Heterogeneous Feature-Aware Graph Neural Network for Tabular Data

Hongxiao Fei[1], Jinqi Hu[1], Liu Yang[1] *, Tingxuan Chen[1] , Huayou Su[2], and Zanqun Liu[3]

[1] School of Computer Science and Engineering, Central South University, Changsha, China
[2] School of Computer Science and Technology, National University of Defense Technology, Changsha, China
[3] School of Civil Engineering, Central South University, Changsha, China

Abstract. Deep learning for tabular data advance in extracting valuable information from column feature interactions. However, tabular data consists of various column feature types such as numerical, categorical and binary, each with distinct semantics, ranges and distributions. Existing methods fail to distinguish between the heterogeneity of features and their interactions, leading to irreversible information loss. To address this, we propose the Heterogeneous Feature Interaction Network (HFIN). HFIN represents column features as different types of nodes in a relational graph and enhances heterogeneous information in feature interactions through relational message passing. To preserve key information from heterogeneous features, we embed numerical, categorical, and binary features separately and integrate sequential information to improve numerical feature embeddings. We use mutual information and symmetric matrix factorization to estimate a global feature interaction graph, capturing both semantic and topological structures of feature interactions. Additionally, we introduce a relational attention-based message passing mechanism, which dynamically adjusts edge weights to capture critical information, further strengthening the expression of heterogeneous interactions. Experimental results on six datasets show that HFIN outperforms most DNN methods in classification and regression tasks, while providing better interpretability.

Keywords: tabular data · heterogeneous graph neural networks · deep learning.

1 Introduction

Tabular data, one of the most common data formats, comprises independent samples (rows) and diverse attributes (columns) [1]. Unlike unstructured data such as images and text, tabular data directly represents real-world phenomena

* Corresponding author

in a structured form, making it crucial for decision-making processes in data-driven fields such as intelligent healthcare, recommendation systems, and materials genomics.

The success of deep learning in fields like image, audio, and text has sparked interest in applying it to tabular data [2]. While deep neural networks exhibit strong representational capabilities and perform well on homogeneous datasets, the heterogeneity of tabular data presents significant challenges[3]. In contrast, graph neural networks (GNNs) represent features or samples as nodes and interactions as edges, enabling more intuitive extraction of feature relationships. However, current methods typically simplify all feature interactions into a single type of information, neglecting the rich semantics and relational patterns inherent in interactions between different feature types, which leads to irreversible loss of feature information.

When addressing the heterogeneity of features and feature interactions in tabular data, using graph neural networks faces a series of challenges. First, graph neural networks (GNNs) are primarily used for modeling graph-structured data, such as node and graph classification, with limited exploration in regression tasks. Secondly, tabular data lacks an explicit graph structure, and the interactions between heterogeneous features vary, making it difficult to effectively extract the semantic relationships and topological structure among them. In addition, tabular data consists of numerical, categorical, and binary features, which have significant differences in semantics, range, and distribution, and extracting and leveraging such heterogeneous information presents a major challenge[4].

To address the above issue, we propose a Heterogeneous Feature Interaction Network (HFIN) for tabular data, which represents feature interactions as a relational graph and enhances the expression of heterogeneous feature interaction information through relational message passing. HFIN consists of three core modules: embedding enhancement, graph structure learning, and relational message passing. In embedding enhancement module, we optimize numerical embeddings to extract richer information; in relational graph structure learning module, we estimate global feature relationships using mutual information and construct a sparse relational graph through topology optimization to capture the topological and semantic relationships in tabular data; in the relational message passing module, we introduce a relational attention mechanism to dynamically adjust edge message weights, further leveraging the rich information in heterogeneous feature interactions. These innovations enable HFIN to achieve higher accuracy and interpretability in tabular data modeling.

Overall, the contributions of this study include:

- We propose an optimized embedding method for numerical variables, significantly improving regression performance on tabular data.
- We design a simple relation graph structure learning method that effectively captures the semantic relationships and topological structures of feature interactions in tabular data.

- We design a relation-based message passing mechanism with relational attention, effectively capturing key interactions between feature types and improving performance in tabular classification tasks.
- Experimental results on six public tabular datasets demonstrate that the method is highly effective in handling heterogeneous interactions, outperforming most deep tabular learning models.

2 Related Works

2.1 Deep Learning for Tabular Data

Transformer-based Models: Transformer has demonstrated exceptional learning ability in various applications, making it an important tool in tabular data research. For instance, TabNet [5] was the first to integrate self-attention with decision tree structures, enabling feature selection and importance evaluation. FT-Transformer [2] separately embeds categorical and numerical features, and captures complex interactions between features using Transformer layers, significantly improving performance. However, it does not further address heterogeneous feature interactions. SAINT [6] combines self-attention and inter-sample attention to capture both feature interactions and relationships among samples. However, Transformer-based models often suffer from computational redundancy, lack intuitive interpretability, and, most importantly, fail to effectively extract heterogeneous information from interactions between different feature types.

GNNs-Based Models: Graph Neural Networks (GNNs) have been applied to model tabular data in recent years, since they can handle row-column exchange invariance and more flexibly capture feature interactions compared to DNN-based models. T2G-Former [7] constructs graph structures by integrating semantic relationships and topological information, while enhancing Transformer attention layers with a graph estimator. INCE [8] uses a message passing mechanism on a complete feature graph to extract and process feature interactions, offering a new framework for tabular learning. DRSA-Net [9] adaptively learns a sparse graph structure and introduces a dual-path information propagation mechanism, considering both feature similarity and topological structure, effectively capturing direct and indirect feature interactions. Unfortunately, these methods overlook the heterogeneity of feature interactions, resulting in incomplete extraction of interaction information and consequently limiting the expressive capacity of models.

3 Problem Statement and Notation Definition

Tabular data refers to structured data with N rows and M columns, where each column represents a feature. Features are categorized into numerical, categorical, and binary types, with F_n, F_c, F_b denoting the feature type sets. Tabular learning focuses on two supervised prediction tasks in tabular data:

regression and classification. Mathematically, given a tabular dataset $D = \{x_i^1, ..., x_i^j, ..., x_i^N, y_i\}_{i\in\{1,...,N\}}$, where x_i^j represents the j-th column feature of the i-th row sample, and y_i represents the label of the i-th row sample.

In relational graph, tabular columns are represented as graph nodes, categorized into numerical, categorical, and binary nodes, and the intensity of columns feature interactions represented as edge weights. Specifically, edge weights do not consider direction, while directionality is handled during relational message passing. The feature interaction relational graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{W}, \mathcal{R})$, $\mathcal{V}$ representing the node set with numerical, categorical, and binary nodes, $\mathcal{E}$ denotes the set of edges with size $|\mathcal{E}|$, $\mathcal{W}$ representing the edge weight matrix, and $\mathcal{R} = \{r_{nn}, r_{bb}, r_{cc}, r_{nc}, r_{nb}, r_{cb}\}$, the set of relationships defining interaction types among different feature categories, such as r_{nn}, representing interactions between numerical nodes. For an edge $e_{ij} \in \mathcal{E}$ connecting node $v_i \in F_n$ and node $v_j \in F_b$, the edge type is $r_{ij} = r_{nb}$, with edge weight $\mathcal{W}_{ij}$.

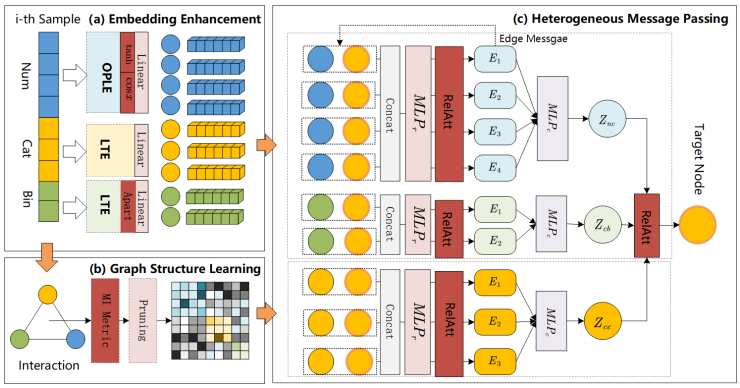

Fig. 1: Overview of HFIN. Embedding Enhancement improves the expressiveness of feature embeddings, Graph Structure Learning captures heterogeneous feature semantics and topology information, Relational Message Passing enhances the representation of heterogeneous feature interactions.

4 Heterogeneous Graph Neural Networks

To capture heterogeneous information and relational patterns from different types of interactions, we propose Heterogeneous Feature Interaction Network (HFIN), as shown in Fig. 1, inculding the modules of Embedding Enhancement, Graph Structure Learning, and Relational Message Passing.

4.1 Embedding Enhancement

A unified embedding of heterogeneous features in tabular data into a dense space inevitably result in the loss of critical feature information. To preserve

the heterogeneous information of different feature types, we embed numerical, categorical, and binary features separately.

Ordinal and Periodic Learning Embedding (OPLE): Numerical features contain rich information and ordinal relationships but present significant challenges for embedding. The PLR method [10] enhances the differentiation of numerical features using periodic activation functions, improving regression accuracy. However, it fails to preserve ordinal relationships effectively and limits embedding differentiation across value ranges. To address this, we propose Ordinal and Periodic Learning Embedding (OPLE), which integrates ordinal and periodic information to enhance both intra- and inter-feature embedding differentiation. The numerical mapping function is as follows:

$$f_n\left(x^j\right) = concat\left[cos\left(v\right)v, tanh\left(v\right)\right], v = \left[2\pi c_1 x^j, ..., 2\pi c_k x^j\right] \tag{1}$$

where $c_i, i \in \{1, ..., k\}$ is a trainable parameter initialized from $N(0, \delta)$, k represents the number of sampling points, both δ and k are learnable parameters.

Lookup Table Embedding (LTE) to preserve discrete semantic information while improving computational efficiency, the categorical and binary mapping functions are described as follows:

$$f_c(x^j) = E_c(x_j), E_c \in R^{C \times d} \tag{2}$$

$$f_b(x^j) = E_b(x_j), E_b \in R^{2 \times d} \tag{3}$$

where E_c and E_b are the embedding matrices, C counts the amount of categorical features, and a symmetry constraint $E_b[1] = -E_b[0]$ is applied to the binary feature embeddings to keep two binary classifications apart.

4.2 Graph Structure Learning

Tabular data lacks an explicit graph structure, so we treat column features as nodes and use the interaction strength between features as edge weights to uncover the implicit graph structure. Considering feature heterogeneity, we classify nodes into numerical, categorical, and binary types, and define six types of edge relationships: r_{nn}, r_{cc} and, r_{bb} represent interactions between nodes of the same type, while r_{nb}, r_{nc}, and r_{cb} describe interactions between nodes of different types, temporarily ignoring the edge directions. Additionally, the graph structure is continuously optimized during message passing, providing interpretability for predictions.

Mutual Information Metric (MIM): To extract heterogeneous global information from tabular data, we utilize mutual information to measure the original feature data. The edge weight between two numerical features x^i and x^j can be calculate using the continuous form of mutual information, and the mathematical formula is as follows:

$$W\left[i, j\right] = I\left(x^i, x^j\right) = \int_{u \in x^i} \int_{v \in x^j} P\left(u, v\right) \log \frac{P\left(u, v\right)}{P\left(u\right) P\left(v\right)} du dv \tag{4}$$

where $P(u, v)$ is the joint probability density of u and v, and $P(u)$ and $P(v)$ are the marginal probability densities of u and v, respectively. When dealing with categorical and binary features, the interaction strength can be computed using the discrete form of mutual information.

Topology Optimization (TO): We utilize the interaction strength between column features as edge weights to construct a weighted symmetric graph. The graph topology is represented by an adjacency matrix $A \in R^M$. To avoid redundant computations and overfitting caused by a fully connected graph, we use matrix symmetric decomposition with learnable parameters to optimize topology A:

$$A = \sigma(UU^T + b > \tau) \tag{5}$$

where $U \in R^{M \times k} (k << M)$ is a learnable parameter, b is a learnable bias term, and σ is an activation function. Furthermore, to avoid redundant feature computations and emphasize information from neighboring nodes, we set the self-loops in the weight matrix to zero ($A[i, i] = 0, i \in \{1, ..., M\}$).

By integrating semantic relationships measured by mutual information with the topology optimized via parameter learning, we can derive the initial heterogeneous weight matrix $\mathcal{W}$.

$$\mathcal{W} = A \odot W \tag{6}$$

where W is the edge weight matrix measured by mutual information, and $\odot$ denotes the Hadamard product.

4.3 Heterogeneous Message Passing

We introduce a relational attention mechanism into the weighted graph structure, enhancing the model's ability to capture heterogeneous feature interactions and improving interpretability through edge weights. To better extract global information for prediction, we initialize a CLS global node based on the task type (numerical, categorical, or binary) and set edge weights to their average.

According to the above graph structure, we define the initial state of each node and edge weight as:

$$v_j^0 = c^j, a_{ij}^0 = \mathcal{W}_{ij} \tag{7}$$

where c^j is the embedding of feature x^j, and $\mathcal{W}_{ij} \in \mathcal{W}$ is the edge weight of the heterogeneous graph.

We assume the target node $v_j \in F_c$, with all connected types corresponding to r_{cc}, r_{nc} and r_{cb}. To more accurately capture the heterogeneous information in interactions between different types of features, we introduce intra-relational attention, dynamically enhancing the weights of critical information within the same type to improve edge information extraction. The representation of the target node at the l-th layer is denoted as v_j^l, and the local interaction information at layer l from node v_i^l to node v_j^l is represented by edge information $\hat{e}_{ij}^{l+1}$. The computation process for extracting the edge information is as follows:

$$\hat{e}_{ij}^{l+1} = \alpha_{ij}^{l+1} MLP_r \left(Concat \left(v_i^l, v_j^l, e_{ij}^l \right) \right), \tag{8}$$

$$\alpha_{ij}^{l+1} = \frac{\exp\left(\sigma\left(\mathbf{a_r}\left[\hat{e}_{ij}^{l} \parallel W_r v_i^l \parallel W_r v_j^l\right] + \alpha_{ij}^{l}\right)\right)}{\sum_{k\in\mathcal{N}_r^j}\exp\left(\sigma\left(\mathbf{a_r}\left[\hat{e}_{kj}^{l+1} \parallel W_r v_k^l \parallel W_r v_j^l\right] + \alpha_{kj}^{l}\right)\right)} \tag{9}$$

where MLP_r is learner that extracts feature interaction information, σ is Leaky ReLU activation function. $\mathbf{a_r}$ and W_r is learnable parameters corresponding to edge types r, and e_{ij}^l represents edge feature for feature interaction from node v_i^l to v_j^l in the l-th layer.

For the target node, all edge information is aggregated by type, resulting in interaction aggregation information $z_{r_{cc}}^l$, $z_{r_{cb}}^l$ and $z_{r_{nc}}^l$.

$$z_r^l = MLP_c\left(\sum_{i\in\mathcal{N}_r^j} \hat{e}_{ij}^l, v_j^l\right) \tag{10}$$

where r_{ij} is abbreviated as r, and MLP_c is an information aggregation learner designed for target nodes with categorical features, used to aggregate all neighboring edge information of the same interaction type.

The edge features are updated using edge messages, as mathematically described below:

$$e_{ij}^{l+1} = e_{ij}^l + \hat{e}_{ij}^{l+1} \tag{11}$$

To enhance the representational capacity of interactions across different types, we further introduce an inter-relational attention mechanism to adaptively learn weights for updating the target node:

$$v_j^{l+1} = v_j^l + \beta_{r_{cc}} z_{r_{cc}}^l + \beta_{r_{nc}} z_{r_{nc}}^l + \beta_{r_{cb}} z_{r_{cb}}^l \tag{12}$$

$$\beta_r = \frac{\exp\left(\sigma\left(\mathrm{b_r}\left[W_r z_r^l || W_r v_j^l\ \right]\right)\right)}{\sum_{\hat{r}\in\mathcal{R}}\exp\left(\sigma\left(\mathrm{b_{\hat{r}}}\left[W_{\hat{r}} z_{\hat{r}}^l || W_{\hat{r}} v_j^l\ \right]\right)\right)} \tag{13}$$

where σ is the activation function. W_r adopts shared weight parameters, and $\mathrm{b_r}$ is the learnable attention parameter for relation type r.

The CLS node is treated as a type based on the predicted label feature category, with the label being a category variable($CLS \in F_c$). The global node is then included in the message passing on the heterogeneous graph, and finally CLS node is extracted for prediction. The prediction layer typically consists of two or three layers of MLP.

5 Experiments

5.1 Dataset and Implementation Details

We select six classic open-source datasets, including three purely numerical datasets and three mixed-type datasets, covering both classification and regression tasks. The datasets are Gesture Phase Prediction (GE), Churn Modeling (CM), Eye Movements (EY), California Housing (CA), House 16H (HO), and Adult (AD). Dataset statistics are presented in Table 1.

Table 1: Statistical details of the six public datasets

dataset	GE	CM	EY	CA	HO	AD
Instance	9873	10000	10936	20640	22784	48842
Num	32	6	22	7	16	6
Cat	0	1	3	0	0	1
Bin	0	3	1	0	0	7
Classes	2	2	3	-	-	2
Metric	ACC.	ACC.	ACC.	RMSE.	RMSE.	ACC.

We compare HFIN with models like NODE, AutoInt, TabNet, FT-Transformer, DANets, T2G-Former, INCE, and benchmark models MLP and XGBoost. The dataset is split 8:1:1 for training, testing, and validation, with a learning rate of 0.001, batch size 256, and 100 epochs. Early stopping is applied after 10 steps. All experiments are run on an NVIDIA RTX 4060 with PyTorch and Python 3.10, with results averaged over 15 trials.

5.2 Prediction Performance

We compared HFIN with 10 models across six classic datasets, as shown in Table 2, our proposed HFIN achieves the best performance on two datasets and ranks second on four others.

Table 2: Performance comparison on the 6 public tabular datasets. ↓ represents the RMSE metric and ↑ represents accuracy.

Dataset	GE↑	CM↓	EY↑	CA↓	HO↓	AD↑
XGBoost	**68.42**	85.92	72.51	**0.436**	3.169	**87.30**
MLP	58.64	85.77	61.10	0.499	3.173	85.35
TabNet	60.01	85.01	62.08	0.513	3.252	84.84
DANet-28	61.63	85.10	60.53	0.524	3.236	85.00
Node	53.94	85.86	65.54	0.463	3.216	85.77
AutoInt	58.33	85.51	61.07	0.472	3.147	85.66
DCNv2	55.72	85.68	61.37	0.489	3.172	85.48
FT-Transformer	61.25	86.07	70.84	0.460	**3.124**	85.72
T2G-Former	65.57	<u>86.21</u>	<u>78.08</u>	0.458	3.138	85.96
INCE	60.48	85.82	76.17	0.470	3.192	85.66
HFIN	<u>65.63</u>	**86.38**	**78.16**	<u>0.449</u>	<u>3.132</u>	<u>86.46</u>

(The best results are marked in **bold** and the second best results are underlined.)

HFIN demonstrates exceptional performance on mixed-type data, outperforming all deep tabular learning methods on three mixed datasets, with only one dataset where it falls short of XGBoost. Additionally, HFIN delivers excellent results on numerical tasks, significantly improving regression accuracy compared

to graph neural network model INCE. All models were hyperparameter-tuned using Optuna to select the best validation results.

5.3 Ablation Experiments

To evaluate the effectiveness of the proposed modules in HFIN, we conduct ablation experiments. These experiments aim to assess the impact of embedding enhancement, graph structure learning and relational attention mechanisms. We select three representative datasets for the ablation analysis, including California Housing, Customer Churn Model and the Adult dataset.

Table 3: Performance Comparison under different ablation settings

Methods	CA↓	CM ↑	AD↑
OPLE-(Linear)	0.471	85.86	85.93
PLR	0.458	86.15	86.27
FC+CS	0.468	85.87	86.02
FC+MIM	0.457	85.91	85.93
TO+CS	0.452	86.26	86.35
RelAtt-	0.448	85.76	85.74
HFIN	0.449	86.38	86.46

Table 3 show that the OPLE module significantly improves the performance of graph neural networks in regression tasks, achieving 4.68% and 1.97% improvements over Linear and PLR, respectively. We also compared three graph structures: FC + CS, TO + CS, and FC + MIM (where FC: fully connected networks, TO: topology optimization, CS: cosine similarity, and MIM: mutual information). The results indicate that MIM provides a small improvement, while the TO module shows the most significant effect by reducing overfitting caused by fully connected graphs, thus preventing early stopping. The combination of TO and MIM yields the best performance. Additionally, we compared the case without relational attention (RelAtt-), where all aggregation weights are set to 1. The results show that RelAtt performs well in classification tasks, improving by 0.67%.

6 Conclusion

This paper presents a Heterogeneous Graph Neural Network (HFIN) for tabular data modeling, addressing the challenges of limited regression capacity and heterogeneous feature interactions in graph networks. We separately embed numerical, categorical, and binary features, enhancing numerical embeddings by integrating sequential information. Mutual information is used to capture global feature interactions, while symmetric matrix decomposition improves graph sparsity. HFIN performs message passing based on feature interaction

types, incorporating a relational attention mechanism to better represent heterogeneous interactions. Compared to existing deep tabular learning methods, HFIN demonstrates superior performance and interpretability on heterogeneous tabular data. Future research could focus on more effective methods for handling feature heterogeneity to further enhance the generalization and practicality of tabular models.

Acknowledgments. We would like to thank the reviewers for their valuable comments on this paper. This work is supported by the National Key Research and Development Program of China (No. 2022YFB2602602) and National Natural Science Foundation of China (No. 62172451).

References

1. Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.
2. Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.
3. Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. Deep neural networks and tabular data: A survey. *IEEE transactions on neural networks and learning systems*, 2022.
4. Yael Mathov, Eden Levy, Ziv Katzir, Asaf Shabtai, and Yuval Elovici. Not all datasets are born equal: On heterogeneous tabular data and adversarial examples. *Knowledge-Based Systems*, 242:108377, 2022.
5. Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.
6. Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.
7. Jiahuan Yan, Jintai Chen, Yixuan Wu, Danny Z Chen, and Jian Wu. T2g-former: organizing tabular features into relation graphs promotes heterogeneous feature interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10720–10728, 2023.
8. Mario Villaizán-Vallelado, Matteo Salvatori, Belén Carro, and Antonio Javier Sanchez-Esguevillas. Graph neural network contextual embedding for deep learning on tabular data. *Neural Networks*, 173:106180, 2024.
9. Qinghua Zheng, Zhen Peng, Zhuohang Dang, Linchao Zhu, Ziqi Liu, Zhiqiang Zhang, and Jun Zhou. Deep tabular data modeling with dual-route structure-adaptive graph networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9715–9727, 2023.
10. Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.