UNIFIED EVALUATION OF TABLE EMBEDDING METHODS ACROSS MULTIPLE BENCHMARK SCENARIOS

Ali Younes[1,2] * **Saeed Ghoorchian**[1] **Maximilian Schambach**[1] **Johannes Höhne**[1]
[1]SAP SE, Berlin, Germany
[2]TU Darmstadt, Darmstadt, Germany
Correspondence to: ali.younes@tu-darmstadt.de

ABSTRACT

We introduce a unified evaluation framework for table-level embeddings, that is, methods that encode an entire table into a single vector. The proposed framework targets operations such as table indexing, clustering, retrieval, as well as data curation primitives for training tabular foundation models with various objectives, including overlap estimation, approximate deduplication, filtering, and sampling. While feature representations in vision and language have enabled scalable retrieval and transfer, tabular representation learning is typically evaluated at finer granularities (rows/cells) or via downstream prediction, leaving table-level embedding quality and robustness under-specified. We benchmark diverse embedding families, including schema and statistical fingerprints, text-serialization encoders, specialized table encoders, and pooled representations from tabular foundation models, on both controlled synthetic table families with known generative factors and real open-source data. Following a set of desiderata, we evaluate consistency under partial views, discriminability across label granularities, robustness to benign perturbations, and efficiency, without downstream fine-tuning. Our results show that simple hashing and lightweight serialization methods are highly competitive and often outperform pooled representations from foundation models. This exposes a representation-prediction tension: strong predictive models do not necessarily yield stable, discriminative table-level geometry after pooling, motivating objectives that explicitly optimize robust table-level embeddings.

1 INTRODUCTION

The evolution of Foundation Models (FMs) has fundamentally transformed Natural Language Processing (NLP) and Computer Vision (CV). Models such as BERT (Devlin et al., 2019) and CLIP (Radford et al., 2021) shifted the paradigm from training task-specific models to leveraging general-purpose, task-agnostic representations. These models encode profound knowledge into compact, latent feature representations that enable seamless adaptation to downstream tasks, such as zero-shot classification or semantic retrieval (Caron et al., 2021). However, while text and image modalities have successfully decoupled representation learning from specific applications, tabular deep learning remains largely entangled with conventional direct target prediction.

Current Tabular Foundation Models (TFMs) predominantly adopt an In-Context Learning (ICL) paradigm based on Prior-Data Fitted Networks (PFNs) (Müller et al., 2022; Hollmann et al., 2023; 2025). While effective for specific classification and regression tasks, these models are trained to optimize row-level predictive performance within a limited context window, rather than learning a holistic, task-agnostic representation of the data structure. This row or even cell-centric focus overlooks the native compositional nature of tables, where meaningful semantics exist at multiple granularities, such as cell, column, row, and table. Tasks involving multi-table understanding, Retrieval-Augmented Generation (RAG) on databases, or schema matching require robust *table-level* embeddings. Beyond retrieval, table-level embeddings support corpus-scale data operations, such as overlap estimation, approximate deduplication, filtering, and sampling, that increasingly shape how tabular foundation models are trained and curated. While recent work suggests that compressing large

*Work done during an internship at SAP

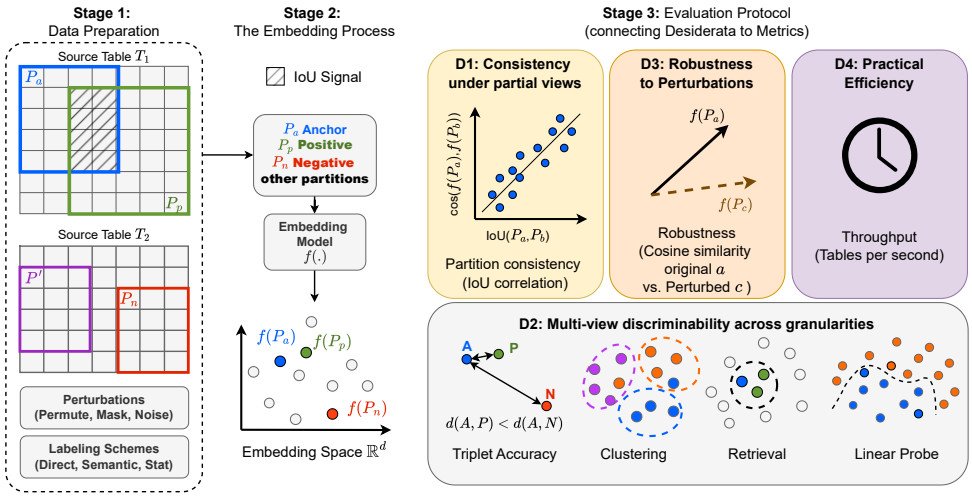

Figure 1: The proposed evaluation pipeline linking table embedding desiderata to concrete evaluation: In stage 1, we sample partitions from source tables and assign them apply predefined perturbation and labeling schemes. Stage 2 involves an embedding model $f(\cdot)$ that maps partitions into a vector space. Stage 3 defines four desiderata (D1-D4) and their corresponding evaluation metrics.

datasets into a smaller learned context (i.e., conditioning a table-level representation) can significantly enhance classification performance (Feuer et al., 2024), the majority of research remains focused on fine-grained cell or row representations (Hoppe et al., 2025), leaving the quality and utility of whole-table representations largely underexplored.

In this work, we propose a unified evaluation framework designed to rigorously assess table-level embeddings based on a set of principled desiderata (illustrated in Figure 1). To facilitate representative evaluation, we frame table-level proxy tasks by sampling partitions from tables and labeling them according to diverse schemes. We benchmark a comprehensive collection of embedding models, ranging from lightweight hashing and statistical baselines to pooled representations from state-of-the-art TFMs, including TabPFN (Hollmann et al., 2023) and ConTextTab (Spinaci et al., 2025). Additionally, we evaluate distinct architectural approaches, such as the hypergraph-enhanced representations of HyTREL (Chen et al., 2023) and vectorized table encodings via Skrub (skrub data, 2023). Our evaluation protocol is not limited to simple downstream accuracy. Instead, over framework extends towards measuring the semantic consistency between Intersection over Union (IoU) and embedding similarity, operating on the assumption that overlapping table partitions should remain close in the latent space (Pugnaloni et al., 2025). We further assess the discriminability of the embeddings in local neighborhoods via triplet comparisons, global clustering, retrieval performance, and linear probe accuracy as a proxy for semantic information decodability. Moreover, we evaluate robustness against permutations, masking, and additive noise, alongside efficiency.

To ensure our analysis is both controlled and empirically valid, we utilize a spectrum of data sources. We first construct a novel collection of synthetic datasets with known generative factors, incorporating physics formulas, geometric shapes, temporal reasoning, and Structural Causal Model (SCM) priors (Qu et al., 2025). We then evaluate our approach on these synthetic datasets as well as on established real-world benchmarks, including CARTE (Kim et al., 2024), OpenML-CC18 (Bischl et al., 2021), and OpenML-CTR23 (Fischer et al., 2023). Our results highlight noticeable differences in the evaluations of different methods, with some findings defying the intuitive expectation of a correlation between a model's predictive power and the quality of general-purpose representations, suggesting the need for a new class of tabular foundation models capable of providing robust general-purpose table-level embeddings without compromising downstream utility.

Our contributions are summarized as follows:

- We introduce a unified benchmarking framework for evaluating general-purpose table-level embeddings, focusing on semantic consistency, clustering quality, retrieval, and robustness.
- We conduct a comprehensive analysis comparing simple baselines against state-of-the-art TFMs across controlled synthetic environments and diverse real-world datasets.
- We identify a significant gap in current TFM capabilities, namely, a powerful ICL TFM with highly predictive performance does not necessarily produce high-quality, task-agnostic table representations.

2 DESIDERATA FOR TABLE-LEVEL EMBEDDINGS

We treat table-level embeddings as reusable interfaces for corpus-scale operations such as retrieval, ranking, and lightweight supervision, without requiring downstream fine-tuning. Due to the lack of existing comparable benchmarks and the scarcity of labeled ground truth data, we design a set of proxy tasks that address this gap and provide a sound evaluation pipeline. The desiderata below motivate the proxy task definitions and metrics introduced in Section 4. Figure 1 depicts the connection between the desiderata and the evaluation metrics.

D1: Consistency under partial views. In many real-world applications, a table is rarely observed in full; it may be accessed via cached extracts, subsampled partitions, or truncated context windows. A table embedding should therefore be stable across partial views of the same table, with similarity increasing monotonically with view overlap.This property directly supports overlap estimation and approximate deduplication in large corpora, where near-duplicate tables often share only partial content, and enables view-consistent filtering and sampling when curating training data for tabular foundation models.

D2: Discriminability across label granularities. A table embedding should support grouping and separation under different label granularities. This is important for training-time curation, where different operations require different equivalence relations, with identity-sensitive signals help deduplication, semantic groupings support stratified sampling across domains and tasks, and coarse statistical groupings enable type-aware filtering and balancing. The evaluation should characterize both local neighborhood structure (triplet comparisons) and global grouping behavior (clustering), alongside operational proxies such as retrieval performance and linear probe decodability of labels.

D3: Robustness to benign perturbations. The table representation should be invariant under row and column permutations, as well as should remain stable under mild corruption such as masking and small additive noise . Robustness is essential not only for reliable indexing and nearest-neighbor retrieval, but also for deduplication and corpus curation over automatically extracted tables.

D4: Practical efficiency. Target applications often involve large corpora; therefore, embedding extraction must be computationally efficient. We measure the throughput (tables per second). An efficient table embedding method balances high throughput with robustness under the aforementioned desiderata, capturing the quality-cost trade-off.

3 TABLE-LEVEL EMBEDDING

A table-level embedding model is a function f that maps a table T to a fixed-dimensional vector $f(T) \in \mathbb{R}^d$. To compare two tables under a fixed embedding method, we measure similarity between their corresponding embedding vectors. In this paper, we use cosine similarity as follows $\mathrm{sim}(T_a, T_b) = \cos(f(T_a), f(T_b))$. Although embedding models conceptually operate on full tables, our evaluation applies $f(\cdot)$ to *table partitions* generated from the same source table, which are themselves valid subtables. This yields a unified interface for benchmarking diverse embedding families under controlled partial-views and perturbation regimes (Section 4).

3.1 TABLE EMBEDDING MODELS

We evaluate table embedding models from multiple families with distinct inductive biases and input preprocessing. Methods that emit row- or cell-level representations are mapped to a single table vector via a fixed pooling operator. For serialization-based methods, we calculate a deterministic CSV-based string serialization $s(T)$ using a fixed number of rows. Unless stated otherwise, we apply mean pooling: for row outputs $L \in \mathbb{R}^{n \times d}$, with $L_i \in \mathbb{R}^d$ the i-th row vector, we obtain the final embedding as $f(T) = \frac{1}{n} \sum_{i=1}^{n} L_i$. For cell outputs $X \in \mathbb{R}^{n \times m \times d}$, with $X_{ij} \in \mathbb{R}^d$ a cell vector, we obtain the final embedding as $f(T) = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} X_{ij}$, where n is the number of rows and m is the number of columns. All embedding models are evaluated under the same observation interface, meaning that if a method requires a fixed input budget (e.g., due to token/row limits), the limits are applied to the same table view used elsewhere in evaluation. Hyperparameters are reported in Appendix A.1 (Table 3). Below, we list all table embedding models used in our experiments and group them into four distinct categories based on their underlying design principles.

(A) Structural & statistical information. These methods are deterministic and emphasize schema and distributional signatures.

- **HashingSchema.** Schema tokens consist of column titles paired with inferred data types, serialized as $\tau(T) = \{(\text{col}_j, \text{dtype}_j)\}_{j=1}^{m}$, and mapped by a hashing vectorizer h with d bins (Pedregosa et al., 2011), yielding $f(T) = h(\tau(T))$.
- **SchemaContent.** Schema features are first serialized and hashed and subsequently concatenated with metadata features: $f(T) = [\phi_{\text{meta}}(T) \parallel h(\tau(T))]$, where ϕ_{meta} is a fixed-size vector that includes size, missingness, and type ratios.
- **TableStatistics.** A global feature vector $\phi(T)$ is formed from row/column counts, missingness ratios, average missing per row/column, and mean and standard deviation of per-column unique counts. Coarse type ratios (numeric/categorical/datetime) are appended, and the vector is padded or truncated to a fixed dimension.
- **StatisticalSummary.** For numeric columns C_{num}, per-column moments (min, max, mean, standard deviation, skew, median) are computed and aggregated by mean and standard deviation across columns. For categorical columns C_{cat}, cardinality, mode frequency, and entropy are computed and aggregated. For datetime columns C_{time}, min/max timestamps are aggregated. The concatenation yields $f(T)$.
- **MatrixFactorization.** Let $M_{\text{num}} \in \mathbb{R}^{n \times m'}$ be the numeric submatrix after mean imputation and column centering. Its singular values $\sigma_1 \geq \cdots \geq \sigma_k$ define $f(T) = [\sigma_1, \ldots, \sigma_k]$; if no numeric columns exist, a zero vector is returned.

(B) Serialization-based text embeddings. These methods treat tables as sequences and encode them with hashing or text models.

- **HashingText.** The table is serialized to a CSV-style string $s(T)$ and embedded as $f(T) = h(s(T))$ using a hashing vectorizer h (Pedregosa et al., 2011).
- **LLMText.** A sentence-transformer encoder $e(\cdot)$ (Reimers & Gurevych, 2019) is applied to the serialized table, $f(T) = e(s(T))$, mapping tabular content into a semantic text space.

(C) Specialized representation methods. These methods encode explicit relational structure beyond flat vectors or train a model to optimize a representation objective.

- **TableVectorizer (skrub data, 2023).** Each row is mapped by Skrub's mixed-type vectorizer $g(\cdot)$ to a feature vector; the table embedding is mean pooled, $f(T) = \frac{1}{n}\sum_i g(\text{row}_i)$, with padding or truncation to a fixed dimension.
- **HyTREL (Chen et al., 2023).** A hypergraph is constructed with cell nodes and hyperedges for the table, columns, and rows. The encoder returns hyperedge embeddings. We take the table-level hyperedge vector as $f(T)$. We employ the provided model checkpoint pretrained with contrastive and discriminative objectives.
- **Armadillo (Pugnaloni et al., 2025).** Tables are converted to a graph with row, column, and value nodes; each value node connects to its corresponding row and column. Initial node features are produced by a token/value encoder, and a GNN propagates them across the graph. The table embedding is the global mean pool of node representations, $f(T) = \frac{1}{|V|}\sum_{v \in V} z_v$. We use the provided model checkpoint pretrained to estimate the overlap ratio between tables.

(D) Foundation model-derived embeddings. We extract final-layer hidden states from pretrained state-of-the-art foundation models and pool them to obtain a table vector.

- **TabPFN (Hollmann et al., 2023).** After preprocessing to obtain (X, y), we follow the extension proposed in Ye et al. (2025) to extract row-level embeddings $L \in \mathbb{R}^{n \times d}$ conditioned on a target column; then we aggregate to get the full table embedding $f(T) = \frac{1}{n}\sum_{i=1}^{n} L_i$. If a target is absent, a valid non-degenerate column is selected.
- **ConTextTab (Spinaci et al., 2025).** The encoder outputs a contextual grid $X \in \mathbb{R}^{n \times m \times d}$ over rows and columns for a dummy randomly generated target. The dummy target token is removed and the grid is pooled, $f(T) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} X_{ij}$.

4 EVALUATION PROTOCOL

To evaluate the desiderata mentioned in Section 2, we propose a unified framework relying on proxy tasks that do not require downstream fine-tuning. We evaluate embeddings on *table partitions* sampled as partial views of each source table, and attach multi-view supervision signals to probe different notions of similarity and grouping. All sampling and evaluation hyperparameters are fixed across methods and reported in the Appendix (Tables 4 and 5).

4.1 PARTITION SAMPLING AND OVERLAP

For each source table T with n rows and m columns, we sample K_p partitions under a fixed observation budget (maximum number of rows/columns) applied consistently across all methods.[1] We model a table as a typed relation $T = (R, C, X)$ with row index set $R = \{1, \dots, n\}$, column index set $C = \{1, \dots, m\}$, and cell values $X \in \mathcal{V}^{n \times m}$. A *partition* is a subtable induced by row and column subsets, $P = T[R', C']$. Let $\mathcal{P}(T) = \{P_1, \dots, P_{K_p}\}$ denote the sampled partitions from T. We represent the extent of each partition by its coordinate set

$$\mathcal{S}(P) = \{(i, j) \mid i \in R', j \in C'\},$$

and define the overlap between two partitions P_a and P_b using the Intersection over Union (IoU):

$$\mathrm{IoU}(P_a, P_b) = \frac{|\mathcal{S}(P_a) \cap \mathcal{S}(P_b)|}{|\mathcal{S}(P_a) \cup \mathcal{S}(P_b)|}.$$

This overlap signal induces a natural notion of similarity between partial views of the same table and directly supports D1.

4.2 LABELING SCHEMES

To assess different granularities of grouping, each partition P is assigned labels under four schemes:

- **Direct:** Table identity, $y_{\text{dir}}(P) = \mathrm{id}(T)$, i.e., all partitions coming from the same table share the same label.
- **Semantic:** High-level semantic labels, $y_{\text{sem}}(P) = g(T)$, where g encodes higher-level semantics from metadata or generative factors. For synthetic data, tuples of generative factors (e.g., `physics formula` + `density`) are mapped to distinct labels, while for real data tuples are formed from benchmark source and dataset identifier (e.g., `carte` + `nba draft`).
- **Semantic+Difficulty:** Semantic labels with an additional difficulty distinction for synthetic data (e.g., `physics formula` + `density` + `medium`), where the difficulty is a generative factor that controls the complexity of table values without altering other features.
- **Stat:** Coarse statistical signature, $y_{\text{stat}}(P) = l(P)$, where l returns the majority coarse column type if its fraction exceeds a predefined threshold. A partition label defaults to a *mixed label* if no coarse type fraction exceeds the threshold.

4.3 METRICS

To evaluate the performance of the different embedding models, we present a set of evaluation metrics that align with the desiderata (Figure 1). Unless stated otherwise, retrieval, triplet accuracy, and the linear probe accuracy are computed under each labeling scheme (Direct/Semantic/Semantic+Difficulty/Stat). We report scores averaged over label types for label-dependent metrics (D2 metrics) in the main Tables 1 and 2, with extended per-label breakdown in Figures 2 and 3 in the Appendix A.2.

D1: Partition consistency (IoU correlation). We measure the Spearman correlation between the partition overlap $\mathrm{IoU}(P_a, P_b)$ and the embedding similarity $\cos(f(P_a), f(P_b))$ over within-table pairs $\{(P_a, P_b) \mid P_a, P_b \in \mathcal{P}(T), a \neq b\}$. A higher correlation indicates that the embedding space preserves the similarity structure induced by overlapping partial views. We additionally report Pearson correlation and mean-squared-error (MSE) in Figures 2 and 3.

[1] Exact budgets and sampling hyperparameters are reported in Appendix A.1.

D2: Triplet ranking (TR). Given an anchor partition A, a positive partition B with the same label ($y(B) = y(A), B \neq A$), and a negative partition C with a different label ($y(C) \neq y(A)$), we evaluate whether the embedding places the positive closer to the anchor than the negative. We report the fraction of triplets satisfying $\cos(f(A), f(B)) > \cos(f(A), f(C))$ as a proxy for ranking capability. We report multiple versions of the triplet ranking depending on how the negative partition is sampled: **TR-R** (random negative), samples C uniformly from the negative pool. **TR-H** (hardest negative), selects C to be the most similar negative by cosine similarity. **TR-CH** (cluster-hard negative), selects C as the most similar negative restricted to the anchor's MiniBatch k-means cluster (with a different label). We aggregate **TR-Avg** $= \frac{1}{3}(\text{TR-R} + \text{TR-H} + \text{TR-CH})$, and report the different triplet metrics in Appendix A.2.

D2: Clustering alignment (CL). We cluster embeddings using MiniBatch k-means and evaluate alignment with labels using Purity, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). We aggregate **CL-Avg** $= \frac{1}{3}(\text{Purity} + \text{NMI} + \text{ARI})$, and report the different clustering metrics in Appendix A.2.

D2: Retrieval (Recall@K_r). For each anchor partition, we retrieve its K_r nearest neighbors by cosine similarity, excluding the anchor itself. A retrieval query is successful if at least one of the top-K_r neighbors shares the same label as the anchor. We report Recall@K_r as the percentage of successful queries across all anchors.

D2: Linear probe accuracy (LP). We train a logistic regression classifier on top of the frozen partition embeddings to predict labels, assessing whether the embedding captures discriminative semantic or statistical information about the table.

D3: Robustness. We perturb each partition by (i) permuting rows and columns, (ii) masking a subset of cells, or (iii) adding small noise to cell values, then measure the cosine similarity between the embeddings of the original and perturbed partitions.

D4: Efficiency (Throughput). We report embedding throughput in tables per second (Tbl/s), measured as the wall-clock rate for computing embeddings over partitions measured on the same compute hardware. This enables comparison of the quality-cost trade-off across embedding methods.

5 Experiments and results

5.1 Data sources

We evaluate on both synthetic and real open-source corpora to balance controlled semantics with real-world variability. Across both settings, we sample 100 source tables and compute the metrics on $K_p = 10$ partitions per table under a shared observation budget (Table 5).

Synthetic data. Synthetic corpora are generated from predefined rules with controllable difficulty levels, enabling systematic variation in scale and nonlinearity while preserving table structure. Each generator family provides generative factors that induce higher-level groupings in the embedding space, which are used for semantic (and semantic+difficulty) labels defined in Section 4. We define four dataset families:

- **Physics formulas.** Fixed schema with numeric relations (e.g., density, ideal gas, kinetic energy, Ohm's law) , emphasizing intra-column numerical reasoning.
- **Geometric shapes.** Mixed numeric/categorical tables with closed-form area/perimeter rules for different geometric shapes; additional random colors act as distractors.
- **Temporal reasoning.** Timestamped event sequences for process lifecycles (HR, order processing, invoicing). Each table is generated under a company-specific random profile with distinct evolution cycles.
- **SCM prior.** Data generated by structural causal models adapted from TabICL (Qu et al., 2025) implementation, with controlled complexity and regression/classification targets.

Table 1: Evaluation results for synthetic data (mean ± 95% CI over 10 seeds).

Embedder	D1 Consistency		D2 Label-Based - Avg					D3 Robustness				D4 Efficiency		Overall
	Spearman	Rank	TR-Avg	CL-Avg	R@5	LP	Rank	Perm	Noise	Mask	Rank	Tbl/s	Rank	Avg Rank
HashingSchema	**0.59±0.02**	1.00	**0.53±0.01**	0.54±0.02	0.72±0.01	0.61±0.01	4.25	**1.00±0.00**	0.98±0.00	0.99±0.00	5.17	**389.67±3.48**	1.00	2.85
SchemaContent	0.29±0.02	9.00	0.01±0.00	0.26±0.01	0.78±0.01	0.59±0.02	8.50	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	3.33	336.66±6.87	3.00	5.96
TableStatistics	0.34±0.01	6.00	0.13±0.01	0.34±0.01	0.71±0.01	0.51±0.02	9.00	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	3.50	338.88±6.39	2.00	5.12
StatisticalSummary	0.35±0.04	5.00	0.36±0.01	0.45±0.03	0.90±0.01	0.57±0.02	4.25	**1.00±0.00**	**0.99±0.00**	**1.00±0.00**	4.17	192.32±1.72	6.00	4.85
MatrixFactorization	0.40±0.02	4.00	0.23±0.01	0.32±0.01	0.76±0.03	0.49±0.03	8.00	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	2.17	221.83±2.31	5.00	4.79
HashingText	0.56±0.04	2.00	0.47±0.01	0.47±0.02	**1.00±0.00**	**0.99±0.01**	2.00	**1.00±0.00**	0.64±0.01	0.97±0.00	7.50	332.94±4.60	4.00	3.88
LLMText	0.31±0.02	7.00	0.51±0.02	**0.56±0.03**	0.87±0.01	0.76±0.02	2.25	0.97±0.00	0.98±0.00	**0.99±0.00**	7.00	174.54±4.65	7.00	5.81
TableVectorizer	0.30±0.02	8.00	0.18±0.01	0.22±0.01	0.74±0.01	0.38±0.02	10.25	0.30±0.01	**1.00±0.00**	0.80±0.02	9.00	49.14±2.25	9.00	9.06
HyTREL	0.18±0.05	10.00	0.25±0.01	0.32±0.02	0.84±0.01	0.74±0.01	5.00	**0.99±0.00**	0.98±0.00	0.89±0.01	8.67	10.00±0.23	10.00	8.42
Armadillo	0.43±0.04	3.00	0.31±0.01	0.30±0.01	0.88±0.01	0.74±0.02	5.00	**1.00±0.00**	0.47±0.01	0.87±0.00	8.83	63.05±1.07	8.00	6.21
TabPFN	-0.12±0.04	12.00	0.16±0.01	0.25±0.01	0.72±0.01	0.49±0.03	10.25	0.75±0.02	0.93±0.01	0.99±0.00	9.33	6.49±0.15	12.00	10.90
ConTextTab	0.04±0.03	11.00	0.18±0.00	0.27±0.02	0.71±0.01	0.57±0.02	9.25	0.97±0.00	0.97±0.00	0.97±0.00	9.33	9.97±1.14	11.00	10.15

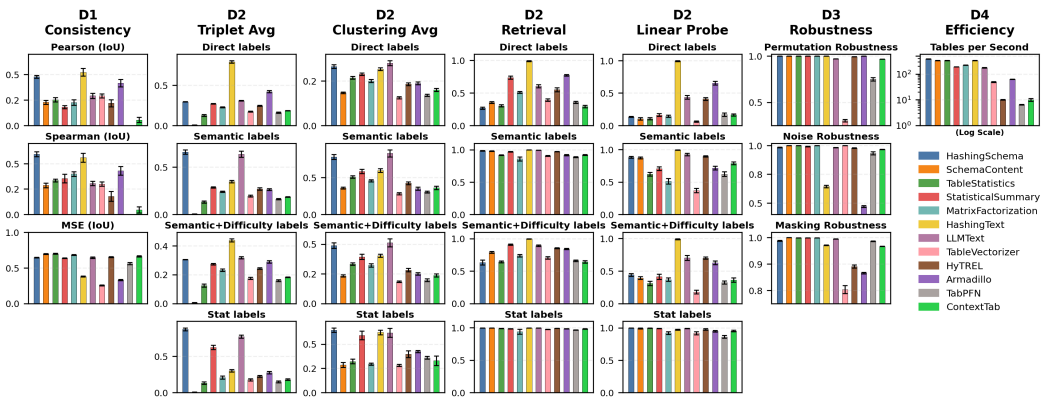

Figure 2: Desiderata bar chart results for synthetic data (mean ± 95% CI over 10 seeds).

Difficulty levels control numeric ranges and the number of SCM layers, yielding progressively harder variations without changing the labeling interface.

Real open-source data. Real tables are sampled from CARTE (Kim et al., 2024), OpenML-CC18 (Bischl et al., 2021), and OpenML-CTR23 (Fischer et al., 2023). For each benchmark task, we form a table by concatenating the feature matrix with a single target column and evaluate on fixed-size partitions under the same budget as synthetic (Table 5). We retain task metadata (benchmark source, dataset identifier; e.g., `carte+nba_draft`) to derive semantic labels.

5.2 Synthetic Data Results

On synthetic data (Table 1, Figure 2), lightweight hashing and statistical-summary embedders dominate the overall ranking. HashingSchema achieves the best overall average rank, with the strongest D1 consistency and the highest D4 throughput. HashingText provides the strongest D2 retrieval and linear probe signals (near-ceiling Recall@5 and LP), while LLMText yields the strongest D2 clustering alignment (Table 1).

D2 metrics are label-dependent (Appendix A.2). For *Direct* identity labels, HashingText achieves the strongest triplet ranking under hard and conditional-hard negatives (Table 6 in Appendix). For the *Stat* labeling scheme, HashingSchema is near-ceiling on both triplet and clustering metrics (Table 9 in Appendix). Increasing label granularity from *Semantic* to *Semantic+Difficulty* reduces hard-negative triplet accuracy across most schema-centric and statistical summary methods (Table 8 in Appendix), while text/content-based methods remain comparatively stronger on strict triplet ranking.

For D3 robustness, most methods saturate on permutation robustness, with TableVectorizer as the clear exception due to its ordering sensitivity (Table 1). Noise robustness is most challenging for value-/content-sensitive methods (Figure 2). Finally, embeddings pooled from prediction-oriented TFMs (TabPFN, ConTextTab) rank poorly on D1/D2 under this protocol despite strong robustness on several D3 perturbations.

Table 2: Evaluation results for real data (mean ± 95% CI over 10 seeds).

Embedder	D1 Consistency		D2 Label-Based - Avg					D3 Robustness				D4 Efficiency		Overall
	Spearman	Rank	TR-Avg	CL-Avg	R@5	LP	Rank	Perm	Noise	Mask	Rank	Tbl/s	Rank	Avg Rank
HashingSchema	0.32±0.04	2.00	**0.57±0.02**	0.36±0.02	**0.97±0.00**	0.88±0.02	2.25	**1.00±0.00**	0.94±0.01	0.95±0.00	7.17	64.61±24.25	4.00	3.85
SchemaContent	0.27±0.02	3.00	0.10±0.02	0.17±0.01	0.81±0.01	0.89±0.02	8.00	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	3.00	**68.64±26.47**	2.00	4.00
TableStatistics	0.18±0.03	7.00	0.08±0.01	0.23±0.01	0.76±0.01	0.59±0.02	8.25	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	3.17	**68.91±18.31**	1.00	4.85
StatisticalSummary	0.19±0.03	5.00	0.31±0.01	0.29±0.02	0.85±0.01	0.55±0.02	5.75	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	3.17	38.53±16.61	7.00	5.23
MatrixFactorization	0.17±0.02	8.00	0.20±0.00	0.19±0.01	0.65±0.02	0.27±0.01	10.00	**1.00±0.00**	**1.00±0.00**	**1.00±0.00**	3.17	43.46±20.40	6.00	6.79
HashingText	**0.42±0.03**	1.00	0.50±0.02	0.30±0.03	**0.97±0.01**	**0.90±0.02**	2.25	**1.00±0.00**	0.75±0.02	0.98±0.00	6.83	67.35±23.68	3.00	3.27
LLMText	0.15±0.05	10.00	0.55±0.02	**0.38±0.03**	**0.98±0.00**	0.87±0.02	2.00	0.92±0.01	0.94±0.00	0.99±0.00	8.00	55.74±16.02	5.00	6.25
TableVectorizer	0.04±0.02	12.00	0.18±0.01	0.18±0.01	0.57±0.01	0.29±0.02	10.75	0.33±0.01	0.98±0.00	0.84±0.01	9.33	11.95±4.11	8.00	10.02
HyTREL	0.17±0.05	9.00	0.25±0.01	0.21±0.01	0.85±0.01	0.77±0.02	5.50	**0.99±0.00**	0.89±0.01	0.96±0.00	9.00	5.37±0.40	9.00	8.12
Armadillo	0.23±0.04	4.00	0.35±0.01	0.21±0.01	0.85±0.01	0.69±0.02	6.25	**1.00±0.00**	0.60±0.02	0.78±0.01	9.17	4.66±2.30	11.00	7.60
TabPFN	0.06±0.04	11.00	0.14±0.01	0.19±0.02	0.60±0.02	0.43±0.02	10.00	0.77±0.01	0.97±0.00	0.97±0.00	8.00	5.32±0.31	10.00	9.75
ConTextTab	0.19±0.03	6.00	0.24±0.01	0.22±0.03	0.75±0.01	0.69±0.02	7.00	0.97±0.00	0.97±0.00	0.97±0.00	8.00	3.49±0.60	12.00	8.25

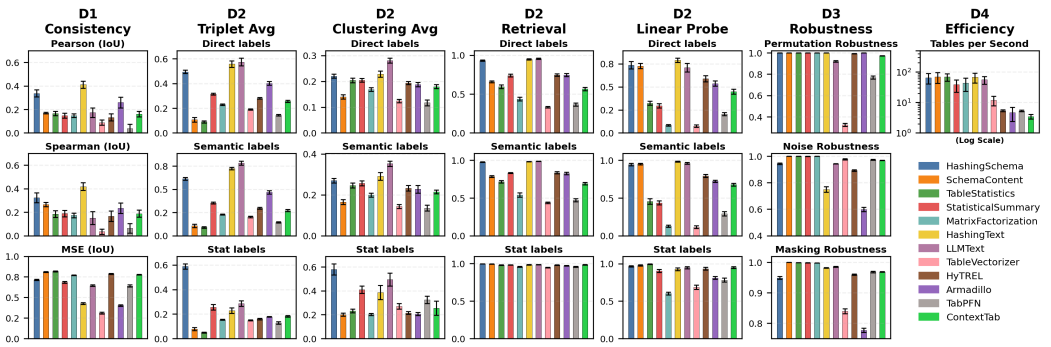

Figure 3: Desiderata bar chart results for real data (mean ± 95% CI over 10 seeds).

5.3 Real open-source data results

On real open-source data (Table 2, Figure 3), lightweight hashing and statistical-summary embedding models remain consistently among the top-performing methods. HashingText ranks best overall, combining the strongest D1 consistency with strong D2 retrieval and linear probe performance, while HashingSchema follows closely, reflecting the continued importance of schema/type cues in heterogeneous benchmark suites. LLMText achieves strong D2 clustering and retrieval signals but ranks poorly on D1 (Table 2).

The per-label D2 breakdown (Appendix A.2) shows that the *Direct* labeling scheme is substantially harder on real tables than on synthetic data: hard-negative triplet scores are low across most methods (Table 11 in Appendix), even when retrieval and linear probe remain high. *Stat* labels are the easiest and align strongly with schema cues (Table 13 in Appendix), while *Semantic* labels score in between identity and coarse type groupings (Table 12 in Appendix). Qualitatively, the t-SNE projections (Figure 5 in Appendix) mirror this trend: *Stat* labels form visibly separated regions for schema-driven methods, while *Direct* labels exhibit substantial mixing for most embedding models.

As in synthetic data, D3 permutation robustness is near-saturated for most methods except TableVectorizer (Table 2). Noise robustness again differentiates value-/content-sensitive methods (Figure 3). Finally, pooled representations from TFMs remain weak as table embeddings under D1/D2, with ConTextTab outperforming TabPFN. Moreover, TFMs are substantially slower than lightweight baselines (D4).

6 Discussion

Overall findings and the quality–cost frontier. Across both synthetic families (Table 1) and open-source corpora (Table 2), we observe a consistent separation between (i) lightweight fingerprint-style embedders (hashing, summaries, serialization) that perform well on D1/D2 reliably under shared observation budgets, and (ii) pooled representations extracted from prediction-oriented tabular foundation models, which are not consistently strong as table-level embeddings under our protocol. A key driver is objective and interface alignment, where lightweight methods encode cues that remain stable across subsampled table views (shared schema tokens, shared lexical/value tokens, low-order statistics), while TFMs are trained for row-/cell-level predictive inference and are not optimized to preserve a global table geometry under partial views. These representational differences coexist with

large efficiency gaps (D4), where methods with similar retrieval or probe accuracy can differ by orders of magnitude in throughput, materially shifting the practical quality-cost frontier for corpus-scale indexing and search.

Multi-view discriminability desideratum (D2) is label-dependent because each label type assesses a different representation capability. The per-label breakdown (Appendix A.2) shows that embedding models do not perform uniformly under D2. More precisely, the *Direct* labeling rewards fine-grained content distinctiveness, *Stat* labeling rewards coarse type/distribution signatures, and *Semantic* labeling sits between them and often correlates with benchmark conventions. This explains the systematic gap we observe between metrics, where retrieval and linear probe scores can be high when a method provides a *coarse* neighborhood signal useful for candidate generation, yet hard/conditional-negative triplet scores remain low when the embedding cannot reliably distribute confusing content in the embedding space. On real open-source tables, solving proxy tasks for direct labeling scheme becomes substantially harder (Appendix Table 11), consistent with shared templates and recurring feature sets that create near-collisions under partial views. While averaging the results under D2 provides useful collective insights, looking at individual scores unveils additional insights regarding per-label type performance. For detailed scores, please refer to Tables 1–2 jointly with Appendix A.2.

Difficulty levels in the semantic labeling scheme distinguish embedding models that can capture numerical scale and complexity. Injecting difficulty into the semantic label (*Semantic+Difficulty* on synthetic) exposes a concrete failure mode for schema-centric methods. While *Semantic* groupings are often captured by schema- and text-driven fingerprints, hard-negative triplet accuracy drops under the *Semantic+Difficulty* labeling scheme for methods that rely primarily on schema/type cues (Table 8 in Appendix). This is consistent with difficulty altering value scale/nonlinearity and interaction structure without changing schema templates. The t-SNE projections (Figure 4 in Appendix) reflect the same trend.

Why do simple baselines achieve strong performance, and under what conditions is this desirable rather than misleading? In many practical settings, lexical and structural features (column names/types, token patterns, low-order statistics) are part of the operational definition of similarity. Tables with similar schemas and recognizable value types are useful neighbors. Under this lens, strong performance of schema-/serialization-based methods on open-source corpora is not inherently misleading. However, synthetic generators can exhibit shallow alignment, where factors may correlate with schema templates or stable token patterns, allowing lightweight methods to succeed on some label types without recovering deeper interaction structure. The label-wise D2 decomposition helps separate these regimes, where *Stat* labeling scheme can often be solved from schema/type cues, whereas *Direct* and *Semantic+Difficulty* labeling schemes under hard negatives directly stress value- and interaction-sensitive information.

Prediction performance does not imply table-level embedding utility. TabPFN and ConTextTab are optimized for predictive inference (row-/cell-level objectives, in-context learning, limited context windows). Pooling their internal states into a single table vector is therefore not guaranteed to preserve the global geometry required by D1–D2 (partition consistency, stable neighborhoods, discriminability across granularities). Empirically, TFMs can look robust on several D3 perturbations while remaining weak on D1/D2 (Tables 1 and 2), reinforcing that predictive competence does not automatically yield a usable table-level similarity space under fixed budgets. This points to the need of objective-level alignment, where TFMs likely require explicit training for representation extraction that shape table-level geometry rather than relying on pooled predictive states.

Desiderata-aligned supervision can improve representation quality. Armadillo, which is pretrained with an explicit overlap-prediction objective aligned with our D1 signal, yields improved representation quality. Under the same evaluation interface, this desiderata-aligned supervision yields non-trivial table-level structure relative to methods that do not optimize for an overlap-aware objective. This provides evidence that the gap between TFMs and strong table embedding models is not purely architectural; rather, the two are not fully aligned in their objectives. As a result, prediction-oriented TFMs may benefit from auxiliary objectives that explicitly target table-level geometry, such as (i) overlap/IoU prediction between partial views, (ii) view-agreement losses that regress similarity to

overlap, or (iii) contrastive learning where positives are high-overlap views and hard negatives are low-overlap or semantically confusable partitions. Such objectives can be added while preserving the downstream predictive performance, offering a principled path from TFMs to reliable table-level embedding models.

Robustness metrics: strong performance for most methods, with some exceptions. Permutation robustness (D3) often saturates because many methods are invariant to column/row permutations by construction or because the perturbation does not challenge their feature extraction. A clear exception is TableVectorizer, which degrades under permutations (Tables 1 and 2), consistent with ordering sensitivity. Noise/masking robustness is more discriminative for value-/content-sensitive methods, where methods that exploit fine-grained value tokens can exhibit stronger strict D2 behavior but become brittle under corruption, which matters for large-scale datasets with potential decoding noise. Future robustness evaluations may benefit from stronger or more realistic edits (e.g., column renaming, unit scaling, value rounding, or row subsampling with distribution shift).

Collective implications of the evaluation results. Taken together, these results motivate (i) standardized evaluation protocols that report per-label D2 behavior (including hard negatives), robustness, and efficiency under shared budgets, and (ii) model designs and training objectives that explicitly optimize *table-level* geometry rather than relying on pooled predictive states. More broadly, the observed specialization suggests that table similarity is inherently multi-faceted, which we capture via different labeling schemes. Practical systems should therefore either (a) select embedding models aligned with the intended notion of similarity, or (b) combine complementary signals, such as schema/type fingerprints for candidate generation with value-sensitive ranking. We view the proposed framework as a stepping stone toward designing tabular foundation models that are capable of producing effective, reliable, and reusable table-level embeddings for vector-based indexing, retrieval, and lightweight supervision.

7 Conclusion

We introduced a unified evaluation framework for table-level embeddings, evaluating their performance based on four desiderata: consistency under partial views (D1), discriminability across label granularities (D2), robustness (D3), and efficiency (D4). To measure these desiderata across embedding models, we designed proxy tasks that require no downstream fine-tuning. We benchmarked diverse embedding models, including structural and statistical embedders, serialization-based text encoders, specialized table encoders, and pooled representations from tabular foundation models (TFMs) on controlled synthetic tables and real open-source corpora (CARTE, OpenML-CC18, OpenML-CTR23) under shared budgets. These desiderata directly support corpus-scale applications and data curation for training tabular foundation models, including overlap estimation and approximate deduplication (D1), filtering and clustering under multiple similarity notions (D2), robustness to noisy or partially observed tables (D3), and efficient large-scale embedding extraction for sampling and retrieval (D4).

Across our evaluation settings, lightweight fingerprints and serialization methods form the strongest quality-cost frontier, while pooled representations from prediction-oriented TFMs do not yield reliable table-level embeddings. Our results suggest that TFMs may benefit from desiderata-aligned supervision (e.g., overlap-aware view agreement or contrastive objectives) to explicitly enhance table-level representations.

Our proxy tasks target core table-level properties, but do not cover all downstream requirements, such as multi-hop reasoning across multiple tables; extending the benchmark with composition- and reasoning-aware evaluations is a natural next step. Results also depend on the observation interface (partition sampling, serialization budgets) and on pooling choices for model-derived representations; future work should study pooling/aggregation mechanisms and budget scaling more systematically, and evaluate whether learned pooling can close the D1-D2 gap for TFMs. Finally, robustness is evaluated on benign perturbations; incorporating more realistic edits, such as column renaming, unit scaling, rounding, and distribution-shift subsampling would better characterize brittleness in practice.

ACKNOWLEDGMENTS

We would like to thank Sam Thelin, Marco Spinaci, Markus Kohler, and Johannes Hoffart for their insightful comments and suggestions throughout the development of this work, which have greatly contributed to shaping its direction and quality.

REFERENCES

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan N. van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. In *NeurIPS 2021 Datasets and Benchmarks Track*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/hash/c7e1249ffc03eb9ded908c236bd1996d-Abstract-round2.html.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9630–9640. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00951. URL https://doi.org/10.1109/ICCV48922.2021.00951.

Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. HyTrel: Hypergraph-enhanced tabular data representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/66178beae8f12fcd48699de95acc1152-Abstract-Conference.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019. URL https://aclanthology.org/N19-1423/.

Benjamin Feuer, Robin Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. TuneTables: Context optimization for scalable prior-data fitted networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/97dc07f1253ab33ee514f395a82fa7cc-Abstract-Conference.html.

Sebastian Felix Fischer, Matthias Feurer, and Bernd Bischl. OpenML-CTR23 – a curated tabular regression benchmarking suite. In *AutoML 2023 Workshop Track*, 2023. URL https://openreview.net/forum?id=HebAOoMm94.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

Frederik Hoppe, Lars Kleinemeier, Astrid Franz, and Udo Göbel. Comparing task-agnostic embedding models for tabular data, 2025. URL https://arxiv.org/abs/2511.14276.

Myung Jun Kim, Léo Grinsztajn, and Gaël Varoquaux. CARTE: Pretraining and transfer for tabular learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23843–23866. PMLR, 2024. URL https://proceedings.mlr.press/v235/kim24d.html.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=KSugKcbNf9.

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Francesco Pugnaloni, Luca Zecchini, Matteo Paganelli, Matteo Lissandrini, Felix Naumann, and Giovanni Simonini. Table overlap estimation through graph embeddings. *Proc. ACM Manag. Data*, 3(3), June 2025. doi: 10.1145/3725365. URL https://doi.org/10.1145/3725365.

Jingang Qu, David Holzmueller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025. URL https://proceedings.mlr.press/v267/qu25a.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*. PMLR, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL https://aclanthology.org/D19-1410/.

skrub data. skrub: Machine learning with dataframes, 2023. URL https://skrub-data.org/.

Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. Contexttab: A semantics-aware tabular in-context learner. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=kGMRb4jbTP.

Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabPFN v2: Understanding its strengths and extending its capabilities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=lQYNmlTwtc.

A APPENDIX

A.1 FURTHER EXPERIMENTAL DETAILS

Table 3: Embedding model hyperparameters.

Embedder	Dim	Key Hyperparameters
HashingSchema	1024	`n_features`=1024 (feature hashing on schema)
SchemaContent	256	`n_features`=256
TableStatistics	14	`output_dim`=14 (fixed statistical features)
StatisticalSummary	22	Numeric: min, max, mean, std, skew, median; Categorical: cardinality, entropy
MatrixFactorization	16	`n_components`=16, `center`=True
HashingText	1024	`n_features`=1024 (feature hashing on text)
LLMText	384	`model`=all-MiniLM-L6-v2, `max_rows`=32
TableVectorizer	512	`output_dim`=512, `cardinality_threshold`=10
HyTREL	768	`model`=bert-base-uncased, `batch_size`=16 `max_token_len`=64, `max_col_len`=20, `max_row_len`=30
Armadillo	300	`gnn_type`=GraphSAGE, `num_layers`=3 `hidden_channels`=300, `initial_embedding`=sha256
TabPFN	192	`n_estimators`=1, `max_samples`=50k, `max_features`=2000 `max_classes`=10, `data source`=test
ConTextTab	768	`model_size`=base, `classification`=cross-entropy `regression`=l2, `num_regression_bins`=10

Table 4: Evaluation metrics and their parameters.

Desideratum	Metric	Configuration
D1: Consistency	Pearson correlation Spearman correlation MSE	IoU vs. cosine similarity IoU vs. cosine similarity IoU vs. cosine similarity
D2: Triplet	TR-R (Random) TR-H (Hard) TR-CH (Cluster-Hard) TR-Avg	Random negative, margin=0.01 Hardest negative mining Hard negatives within clusters Mean of TR-R, TR-H, TR-CH
D2: Clustering	Purity NMI ARI CL-Avg	KMeans, n_cluster=10 Normalized mutual information Adjusted Rand index Mean of Purity, NMI, ARI
D2: Retrieval	Recall@5	Top-5 nearest neighbors
D2: Linear Probe	Accuracy	Logistic regression (max_iter=1000)
D3: Robustness	Permutation Noise Masking	Row/column shuffle invariance σ=[0.0, 0.01, 0.05, 0.1] Gaussian noise [0,5,10,25]% random cell masking
D4: Efficiency	Tables/second	Embedding throughput (measured on Nvidia T4 GPU)

Table 5: Data sampling and partitioning configuration.

Parameter	Value	Parameter	Value	Parameter	Value
Tables	100	Partitions/table	10	Overlap ratio	0.5
Row fraction	[0.2, 0.5]	Column fraction	[0.2, 0.5]	Min partition	10×5
Seeds	10	Rows (synthetic)	100	Rows (real)	varies

A.2 EXTENDED D2 PER-LABEL RESULTS

This appendix provides a label-wise decomposition of D2, complementing the label-averaged D2 scores reported in the main paper. For each label type, we report triplet ranking under three negative-sampling strategies (TR-R / TR-H / TR-CH), clustering alignment (Purity / NMI / ARI), retrieval (Recall@5), and linear-probe accuracy. TR-H and TR-CH are typically harder than TR-R because they focus on high-similarity negatives (hard negatives) and confusable negatives from the anchor's embedding cluster (cluster-hard negatives), respectively. The *Avg Rank* column summarizes performance across the individual D2 components for that label type.

A.2.1 SYNTHETIC DATA RESULTS

Direct labeling exposes identity collisions under hard and cluster-hard negatives for most methods. Under Direct labeling (Table 6), several methods achieve high TR-R yet collapse to near-zero on TR-H and TR-CH, confirming that random negatives are often trivial, while the hard and cluster-hard negatives reveal embedding collisions. HashingText is the clear outlier, remaining strong on both TR-H and TR-CH, consistent with preserving fine-grained identity features from serialized content. Notably, Armadillo is the only other method with non-trivial TR-H and TR-CH (while still trailing HashingText), indicating partial identity preservation beyond schema/statistical fingerprints. By contrast, schema-only and coarse statistical baselines (e.g., HashingSchema, TableStatistics, StatisticalSummary) retain reasonable TR-R but fail under hard negatives, consistent with many partitions sharing near-identical schema/type signatures. Finally, pooled TFM embeddings (TabPFN, ConTextTab) show moderate TR-R performance but collapse under hard negatives, suggesting that pooling produces weak identity geometry even when some coarse similarity is recoverable.

Table 6: D2 metrics for synthetic data - Direct labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Direct)					Clustering (Direct)					Retrieval/Probe (Direct)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	0.88±0.01	0.00±0.00	0.00±0.00	0.30±0.00	5.75	**0.12±0.01**	0.57±0.01	0.10±0.01	0.27±0.01	1.75	0.27±0.02	0.13±0.01	10.50	6.00
SchemaContent	0.02±0.00	0.00±0.00	0.00±0.00	0.01±0.00	11.62	0.07±0.00	0.34±0.01	0.03±0.00	0.15±0.00	10.00	0.35±0.01	0.11±0.02	9.50	10.38
TableStatistics	0.37±0.03	0.00±0.00	0.00±0.00	0.12±0.01	11.12	0.09±0.01	0.48±0.01	0.07±0.00	0.21±0.01	5.75	0.30±0.01	0.10±0.02	10.50	9.12
StatisticalSummary	0.80±0.01	0.01±0.00	0.01±0.00	0.27±0.00	4.50	0.10±0.01	0.51±0.01	0.07±0.00	0.23±0.01	4.25	0.74±0.02	0.16±0.02	5.00	4.58
MatrixFactorization	0.68±0.02	0.00±0.00	0.00±0.00	0.23±0.01	8.75	0.11±0.01	0.45±0.01	0.04±0.00	0.20±0.01	6.25	0.51±0.01	0.14±0.02	7.00	7.33
HashingText	**0.98±0.00**	**0.69±0.02**	**0.70±0.02**	**0.79±0.01**	1.00	0.12±0.01	0.56±0.01	0.08±0.00	0.25±0.01	3.00	**0.99±0.01**	**0.99±0.01**	1.00	1.67
LLMText	0.91±0.01	0.01±0.00	0.01±0.00	0.31±0.00	2.75	**0.12±0.01**	**0.60±0.02**	**0.11±0.01**	**0.28±0.01**	1.25	0.60±0.02	0.43±0.03	3.50	2.50
TableVectorizer	0.51±0.02	0.00±0.00	0.00±0.00	0.17±0.01	7.00	0.07±0.00	0.28±0.01	0.02±0.00	0.13±0.00	12.00	0.39±0.02	0.06±0.01	9.50	9.50
HyTREL	0.73±0.02	0.00±0.00	0.00±0.00	0.24±0.01	6.00	0.09±0.01	0.41±0.01	0.05±0.00	0.19±0.01	7.25	0.55±0.03	0.41±0.02	4.50	5.92
Armadillo	0.84±0.01	0.20±0.02	0.22±0.02	0.42±0.01	2.50	0.10±0.01	0.41±0.01	0.06±0.00	0.19±0.01	6.75	0.77±0.01	0.65±0.03	2.00	3.75
TabPFN	0.48±0.02	0.00±0.00	0.00±0.00	0.16±0.01	8.50	0.07±0.00	0.31±0.01	0.02±0.00	0.14±0.00	11.00	0.36±0.02	0.17±0.03	6.50	8.67
ConTextTab	0.55±0.01	0.00±0.00	0.00±0.00	0.18±0.00	8.50	0.08±0.00	0.36±0.01	0.04±0.00	0.16±0.01	8.75	0.29±0.02	0.16±0.02	8.50	8.58

Semantic labeling highlights global grouping strength of serialization-based text embedding models. Under Semantic labeling (Table 7), HashingSchema achieves the strongest triplet performance, implying that schema tokens correlate with the generative factors used to define the semantic labels. However, LLMText provides the best clustering alignment by a substantial margin, indicating that semantic text encoders better preserve the global grouping structure induced by the generators, beyond local ranking. Retrieval and linear probe scores are generally high in this regime, but remain discriminative, with HashingText reaching near-perfect retrieval and linear probe performance, reflecting that content-level fingerprints preserve both separability and linear decodability for these semantic groupings.

Table 7: D2 metrics for synthetic data - Semantic labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Semantic)					Clustering (Semantic)					Retrieval/Probe (Semantic)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	**0.96±0.01**	**0.50±0.03**	**0.51±0.03**	**0.66±0.02**	1.25	0.78±0.03	0.85±0.02	0.70±0.05	0.78±0.03	2.00	0.98±0.00	0.88±0.01	3.50	2.25
SchemaContent	0.02±0.00	0.00±0.00	0.00±0.00	0.01±0.00	11.75	0.42±0.02	0.45±0.02	0.21±0.01	0.36±0.01	9.25	0.98±0.01	0.87±0.01	4.50	8.50
TableStatistics	0.40±0.03	0.00±0.00	0.00±0.00	0.13±0.01	11.25	0.52±0.02	0.60±0.01	0.40±0.02	0.51±0.02	5.25	0.92±0.01	0.62±0.03	9.00	8.50
StatisticalSummary	0.82±0.01	0.01±0.01	0.02±0.01	0.29±0.01	5.25	0.60±0.04	0.67±0.01	0.48±0.04	0.58±0.03	3.75	0.97±0.00	0.71±0.03	7.00	5.33
MatrixFactorization	0.71±0.02	0.00±0.00	0.01±0.01	0.24±0.01	7.75	0.54±0.02	0.56±0.01	0.28±0.02	0.46±0.02	5.75	0.86±0.03	0.51±0.04	11.50	8.33
HashingText	0.78±0.01	0.11±0.01	0.15±0.02	0.35±0.01	3.25	0.58±0.03	0.69±0.02	0.51±0.03	0.60±0.03	3.25	**1.00±0.00**	**0.99±0.00**	1.00	2.50
LLMText	**0.96±0.01**	0.46±0.04	0.48±0.05	0.64±0.03	1.75	**0.82±0.04**	**0.90±0.03**	**0.76±0.07**	**0.83±0.05**	1.00	**0.99±0.00**	0.93±0.02	2.00	1.58
TableVectorizer	0.52±0.02	0.03±0.01	0.03±0.01	0.19±0.01	6.75	0.39±0.02	0.31±0.02	0.15±0.01	0.28±0.01	11.75	0.90±0.01	0.37±0.04	11.00	9.83
HyTREL	0.75±0.03	0.03±0.01	0.03±0.01	0.27±0.01	5.50	0.49±0.02	0.50±0.02	0.28±0.02	0.42±0.02	7.00	0.97±0.00	0.90±0.01	4.00	5.50
Armadillo	0.68±0.02	0.05±0.01	0.06±0.01	0.26±0.01	5.25	0.42±0.02	0.38±0.02	0.25±0.03	0.35±0.02	9.25	0.92±0.01	0.71±0.03	8.00	7.50
TabPFN	0.48±0.02	0.00±0.00	0.00±0.00	0.16±0.01	9.25	0.37±0.01	0.37±0.01	0.17±0.01	0.30±0.01	11.25	0.89±0.01	0.62±0.03	10.00	10.17
ConTextTab	0.55±0.01	0.00±0.00	0.00±0.00	0.18±0.00	9.00	0.43±0.02	0.41±0.03	0.25±0.02	0.36±0.02	8.50	0.92±0.01	0.79±0.02	6.50	8.00

Semantic+Difficulty labeling reveals schema-only limitations under value-complexity shifts. Under Semantic+Difficulty labeling (Table 8), we refine semantic groups by adding a difficulty factor,

so tables with the same schema can receive different labels depending on content complexity. Triplet scores drop broadly, indicating that many methods fail to distinguish partitions based on value-level complexity. The degradation is most pronounced for schema-centric hashing, with HashingSchema largely collapsing on TR-H and TR-CH, suggesting that difficulty is not recoverable from schema tokens alone and is instead expressed through value distributions and nonlinear relationships. In this harder setting, HashingText provides the strongest triplet ranking (including hard and cluster-hard negatives), whereas LLMText again achieves the best clustering alignment. This reinforces a recurring trade-off between content fingerprints which excel at fine-grained neighbor ranking, and serialization-based methods that stabilizes global grouping. The Semantic+Difficulty labeling scheme supports one of the central narratives of this paper, namely that incorporating difficulty granularity into semantic labeling renders evaluation sensitive to whether embeddings capture numerical scale and complexity, as opposed to merely shallow schema-derived features.

Table 8: D2 metrics for synthetic data - Semantic+Difficulty labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Semantic+Difficulty)					Clustering (Semantic+Difficulty)					Retrieval/Probe (Semantic+Difficulty)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	0.91±0.00	0.00±0.00	0.00±0.00	0.31±0.00	4.50	0.40±0.03	0.70±0.02	0.36±0.03	0.49±0.02	2.00	0.63±0.04	0.44±0.02	8.50	5.00
SchemaContent	0.02±0.00	0.00±0.00	0.00±0.00	0.01±0.00	11.75	0.21±0.01	0.38±0.01	0.11±0.01	0.23±0.01	9.50	0.79±0.01	0.39±0.02	6.50	9.25
TableStatistics	0.37±0.03	0.00±0.00	0.00±0.00	0.12±0.01	11.25	0.28±0.02	0.51±0.01	0.21±0.01	0.33±0.01	5.25	0.64±0.01	0.31±0.03	10.50	9.00
StatisticalSummary	0.80±0.01	0.01±0.01	0.01±0.01	0.27±0.01	4.25	0.33±0.03	0.59±0.02	0.26±0.02	0.39±0.02	3.75	0.91±0.01	0.41±0.04	4.00	4.00
MatrixFactorization	0.69±0.02	0.00±0.00	0.01±0.01	0.23±0.01	7.50	0.31±0.02	0.51±0.01	0.15±0.02	0.32±0.01	6.25	0.74±0.02	0.37±0.03	7.50	7.08
HashingText	0.83±0.01	**0.23±0.01**	**0.25±0.01**	**0.44±0.01**	1.50	0.32±0.02	0.62±0.01	0.26±0.01	0.40±0.01	3.25	**1.00±0.00**	**0.99±0.01**	1.00	1.92
LLMText	**0.93±0.01**	0.01±0.01	0.01±0.01	0.32±0.01	2.25	**0.41±0.04**	**0.74±0.03**	**0.39±0.04**	**0.51±0.03**	1.00	0.89±0.01	0.70±0.04	2.50	1.92
TableVectorizer	0.51±0.02	0.01±0.00	0.01±0.00	0.17±0.01	7.00	0.20±0.01	0.28±0.01	0.07±0.01	0.18±0.01	11.75	0.70±0.02	0.18±0.03	10.00	9.58
HyTREL	0.72±0.02	0.00±0.00	0.00±0.00	0.24±0.01	6.75	0.26±0.01	0.44±0.02	0.15±0.02	0.28±0.01	6.75	0.85±0.01	0.70±0.02	3.50	5.67
Armadillo	0.71±0.02	0.07±0.01	0.08±0.01	0.29±0.01	3.50	0.23±0.01	0.37±0.02	0.15±0.01	0.25±0.01	8.00	0.84±0.01	0.63±0.03	4.50	5.33
TabPFN	0.48±0.02	0.00±0.00	0.00±0.00	0.16±0.01	9.00	0.19±0.01	0.32±0.01	0.08±0.01	0.20±0.01	11.25	0.66±0.01	0.32±0.03	9.50	9.92
ConTextTab	0.54±0.01	0.00±0.00	0.00±0.00	0.18±0.00	8.75	0.22±0.01	0.36±0.02	0.13±0.01	0.24±0.01	9.25	0.64±0.02	0.36±0.03	10.00	9.33

Stat labeling shows that retrieval and linear probing for majority-type labels are near-perfect, leaving triplet and clustering to separate methods. Under the Stat labeling scheme (Table 9), retrieval and linear probe are near-saturated for many methods, indicating that coarse statistical type signatures are easy to identify under the current partition budget. Therefore, triplet ranking and clustering become the primary discriminators in this regime. HashingSchema dominates overall, consistent with type- and schema-token features being sufficient for coarse statistical grouping. LLMText remains competitive, especially in triplet ranking, while content- and factorization-based embedding models vary more widely, reflecting that some methods preserve coarse type separability without forming clean global cluster geometry.

Table 9: D2 metrics for synthetic data - Stat labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Stat)					Clustering (Stat)					Retrieval/Probe (Stat)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	**0.96±0.01**	**0.83±0.03**	**0.83±0.03**	**0.87±0.02**	1.00	**0.96±0.01**	**0.55±0.03**	**0.42±0.04**	**0.65±0.02**	1.00	**1.00±0.00**	**1.00±0.00**	1.00	1.00
SchemaContent	0.02±0.00	0.00±0.00	0.00±0.00	0.01±0.00	12.00	0.78±0.04	0.04±0.04	0.04±0.02	0.29±0.03	10.50	**1.00±0.00**	**0.99±0.01**	3.00	8.50
TableStatistics	0.25±0.02	0.07±0.02	0.07±0.02	0.13±0.02	9.00	0.79±0.03	0.19±0.02	-0.02±0.04	0.32±0.02	8.75	**0.99±0.00**	**1.00±0.00**	4.00	7.25
StatisticalSummary	0.86±0.03	0.48±0.03	0.52±0.06	0.62±0.03	3.00	0.93±0.01	0.44±0.05	0.40±0.07	0.59±0.04	3.75	0.99±0.00	**0.99±0.00**	6.00	4.25
MatrixFactorization	0.44±0.03	0.09±0.02	0.09±0.02	0.21±0.02	6.75	0.77±0.02	0.15±0.02	-0.03±0.03	0.29±0.01	10.50	0.94±0.04	0.92±0.02	11.00	9.42
HashingText	0.54±0.02	0.13±0.02	0.23±0.04	0.30±0.02	4.50	0.95±0.01	0.50±0.03	0.41±0.05	0.62±0.02	2.50	**1.00±0.00**	0.97±0.01	5.00	4.00
LLMText	0.94±0.01	0.69±0.03	0.69±0.03	0.77±0.02	2.00	**0.96±0.01**	0.52±0.05	0.37±0.09	0.62±0.05	2.75	**1.00±0.00**	**0.99±0.00**	3.50	2.75
TableVectorizer	0.40±0.02	0.05±0.02	0.07±0.02	0.18±0.01	8.75	0.77±0.03	0.08±0.02	-0.01±0.02	0.28±0.01	11.00	0.98±0.00	0.92±0.02	10.50	10.08
HyTREL	0.57±0.03	0.05±0.01	0.05±0.01	0.23±0.01	7.25	0.84±0.03	0.19±0.04	0.16±0.05	0.40±0.03	6.50	**0.99±0.00**	0.98±0.01	5.50	6.42
Armadillo	0.61±0.02	0.10±0.02	0.11±0.02	0.28±0.02	4.75	0.86±0.02	0.23±0.02	0.18±0.02	0.42±0.01	5.00	0.98±0.00	0.95±0.01	8.50	6.08
TabPFN	0.41±0.03	0.02±0.00	0.02±0.00	0.15±0.01	10.25	0.80±0.02	0.11±0.02	0.17±0.02	0.36±0.01	7.75	0.97±0.00	0.86±0.02	11.50	9.83
ConTextTab	0.47±0.02	0.02±0.01	0.05±0.02	0.18±0.01	8.75	0.81±0.04	0.14±0.07	0.04±0.05	0.33±0.05	8.00	0.98±0.00	0.95±0.01	8.50	8.42

Averaging across labels hides specialization, but preserves the big picture. Averaging over label types (Table 10) clarifies why no single method consistently performs best across D2 proxy tasks. LLMText achieves the best overall D2 ranking due to consistently strong clustering under the Semantic labeling scheme, while HashingText is the best on retrieval, linear probe and remains most reliable for Direct-style identity ranking under hard negatives. In contrast, HashingSchema ranks best on average triplet performance but is comparatively weaker on retrieval and linear probe, highlighting that schema fingerprints can match semantic groupings without yielding robust instance-level neighborhoods. This decomposition explains the main-table D2 average scores, where different labeling schemes reflect different notions of table similarity, and Semantic+Difficulty specifically reveals where schema features stop being sufficient.

Table 10: Detailed D2 metrics for synthetic data - Averaged over label types (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Avg over Labels)					Clustering (Avg over Labels)					Retrieval/Probe (Avg over Labels)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	**0.93±0.01**	**0.33±0.02**	**0.34±0.02**	**0.53±0.01**	1.25	0.57±0.02	0.67±0.02	0.40±0.03	0.54±0.02	2.00	0.72±0.01	0.61±0.01	7.00	3.42
SchemaContent	0.02±0.00	0.00±0.00	0.00±0.00	0.01±0.00	12.00	0.37±0.02	0.30±0.02	0.10±0.01	0.26±0.01	10.25	0.78±0.01	0.59±0.02	6.00	9.42
TableStatistics	0.35±0.03	0.02±0.00	0.02±0.00	0.13±0.01	10.00	0.42±0.02	0.45±0.01	0.17±0.02	0.34±0.01	5.25	0.71±0.01	0.51±0.02	10.00	8.42
StatisticalSummary	0.82±0.02	0.13±0.01	0.14±0.02	0.36±0.01	3.75	0.49±0.02	0.55±0.02	0.30±0.03	0.45±0.03	4.00	0.90±0.01	0.57±0.02	4.50	4.08
MatrixFactorization	0.63±0.02	0.02±0.01	0.03±0.01	0.23±0.01	7.00	0.43±0.02	0.42±0.01	0.11±0.02	0.32±0.01	7.00	0.76±0.03	0.49±0.03	9.00	7.67
HashingText	0.78±0.01	0.29±0.01	0.33±0.02	0.47±0.01	3.00	0.49±0.02	0.59±0.02	0.32±0.02	0.47±0.02	3.00	**1.00±0.00**	**0.99±0.01**	1.00	2.33
LLMText	**0.93±0.01**	0.29±0.02	0.30±0.02	0.51±0.02	2.00	**0.58±0.02**	**0.69±0.03**	**0.41±0.05**	**0.56±0.03**	1.00	0.87±0.01	0.76±0.02	3.00	2.00
TableVectorizer	0.48±0.02	0.02±0.01	0.03±0.01	0.18±0.01	7.50	0.36±0.02	0.24±0.01	0.06±0.01	0.22±0.01	11.75	0.74±0.01	0.38±0.02	10.00	9.75
HyTREL	0.69±0.02	0.02±0.00	0.02±0.01	0.25±0.01	7.00	0.42±0.02	0.39±0.02	0.16±0.02	0.32±0.02	6.50	0.84±0.01	0.74±0.01	4.00	5.83
Armadillo	0.71±0.02	0.11±0.01	0.12±0.01	0.31±0.01	5.00	0.40±0.01	0.35±0.02	0.16±0.02	0.30±0.01	7.75	0.88±0.01	0.74±0.02	3.50	5.42
TabPFN	0.46±0.02	0.01±0.00	0.01±0.00	0.16±0.01	10.25	0.36±0.01	0.28±0.01	0.11±0.01	0.25±0.01	10.75	0.72±0.01	0.49±0.03	10.00	10.33
ConTextTab	0.53±0.01	0.01±0.00	0.01±0.01	0.18±0.00	9.25	0.38±0.02	0.32±0.03	0.12±0.02	0.27±0.02	8.75	0.71±0.01	0.57±0.02	10.00	9.33

A.2.2 REAL DATA RESULTS

Across open-source corpora, D2 scores remain strongly label-dependent and separate methods by *what notion of similarity they preserve*: instance identity (Direct), benchmark/task semantics (Semantic), or coarse type signatures (Stat). We therefore report label-wise D2 tables below and the averaged summary (Table 14) to make the specialization pattern explicit rather than relying on a single aggregated number.

Direct labeling shows partial collapse for hard and cluster-hard negatives rather than the complete collapse. Under Direct labeling scheme (Table 11), several methods attain non-trivial performance on TR-H and TR-CH, indicating that the identity task is challenging but not degenerate for open-source tables. Overall, LLMText attains the strongest TR-Avg, while HashingText achieves the best retrieval and linear-probe performance, consistent with content-aware signals being the most reliable for separating confusable tables. Schema-driven fingerprints (e.g., HashingSchema) remain competitive and substantially outperform purely statistical baselines under hard negatives, but still trail the content/serialization methods, suggesting that schema features alone cannot fully resolve similar tables that share column types and structural templates. Model-pooled embeddings (TabPFN, ConTextTab) remain weak on hard negatives and clustering, indicating that pooling yields coarse neighborhoods that are insufficient for strict identity ranking even when some separability exists.

Table 11: D2 metrics for real data - Direct labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Direct)					Clustering (Direct)					Retrieval/Probe (Direct)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	0.96±0.00	0.26±0.02	0.27±0.02	0.49±0.01	3.00	0.11±0.01	0.49±0.01	0.05±0.01	0.22±0.01	4.25	0.93±0.01	0.74±0.04	2.50	3.25
SchemaContent	0.32±0.05	0.00±0.00	0.00±0.00	0.11±0.02	11.25	0.08±0.00	0.32±0.02	0.02±0.00	0.14±0.01	10.00	0.66±0.01	0.73±0.03	5.00	8.75
TableStatistics	0.27±0.03	0.00±0.00	0.00±0.00	0.09±0.01	11.75	0.11±0.01	0.44±0.02	0.06±0.01	0.20±0.01	4.50	0.59±0.02	0.32±0.02	8.00	8.08
StatisticalSummary	0.81±0.01	0.06±0.01	0.07±0.01	0.32±0.01	5.00	0.11±0.01	0.44±0.01	0.06±0.01	0.20±0.01	4.25	0.74±0.02	0.30±0.02	7.50	5.58
MatrixFactorization	0.69±0.02	0.00±0.00	0.00±0.00	0.23±0.01	9.00	0.10±0.01	0.37±0.01	0.04±0.00	0.17±0.01	9.00	0.44±0.03	0.08±0.01	10.50	9.50
HashingText	0.96±0.01	0.34±0.04	**0.37±0.04**	0.56±0.03	2.00	0.12±0.01	0.50±0.02	0.06±0.01	0.23±0.01	2.50	**0.95±0.01**	**0.79±0.02**	1.50	2.00
LLMText	**0.98±0.00**	**0.36±0.05**	**0.37±0.05**	**0.57±0.03**	1.00	**0.13±0.01**	**0.60±0.01**	**0.11±0.01**	**0.28±0.01**	1.00	**0.95±0.01**	0.71±0.05	2.50	1.50
TableVectorizer	0.52±0.01	0.03±0.00	0.03±0.00	0.19±0.01	8.00	0.07±0.00	0.28±0.01	0.02±0.00	0.12±0.01	11.25	0.33±0.01	0.08±0.01	12.00	10.42
HyTREL	0.79±0.01	0.02±0.01	0.03±0.01	0.28±0.01	7.00	0.11±0.01	0.42±0.01	0.06±0.00	0.19±0.01	5.75	0.74±0.02	0.59±0.03	4.50	5.75
Armadillo	0.83±0.01	0.18±0.02	0.20±0.02	0.40±0.02	4.00	0.11±0.01	0.40±0.02	0.05±0.01	0.19±0.01	5.75	0.74±0.02	0.54±0.03	5.50	5.08
TabPFN	0.42±0.02	0.01±0.00	0.01±0.00	0.14±0.01	9.50	0.07±0.01	0.26±0.02	0.02±0.01	0.12±0.01	11.75	0.37±0.02	0.21±0.02	10.50	10.38
ConTextTab	0.71±0.02	0.03±0.01	0.03±0.01	0.26±0.01	6.50	0.10±0.01	0.39±0.01	0.05±0.00	0.18±0.01	8.00	0.57±0.02	0.45±0.03	8.00	7.50

Semantic labeling highlights the global grouping strength of serialization-based text embeddings for benchmark and task structure For Semantic labeling (Table 12), LLMText is the clear winner on both triplet ranking and clustering alignment, indicating that semantic encoders best preserve the global grouping induced by benchmark and task semantics. HashingText remains highly competitive and achieves near-top retrieval and linear probe, while schema-oriented methods (HashingSchema) retain strong triplet performance but weaker clustering, consistent with capturing coarse correlates of benchmark structure without producing globally coherent clusters.

Stat labeling shows that retrieval and linear probing for majority-type labels are near-perfect, with schema features dominating the remaining signal. For the Stat labeling scheme (Table 13), retrieval and linear probe accuracy are near-saturated for many methods, so the more discriminative signal comes from triplet and clustering. In this regime, HashingSchema dominates overall, consistent with majority-type grouping being largely determined by column-type and schema tokens, while LLMText remains the strongest non-schema alternative.

Table 12: D2 metrics for real data - Semantic labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Semantic)					Clustering (Semantic)					Retrieval/Probe (Semantic)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	0.96±0.00	0.43±0.02	0.45±0.02	0.61±0.01	2.75	0.20±0.01	0.52±0.01	0.09±0.01	0.27±0.01	4.25	0.98±0.00	0.94±0.02	3.50	3.50
SchemaContent	0.32±0.05	0.00±0.00	0.00±0.00	0.11±0.02	11.12	0.13±0.01	0.32±0.02	0.04±0.01	0.16±0.01	10.00	0.79±0.01	0.95±0.01	5.00	8.71
TableStatistics	0.27±0.02	0.00±0.00	0.00±0.00	0.09±0.01	11.88	0.17±0.01	0.46±0.02	0.10±0.01	0.25±0.01	5.00	0.72±0.01	0.45±0.04	8.00	8.29
StatisticalSummary	0.81±0.01	0.12±0.01	0.13±0.01	0.35±0.01	5.00	0.20±0.01	0.46±0.01	0.11±0.01	0.26±0.01	3.25	0.83±0.01	0.44±0.03	7.00	5.08
MatrixFactorization	0.68±0.02	0.00±0.00	0.00±0.00	0.23±0.01	9.00	0.16±0.01	0.37±0.01	0.06±0.01	0.20±0.01	9.00	0.54±0.03	0.13±0.01	10.50	9.50
HashingText	0.96±0.01	0.59±0.02	0.62±0.02	0.72±0.01	2.25	0.23±0.02	0.54±0.02	0.10±0.02	0.29±0.02	2.50	**0.98±0.00**	**0.98±0.01**	1.50	2.08
LLMText	**0.99±0.00**	**0.67±0.03**	**0.69±0.03**	**0.78±0.02**	1.00	**0.24±0.01**	**0.64±0.01**	**0.18±0.02**	**0.35±0.01**	1.00	**0.99±0.00**	0.96±0.01	1.50	1.17
TableVectorizer	0.52±0.01	0.04±0.01	0.05±0.01	0.20±0.01	8.25	0.12±0.01	0.27±0.02	0.04±0.01	0.14±0.01	11.00	0.44±0.01	0.11±0.02	12.00	10.42
HyTREL	0.80±0.02	0.04±0.01	0.05±0.01	0.30±0.01	6.75	0.18±0.02	0.43±0.02	0.09±0.01	0.23±0.01	5.75	0.84±0.02	0.79±0.02	4.50	5.67
Armadillo	0.83±0.01	0.27±0.02	0.29±0.02	0.47±0.02	4.00	0.19±0.02	0.40±0.02	0.09±0.02	0.23±0.02	6.50	0.83±0.02	0.72±0.02	6.00	5.50
TabPFN	0.41±0.02	0.01±0.00	0.01±0.00	0.14±0.01	9.50	0.11±0.01	0.27±0.02	0.03±0.01	0.14±0.01	12.00	0.47±0.02	0.29±0.03	10.50	10.67
ConTextTab	0.72±0.02	0.04±0.01	0.05±0.01	0.27±0.01	6.50	0.16±0.01	0.41±0.02	0.08±0.01	0.21±0.01	7.75	0.69±0.02	0.68±0.02	8.00	7.42

Table 13: D2 metrics for real data - Stat labeling (mean ± 95% CI over 10 seeds)

Embedder	Triplet (Stat)					Clustering (Stat)					Retrieval/Probe (Stat)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	**0.74±0.02**	**0.50±0.02**	**0.52±0.03**	**0.59±0.02**	1.00	**0.85±0.02**	**0.40±0.06**	**0.49±0.06**	**0.58±0.05**	1.00	**1.00±0.00**	0.97±0.01	2.00	1.33
SchemaContent	0.22±0.03	0.00±0.01	0.01±0.01	0.08±0.01	10.00	0.59±0.04	0.03±0.02	-0.02±0.01	0.20±0.01	11.25	**1.00±0.00**	0.98±0.01	2.00	7.75
TableStatistics	0.15±0.01	0.00±0.00	0.00±0.00	0.05±0.00	12.00	0.63±0.03	0.06±0.03	0.01±0.02	0.23±0.02	8.00	0.98±0.00	**0.99±0.00**	4.50	8.17
StatisticalSummary	0.65±0.03	0.06±0.02	0.06±0.02	0.26±0.02	3.50	0.76±0.02	0.21±0.03	0.26±0.04	0.41±0.03	3.25	0.98±0.00	0.90±0.02	7.00	4.58
MatrixFactorization	0.46±0.01	0.00±0.00	0.00±0.00	0.15±0.00	9.25	0.58±0.02	0.01±0.01	0.01±0.01	0.20±0.01	10.50	0.96±0.01	0.60±0.02	11.50	10.42
HashingText	0.55±0.03	0.06±0.02	0.07±0.03	0.23±0.02	3.50	0.73±0.04	0.23±0.06	0.21±0.08	0.39±0.06	3.75	0.99±0.00	0.93±0.02	5.50	4.25
LLMText	0.67±0.02	0.09±0.02	0.10±0.03	0.29±0.02	2.00	0.80±0.03	0.31±0.05	0.37±0.08	0.49±0.06	2.00	**0.99±0.00**	0.95±0.01	4.00	2.67
TableVectorizer	0.42±0.01	0.01±0.00	0.01±0.00	0.15±0.00	7.75	0.65±0.02	0.08±0.02	0.08±0.03	0.27±0.02	6.25	0.95±0.01	0.69±0.03	11.50	8.50
HyTREL	0.46±0.01	0.01±0.01	0.01±0.01	0.16±0.01	7.50	0.61±0.03	0.04±0.03	-0.00±0.02	0.21±0.01	9.50	0.98±0.00	0.93±0.02	6.50	7.83
Armadillo	0.50±0.00	0.02±0.00	0.02±0.00	0.18±0.00	5.75	0.61±0.04	0.01±0.01	0.00±0.01	0.21±0.01	10.50	0.97±0.00	0.81±0.02	9.00	8.42
TabPFN	0.38±0.03	0.00±0.00	0.00±0.00	0.13±0.01	10.00	0.70±0.03	0.11±0.02	0.16±0.04	0.32±0.03	5.00	0.96±0.01	0.78±0.03	10.00	8.33
ConTextTab	0.52±0.01	0.01±0.00	0.02±0.01	0.18±0.01	5.75	0.62±0.05	0.10±0.06	0.04±0.08	0.26±0.06	7.00	0.99±0.00	0.95±0.01	4.50	5.75

Averaging over label types hides specialization, but still provides useful intuition. When averaged over label types (Table 14), HashingSchema ranks best on average triplet ranking, LLMText ranks best via clustering alignment, and HashingText ranks best on retrieval and probe. This decomposition matches the interpretation in the main paper, as different labeling schemes lead to different similarity notions, so a single averaged D2 score can hide meaningful specialization.

Table 14: Detailed D2 metrics for real data - Averaged over label types (mean ± 95% CI)

Embedder	Triplet (Avg over Labels)					Clustering (Avg over Labels)					Retrieval/Probe (Avg over Labels)			Overall
	TR-R	TR-H	TR-CH	TR-Avg	Rank	Purity	NMI	ARI	CL-Avg	Rank	R@5	LP	Rank	Avg Rank
HashingSchema	**0.89±0.01**	**0.40±0.02**	**0.41±0.02**	**0.57±0.02**	1.00	0.39±0.01	0.47±0.03	0.21±0.03	0.36±0.02	2.00	**0.97±0.00**	0.88±0.02	3.00	2.00
SchemaContent	0.29±0.05	0.00±0.00	0.00±0.00	0.10±0.02	10.50	0.27±0.02	0.22±0.02	0.02±0.01	0.17±0.01	11.50	0.81±0.01	0.89±0.02	4.50	8.83
TableStatistics	0.23±0.02	0.00±0.00	0.00±0.00	0.08±0.01	12.00	0.30±0.02	0.32±0.02	0.06±0.01	0.23±0.01	5.25	0.76±0.01	0.59±0.02	8.00	8.42
StatisticalSummary	0.76±0.02	0.08±0.01	0.08±0.02	0.31±0.01	4.75	0.36±0.01	0.37±0.02	0.14±0.02	0.29±0.02	3.75	0.85±0.01	0.55±0.02	7.00	5.17
MatrixFactorization	0.61±0.01	0.00±0.00	0.00±0.00	0.20±0.00	9.50	0.28±0.01	0.25±0.01	0.04±0.01	0.19±0.01	10.00	0.65±0.02	0.27±0.01	11.00	10.17
HashingText	0.82±0.01	0.33±0.03	0.36±0.03	0.50±0.02	3.00	0.36±0.02	0.42±0.03	0.12±0.04	0.30±0.03	3.25	**0.97±0.01**	**0.90±0.02**	1.50	2.58
LLMText	**0.88±0.01**	0.37±0.03	0.39±0.04	0.55±0.02	2.00	**0.39±0.02**	**0.52±0.03**	**0.22±0.04**	**0.38±0.03**	1.00	**0.98±0.00**	0.87±0.02	2.50	1.83
TableVectorizer	0.49±0.01	0.03±0.00	0.03±0.00	0.18±0.01	8.00	0.28±0.01	0.21±0.02	0.05±0.01	0.18±0.01	11.00	0.57±0.01	0.29±0.02	11.50	10.17
HyTREL	0.68±0.01	0.03±0.01	0.03±0.01	0.25±0.01	7.00	0.30±0.02	0.29±0.02	0.05±0.01	0.21±0.01	7.25	0.85±0.01	0.77±0.02	4.50	6.25
Armadillo	0.72±0.01	0.16±0.02	0.17±0.02	0.35±0.01	4.25	0.30±0.02	0.27±0.02	0.05±0.01	0.21±0.01	7.75	0.85±0.01	0.69±0.02	6.50	6.17
TabPFN	0.40±0.02	0.01±0.00	0.01±0.00	0.14±0.01	9.50	0.30±0.02	0.21±0.02	0.07±0.02	0.19±0.02	8.25	0.60±0.02	0.43±0.02	10.50	9.42
ConTextTab	0.65±0.02	0.03±0.01	0.03±0.01	0.24±0.01	6.50	0.30±0.02	0.30±0.03	0.05±0.03	0.22±0.03	7.00	0.75±0.01	0.69±0.02	7.50	7.00

A.3 TSNE PLOTS

We provide t-SNE projections of partition embeddings as a qualitative sanity check for the label-wise D2 decomposition (Appendix A.2). Each row corresponds to an embedding model, and columns show the same embeddings colored by different labeling schemes. We emphasize that t-SNE is not metric-faithful; global distances are distorted and apparent cluster sizes are not comparable across methods. Nevertheless, label-homogeneous neighborhoods and visually coherent group structures often mirror the separability trends measured by triplet and clustering scores in the per-label D2 tables.

Synthetic. For synthetic corpora (Figure 4), the t-SNE projections qualitatively mirror the label-type dependence observed in the D2 decomposition (Appendix A.2). Under Direct labeling, methods that sustain hard-negative identity ranking (notably HashingText) exhibit more locally label-consistent neighborhoods, whereas pooled TFM representations (e.g., TabPFN, ConTextTab) form mixed-color manifolds with limited fine-grained separation. Under Semantic labeling, schema- and text-driven methods yield clearer group structure, aligning with their higher clustering scores. Crucially, moving from Semantic to Semantic+Difficulty visibly reduces separability for schema-centric fingerprints (e.g., HashingSchema), consistent with the Semantic+Difficulty triplet degradation in Table 8; this supports the interpretation that the added difficulty factor is weakly expressed in schema tokens and

is instead reflected in value distributions and interaction patterns. We therefore treat t-SNE strictly as a qualitative diagnostic to support the quantitative D2 conclusions reported in Tables 6–8.

Real open-source. For open-source tables (Figure 5), the qualitative structure mirrors the D2 label dependence in Tables 11–13. Under Stat labeling, many methods exhibit visually separated regions, consistent with the relative ease of coarse type grouping. Under Direct labels, most embedding models show substantial color mixing, consistent with stricter identity ranking remaining challenging. Triplet ranking under hard and cluster-hard negatives exposes near-collisions even when retrieval and linear probe can be high. Semantic labels typically sit between these extremes, with semantic- and content-driven methods showing clearer group structure than purely schema- and statistical baselines.

A.4 Large language model (LLM) usage disclosure

We used language models (mainly ChatGPT from OpenAI) to improve grammar and clarity of the manuscript and to suggest code refactorings during implementation. All benchmark design decisions, experiments, results, and conclusions are the authors' own; all LLM outputs were reviewed and validated, and the authors take full responsibility for the content.

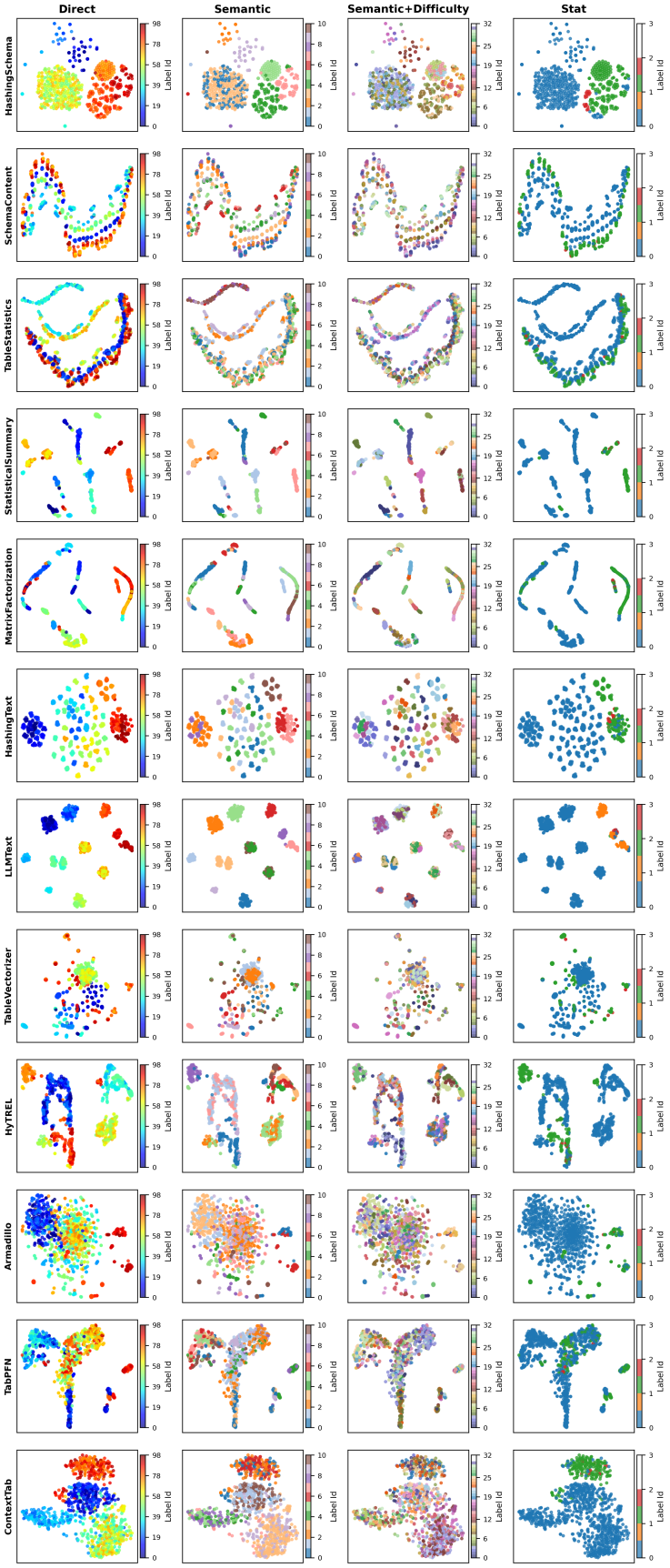

Figure 4: TSNE plots of the synthetic data with different labels

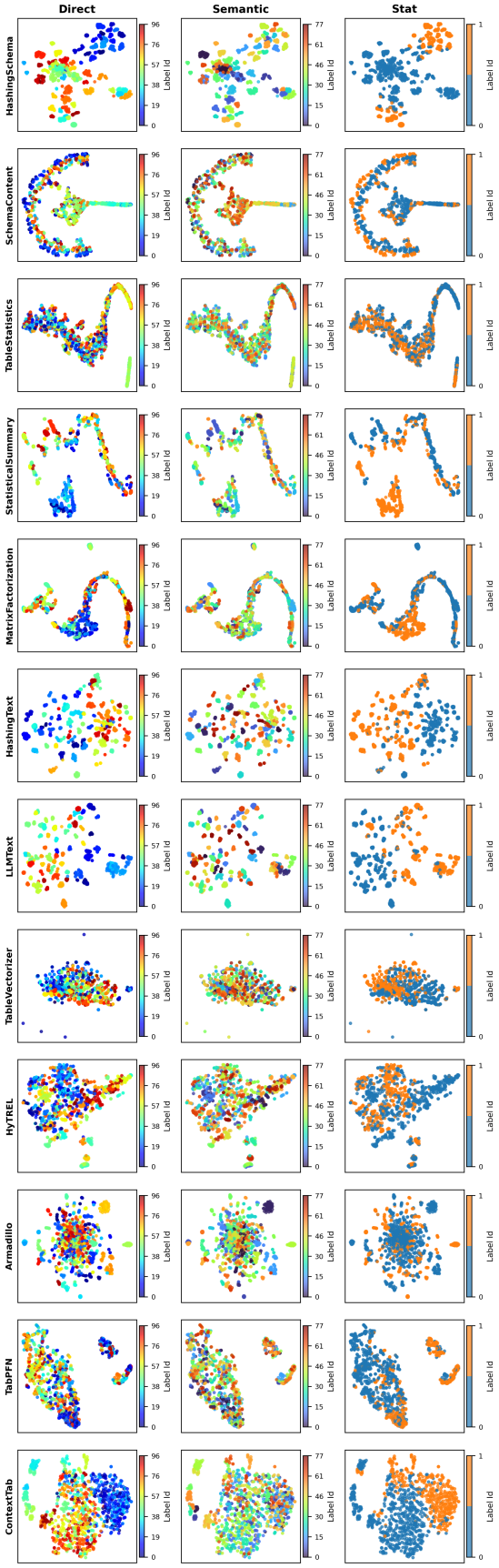

Figure 5: TSNE plots of the real data with different labels