transformers instead score negative since their attention tracking surface token order over tabular structure, so the whole-row/column permutation behind the positive disturbs that order more than the localized value-swaps behind the negative.

The ECB dataset provides the clearest diagnostic case. Its statistical content carries minimal lexical discriminability, so any positive-versus-negative separation must arise from structural encoding rather than lexical cues. Transformer-based approaches and the bag-of-words baseline collapse entirely on ECB with 0% accuracy, consistent with the distinct failure modes above. The structure-aware encoder maintains near-perfect accuracy, directly isolating the structural contribution by eliminating the lexical shortcut.

Figure 7 (Appendix A.2) reveals encoder-specific directional asymmetries: row reordering and column reordering yield distinct accuracy profiles per approach, indicating that different encoders represent the two structural axes differently.

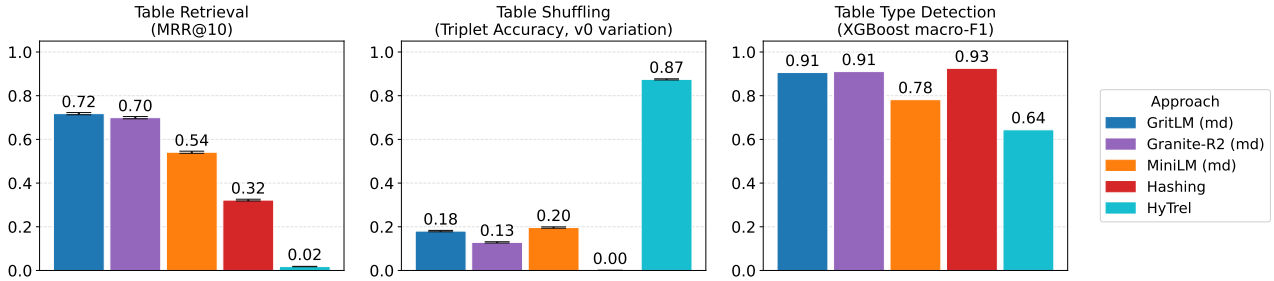

Figure 3: Per-task scores for all five approaches, averaged over each task's datasets. Markdown serialization was used for text-serialization transformers. For each task a different model works best: GritLM for Retrieval, HyTrel for Shuffling, Hashing for TTD. Error bars on the Retrieval and Shuffling panels show bootstrapped 95% CIs.

3.4 Table Type Detection

We train three probe classifiers (XGBoost [4], MLP, KNN) on frozen embeddings and evaluate at markdown and CSV serialization, as described in Section 2.3. Figure 4 reports macro-F1 across classifiers. Classifier choice affects absolute scores but not approach rankings, which are consistent across probe types. The term-frequency baseline (Hashing) leads the tree-based classifier, confirming that table type is recoverable from surface token patterns alone. Dense transformer approaches achieve stronger performance under neural probes, indicating their representations compress type-relevant semantic structure beyond what token frequencies encode. The structure-aware encoder underperforms across all classifiers, consistent with pretraining objectives that optimize structural relationships rather than schema-type signal. Complete per-classifier results are in Table 6 (Appendix).

3.5 Cross-Task Overview

Table 7 (Appendix A.4) reports per-task ranks and an average rank across the three tasks. Rankings diverge sharply across tasks, with no single encoder leading on all three axes. This directly motivates the need for models with broad, multi-dimensional table-level capabilities that do not yet exist. The comparison of CSV and markdown serialization is shown in Figure 8 (Appendix A.4). The results vary for each approach and task combination, confirming that there is no universally superior format between CSV and markdown. In addition to measuring performance, we also measured execution time, as visualized in Figure 9 (Appendix A.4). A general trend holds where higher processing cost yields better pooled quality, with Hashing as the exception since its overall quality is poor despite high embedding cost, consistent with its nature as a classical lexical-only baseline. Granite-R2 achieves strong pooled quality despite being substantially smaller than GritLM, owing to its training on a large corpus of relational table data [1].

4 RELATED WORK

TARGET [8] benchmarks retrieval for generative tasks such as question answering, fact verification, and text-to-SQL. LakeBench [15] evaluates data lake discovery across multiple tasks. Observatory [7] takes a diagnostic approach, defining eight primitive properties grounded in relational invariants and data distribution considerations. It measures how embeddings respond to controlled perturbations at the row and column level, quantifying embedding drift

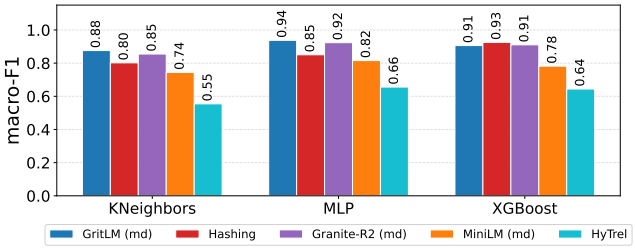

Figure 4: Table Type Detection: Macro-F1 per approach and classifier on header-stripped WDC Schema.org tables. Markdown-serialization was used for the text-serialization transformers.

across the shuffles. Younes et al. [18] similarly focus on robustness to benign perturbations, applying row and column permutations and measuring the similarity between the original and perturbed embeddings. Both frameworks confirm that encoders vary in their sensitivity to reordering. In a permutation probe, however, the anchor and the permuted table share the exact same multiset of tokens, so an encoder operating as a bag-of-words model passes by matching token overlap without encoding structural information. Our value-shuffled negative fills this gap by preserving each column's token multiset while scrambling cross-column associations, so any separation must arise from structural encoding alone.

5 CONCLUSION

TEmBed's [16] table-level evaluation axis is limited to a single retrieval task, leaving the intrinsic properties a table encoder must possess entirely untested. We extend it into a three-task diagnostic suite grounded in both the fundamental characteristics a table embedder needs to have and the application-level properties it is expected to deliver in practice. A model that performs well across all three axes can be considered a genuinely capable table-level encoder. We evaluated the suite through a range of experiments and ablation sweeps that demonstrate the benchmark's ability to probe encoders along independent diagnostic axes. Per-task rankings diverge substantially across all approaches, and no current model performs well on all three axes simultaneously. This confirms that table-level embedding quality is multi-dimensional, and building an encoder that covers the full capability profile remains an open problem.

ACKNOWLEDGMENTS

This work was funded by the BMBF and the state of Hesse as part of the NHR program, by the LOEWE Spitzenprofessur (III 5-519/05.00.003-(0005)), by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), and under Germany's Excellence Strategy (EXC-3057/1 "Reasonable Artificial Intelligence", Project No. 533677015). We also thank DFKI and hessian.AI.

REFERENCES

[1] Parul Awasthy, Aashka Trivedi, Yulong Li, Meet Doshi, Riyaz A. Bhat, Vignesh P, Vishwajeet Kumar, Yushu Yang, Bhavani Iyer, Abraham Daniels, Rudra Murthy, Ken Barker, Martin Franz, Madison Lee, Todd Ward, Salim Roukos, David Cox, Luis A. Lastras, Jaydeep Sen, and Radu Florian. 2025. Granite Embedding R2 Models. *CoRR* abs/2508.21085 (2025). https://doi.org/10.48550/ARXIV.2508.21085 arXiv:2508.21085

[2] Gilbert Badaro, Mohammed Saeed, and Paolo Papotti. 2023. Transformers for Tabular Data Representation: A Survey of Models and Applications. *Trans. Assoc. Comput. Linguistics* 11 (2023), 227–249. https://doi.org/10.1162/TACL_A_00544

[3] Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. 2023. HyTrel: Hypergraph-enhanced Tabular Data Representation Learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.). http://papers.nips.cc/paper_files/paper/2023/hash/66178beae8f12fcd48699de95acc1152-Abstract-Conference.html

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 785–794. https://doi.org/10.1145/2939672.2939785

[5] Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021. Open Question Answering over Tables and Text. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=MmCRswl1UYl

[6] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=rkeJRhNYDH

[7] Tianji Cong, Madelon Hulsebos, Zhenjie Sun, Paul Groth, and H. V. Jagadish. 2023. Observatory: Characterizing Embeddings of Relational Tables. *Proc. VLDB Endow.* 17, 4 (2023), 849–862. https://doi.org/10.14778/3636218.3636237

[8] Xingyu Ji, Parker Glenn, Aditya G. Parameswaran, and Madelon Hulsebos. 2024. TARGET: Benchmarking Table Retrieval for Generative Tasks. *NeurIPS 2024 Third Table Representation Learning Workshop* (2024). https://openreview.net/pdf?id=gGGvnjFUfL

[9] Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 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 (2024).

[10] Niklas Muennighoff, SU Hongjin, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative representational instruction tuning. In *The Thirteenth International Conference on Learning Representations*.

[11] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*. 2014–2037.

[12] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryscinski, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir R. Radev. 2022. FeTaQA: Free-form Table Question Answering. *Trans. Assoc. Comput. Linguistics* 10 (2022), 35–49. https://doi.org/10.1162/TACL_A_00446

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[14] Ralph Peeters, Alexander Brinkmann, and Christian Bizer. 2024. The Web Data Commons Schema.org Table Corpora. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, Tat-Seng Chua, Chong-Wah Ngo, Roy Ka-Wei Lee, Ravi Kumar, and Hady W. Lauw (Eds.). ACM, 1079–1082. https://doi.org/10.1145/3589335.3651441

[15] Kavitha Srinivas, Julian Dolby, Ibrahim Abdelaziz, Oktie Hassanzadeh, Harsha Kokel, Aamod Khatiwada, Tejaswini Pedapati, Subhajit Chaudhury, and Horst Samulowitz. 2023. LakeBench: Benchmarks for Data Discovery over Data Lakes. *CoRR* abs/2307.04217 (2023). https://doi.org/10.48550/ARXIV.2307.04217 arXiv:2307.04217

[16] Liane Vogel, Kavitha Srinivas, Niharika S. D'Souza, Sola Shirai, Oktie Hassanzadeh, and Horst Samulowitz. 2026. Towards Universal Tabular Embeddings: A Benchmark Across Data Tasks. *CoRR* abs/2604.21696 (2026). https://doi.org/10.48550/ARXIV.2604.21696 arXiv:2604.21696

[17] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[18] Ali Younes, Saeed Ghoorchian, Maximilian Schambach, and Johannes Höhne. 2026. Unified Evaluation of Table Embedding Methods Across Multiple Benchmark Scenarios. *3rd DATA-FM Workshop at ICLR* (2026).

[19] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 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, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 3911–3921. https://doi.org/10.18653/V1/D18-1425

A APPENDIX

This appendix provides full experimental detail supporting the results summarized in Section 3: per-dataset and per-serialization breakdowns for Table Retrieval (A.1), the complete variation grid and ablations for Table Shuffling (A.2), per-classifier results for Table Type Detection (A.3), and cross-task comparisons of serialization, ranking, and cost-quality trade-offs (A.4).

A.1 Table Retrieval

While Figure 1 reports the results using markdown serialization for Granite-R2 [1], GritLM [10], and MiniLM [17], we report the full results of the format ablations per dataset in Table 1. Results vary by dataset, but markdown tends to achieve higher results overall, especially for GritLM, whereas MiniLM scores slightly higher on average with CSV serialization.

A.2 Table Shuffling: Variation Grid & Ablations

The Table Shuffling evaluation is structured as a grid of 15 variations across three axes: permutation type (both, row reorder, column reorder), perturbation magnitude, and table window size, as described in Section 2.2.

Variation grid. Table 2 lays out the main magnitude grid (v0–v8). For high positive perturbation we use $d_+ = 0.75$; for low positive, $d_+ = 0.25$. For high negative perturbation we use $f_- = 0.7$, $d_- = 0.8$; for low negative, $f_- = 0.2$, $d_- = 0.2$. The canonical setting (v0) fixes $d_+ = 0.75$, $f_- = 0.7$, $d_- = 0.8$. hi-pos/lo-neg is the most challenging regime: the positive undergoes maximal surface disruption while the negative carries only minimal integrity-breaking change, forcing the encoder to rank a heavily permuted but structurally faithful table above a barely corrupted one. lo-pos/hi-neg is the inverse and the easiest regime. We select the balanced hi-pos/hi-neg setting with both permutation types as the canonical variation v0, reported in the main paper.

Approach	bird-validation	fetaqa	ottqa	spider-train	tabfact	Mean
Granite-R2 (csv)	0.8231 [0.809, 0.837]	0.5015 [0.482, 0.521]	0.7675 [0.753, 0.782]	**0.6090** [0.599, 0.619]	0.5500 [0.542, 0.558]	0.6502
Granite-R2 (md)	0.8257 [0.811, 0.840]	0.5112 [0.492, 0.531]	0.7777 [0.763, 0.792]	0.5932 [0.583, 0.603]	0.5673 [0.560, 0.575]	0.6550
GritLM (csv)	0.7810 [0.765, 0.797]	**0.6196** [0.601, 0.638]	0.8569 [0.845, 0.868]	0.5219 [0.512, 0.532]	**0.6183** [0.611, 0.626]	0.6795
GritLM (md)	**0.8290** [0.814, 0.843]	0.6086 [0.590, 0.627]	**0.8605** [0.849, 0.872]	0.5527 [0.543, 0.562]	0.6161 [0.609, 0.624]	**0.6934**
MiniLM (csv)	0.6913 [0.673, 0.709]	0.2980 [0.280, 0.316]	0.5086 [0.491, 0.526]	0.4942 [0.484, 0.504]	0.4291 [0.422, 0.437]	0.4842
MiniLM (md)	0.6986 [0.680, 0.716]	0.2819 [0.264, 0.299]	0.4983 [0.481, 0.516]	0.4825 [0.473, 0.492]	0.4072 [0.400, 0.415]	0.4737

Table 1: Table Retrieval MRR@10: markdown vs CSV serialization (rl=100, transformers only; spider validation and test splits showed near-identical trends and are omitted for space). Bracketed values show bootstrapped 95% CIs.

	hi-pos hi-neg	**lo-pos hi-neg**	**hi-pos lo-neg**	**BIG (hi-pos/hi-neg)**	**SMALL (hi-pos/hi-neg)**
both	v0	v1	v2	v9	v12
row_reorder	v3	v4	v5	v10	v13
col_reorder	v6	v7	v8	v11	v14

Table 2: Shuffling variation grid: Permutation × Magnitude (v0–v8) and Permutation × Size (v9–v14, fixed hi-pos/hi-neg). v0 is the canonical setting.

Approach	hi-pos hi-neg	lo-pos hi-neg	hi-pos lo-neg	Mean
Granite-R2 (md)	0.1286	0.5705	0.0091	0.2361
GritLM (md)	0.1802	0.8426	0.0036	0.3422
Hashing	0.0000	0.0000	0.0000	0.0000
HyTrel	**0.8747**	**0.9712**	**0.8277**	**0.8912**
MiniLM (md)	0.1965	0.6376	0.0036	0.2792

Table 3: Table Shuffling: Magnitude grid. Accuracy averaged over all 6 datasets (perturbation=both, default window). Rows = positive-magnitude / negative-magnitude. Best in bold.

Table 3 reports accuracy across magnitude regimes and permutation types. Discrimination difficulty increases monotonically with positive perturbation magnitude and decreases with negative perturbation magnitude, because the two axes operate on the same underlying margin. Larger positive perturbation drives the positive sample further from the anchor in surface-text space, compressing the gap an encoder must bridge to score it above the negative. Conversely, smaller negative perturbation keeps the negative textually close to the anchor, shrinking the margin from the other side. Both effects independently tighten the structural signal an encoder must exploit to succeed.

Size Ablation. The same grid (Table 2) crosses permutation type with two extreme table window sizes (BIG and SMALL), fixing hi-pos/hi-neg, where variations v9–v11 use large windows and v12–v14 use small windows.

We test whether structural sensitivity depends on table dimensions by comparing accuracy across BIG and SMALL windows for each permutation type. Table 4 shows this robustness holds broadly, with approaches ranking consistent across window sizes. One asymmetry stands out: HyTrel's accuracy on column reordering trails row reordering, suggesting that the structure-aware encoder's representations are not completely column-position independent.

Approach	Both		Row reorder		Col reorder		
	BIG	SMALL	BIG	SMALL	BIG	SMALL	Mean
Granite-R2 (md)	0.1856	0.2114	0.4473	0.4507	0.4781	0.4973	0.3784
GritLM (md)	0.2375	0.2619	0.5214	0.5646	0.6040	0.5554	0.4575
Hashing	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
HyTrel	**0.9654**	**0.9850**	**0.9995**	**0.9995**	**0.9694**	**0.9824**	**0.9835**
MiniLM (md)	0.2434	0.2345	0.3906	0.4270	0.6821	0.6403	0.4363

Table 4: Table Shuffling: Perturbation-type breakdown and size ablation (hi-pos/hi-neg). Averaged over fetaqa, tabfact, ottqa, spider-train. CKAN and ECB excluded (no tables pass SMALL window filter).

Per-dataset Accuracy. The v0 setting (hi-pos/hi-neg magnitude, both row+column permutation, default window; Table 2) gives one aggregate accuracy per approach, but this could mask dataset-specific failure or success. We therefore report the full per-dataset breakdown at v0 in Table 5 to check whether trends hold uniformly or are driven by a subset of corpora. HyTrel's advantage holds everywhere, most strikingly on ECB and TabFact where it reaches near-perfect accuracy while every transformer and Hashing score exactly zero.

Full Grid Overview. Having confirmed per-dataset consistency at v0, we check whether these trends extend across all 15 variations rather than just the canonical setting; the results are visualized in Figure 5. HyTrel [3] maintains uniformly high accuracy throughout the grid, while transformer columns exhibit variation-dependent sensitivity with performance dropping in higher perturbation regimes, and the hashing approach predictably scores zero as the extreme failure mode of this task.

Silhouette Score. Accuracy alone doesn't reveal how confidently an approach separates positives from negatives, so we compute the silhouette score (Eq. 1) per approach, averaged across datasets at v0. Figure 6 shows Hashing sits at exactly zero, reflecting its permutation invariance rather than weak discrimination — it cannot represent anchor, positive, and negative as distinct vectors at all. The transformers score negative, indicating systematic inversion rather than merely weak signal, while HyTrel is the only approach with a clearly positive margin.

Row vs. Column Sensitivity. Aggregate accuracy conflates row and column reordering, which could affect encoders differently. Figure 7 compares per-approach accuracy on row-only (v3) vs. column-only (v6) variations: HyTrel sits near the top-right corner but slightly below the diagonal, indicating mildly weaker robustness to column reordering, while transformers scatter off-diagonal, each showing its own directional asymmetry.

Approach	ckan_subset	ecb	fetaqa	ottqa	spider-train	tabfact	Mean
Granite-R2 (csv)	0.0549 [0.049, 0.061]	0.0000 [0.000, 0.000]	0.2200 [0.184, 0.256]	0.2285 [0.192, 0.267]	0.1344 [0.104, 0.165]	0.1342 [0.118, 0.151]	0.1287
Granite-R2 (md)	0.0585 [0.052, 0.065]	0.0000 [0.000, 0.000]	0.2340 [0.196, 0.272]	0.1984 [0.164, 0.234]	0.1222 [0.094, 0.153]	0.1584 [0.141, 0.176]	0.1286
GritLM (csv)	0.1719 [0.161, 0.182]	0.0000 [0.000, 0.000]	0.3160 [0.276, 0.356]	0.2545 [0.216, 0.293]	0.1141 [0.088, 0.143]	0.1348 [0.118, 0.151]	0.1652
GritLM (md)	0.1585 [0.149, 0.169]	0.0000 [0.000, 0.000]	0.3680 [0.326, 0.412]	0.2766 [0.238, 0.317]	0.1181 [0.090, 0.147]	0.1602 [0.143, 0.178]	0.1802
Hashing	0.0000 [0.000, 0.000]	0.0000 [0.000, 0.000]	0.0000 [0.000, 0.000]	0.0000 [0.000, 0.000]	0.0000 [0.000, 0.000]	0.0000 [0.000, 0.000]	0.0000
HyTrel	**0.3824** [0.369, 0.396]	**1.0000** [1.000, 1.000]	**0.8740** [0.844, 0.902]	**0.9940** [0.986, 1.000]	**0.9980** [0.994, 1.000]	**1.0000** [1.000, 1.000]	**0.8747**
MiniLM (csv)	0.1862 [0.176, 0.197]	0.0000 [0.000, 0.000]	0.2380 [0.202, 0.276]	0.2104 [0.174, 0.248]	0.1446 [0.114, 0.177]	0.1507 [0.134, 0.167]	0.1550
MiniLM (md)	0.2160 [0.205, 0.227]	0.0000 [0.000, 0.000]	0.2820 [0.242, 0.322]	0.3086 [0.269, 0.349]	0.1629 [0.132, 0.196]	0.2092 [0.190, 0.229]	0.1965

Table 5: Table Shuffling: Accuracy per dataset (v0: hi-pos/hi-neg, both row+col perturbation, default window). Best per column in bold, second-best underlined. Bracketed values show bootstrapped 95% CIs.

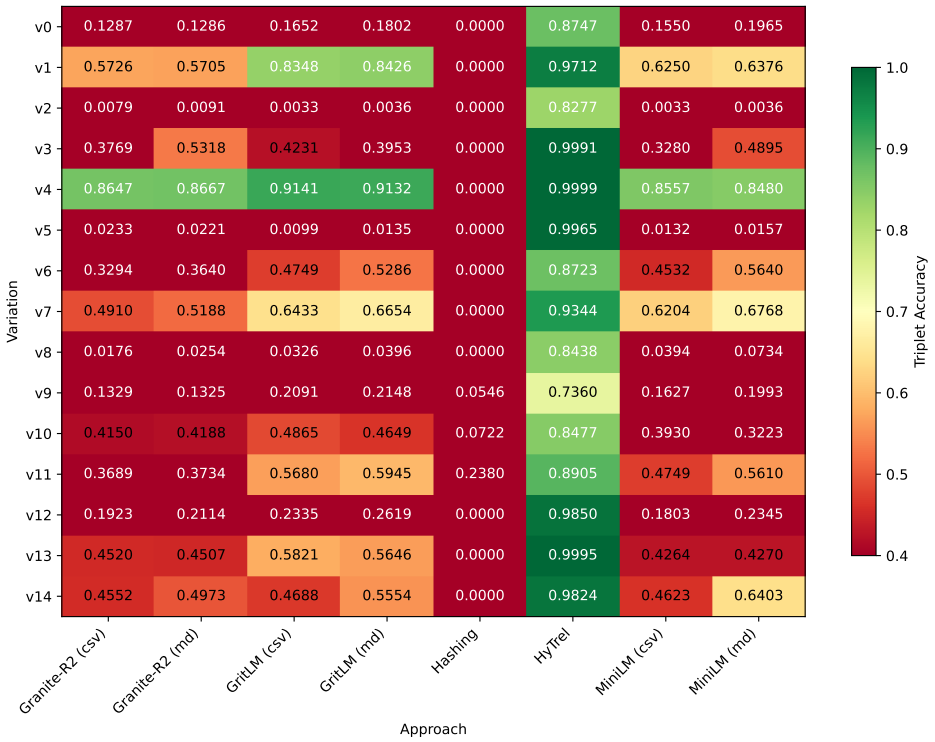

Figure 5: Table Shuffling: Accuracy heatmap across all variations v0–v14 (defined in Table 2) and approaches, averaged over datasets. The structure-aware encoder maintains uniformly high accuracy; transformer columns show variation-dependent structural sensitivity.

A.3 Table Type Detection

Per-Classifier Results. Section 3.4 reports macro-F1 aggregated across classifiers, which could mask whether rankings depend on the probe used. Table 6 breaks down accuracy and macro-F1 by classifier. The dense transformer approaches (GritLM, Granite-R2, MiniLM) rank consistently relative to one another across XGBoost, MLP, and KNeighbors, but Hashing shows a pronounced classifier dependence: it leads under XGBoost yet trails all transformer encoders under MLP and KNeighbors, consistent with its bag-of-words representation being well-suited to tree-based decision boundaries but less effective for neural or distance-based classifiers. HyTrel ranks last across all three probes.

Approach	XGBoost accuracy	XGBoost macro-F1	MLP accuracy	MLP macro-F1	KNeighbors accuracy	KNeighbors macro-F1
Granite-R2 (csv)	0.8980	0.8959	0.9090	0.9078	0.8560	0.8507
Granite-R2 (md)	0.9110	0.9103	0.9240	0.9233	0.8600	0.8549
GritLM (csv)	0.9090	0.9083	0.9330	0.9324	0.8690	0.8627
GritLM (md)	0.9060	0.9059	**0.9380**	**0.9375**	**0.8820**	**0.8758**
Hashing	**0.9358**	**0.9252**	0.8684	0.8503	0.8376	0.8017
HyTrel	0.6490	0.6439	0.6610	0.6557	0.5630	0.5545
MiniLM (csv)	0.7910	0.7888	0.8100	0.8070	0.7680	0.7519
MiniLM (md)	0.7850	0.7818	0.8180	0.8154	0.7600	0.7443

Table 6: Table Type Detection: Accuracy and macro-F1 per classifier on WDC Schema.org (header-stripped, frozen embeddings). Best per column in bold, second-best underlined.

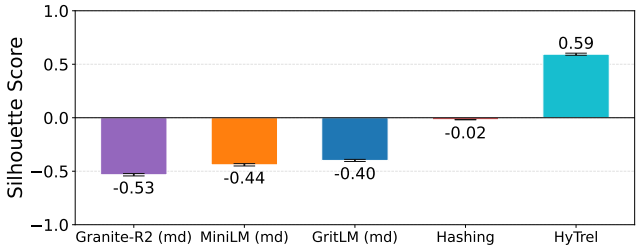

Figure 6: Table Shuffling: Silhouette score per approach, averaged across all datasets at the v0 variation (pos_type=both, hi-pos/hi-neg, markdown serialization). Error bars show bootstrapped 95% CIs, pooled across datasets. Hashing's near-zero score reflects its permutation invariance: the embedding cannot distinguish the anchor, positive, and negative as different vectors, confirming the evaluation protocol does not reward bag-of-words representations.

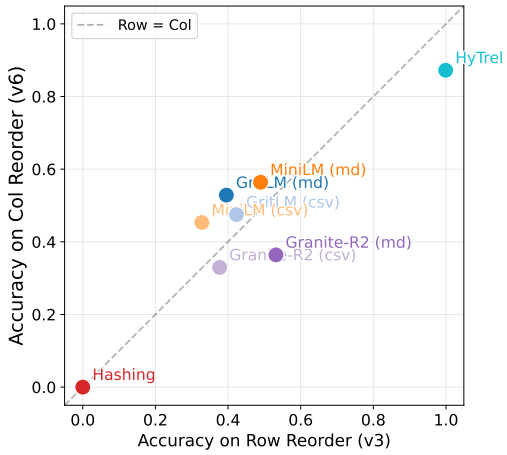

Figure 7: Table Shuffling: Row vs. column perturbation accuracy (hi-pos/hi-neg, default table window, averaged over datasets). Points above the diagonal are column-sensitive; points below are row-sensitive. HyTrel sits near the top-right corner, indicating slightly weaker robustness to column than row reordering. Transformer points spread off-diagonal, revealing encoder-specific asymmetries.

A.4 Cross-Task Results

Serialization Effects Across Tasks. Section 3.2 analyzed serialization effects for table retrieval; here we extend the comparison across all three tasks. Figure 8 confirms that the sign and magnitude of the delta between CSV and markdown differ across the approach × task grid, with no format uniformly preferable, though shuffling shows the largest swings, especially for MiniLM [17].

Overall Ranking. Section 3.5 argues no single approach dominates all axes; Table 7 makes this concrete by ranking each approach per task and averaging. GritLM has the best average rank overall, but no approach ranks first on more than one task, reinforcing that per-task and aggregate rankings disagree.

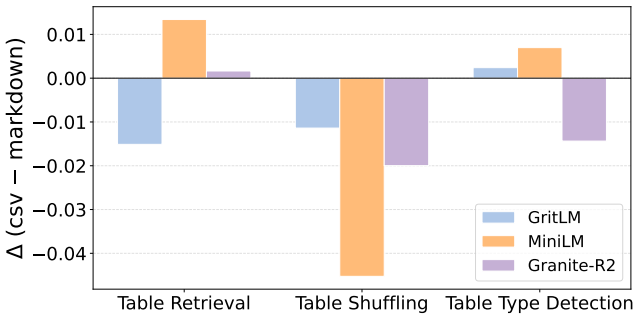

Figure 8: Serialization deltas (Δ = CSV − markdown; positive means CSV scores higher) across tasks for transformer approaches. Sign and magnitude vary across the approach × task grid.

Approach	Table Retrieval (MRR@10)	Table Shuffling (Accuracy)	Table Type Detection (XGB F1)	Overall
GritLM (md)	**1.00**	3.00	3.00	**2.33**
Granite-R2 (md)	2.00	4.00	2.00	2.67
MiniLM (md)	3.00	2.00	4.00	3.00
Hashing	4.00	5.00	**1.00**	3.33
HyTrel	5.00	**1.00**	5.00	3.67

Table 7: Per-task and overall rank of each approach (1=best, 5=worst). No approach ranks first on more than one task, underscoring that no single embedding has all properties.

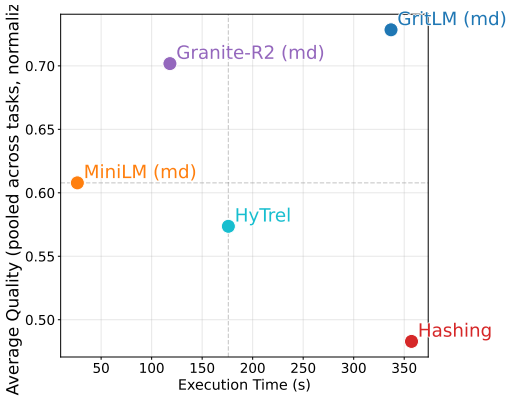

Figure 9: Cost–quality trade-off: Quality is the mean of each approach's three per-task scores, cost is the total embedding time in seconds.

Cost-Quality Trade-off. Beyond raw quality, practical model selection depends on embedding cost. Figure 9 relates pooled quality to embedding time: cost and quality are not consistently related. GritLM's higher cost does pay off in higher quality, but Hashing incurs high cost for the lowest quality (due to its 32,768-dimensional sparse vectors), and Granite-R2 offers the best cost–quality trade-off in the benchmark, indicating quality is driven more by architecture and training than by compute spent at inference.