Can a single tabular embedding model service different tasks?

Niharika S. D'Souza
IBM Silicon Valley Labs
San Jose, California
Niharika.Dsouza@ibm.com

Liane Vogel
Technical University of Darmstadt
Germany

Kavitha Srinivas
IBM Yorktown Heights
Yorktown Heights, New York

Sola Shirai
IBM Yorktown Heights
Yorktown Heights, New York

Oktie Hassanzadeh
IBM Yorktown Heights
Yorktown Heights, New York

Horst Samulowitz
IBM Yorktown Heights
Yorktown Heights, New York

ABSTRACT

Universal text embedding models demonstrate that a single pretrained model can produce representations effective across tasks such as classification, clustering, and retrieval. In contrast, existing tabular foundation models are largely task-specific. In this work, we investigate whether a **single tabular embedding model** can generalize effectively across tasks. We propose an initial approach that first aligns heterogeneous table-cell representations into a shared space using *Hirschfeld–Gebelein–Rényi maximal correlation* (HGR), enabling numerical and non-numerical cells co-occurring in the same row to be mapped consistently. We then perform message passing with All-Set Transformer modules within a Hypergraph Transformer architecture to preserve row and column permutation invariance. Both stages are trained using self-supervised objectives to learn consistent representations at multiple granularities, including the cell, row, column, and table levels. Without additional training on new datasets, the model produces table embeddings that generalize across tasks. We evaluate the approach on row similarity, column similarity, cell semantic retrieval, and predictive machine learning tasks, and find that it generalizes competitively compared to models specifically designed for individual tasks.

VLDB Workshop Reference Format:
Niharika S. D'Souza, Liane Vogel, Kavitha Srinivas, Sola Shirai, Oktie Hassanzadeh, and Horst Samulowitz. Can a single tabular embedding model service different tasks?. VLDB 2026 Workshop: Tabular Data Analysis (TaDA).

1 INTRODUCTION

The power of universal language-model-based text embedding models lies in their ability to use a single pretrained encoder across many downstream tasks. For example, the widely used MTEB benchmark [24] evaluates embedding models on a diverse collection of tasks, including text classification, document clustering, and information retrieval. In contrast, the emerging field of tabular foundation models has so far produced models that are largely specialized for individual tasks.

Proceedings of the VLDB Endowment. ISSN 2150-8097.

Limitations of current tabular foundation models. Recent work [15, 17, 25] has demonstrated that large-scale pretraining on diverse tabular datasets can substantially improve tabular machine learning performance. However, these models are primarily designed for supervised prediction rather than for learning reusable, task-agnostic embeddings. Likewise, application-specific approaches for problems such as table retrieval in data lakes [16] or table question answering [14] do not produce universal row or column representations that transfer across tasks. This motivates the central question of our work: can we build a **single tabular embedding model** that generalizes across a broad range of tabular tasks, including both table understanding and predictive modeling tasks such as classification and regression?

Challenges of tabular data. Tabular data is inherently heterogeneous. A single table may contain numerical, categorical, ordinal, temporal, and free-text features simultaneously. In addition, tables exhibit strong semantic heterogeneity: column names and categorical values are highly context dependent, often abbreviated, and may vary substantially across domains. Such diversity makes it difficult to learn representations that generalize across tables and tasks, and is one reason why existing tabular foundation models remain largely task-specific. A further challenge arises from the tension between modeling tables as collections of numerical feature values with well-defined interactions, and modeling the semantic information encoded in column names, categorical labels, and textual entries. For predictive tasks, methods that focus primarily on numerical feature values [15] have achieved strong performance, suggesting that rich semantic modeling is not always necessary for classification and regression. However, purely numerical representations discard information essential for broader table understanding tasks such as entity matching or column similarity detection.

Limitations of applying text embedding models to tables. Language-model-based embedding models are designed as universal text encoders that map text from diverse domains into representations useful across many tasks. These models benefit from the inherently sequential nature of language, where token order provides strong syntactic and semantic inductive biases. Tables, however, differ fundamentally from text: they consist of heterogeneous rows and columns that do not possess a natural linear ordering. Naively linearizing tables into text can obscure relational structure, introduce spurious dependencies, and ignore feature-specific inductive biases. Moreover, mixed data types do not naturally align with text tokenization. Numerical values may be fragmented into multiple tokens, increasing computational cost while weakening the

representation of quantitative relationships. As a result, directly applying language models to tabular data is often insufficient, motivating architectures that explicitly model tabular structure and heterogeneity.

Towards universal tabular embeddings. We investigate whether a single embedding model can support both table understanding tasks—such as identifying similar rows or columns—and predictive tasks, where successful approaches often treat tables primarily as numerical feature collections. To build such a representation, we address two key challenges. First, we learn an alignment between numerical and textual cells within the same row of a table, encouraging semantically related cells that co-occur in a row to occupy nearby regions of the embedding space. Second, we design aggregation mechanisms that operate across multiple granularities, enabling the same cell-level representations to be composed into row embeddings for tasks such as entity matching and anomaly detection, and column embeddings for tasks such as schema matching and joinability detection. We evaluate the resulting embeddings on four core tasks that underlie many real-world tabular applications: row similarity search, column similarity search, cell semantic retrival, and predictive modeling. Our results suggest that a single embedding model can generalize competitively across diverse settings and support multiple tasks simultaneously.

1.1 Related Work

LLM-Based Table Models. Recent work has explored the use of Large Language Models (LLMs) for tabular data by serializing table rows into sequences of language tokens [9, 13]. For example, TabuLA 8B [12] fine-tunes Llama 3–8B on serialized tables and achieves competitive performance with tree-based methods in few-shot settings. However, such approaches are constrained by the context window limitations of LLMs, which restrict the size of tables that can be processed effectively. In addition, LLMs may have been pretrained on widely used benchmark datasets, raising concerns about evaluation leakage and fairness in tabular benchmarks [3].

Other approaches, such as Hytrel [4], avoid row serialization by preserving the two-dimensional grid structure of tables. However, these models still encode numerical values as text, which can be problematic because language models are known to struggle with numerical reasoning and quantitative representation [31].

In-Context Learning for Tabular Data. At the other end of the spectrum are models such as TabPFN [15], MITRA [36], and TabICL [25], which treat tables primarily as numerical feature collections, similar to traditional boosting approaches [5, 11]. These models are trained largely on synthetically generated numerical datasets, with categorical variables represented through indexing schemes.

Although synthetic data enables scalable and highly diverse pretraining, it omits many semantically meaningful signals present in real-world tables, including descriptive column names, categorical labels, free-text entries, and structured types such as dates and timestamps. As a result, semantic information cannot be effectively leveraged at inference time, even when it is available. In particular, column names are ignored, while categorical variables are represented through one-hot or ordinal encodings that discard their semantic relationships.

Semantic-Aware Models. Hybrid approaches such as SAP-RPT-1-OSS (formerly ContexTab) [29] are better suited to real-world settings where semantic context is informative. These models treat contextual and content information as separate modalities, extending the TabPFN architecture with modality-specific processing followed by fusion through additive combination. While this design performs well for predictive machine learning tasks, additive fusion may obscure subtle distinctions between representations. For example, columns with different names but similar semantics may not be aligned effectively, while near-duplicate rows with small but important differences may become difficult to distinguish.

TARTE [17] addresses this issue through knowledge pretraining that jointly encodes column names and cell values to produce semantically enriched embeddings. However, despite leveraging semantic information, both SAP-RPT-1-OSS and TARTE remain primarily optimized for specific predictive tasks rather than learning general-purpose table embeddings. More broadly, most foundation models for tabular data focus predominantly on predictive machine learning benchmarks, with relatively little attention paid to representation learning at other granularities, such as cell-, row-, column-, or table-level understanding tasks.

1.2 Our Contributions

To address heterogeneity in tabular data, we introduce an alignment step that projects disparate table-cell representations into a shared embedding space. We formulate this alignment using *Hirschfeld–Gebelein–Rényi maximal correlation* (HGR), which aligns numerical and non-numerical cells (e.g., text or categorical values) that co-occur within the same row. This objective preserves the statistical properties of each feature type while capturing shared semantic information across modalities.

To learn universal embeddings for cells, rows, columns, and tables while preserving native row–column structure, we represent each table as a hypergraph. Cells correspond to nodes, while rows, columns, and tables define hyperedges connecting multiple cells simultaneously. A Hypergraph Transformer with All-Set Transformer modules performs message passing over this structure, producing contextualized embeddings while preserving row and column permutation invariance.

Both stages are trained using self-supervised learning: the alignment module with a soft-HGR objective, and the Hypergraph Transformer with a combination of contrastive learning, masked cell modeling and embedding reconstruction. The resulting representations are label-agnostic and task-agnostic, making them transferable across downstream tabular tasks. Our framework, named **HGR-TabE** also naturally supports in-context learning for predictive modeling by formulating it as link prediction in a transductive hypergraph setting. Overall, HGR-TabE is a flexible tabular representation learning framework that supports a broad range of tasks and achieves competitive out-of-the-box performance without additional task-specific fine-tuning.

2 HGR-TABE

Figure 1 provides an overview of the HGR-TabE framework. The model consists of two main stages. First, cells within the same row

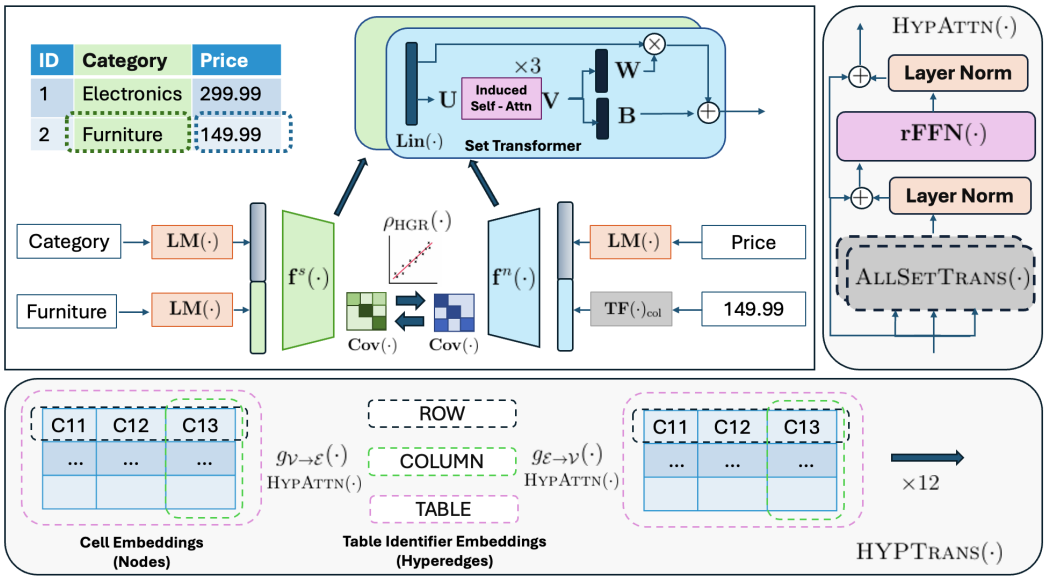

Figure 1: Illustrating our HGR-TabE framework. *(Top Left)* Common-space alignment of input tables. Row-wise cell values are encoded by type—LMs for non-numeric/semantic data (projected via $f^s(\cdot)$) and TabICL-v2's column encoder for numeric data (projected using $f^n(\cdot)$) to maximize HGR correlation. *(Bottom)* A hypergraph transformer produces context-aware embeddings for cells, rows, columns, and table headers. *(Top Right)* The AllSetTransformer architecture that enforces structural invariance to row and column permutations.

are projected into a shared embedding space through a common-space alignment module trained with a soft-HGR objective. This step aligns numerical and non-numerical representations pairwise, enabling heterogeneous cell types to become directly comparable within a unified embedding space.

Next, the aligned table is transformed into a hypergraph, where cells correspond to nodes and row, column, and table memberships define the hyperedges. A Hypergraph Transformer then performs message passing over this structure to produce table-aware contextualized embeddings for cells, rows, columns, and entire tables. These representations can subsequently be used across a range of downstream tabular learning tasks.

2.1 Table-Structure-Aware Hypergraph

At the core of HGR-TabE is a hypergraph representation that explicitly captures the relational structure of a table. Formally, a table is represented as $\mathcal{T} = [\mathbf{t}^H, \mathbf{C}, \mathbf{R}]$, where $\mathbf{t}^H$ denotes the table header. Let N and M denote the number of rows and columns, respectively.

Column information is represented as $\mathbf{C} = \{\mathbf{C}^H, \mathbf{C}^C\}$, where $\mathbf{C}^H = [\mathbf{c}_1^H, \mathbf{c}_2^H, \ldots, \mathbf{c}_M^H]$ contains the column headers, and $\mathbf{c}_j^C = [\mathbf{v}_{1j}; \mathbf{v}_{2j}; \ldots; \mathbf{v}_{Nj}]$ contains the cell values associated with column j. Similarly, row information is represented as $\mathbf{R} = \{\mathbf{R}^H, \mathbf{R}^C\}$, where $\mathbf{r}_i^C = [\mathbf{v}_{i1}, \mathbf{v}_{i2}, \ldots, \mathbf{v}_{iM}]$ denotes the cells in row i.

As illustrated in Figure 1 (bottom), the table $\mathcal{T}$ is transformed into a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,, where each cell $\mathbf{v}_{ij} \in \mathcal{V}$ forms a node, and the hyperedges $\mathcal{E} = \{\{\mathcal{R}_i\}, \{C_j\}, \mathcal{T}^H\}$ encode row, column, and table memberships. Each cell node is associated with a node embedding $\tilde{\mathbf{v}}_{ij} \in \mathbb{R}^{D\times 1}$. Column hyperedges are associated with embeddings $\{\tilde{\mathbf{e}}_j^c \in \mathbb{R}^{D\times 1}\}$, row hyperedges with $\{\tilde{\mathbf{e}}_i^r \in \mathbb{R}^{D\times 1}\}$,, and the table hyperedge with $\tilde{\mathbf{e}}^t \in \mathbb{R}^{D\times 1}$. Finally, the hypergraph is represented using a binary incidence matrix $\mathcal{A} \in \{0,1\}^{MN\times(M+N+1)}$,, where $\mathcal{A}_{lm} = 1$ if node l belongs to hyperedge m, and $\mathcal{A}_{lm} = 0$ otherwise, similar to [4].

2.2 Encoding Table Data Primitives

We encode table entries according to their data types to distinguish between numerical and non-numerical information. As shown in Figure 1 (top left), each cell is represented using a compound embedding with two components: a *context embedding* capturing the cell's role in the table, and a *content embedding* capturing its value and distributional properties.

2.2.1 Cell Representations. Each cell embedding ($\mathbf{x}_{ij}$) combines *context* (column header) and *content* to avoid ambiguity. The context of a cell is defined by its column header. We encode column headers using the lightweight sentence embedding model *all-MiniLM-L6-v2* to capture semantic information efficiently.

Non-Numeric Content Embedding. For non-numeric cells, including free-text and categorical values, we use dense semantic embeddings generated with *all-MiniLM-L6-v2* [27, 33]. This preserves semantic relationships between categorical labels and textual values. For example, a value such as "SF" can be interpreted in the context of related entities such as "city" or "NY" in a common representation. The resulting compound embedding for non-numeric cells is $\mathbf{x}_{ij}^s = [\text{LM}(\mathbf{c}_j^H); \text{LM}(\mathbf{v}_{ij})]$.

Numeric Content Embedding. For numerical cells, we adopt the column-wise embedding mechanism from TabICL-v2 [26], which

maps scalar values within a column to distribution-aware representations. Following [25], all columns share a Set Transformer-based embedding network TF_{col}, illustrated in Figure 1 (top left):

$$\mathbf{W}_{\text{TF}}, \mathbf{B}_{\text{TF}} = \text{TF}_{\text{col}}(\{\mathbf{v}_{ij}\}_{i=1}^{N}) \in \mathbb{R}^{N \times d^N},$$
$$\tilde{\mathbf{c}}^n_{:,j} = \mathbf{W}_{\text{TF}} \odot \mathbf{v}_{:,j} + \mathbf{B}_{\text{TF}} \in \mathbb{R}^{N \times d^N},$$

where each cell receives a cell-specific weight and bias. The network treats feature embedding as a set-input problem, using a permutation-invariant Set Transformer to generate column distribution aware parameters from the column values. The column is projected to $d = 128$ dimensions and processed through three induced self-attention blocks (ISABs) before generating $\mathbf{W}_{\text{TF}}$ and $\mathbf{B}_{\text{TF}}$. The final compound embedding for numeric cells is $\mathbf{x}^n_{ij} = [\text{LM}(\mathbf{c}^H_j); \tilde{\mathbf{c}}^n_{ij}]$.

2.2.2 Identifier Representations at the Table, Column, and Row Levels. To maintain consistency during message passing, table-, column-, and row-level identifiers follow the same embedding structure as cells, combining contextual and content information using *all-MiniLM-L6-v2* embeddings. These are projected via an $\text{MLP}_e(\cdot) : \mathcal{R}^{LM} \rightarrow \mathcal{R}^D$ to the same representation space $\{\tilde{\mathbf{e}}^c_j, \tilde{\mathbf{e}}^r_i, \tilde{\mathbf{e}}^t\} \in \mathcal{R}^D$ as the cell embeddings during message passing (hypergraph representation learning).

Column Identifier Embedding. Column hyperedge embeddings are generated from serialized column descriptions of the form:

"`Col Name : Unique Val 1 | Unique Val 2 | ...`".

Let $\mathbf{e}^c_j$ denote the embedding for column j.

Table Identifier Embedding. The table embedding is generated from the table caption. If no caption is available, the column headers are concatenated to form the table description. Let $\mathbf{e}^t$ denote the table embedding.

Row Identifier Embedding. Each row embedding is constructed from the serialized row:

"`| Cell r1 | Cell r2 | . . . | Cell rN |`".

Missing values are left blank during serialization. Let $\mathbf{e}^r_i$ denote the embedding for row i.

2.3 Common-Space Alignment of Table cells across Distinct Data-Types

Recall that the row representations consist of cells with numeric or non-numeric data-types, which we refer to as modalities of the cell data. $\{\mathcal{N}, \mathcal{S}\}$ are the set of columns having numerical and non numerical cell types. Numeric data is encoded using TabICLv2's column transformer embedding concatenated with the column header embedding obtained from *all-miniLM-L6-v2*, giving us a 512 dimensional representation. All non-numeric data is encoded using the cell's *all-miniLM-L6-v2* embedding concatenated with the column header embedding, giving us a 768 dimensional embedding. These embeddings are $\mathbf{x}^n_{ij} \in \mathcal{R}^{512\times 1}, j \in \mathcal{N}$ and $\mathbf{x}^s_{ij} \in \mathcal{R}^{768\times 1}, j \in \mathcal{S}$ respectively.

To obtain a common embedding space for cell representations, we use parallel projection networks $\{\mathbf{f}^n(\cdot), \mathbf{f}^s(\cdot)\}$ on the input embeddings to obtain their latent space representations $\tilde{\mathbf{v}}^n_{ij}$ and $\tilde{\mathbf{v}}^n_{ij}$. We adopt the Hirschfeld–Gebelein–Rényi (HGR) framework [32], which generalizes statistical dependence into abstract, non-linear functional spaces, parameterized using deep neural networks. The HGR maximal correlation is a symmetric bilinear dependence measure obtained by solving the following coupled constrained optimization problem across all rows, $l \in \mathcal{N}, m \in \mathcal{S}$,

$$\sup_{C_E, C_{\text{Cov}}, \forall i} \rho_{\text{HGR}}(\{\mathbf{x}_{i,l}\}, \{\mathbf{x}_{i,m}\}) = \sup_{C_E, C_{\text{Cov}}} \mathbb{E}\Big[[\mathbf{f}^n(\mathbf{x}_{:,l})]^T \mathbf{f}^s(\mathbf{x}_{:,m})\Big] \quad (1)$$

The constraint sets are given by:

$$C_E : \{\mathbb{E}[\mathbf{f}^n(\cdot)] = \mathbb{E}[\mathbf{f}^s(\cdot)] = \mathbf{0}\}$$
$$C_{\text{Cov}} : \{\mathbf{Cov}(\mathbf{f}^n(\mathbf{x}^n)) = \mathbf{Cov}(\mathbf{f}^s(\mathbf{x}^s)) = \mathcal{I}_D\}$$

where $\mathcal{I}_D$ is a $D \times D$ identity matrix. The expectation $\mathbb{E}$ is taken over all pairs of numeric and non-numeric cells belonging to row i.

Methods such as deep CCA [1] can be viewed as special cases of this formulation, in which the whitening constraints are imposed by decomposing the empirical covariance matrix. This is known to suffer from scalability and numerical stability issues for large datasets. For scalable estimation, we use the approximation in [32], which establishes the equivalence offered by the *soft-HGR* (sHGR) loss. Under this relaxation, Eq. (1) can reformulated as an empirical risk minimization problem with the sHGR loss replacing the hard whitening constraint using a matrix trace regularizer.

$$\mathcal{L}_{\text{sHGR}} = -\frac{1}{N_z}\Bigg[\sum_{l\in\mathcal{N}, m\in\mathcal{S}, \forall i} \Big[\mathbb{E}[\mathbf{f}^n(\mathbf{x}_{il})^\top \mathbf{f}^s(\mathbf{x}_{im})] + \frac{1}{2}\text{Tr}[\mathbf{Cov}(\mathbf{f}^n(\mathbf{x}_{:,l})\mathbf{Cov}(\mathbf{f}^s(\mathbf{x}_{:,m})]\Big]\Bigg] \quad (2)$$

Here, $\mathbf{Cov}(\cdot)$ denotes the empirical covariance matrix. Mathematically, the first term is equivalent to computing the expectation (across rows) of the inner product between the average row-wise numerical cell embedding $(\sum_{l\in\mathcal{N}} \mathbf{f}^n(\mathbf{x}_{i,l})/|\mathcal{N}|)$ and the average row-wise non-numerical cell embedding $(\sum_{m\in\mathcal{S}} \mathbf{f}^s(\mathbf{x}_{i,m})/|\mathcal{S}|)$. The second term can be simplified to compute the trace of the product of covariances matrix estimates across the same quantities. This simplification can improve the computational efficiency of the s-HGR loss estimation across large tables. Both transformations $\mathbf{f}^s(\cdot), \mathbf{f}^n(\cdot)$ are parameterized by a cascade of three Induced Self Attention Blocks (ISAB) with number of inducing points as $I = 128$ and dimensionality $d = 512$ and 4 attention heads. $N_z = M(M-1)$ is given by the number of pairs in the batch, which varies for individual table depending on the number of numerical vs text/categorical columns. This design helps us exploit column distributional properties in a memory-efficient fashion. The zero-mean constraints on the functional transformations, $\mathbb{E}[\mathbf{f}^n(\mathbf{x}^n)] = \mathbb{E}[\mathbf{f}^s(\mathbf{x}^s)] = \mathbf{0}$ can be enforced during optimization via step-wise mean subtraction. We also introduce two decoders $\{\mathbf{f}^n_{\text{dec}}(\cdot), \mathbf{f}^s_{\text{dec}}(\cdot)\}$ to reconstruct input representations from the common space $\tilde{\mathbf{v}}^s = \mathbf{f}^s(\mathbf{x}^s), \tilde{\mathbf{v}}^n = \mathbf{f}^n(\mathbf{x}^n)$. With the weighting parameter $\lambda = 0.9$, the common space alignment optimizes the following loss function.

$$\mathcal{L} = \lambda\mathcal{L}_{\text{sHGR}} + (1-\lambda)\mathcal{L}_{MSE} \quad (3)$$

For the subsequent message passing step, the cell embeddings are the $D = 1024$ dimensional representations $\tilde{\mathbf{v}}_{ij} = \mathbf{f}^n(\mathbf{x}_{ij})$ for numerical and $\tilde{\mathbf{v}}_{ij} = \mathbf{f}^s(\mathbf{x}_{ij})$ for non-numerical cells.

2.4 Hypergraph Transformers for Universal Embedding Estimation

Once heterogeneous cell embeddings are aligned into a shared space, we use a Hypergraph Transformer (HGTrans) to jointly model table content, structure, and relationships across table primitives. These primitives include cells, column headers, row identities, and table captions. As shown in Figure 1 (bottom), each HGTrans layer alternates between updating node representations (cells) and hyperedge representations (rows, columns, and tables) using two hypergraph attention blocks (HYPAtt) [4, 6], followed by a fusion step across scales. The initial node representations are the aligned cell embeddings $\mathbf{H}_v^{(0)} = \{\tilde{\mathbf{v}}_{ij}\}$, while the concatenated hyperedge representations are $\mathbf{H}_e^{(0)} = \Big\|\{\{\tilde{\mathbf{e}_j}^c\}, \{\tilde{\mathbf{e}_i}^r\}, \tilde{\mathbf{e}}^t\} = \Big\| \{\{\mathrm{MLP}_e(\mathbf{e}_j{}^c)\}, \{\mathrm{MLP}_e(\mathbf{e}_i^r)\}, \mathrm{MLP}_e(\mathbf{e}^t)\}$. Information is first propagated from nodes to hyperedges using:

$$\mathbf{Z}_{e,:}^{(l+1)} = \mathbf{\Phi}_{\mathrm{R}}\Big(g_{\mathcal{V}\to\mathcal{E}}(\mathcal{A}_{:,e}, \mathbf{H}_v^{(l)})\Big), \tag{4}$$

where $\mathcal{A}_{:,e} = \{v \mid \mathbf{A}_{v,e} = 1\}$ denotes the nodes incident to hyperedge e, $g_{\mathcal{V}\to\mathcal{E}}$ is a hypergraph attention operator, and $\mathbf{\Phi}_{\mathrm{R}}$ is a ReLU activation. This step aggregates information from constituent cells into row-, column-, and table-level representations. The updated hyperedge representations are fused with the previous hyperedge embeddings through an MLP and propagated back to the nodes:

$$\begin{aligned}\tilde{\mathbf{H}}_{v,:}^{(l+1)} &= \mathbf{\Phi}_{\mathrm{R}}\Big[g_{\mathcal{E}\to\mathcal{V}}\Big(\mathcal{A}_{v,:}, \mathbf{\Phi}_D\big(\mathbf{W}_f^l[\mathbf{Z}_{e,:}^{(l+1)}; \mathbf{H}_e^{(l)}]\big)\Big)\Big] \\ \mathbf{H}_{v,:}^{(l+1)} &= \mathrm{LN}\Big(\mathbf{\Phi}_D(\tilde{\mathbf{H}}_{v,:}^{(l+1)} + \mathbf{H}_{v,:}^{(l)})\Big),\end{aligned} \tag{5}$$

where $g_{\mathcal{E}\to\mathcal{V}}$ propagates information from hyperedges back to nodes, and $\mathcal{A}_{v,:} = \{e \mid \mathbf{A}_{v,e} = 1\}$ denotes the hyperedges incident to node v. Here, $\mathbf{\Phi}_D$ denotes dropout regularization, $\mathbf{W}_f \in \mathbb{R}^{D\times 2D}$ is a fusion layer, and LN denotes layer normalization. Residual connections are used throughout to stabilize optimization.

The HYPAtt block (Figure 1, top right) consists of multi-head attention (MHA), a position-wise feed-forward network (rFFN), layer normalization, and residual connections. We adopt the permutation-invariant hypergraph set-attention mechanism from [2], which preserves row and column invariance in tabular data.

$$\begin{aligned} g_{\mathcal{V}\to\mathcal{E}\text{ or }\mathcal{E}\to\mathcal{V}}(\mathbf{I}) &= \mathrm{LN}\left(\tilde{\mathbf{I}} + \mathrm{rFFN}(\tilde{\mathbf{I}})\right), \tilde{\mathbf{I}} = \mathrm{MHA}(\mathbf{I}) \\ &= \mathrm{LN}\left(\boldsymbol{\omega} + \Big\|_{i=1}^{A_h} \mathrm{SoftmaxLR}\left(\boldsymbol{\omega}_i \mathbf{I}\mathbf{W}_i^{K^\top}\right)\mathbf{I}\mathbf{W}_i^V\right),\end{aligned} \tag{6}$$

where $\mathbf{I}$ denotes the input node or hyperedge representations, $\boldsymbol{\omega}$ is a learnable query vector, $\boldsymbol{\omega} = \|_{i=1}^{A_h}\boldsymbol{\omega}_i$, $\mathbf{W}_i^K$ and $\mathbf{W}_i^V$ are key and value projection matrices, and $\|$ denotes concatenation across $A_h = 16$ attention heads. SoftmaxLR$(\cdot)$ applies a LeakyReLU activation ($\alpha = 0.01$) followed by softmax normalization. The rFFN layer in Eq. 2.4 further refines the attended representations. Finally, we stack 12 layers of HGTrans, following architectures similar to [4, 35]. Unlike [4], we explicitly fuse initial and contextualized embeddings at the final layer to preserve both local semantics and global structure.

$$\mathbf{Z}_v^{(L)} = \mathrm{mlp}_{\mathrm{fs}}([\mathbf{H}_v^{(L)}; \mathbf{H}_v^{(0)}]), \quad \mathbf{Z}_e^{(L)} = \mathrm{mlp}_{\mathrm{ft}}([\mathbf{H}_e^{(L)}; \mathbf{H}_e^{(0)}]).$$

This residual fusion MLPs are trained with dropout to mitigate oversmoothing and avoid degeneracy in solutions. Thus, fine-grained cell content (from HGR-aligned embeddings) is retained alongside higher-order relational signals.

2.4.1 Objective Function. HGR-TabE uses a combination of reconstruction and contrastive objectives operating at different granularities of the table. These losses are designed to (i) preserve cell-level semantics, (ii) anchor higher-level representations to meaningful signals, and (iii) enforce invariance across views of the same table. The overall objective combines these components:

$$\mathcal{L} = \mathcal{L}_{\mathrm{cell}} + \lambda_{\mathrm{hyper}}\mathcal{L}_{\mathrm{hyper}} + \lambda_{\mathrm{InfoNCE}}\mathcal{L}_{\mathrm{InfoNCE}}$$

Together, these losses balance local fidelity, global consistency, and cross-view invariance, enabling HGR-TabE to learn representations that generalize across tasks and tables ($\lambda_{\mathrm{hyper}} = 0.5$, $\lambda_{\mathrm{InfoNCE}} = 0.03$)

Cell reconstruction. To ensure that the learned embeddings retain fine-grained content information, we apply masked reconstruction at the cell level. A fraction of cells is randomly masked, and the model is trained to reconstruct their original embeddings. We use separate decoders for numeric and non-numeric cells to respect modality differences. The loss is

$$\mathcal{L}_{\mathrm{cell}} = \frac{n_{\mathrm{num}}\mathcal{L}_{\mathrm{num}} + n_{\mathrm{cat}}\mathcal{L}_{\mathrm{cat}}}{n_{\mathrm{num}} + n_{\mathrm{cat}}}$$

, where $\mathcal{L}_{\mathrm{num}} = \frac{1}{|\mathcal{M}_{\mathrm{num}}|}\sum_{v\in\mathcal{M}_{\mathrm{num}}} \|\mathcal{H}_{\mathrm{num}}(\mathbf{H}_{:,j}^{(L)}) - \mathbf{x}_{:,j}\|_2^2$ ($j \in \mathcal{N}$) and $\mathcal{L}_{\mathrm{cat}} = \frac{1}{|\mathcal{M}_{\mathrm{cat}}|}\sum_{v\in\mathcal{M}_{\mathrm{cat}}} \|\mathcal{H}_{\mathrm{cat}}(\mathbf{H}_{:,j}^{(L)}) - \mathbf{x}_{:,j}\|_2^2$ ($j \in \mathcal{S}$). This objective prevents representation collapse and information loss while encouraging faithful encoding of both numeric distributions and semantic content.

Hyperedge reconstruction. While cell reconstruction preserves local information, it does not directly constrain higher-level representations. To address this, we introduce a hyperedge reconstruction loss

$$\mathcal{L}_{\mathrm{hyper}} = \frac{1}{|\mathcal{E}|}\sum_{e\in\mathcal{E}} \|\mathcal{H}_e(\mathbf{H}_e^{(L)}) - \mathbf{e}\|_2^2$$

, with $\mathbf{e} = \Big\|\{\{\mathbf{e}_j{}^c\}, \{\mathbf{e}_i{}^r\}, \mathbf{e}^t\}$ which anchors row, column, and table embeddings to their initial semantic representations (derived from text encoders). This stabilizes training and ensures that global representations remain interpretable and aligned with input semantics.

Contrastive consistency (InfoNCE). To encourage invariance to stochastic corruption and improve generalization, we apply a contrastive loss on column and table embeddings. Two views (w) of the same table are generated via independent masking and dropout, and corresponding column/table embeddings are treated as positive pairs. The loss is

$$\mathcal{L}_{\mathrm{InfoNCE}} = -\frac{1}{K}\sum_k \log\frac{\exp(\mathbf{z}_k^{(1)}\cdot\mathbf{z}_k^{(2)}/\tau)}{\sum_j \exp(\mathbf{z}_k^{(1)}\cdot\mathbf{z}_j/\tau)}$$

, where $\mathbf{z}_k^{(w)}$ are normalized embeddings. This objective promotes robustness to missing data and encourages consistent representations across different views of the same underlying table.

3 EXPERIMENTS

We train HGR-TabE on tables from the CARTE dataset, which provides rich table structures with semantic text and numerical columns with a total of over a million rows across 50 tables. We evaluate HGR-TabE on four tasks spanning different levels of tabular representations: (i) tabular machine learning (classification/regression), (ii) row similarity search, and (iii) column similarity search (iv)

Task → Metric → Method ↓	Bin. Class. AUC (↑)	Multi. Class. LogL. (↓)	Reg. - RMSE (↓)	Row Sim. MRR (↑)	Col Sim. MRR (↑)	Cell Retr. Acc (↑)
Hytrel	0.50	2.81	38047.99	0.05	0.34	0.30
SAP-RPT-OSS	0.84	0.30	7179.57	0.03	0.08	0.28
all-Mini-LM	0.72	0.51	24117.74	**0.66**	**0.59**	**0.63**
TabICLv2	**0.86**	**0.24**	**6512.21**	0.04	0.15	0.18
TabPFNv2.5	0.84	0.27	6619.37	0.00	-	-
Tabula8b	0.78	0.44	15906.32	0.44	-	-
HGR-TabE	0.74	0.44	21991.60	0.50	0.54	0.30

Table 1: Results aggregated across datasets per task. ↑/↓ indicate whether higher or lower values are better. HGR-TabE achieves balanced performance across predictive and retrieval-oriented tasks, outperforming specialized tabular foundation models on similarity benchmarks while remaining competitive on predictive tasks.

cell semantics. Row embeddings support both predictive modeling and instance-level retrieval (e.g., deduplication), while column embeddings enable schema-level tasks such as joinable column discovery. We compare against tabular foundation models (*Hytrel*, *TabICL*, *TabPFN*, *SAP-RPT-OSS*, *Tabula-8b*), and a general-purpose text embedding model (*all-MiniLM-L6-v2*), evaluating both predictive performance and embedding quality. TabPFN/TabuLA-8b do not support column embeddings, or cell embeddings.

3.1 Experiment Setup

Row Similarity Search. We evaluate our approach on seven entity matching datasets [19, 23] and two entity clustering datasets [28]. For the entity matching benchmarks, each positive entity pair defines a retrieval task in which one row is treated as the query and the matching row as the ground-truth relevant item. For the entity clustering datasets, we randomly sample one row from each cluster as the query and treat all remaining rows in the cluster as relevant targets. This setup guarantees that every query has at least one relevant retrieval result. Rows referring to the same underlying entity are considered relevant. Each row is embedded independently, and retrieval is performed using cosine similarity between embeddings. For the entity matching datasets, paired tables share the same schema, allowing them to be merged into a single evaluation table. We report Mean Reciprocal Rank (MRR), which measures the ranking position of the first relevant row.

Column Similarity Search. Column similarity search retrieves columns that are semantically related to a given query column and is central to tasks such as dataset discovery, schema matching, and data integration. Unlike exact string matching, the task requires models to capture both semantic meaning and distributional characteristics across columns, making it a natural benchmark for evaluating column representations. We evaluate on four public benchmarks for join discovery and schema matching. *Nextia* [10] contains tables from repositories such as Kaggle and OpenML annotated with join-quality labels; following Cong et al. [7], we use the small and medium testbeds. *Valentine* [20] provides schema-matching tasks with exact and noisy joins, and we evaluate on the *semantically-joinable* subset. *OpenData* [18] contains heterogeneous web and government tables with human-annotated semantic joinability labels. Finally, we use *WikiJoin* from the LakeBench suite [30], evaluating on a reduced subset of 659 tables (*WikiJoin-Small*).

Tabular Machine Learning. Beyond capturing semantic or structural relationships, tabular embeddings should also preserve information useful for downstream prediction tasks. For example, embeddings derived from a product table may be used to predict sales performance or price range. Higher-quality embeddings should therefore retain enough information from the original rows to support accurate supervised learning. We evaluate on all 51 datasets from the TabArena-Lite benchmark [8], which contains diverse real-world tabular classification and regression tasks. For each dataset, we follow the predefined TabArena-Lite protocol (fold 0, "lite" setting). We report ROC-AUC for binary classification, log-loss for multiclass classification, and RMSE for regression tasks. For models without native prediction heads (e.g., HyTrel, MiniLM), we train downstream classifiers on frozen row embeddings and report on XGBoost, which performed best, follows established protocols in tabular learning [34] and NLP [21]. For TabuLa-8b, we also use this pipeline due to unreliable label formatting in few-shot outputs. For HGR-TabE, we also evaluate multiple prediction heads and report Logistic Regression (one-vs-rest for multiclass and binned quantile classification with $N_q = 16$ for regression).

Cell Semantic Retrieval Given a query cell, the task is to retrieve the top-k most semantically similar cells from a corpus of tables. Because embedding methods may incorporate different levels of context (e.g., cell, row, column, or table information), retrieval performance provides a direct measure of how well embeddings capture fine-grained entity-level semantics. We evaluate using the S2abEL dataset [22], originally developed for entity linking in scientific tables. It contains 732 result tables extracted from 327 machine learning papers, with cells manually linked to entities in the PaperswithCode taxonomy. Cells associated with the same entity are treated as positives for retrieval. Each test case consists of a query cell and all other cells annotated with the same entity across a set of related tables. Each test case includes at most three tables, always including the table containing the query cell. Entities appearing in only one table are excluded. Overall, the benchmark contains 1000 cell-retrieval test cases with performance measured by accuracy.

4 RESULTS

Overall, no single model (Table 1) consistently outperforms others across all tasks, highlighting the trade-offs between general-purpose embeddings and task-specific tabular models. This underscores the need for a model capable of generalising across tabular tasks. Detailed results are included in Tables 2-7

Predictive Machine Learning. For binary classification (Table 2), HGR-TabE achieves a mean ROC-AUC of 0.74, outperforming HyTrel (0.50) and improving over MiniLM (0.72), while trailing specialized tabular models such as TabICL (0.86) and TabPFN (0.84). This reflects a common trade-off: methods optimized for predictive performance retain an advantage on supervised tasks. On multiclass classification

Dataset	XGBoost	HyTrel*	MiniLM*	SAP-RPT-1	TabICL-v2	TabPFN-v2.5	TabuLa-8B*	HGR-TabE
APSFailure	0.99	-	0.97	0.99	**1.00**	-	0.98	0.66
Amazon_employee.	0.82	0.50	0.72	**0.85**	**0.85**	0.84	0.73	0.68
Bank_Customer_Chu.	0.85	0.50	0.73	0.87	**0.88**	**0.88**	0.80	0.75
Bioresponse	0.88	-	0.60	0.87	0.87	**0.89**	0.78	0.71
Diabetes130US	0.63	-	0.60	0.65	**0.68**	0.67	0.61	0.65
E-CommereShipping.	0.74	0.51	0.67	**0.75**	0.74	0.74	0.70	0.71
Fitness_Club	0.76	0.53	0.68	**0.81**	**0.81**	**0.81**	0.77	0.74
GiveMeSomeCredit	0.86	-	0.77	**0.87**	**0.87**	-	0.84	0.85
HR_Analytics_Job˙	0.80	0.51	0.77	0.81	**0.82**	0.81	0.78	0.80
Is-this-a-good-cust.	0.67	0.50	0.69	**0.75**	0.74	0.74	0.67	0.64
Marketing_Campaign	0.92	0.51	0.75	0.89	**0.94**	**0.94**	0.86	0.73
NATICUSdroid	0.98	0.48	0.88	**0.99**	**0.99**	**0.99**	0.96	0.96
bank-marketing	0.75	0.51	0.70	0.77	**0.78**	0.77	0.72	0.75
blood-transfusion.	0.65	0.45	0.62	0.68	**0.74**	**0.74**	0.69	0.67
churn	0.93	0.52	0.66	0.93	**0.94**	**0.94**	0.78	0.67
coil2000_insurance.	0.71	-	0.61	0.73	**0.76**	**0.76**	0.65	0.71
credit-g	0.77	0.53	0.74	**0.80**	0.79	0.79	0.75	0.76
credit_card_clien.	0.77	0.49	0.67	**0.79**	**0.79**	**0.79**	0.75	0.73
customer_satisfact.	0.99	-	0.92	0.99	**1.00**	-	0.99	0.94
diabetes	0.81	0.44	0.70	0.84	**0.85**	**0.85**	0.80	0.72
hazelnut-spread.	0.97	0.48	0.75	**0.99**	**0.99**	**0.99**	0.82	0.81
heloc	0.77	0.51	0.72	**0.80**	**0.80**	**0.80**	0.76	0.75
in_vehicle_coupon.	0.83	0.49	0.77	0.77	**0.85**	**0.85**	0.78	0.76
jm1	0.73	0.51	0.70	0.75	**0.79**	0.77	0.72	0.70
kddcup09_appetency	0.77	-	0.55	0.81	0.81	**0.82**	0.56	0.70
online_shoppers.	0.92	0.49	0.86	**0.93**	**0.93**	**0.93**	0.89	0.85
polish_companies˙	0.96	0.51	0.71	0.97	0.95	**0.98**	0.88	0.76
qsar-biodeg	0.91	0.47	0.77	**0.94**	**0.94**	**0.94**	0.90	0.78
seismic-bumps	0.74	0.47	0.66	0.78	**0.81**	0.79	0.75	0.68
taiwanese_bankrupt.	0.94	0.53	0.65	0.94	**0.95**	**0.95**	0.84	0.77
Mean	0.83	0.50	0.72	0.84	**0.86**	0.84	0.78	0.74

Table 2: Tabular Prediction : Binary Classification Results per Dataset measured by ROC AUC Score (higher is better). * indicates that the results were obtained by training XGBoost on top of row embeddings.

Dataset	HyTrel*	MiniLM*	SAP-RPT-1	TabICL-v2	TabPFN-v2.5	TabuLa-8B*	HGR-TabE
MIC	1.71	0.97	**0.46**	0.47	**0.46**	0.79	0.78
SDSS17	-	0.33	0.08	**0.07**	-	0.13	0.30
anneal	2.58	0.17	**0.02**	0.03	**0.02**	0.08	0.10
hiva_agnostic	-	0.21	**0.18**	0.19	**0.18**	0.27	**0.18**
maternal_health.	3.49	0.40	0.50	**0.36**	0.42	0.43	0.56
splice	2.48	0.49	0.09	**0.07**	0.08	0.42	0.47
students_dropout˙	2.96	1.07	0.55	0.55	**0.54**	0.93	0.82
website_phishing	3.64	0.46	0.54	**0.20**	0.21	0.43	0.32
Mean	2.81	0.51	0.30	**0.24**	0.27	0.44	0.44

Table 3: Tabular prediction results for multiclass classification measured using log loss (lower is better). * indicates methods evaluated by training XGBoost on top of row embeddings.

(Table 3), HGR-TabE achieves a mean log-loss of 0.44, improving over MiniLM (0.51) but trailing SAP-RPT (0.30), TabPFNv2.5 (0.27) and TabICLv2 (0.24). While not the top performer, it achieves competitive results and matches the best model on certain datasets (e.g., *hiva_agnostic*). For regression (Table 4), HGR-TabE achieves a mean RMSE of 21,991.6. This lags TabICL, SAP-RPT-OSS, and TabPFN, but outperforms HyTrel and MiniLM. The higher mean is heavily influenced by datasets with large target scales (e.g., *miami_housing*), and relative performance is more consistent across datasets..

Semantic Similarity Tasks. On row similarity (Table 5), HGR-TabE achieves a mean MRR of 0.50, outperforming tabular representation learning baselines such as HyTrel (0.05) and TabICL (0.04), while approaching the sentence-transformer MiniLM (0.66). It attains second-best performance on several datasets, indicating robust instance-level representations. On column joins (Table 6), HGR-TabE reaches a mean MRR of 0.54, outperforming tabular baselines and remaining competitive with MiniLM (0.59). These results suggest that HGR-TabE captures both row- and schema-level semantics effectively. For cell-level retrieval (Table 7), HGR-TabE achieves an MRR of 0.30 (tied with HyTrel), outperforming TabICL (0.18). MiniLM still achieves the strongest performance overall (0.63), the results indicate that HGR-TabE can capture fine-grained semantic relationships between cells better than predictive foundation models. Overall, these results suggest that retrieval tasks benefit substantially from converting tables into textual representations, as language-model-based semantic encoders capture relational and contextual meaning more effectively than table-native approaches such as TabICL, HyTrel etc.

Dataset	HyTrel*	MiniLM*	SAP-RPT-1	TabICL-v2	TabPFN-v2.5	TabuLa-8B*	HGR-TabE
Another-Dataset-on.	2357.67	1453.25	701.13	684.34	**683.78**	1084.08	1359.73
Food_Delivery_Tim.	10.65	9.01	7.98	7.66	**7.60**	8.25	8.54
QSAR-TID-11	-	1.48	0.90	0.87	**0.83**	1.40	1.36
QSAR_fish_toxicit.	1.75	1.30	0.90	**0.89**	0.90	1.10	1.41
airfoil_self_nois.	8.93	5.29	1.17	0.99	**0.97**	3.70	4.71
concrete_compress.	22.25	12.84	4.05	**3.66**	3.82	9.79	11.41
diamonds	5441.49	1319.28	574.47	503.70	**503.14**	761.96	1154.98
healthcare_insur.	15026.74	6698.41	4173.69	**4053.12**	4083.82	5739.85	6054.10
houses	0.70	0.48	**0.20**	**0.20**	**0.20**	0.34	0.44
miami_housing	395649.83	304003.34	87855.92	**79390.80**	80753.88	199150.39	277275.23
physiochemical.	6.79	5.99	3.53	**3.01**	3.05	5.18	5.94
superconductivity	-	19.25	9.87	**8.92**	9.20	15.39	12.19
wine_quality	1.12	0.74	0.60	**0.59**	0.62	0.71	1.24
Mean	38047.99	24117.74	7179.57	**6512.21**	6619.37	15906.32	21,991.6

Table 4: Tabular prediction results for regression tasks measured using RMSE (lower is better). * indicates methods evaluated by training XGBoost on top of row embeddings.

Dataset	HyTrel	MiniLM	SAP-RPT-1	TabICL-v2	TabPFN v2.5	TabuLa-8B	HGR-TabE
Amazon-Google	0.05	**0.58**	0.02	0.00	0.00	0.35	0.45
Beer	0.07	**0.86**	0.00	0.00	0.00	0.45	0.40
DBLP-ACM	0.16	**0.96**	0.02	0.00	0.00	0.95	0.96
DBLP-GoogleScholar	0.03	**0.60**	-	0.19	-	0.48	0.29
Fodors-Zagats	0.04	**0.94**	0.08	0.04	0.01	0.87	0.96
MusicBrainz	0.01	**0.96**	0.00	0.00	0.00	0.34	0.64
Walmart-Amazon	0.05	**0.72**	0.10	0.00	0.00	0.50	0.53
geological-settlements	0.00	0.06	0.00	**0.15**	0.00	0.00	0.13
iTunes-Amazon	0.00	**0.23**	-	0.00	-	0.03	0.16
Mean	0.05	**0.66**	0.03	0.04	0.00	0.44	0.50

Table 5: Row Similarity Search: Full Results MRR results per dataset. - indicates that the approach could not be run on the dataset, mostly due to memory constraints.

Dataset	HyTrel	MiniLM	SAP-RPT-1	TabICL-v2	HGR-TabE
Nextia	0.25	**0.26**	0.01	0.08	0.24
OpenData	0.43	**0.58**	0.20	0.07	0.56
Valentine	0.26	**0.59**	0.00	0.42	0.50
Wikijoin Small	0.74	**0.94**	0.19	0.20	0.88
Mean	0.34	**0.59**	0.08	0.15	0.54

Table 6: Column Similarity Search. MRR results per dataset.

Dataset	MiniLM	HyTrel	TabICL-v2	SAP-RPT-1	HGR-TabE
S2Abel	**0.63**	0.30	0.18	0.28	0.30

Table 7: Cell Level Semantic Retrieval : Retrieval Accuracy per dataset.

5 CONCLUSION

We introduced **HGR-TabE**, a unified tabular embedding framework that learns transferable representations across heterogeneous tables without task-specific fine-tuning. By combining HGR-based common-space alignment with hypergraph-based message passing, the model jointly captures semantic, structural, and distributional information at the cell, row, column, and table levels.

Our experiments across predictive modeling, row similarity, column similarity, and cell retrieval demonstrate that a shared embedding model can generalize competitively across diverse tabular tasks. HGR-TabE consistently improves over general-purpose text embedding models on structured prediction tasks, while outperforming several tabular foundation models on retrieval and similarity benchmarks. These results suggest that universal tabular embeddings are feasible and can support both predictive and semantic applications within a common representation space.

At the same time, the results highlight that no single approach dominates every setting. Specialized models such as TabPFN and TabICL remain highly effective for supervised prediction, while language-model-based embeddings continue to excel on semantic retrieval tasks where converting tables into text provides richer contextual signals. This suggests that broadly applicable tabular embeddings are best viewed as a flexible foundation that complements, rather than replaces, specialized systems. More broadly, our work highlights the importance of moving beyond prediction-only evaluations toward tabular foundation models capable of supporting tasks at multiple granularities, including cells, rows, columns, and entire tables. We hope HGR-TabE provides a useful step toward more general-purpose tabular representation learning.

REFERENCES

[1] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. 2013. Deep canonical correlation analysis. In *International conference on machine learning*. PMLR, 1247–1255.

[2] Song Bai, Feihu Zhang, and Philip H. S. Torr. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognit.* 110 (2021), 107637. https://doi.org/10.1016/J.PATCOG.2020.107637

[3] Sebastian Bordt, Harsha Nori, and Rich Caruana. 2023. Elephants Never Forget: Testing Language Models for Memorization of Tabular Data. *Table Representation Learning Workshop at NeurIPS 2023* abs/2403.06644 (2023). https://doi.org/10.48550/ARXIV.2403.06644 arXiv:2403.06644

[4] 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*. 32173–32193. https://openreview.net/forum?id=7vqlzODS28

[5] 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*. ACM, 785–794. https://doi.org/10.1145/2939672.2939785

[6] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. 2021. You are allset: A multiset function framework for hypergraph neural networks. *arXiv preprint arXiv:2106.13264* (2021).

[7] Tianji Cong, James Gale, Jason Frantz, H. V. Jagadish, and Çagatay Demiralp. 2023. WarpGate: A Semantic Join Discovery System for Cloud Data Warehouses. In *13th Conference on Innovative Data Systems Research, CIDR 2023, Amsterdam, The Netherlands, January 8-11, 2023*. www.cidrdb.org. https://vldb.org/cidrdb/2023/warpgate-a-semantic-join-discovery-system-for-cloud-data-warehouses.html

[8] Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. 2025. TabArena: A Living Benchmark for Machine Learning on Tabular Data. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*. https://openreview.net/pdf?id=jZqCqpCLdU

[9] Xi Fang, Weijie Xu, Fiona Anting Tan, Ziqing Hu, Jiani Zhang, Yanjun Qi, Srinivasan H. Sengamedu, and Christos Faloutsos. 2024. Large Language Models (LLMs) on Tabular Data: Prediction, Generation, and Understanding - A Survey. *Trans. Mach. Learn. Res.* 2024 (2024). https://openreview.net/forum?id=IZnrCGF9WI

[10] Javier Flores, Sergi Nadal, and Oscar Romero. 2021. Effective and Scalable Data Discovery with NextiaJD. In *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*. OpenProceedings.org, 690–693. https://doi.org/10.5441/002/EDBT.2021.85

[11] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[12] Josh Gardner, Juan C. Perdomo, and Ludwig Schmidt. 2024. Large Scale Transfer Learning for Tabular Data via Language Modeling. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*. https://openreview.net/forum?id=WH5blx5tZ1

[13] Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David A. Sontag. 2023. TabLLM: Few-shot Classification of Tabular Data with Large Language Models. In *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain (Proceedings of Machine Learning Research)*, Vol. 206. PMLR, 5549–5581. https://proceedings.mlr.press/v206/hegselmann23a.html

[14] Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open Domain Question Answering over Tables via Dense Retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Online, 512–519. https://doi.org/10.18653/v1/2021.naacl-main.43

[15] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. 2025. Accurate predictions on small data with a tabular foundation model. *Nature* (09 01 2025). https://doi.org/10.1038/s41586-024-08328-6

[16] Aamod Khatiwada, Harsha Kokel, Ibrahim Abdelaziz, Subhajit Chaudhury, Julian Dolby, Oktie Hassanzadeh, Zhenhan Huang, Tejaswini Pedapati, Horst Samulowitz, and Kavitha Srinivas. 2025. TabSketchFM: Sketch-Based Tabular Representation Learning for Data Discovery Over Data Lakes. In *41st IEEE International Conference on Data Engineering, ICDE 2025, Hong Kong, May 19-23, 2025*. IEEE, 1523–1536. https://doi.org/10.1109/ICDE65448.2025.00118

[17] Myung Jun Kim, Félix Lefebvre, Gaëtan Brison, Alexandre Perez-Lebel, and Gaël Varoquaux. 2025. Table Foundation Models: on knowledge pre-training for tabular learning. *Trans. Mach. Learn. Res.* 2025 (2025). https://openreview.net/forum?id=QV4P8Csw17

[18] Harsha Kokel, Aamod Khatiwada, Tejaswini Pedapati, Haritha Ananthakrishnan, Oktie Hassanzadeh, Horst Samulowitz, and Kavitha Srinivas. 2025. Evaluating Joinable Column Discovery Approaches for Context-Aware Search. arXiv:2510.24599 [cs.DB] https://arxiv.org/abs/2510.24599

[19] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proc. VLDB Endow.* 3, 1 (2010), 484–493. https://doi.org/10.14778/1920841.1920904

[20] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. 2021. Valentine: Evaluating Matching Techniques for Dataset Discovery. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*. IEEE, 468–479. https://doi.org/10.1109/ICDE51399.2021.00047

[21] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the Sentence Embeddings from Pre-trained Language Models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Online, 9119–9130. https://doi.org/10.18653/v1/2020.emnlp-main.733

[22] Yuze Lou, Bailey Kuehl, Erin Bransom, Sergey Feldman, Aakanksha Naik, and Doug Downey. 2023. S2abEL: A Dataset for Entity Linking from Scientific Tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. 3089–3101.

[23] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. ACM, 19–34. https://doi.org/10.1145/3183713.3196926

[24] 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, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*. Association for Computational Linguistics, 2006–2029. https://doi.org/10.18653/V1/2023.EACL-MAIN.148

[25] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2025. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net. https://openreview.net/forum?id=0VvD1PmNzM

[26] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2026. TabICLv2: A better, faster, scalable, and open tabular foundation model. *CoRR* abs/2602.11139 (2026). https://doi.org/10.48550/ARXIV.2602.11139 arXiv:2602.11139

[27] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 3980–3990. https://doi.org/10.18653/V1/D19-1410

[28] Alieh Saeedi, Eric Peukert, and Erhard Rahm. 2017. Comparative Evaluation of Distributed Clustering Schemes for Multi-source Entity Resolution. In *Advances in Databases and Information Systems - 21st European Conference, ADBIS 2017, Nicosia, Cyprus, September 24-27, 2017, Proceedings (Lecture Notes in Computer Science)*, Vol. 10509. Springer, 278–293. https://doi.org/10.1007/978-3-319-66917-5_19

[29] Marco Spinaci, Marek Polewczyk, Maximilian Schambach, and Sam Thelin. 2025. ConTextTab: A Semantics-Aware Tabular In-Context Learner. In *Advances in Neural Information Processing Systems (NeurIPS)*. https://openreview.net/forum?id=kGMRb4jbTP

[30] 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

[31] Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro A. Szekely. 2021. Representing Numbers in NLP: a Survey and a Vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*. Association for Computational Linguistics, 644–656. https://doi.org/10.18653/V1/2021.NAACL-MAIN.53

[32] Lichen Wang, Jiaxiang Wu, Shao-Lun Huang, Lizhong Zheng, Xiangxiang Xu, Lin Zhang, and Junzhou Huang. 2019. An Efficient Approach to Informative Feature Extraction from Multimodal Data. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 5281–5288. https://doi.org/10.1609/AAAI.V33I01.33015281

[33] 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*. https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[34] Jing Wu, Suiyao Chen, Qi Zhao, Renat Sergazinov, Chen Li, Shengjie Liu, Chongchao Zhao, Tianpei Xie, Hanqing Guo, Cheng Ji, Daniel Cociorva, and Hakan Brunzell. 2024. SwitchTab: Switched Autoencoders Are Effective Tabular

Learners. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*. AAAI Press, 15924–15933. https://doi.org/10.1609/AAAI.V38I14.29523

[35] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314* (2020).

[36] Xiyuan Zhang, Danielle Maddix Robinson, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael Mahoney, et al. 2026. Mitra: Mixed synthetic priors for enhancing tabular foundation models. *Advances in neural information processing systems* 38 (2026), 15795–15840. https://openreview.net/pdf?id=t8YRsWY6HM