TabGLM: Tabular Graph Language Model for Learning Transferable Representations Through Multi-Modal Consistency Minimization

Anay Majee[*1, 2 †], **Maria Xenochristou**[*1], **Wei-Peng Chen**[1]

[1]Fujitsu Research of America
[2]The University of Texas at Dallas
anay.majee@utdallas.edu, mxenochristou@fujitsu.com, wchen@fujitsu.com

Abstract

Handling heterogeneous data in tabular datasets poses a significant challenge for deep learning models. While attention-based architectures and self-supervised learning have achieved notable success, their application to tabular data remains less effective over linear and tree based models. Although several breakthroughs have been achieved by models which transform tables into uni-modal transformations like image, language and graph, these models often underperform in the presence of feature heterogeneity. To address this gap, we introduce **TabGLM** (**Tab**ular **G**raph **L**anguage **M**odel), a novel multi-modal architecture designed to model both structural and semantic information from a table. TabGLM transforms each row of a table into a fully connected graph and serialized text, which are then encoded using a graph neural network (GNN) and a text encoder, respectively. By aligning these representations through a joint, multi-modal, self-supervised learning objective, TabGLM leverages complementary information from both modalities, thereby enhancing feature learning. TabGLM's flexible graph-text pipeline efficiently processes heterogeneous datasets with significantly fewer parameters over existing Deep Learning approaches. Evaluations across 25 benchmark datasets demonstrate substantial performance gains, with TabGLM achieving an average AUC-ROC improvement of up to 5.56% over State-of-the-Art (SoTA) tabular learning methods.

1 Introduction

Real-world applications ranging from predicting sales in e-commerce to diagnosing diseases in healthcare rely on tabular data. These datasets are oftentimes a mix of numerical, categorical, and text values, presenting a unique challenge for machine learning models. Traditional approaches (Breiman 2001; Chen and Guestrin 2016; Prokhorenkova et al. 2018) as well as some early Deep Learning (DL) models (Yoon et al. 2020; Arik and Pfister 2021; Gorishniy et al. 2021; Hollmann et al. 2023) convert textual data into numerical encodings modeling only structural features from an input table, leading to loss of semantic information. Recent trends in tabular DL indicate an increase in approaches attempting modality switch from tabular to image (Sharma et al. 2019; Wang et al. 2019), text (Hegselmann et al. 2023; Arik and Pfister 2021), or graph (Alkhatib et al. 2024; Guo et al. 2021), modeling *either* semantic or structural relationships. These transformations aim to exploit the strengths of established models in vision, language, and graph domains to enhance the representation learning of tabular data. Unfortunately, modeling a single type of relationship through uni-modal transformation limits the ability of DL models in this domain to perform well on heterogeneous datasets. In addition, DL models are often prone to overfitting, especially on datasets with high dimensionality or limited samples. Thus, such models frequently struggle to outperform simple linear and tree based models. This discrepancy highlights a fundamental challenge: *effectively integrating the diverse types of information within tabular data*, while preserving the rich semantic and structural nuances.

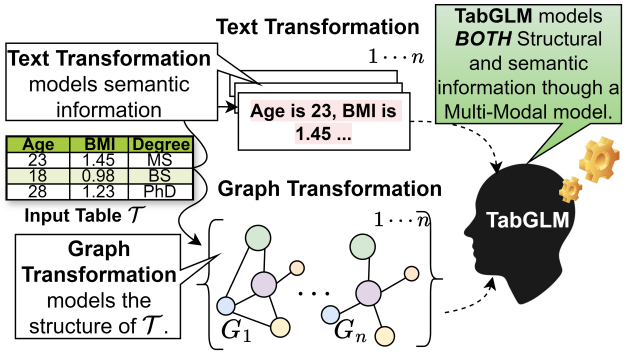

Figure 1: **Semi-Supervised Multi-Modal Tabular Deep Learning in TabGLM.** We propose a joint graph-language method that can effectively learn from heterogeneous, real-world tabular datasets by integrating structural and semantic information.

We bridge this gap by introducing **TabGLM** (**Tab**ular **G**raph **L**anguage **M**odel), a novel multi-modal architecture designed to effectively capture **both structural and semantic information in tabular data**. This is achieved by transforming each row of a tabular dataset into a graph and serialized text, and encoding it using a graph neural network (GNN) and a pretrained text encoder, respectively (Fig-

[*]These authors contributed equally.
[†]Work done as an intern at Fujitsu.

ure 1). Transforming a record into a graph encodes relationships between columns, thus modeling structure, while transforming it into text embeddings captures semantic information. The joint semi-supervised learning strategy in TabGLM, namely MUCOSA (detailed in Section 3.3), aligns the learned representations from both the graph and text encoders while adapting to downstream tasks. This alignment enhances the quality of the learnt representations by leveraging complementary information from both modalities, while acting as a regularization strategy to prevent overfitting.

To the best of our knowledge, we are the *first to introduce a multi-modal learning framework for tabular data*, with the following principal contributions -

- We introduce a **multi-modal method** that transforms each row of a tabular dataset into a graph and serialized text, capturing both structural and semantic features.
- Our joint loss (MUCOSA) assists with information fusion from the 2 modalities, while acting as a regularization mechanism to mitigate overfitting.
- TabGLM's targeted use of frozen and trainable components achieves a **significantly lower (by over 80%) parameter count** compared to State-of-the-Art (SoTA) uni-modal DL approaches.
- Extensive experiments and ablation studies validate the effectiveness of TabGLM and its key components. We **demonstrate an absolute improvement in AUC-ROC scores up to 5.56%, compared to SoTA models across 25 benchmark datasets** detailed in Section 4 of the main paper.

2 Related Work

Traditional Tabular Machine Learning Historically, the realm of tabular data modeling over the past decade has been largely dominated by conventional machine learning methods (Shwartz-Ziv and Armon 2022). Models such as Gradient Boosting (Bentéjac, Csörgő, and Martínez-Muñoz 2021), ExtraTrees (Geurts, Ernst, and Wehenkel 2006), and Random Forests (Breiman 2001) have been pivotal in learning intricate data patterns and enhancing robustness against overfitting. Notable techniques like XGBoost (Chen and Guestrin 2016) and LightGBM (Ke et al. 2017) stand out for their efficiency, optimization techniques, and scalability, making them go-to options in various applications. Logistic regression (Hosmer Jr, Lemeshow, and Sturdivant 2013) has been particularly applied to binary classification tasks due to its simplicity and interpretability. Specialized algorithms like CatBoost (Prokhorenkova et al. 2018), designed to handle categorical features seamlessly, have gained prominence. This diverse set of models contributes to a versatile toolbox, addressing the intricacies of tabular data modeling with distinct strengths and adaptability (Grinsztajn, Oyallon, and Varoquaux 2022). These traditional models provide a solid foundation for tabular data analysis, balancing interpretability, efficiency, and performance essential for real-world applications. Despite their effectiveness, these models are often limited by their reliance on handcrafted feature engineering and their inability to leverage the representation learning capabilities inherent in deep learning models.

Transformers for tabular data Following the popularity of Transformer architectures in vision and language, several methods (Hollmann et al. 2023; Arik and Pfister 2021; Zhu et al. 2023) have adapted transformers for learning from tabular datasets. For instance, FT-Transformer (Gorishniy et al. 2021) showed superior performance in tabular classification and regression tasks by separating numerical and categorical features. Additionally, Saint (Somepalli et al. 2021) introduced row-wise attention, capturing inter-sample interactions, Fastformer (Wu et al. 2021) suggested the use of additive attention which is lightweight with linear complexity, while TransTab (Wang and Sun 2022) incorporated transfer learning in tabular tasks, all using transformers as backbones. Recent advancements have specifically tailored the transformer architecture to address challenges in data imputation and cross-table learning, incorporating modifications to the attention mechanism and embedding layers (Badaro, Saeed, and Papotti 2023).

Self-supervised pretraining Furthermore, the emergence of self-supervised pretraining in the tabular domains has paved the way for novel approaches to feature extraction and representation learning, reducing the reliance on labeled data (Liu et al. 2021). Specifically, drawing inspiration from the success of pretraining in vision and language, previous studies have delved into tabular self-supervised learning (Yoon et al. 2020; Ucar, Hajiramezanali, and Edwards 2021; Somepalli et al. 2021; Bahri et al. 2021; Majmundar et al. 2022; Rubachev et al. 2022; Wang and Sun 2022). Authors in (Yoon et al. 2020; Ucar, Hajiramezanali, and Edwards 2021) introduced an auto-encoder framework with a pretext task focused on reconstructing missing elements in a table while (Bahri et al. 2021) utilized contrastive learning (Chen et al. 2020) as pretraining objective for improving generalizability of trained architectures in tabular tasks. Additionally, (Rubachev et al. 2022; Wang and Sun 2022) created a target-aware objective by incorporating label columns of tabular tasks in pretraining. Although these innovations have largely improved performance over traditional machine learning approaches, these models have been shown to particularly underperform in the presence of heterogeneous feature columns (Hegselmann et al. 2023).

Modality switch for Tabular Deep Learning Recent research has explored the conversion of tabular data into orthogonal modalities, such as text, image, and graph. TabLLM (Hegselmann et al. 2023) converted tabular data to text for few-shot classification using large language models. Although it can suffer from context loss and inefficiency when handling high-dimensional data, TabLLM successfully captures the semantic information encapsulated within columns in a table. SuperTML (Wang et al. 2019) introduced a method to transform tabular data into a super ensemble of image-based data points, enabling the use of convolutional neural networks for tabular tasks. DeepInsight (Sharma et al. 2019) proposed projecting tabular data into an image space using t-SNE, enabling the application of image classification models to tabular data. Even though this technique effectively captures underlying feature correlations, the reliance on a single-image representation and t-SNE's specific distance metric limits its ability to capture diverse and multi-

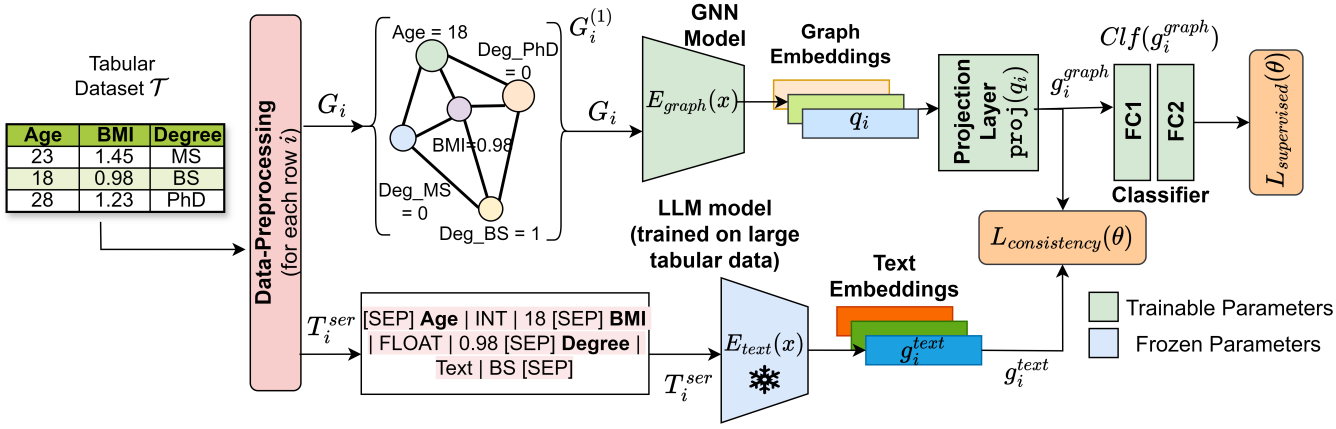

Figure 2: **Overview of our TabGLM framework.** TabGLM introduces Multi-modal Graph-Language Modeling to enable tabular learning on datasets with heterogeneous data types. Our method leverages graph and language embeddings, consistency regularization, and supervised learning to effectively adapt to diverse real-world downstream tasks.

faceted relationships inherent in complex tabular datasets. Table2Graph (Zhou et al. 2022) transforms tabular data into a unified weighted graph and IGNNet (Alkhatib et al. 2024) transforms each record into a fully-connected graph, allowing the application of graph neural networks (GNNs) for tabular data learning. Additionally, GCondNet (Margeloiu et al. 2023) transforms each column into a graph while CARTE (Kim, Grinsztajn, and Varoquaux 2024) mines entities in tables to learn from entity-centric graphs. Furthermore, models like Graph foundation models (Galkin et al. 2024; Zhang 2024) and (Sun 2023) highlight the efficacy of GNNs in capturing relational structures within tabular data. HyTrel (Chen et al. 2023b) enhances tabular data representation by integrating hypergraph structures, which can capture high-order relationships among features, but the complexity of hypergraph construction and the increased computational cost are significant challenges. Despite their innovative approach, these methods often face scalability issues with large datasets and are sensitive to the graph construction method. Additionally, even though graphs can capture the structural relationships among features in a table, they cannot capture the semantic information of the categorical and text columns, as well as the column headers. This information can provide valuable insight, which is especially valuable when learning from small datasets.

Multi-Modal Learning Multi-modal learning integrates data from multiple sources, such as text, image, video, and audio to enhance machine learning models' performance. A pivotal model in this domain is CLIP (Radford et al. 2021), which aligns text and image representations using contrastive learning, enabling effective zero-shot learning and image-text retrieval. Other significant advancements include (Hegde, Jose Valanarasu, and Patel 2023), which adapts CLIP to 3D recognition tasks through prompt tuning for language grounding, as well as (Chen et al. 2023a), which introduces cross-modal knowledge distillation, and (Ramesh et al. 2021), which introduces zero-shot text-to-image generation.

An important lesson from existing literature is that multi-modal models are capable of generalizing to downstream tasks by capturing complementary information from multiple modalities. For instance, they extract complex spatial patterns from images, semantic meaning from text, and structural relationships from graphs. We capitalize on this property to design a multi-modal model for tabular machine learning that combines the richness of graph and text modalities into a unified embedding space to improve performance on downstream ML tasks. To the best of our knowledge, we are the first to introduce multi-modal learning for tabular datasets using a single table as input across several classification based downstream tasks.

3 Method

Datasets in real-world are oftentimes heterogeneous consisting of both numerical and textual features. To this end, we introduce Tabular Graph Language Model (TabGLM) depicted in Figure 2, which tackles the aforementioned challenge by preserving both structural and semantic features enumerated in tabular datasets.

3.1 Problem Definition

Given a tabular dataset $\mathcal{T} \in \mathbb{R}^{n \times m}$ represented as a matrix of n records each with m feature columns and a label y_i, where $i \in |\mathcal{T}|$, we are tasked to predict the probability of a newly introduced record x in the test dataset to be one among the target classes y_i. To achieve this goal we train a feature representation learner $h(\mathcal{T}_i; \theta)$, which learns feature representations g_i from samples (rows) $\mathcal{T}_i$ in a training dataset $\mathcal{T}_{train}$, where $i \in [1, n]$ and total model parameters θ. The learned representations $g_i, \forall i \in [1, n]$ are then passed to a predictor $Clf(g_i)$ for downstream classification tasks. The performance of the model $h(x_i, \theta)$ on unseen records in $\mathcal{T}_{test}$ largely depends on the quality of learned representations g. Particularly, in this paper we tackle the challenges associated with tables $\mathcal{T}$ containing heterogeneous column

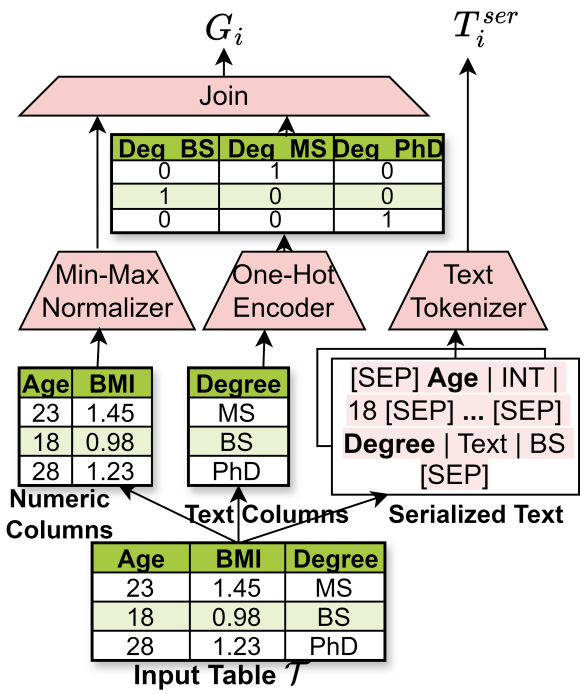

Figure 3: **Multi-Modal Representation of Tables in TabGLM,** depicting the text serialization and graph preprocessing pipelines.

types (both numeric and textual) through a multi-modal architecture as discussed in Section 3.2.

3.2 TabGLM: Tabular Graph Language Model

TabGLM introduces a multi-modal architecture as shown in Figure 2, which encodes each record in an input table into a graph (learning structural features) and serialized text (learning semantic features). As highlighted in Section 2, recent approaches employ uni-modal transformations, encoding either structural or semantic information. A uni-modal model, therefore, lacks the advantages provided by auxiliary modalities (promoting learning of only specific types of features). The multi-modal architecture of TabGLM addresses this gap and demonstrates improvements in downstream tasks by combining two complimentary modalities in a single unified architecture.

We simultaneously transform each record $\mathcal{T}_i \in \mathcal{T}$ into a fully-connected graph G_i and natural language (serialized text) T_i^{ser}. TabGLM then encodes G_i and T_i^{ser} using a graph encoder E_{graph} and a text encoder E_{text}, producing feature vectors g_i^{graph} and g_i^{text} respectively. Finally, we combine the encoded feature vectors, g_i^{graph} and g_i^{text} using a Multi-Modal Consistency Learner (MuCosa) that minimizes the feature separation between complimentary modalities (unsupervised) while adapting to downstream tasks (supervised). We detail the aforementioned components in our TabGLM architecture below, which can be decomposed into the text pipeline and the graph pipeline.

Text Pipeline The text encoder (E_{text}) encodes each record $\mathcal{T}_i \in \mathcal{T}$ into an embedding g_i^{text} with the goal of preserving the semantic information in the cell values. We achieve this by first transforming each row in $\mathcal{T}$ to serialized natural text T^{ser} as shown in Figure 3. Drawing inspiration from the authors in Hegselmann et al. (2023), we adopt the simple `text` serialization which is inexpensive while being representative. An example of this is also depicted in Figure 3 where each row in the table in represented as a templated (Hegselmann et al. 2023; Herzig et al. 2020) text (serialization). The serialized output is tokenized to convert the natural language input (each row in the input table) into tensors T^{ser} before passing to the text encoder E_{text}. The text encoder produces the text embedding g_i^{text} as shown in Equation 1, where $g_i^{text} \in \mathbb{R}^d$. Following the recent success of LLMs in tabular question answering (Liu et al. 2022; Herzig et al. 2020), TabGLM adopts the best performing pretrained text encoders in TAPAS (Herzig et al. 2020) and TAPEX (Liu et al. 2022), trained on a large number of records. The choice of the text encoder presents a trade-off between performance and computational complexity, which is elucidated through the ablation study in Section 4.4. The parameters θ_{text} of the text encoder E_{text} are kept frozen during the training process and used to produce an instance-level (row) embedding g_i^{text}**, encoding context aware features from each record**.

$$g_i^{text} = E_{text}(\texttt{tokenize}(\mathcal{T}_i^{ser}); \theta_{text}) \quad (1)$$

Graph Pipeline The Graph Encoder (E_{graph}) takes as input a fully connected graph $G(v, e)$ to learn embeddings g_i^{graph} corresponding to each row $\mathcal{T}_i \in \mathcal{T}$. The goal of E_{graph} is to encode the latent structural relationships between columns in the underlying table $\mathcal{T}$. Following the authors in Alkhatib et al. (2024), we first transform each record $\mathcal{T}_i \in \mathcal{T}$ into a graph representation $G_i(v, e)$ where each node in v encodes the value associated with a feature column in $\mathcal{T}$ and each edge in e represents the relationship between feature columns (as edge weight). During implementation, a set of edge weights W re-weights each edge in G_i while an adjacency matrix A encodes the structure of G_i. A known limitation of SoTA tabular graph learning approaches (Alkhatib et al. 2024; Zhou et al. 2022) is the ability to encode categorical features. As depicted in Figure 3, TabGLM converts columns with categorical features to numerical encodings as a preprocessing step before passing them as input to E_{graph} which is parameterized by θ_{graph} as shown in Equation 2.

$$g_i^{graph} = \texttt{proj}(E_{graph}(G_i(v, e); \theta_{graph})) \quad (2)$$

TabGLM learns a latent representation q_i for each row in $\mathcal{T}$ by adopting the popular Graph Neural Network (GNN) in Xu et al. (2019); Alkhatib et al. (2024) which employs a sequence of *message passing* (Xu et al. 2019) layers in its architecture to learn node level features. These node level features are further aggregated using *read-out layers* to learn an embedding for the input graph G_i. Several iterations of message passing during model training allows E_{graph} **to learn the structure of the table $\mathcal{T}$ by modeling the relationships between features columns** (represented as nodes v). The output latent representation q_i is projected to

a lower dimensional space using a projection layer `proj` to produce graph embeddings $g_i^{graph} = \texttt{proj}(q_i)$ as depicted in Figure 2 and expressed in Equation 2. This layer projects the graph embedding in the same dimensional space as the text embedding.

During the *training phase E_{graph} is trained from scratch in an end to end fashion while E_{text} remains frozen*, with the total parameter count of the feature extractor h (composed of E_{graph} and E_{text}) being $\theta = \theta_{graph} + \theta_{text}$. This design choice is based on the assumption that the text encoder in TAPAS / TAPEX has learnt generalizable representations from a large volume of records (26.9 billion in Herzig et al. (2020)) it was pretrained on. During *inference, TabGLM omits the forward pass through the text encoder E_{text}*, relying solely on the embeddings learnt from E_{graph}, **significantly boosting inference speeds**. Further, we show through ablation experiments in Section 4.4 that the proposed **TabGLM architecture uses only 336M parameters which is over 80% lower than SoTA approach TabLLM** (Hegselmann et al. 2023).

3.3 MuCosa: Multi-Modal Consistency Learner

The training of TabGLM proceeds in a single stage with a joint sem-supervised learning approach, MuCosa. As discussed in Section 3.2, the representations learnt from both E_{graph} and E_{text} encode orthogonal concepts with the former encoding structure and the latter encoding semantic information. To combine the learnings from both g_i^{graph} and g_i^{text} we minimize the consistency between the two modalities through a consistency loss, $L_{\text{consistency}}$ as shown in Equation 3. $L_{\text{consistency}}$ aligns the text embeddings g_i^{text} with the graph embeddings g_i^{graph} corresponding to each row in T_{train} and vice versa, in a label free fashion.

$$L_{\text{consistency}} = -\frac{1}{2n}\sum_{i=1}^{n}\left[\log\frac{\exp\left(\frac{\hat{g}_i^{\text{text}}\cdot(\overline{\hat{g}_i^{\text{graph}}})^T}{\tau}\right)}{\sum_{j=1}^{n}\exp\left(\frac{\hat{g}_i^{\text{text}}\cdot(\overline{\hat{g}_j^{\text{graph}}})^T}{\tau}\right)} + \log\frac{\exp\left(\frac{\hat{g}_i^{\text{graph}}\cdot(\overline{\hat{g}_i^{\text{text}}})^T}{\tau}\right)}{\sum_{j=1}^{n}\exp\left(\frac{\hat{g}_i^{\text{graph}}\cdot(\overline{\hat{g}_j^{\text{text}}})^T}{\tau}\right)}\right] \tag{3}$$

Here, $\hat{g}_i^{\text{text}} = \frac{g_i^{\text{text}}}{\|g_i^{\text{text}}\|_2}$ and $\hat{g}_i^{\text{graph}} = \frac{g_i^{\text{graph}}}{\|g_i^{\text{graph}}\|_2}$ represents the normalized form of the graph and text embeddings, τ (set to 0.1 following Chen et al. (2020)) denotes the temperature term and $\overline{\hat{g}_i^{\text{text}}}$, $\overline{\hat{g}_i^{\text{graph}}}$ indicates explicitly that gradients are not propagated for those terms. Additionally, we minimize a supervised loss $L_{\text{supervised}}$ between the ground truth y_i and the predicted logits from a classifier head $\hat{y}_i = Clf(g_i^{graph})$ as shown in Figure 2. Note, that the classifier head consumes only the graph embeddings to mimic the inference setting. The supervised loss can be represented as Equation 4.

$$L_{\text{supervised}} = \frac{1}{n}\sum_{i=1}^{n} H(y_i, \hat{y}_i) \tag{4}$$

Finally, TabGLM introduces joint objective, MuCosa (L) as shown in Equation 5 which combines both $L_{\text{supervised}}$ and $L_{\text{consistency}}$ as a weighted sum with the hyper-parameter λ controlling the contribution of each component to the total loss L.

$$L = (1-\lambda)L_{\text{supervised}} + \lambda L_{\text{consistency}} \tag{5}$$

4 Experiments

We conduct our experiments on a wide variety of tabular datasets (refer Section 4.1) with varying levels of data heterogeneity across a variety of downstream classification tasks (refer Section 3.2).

4.1 Datasets

To demonstrate the effectiveness of TabGLM in the presence of heterogeneous feature columns, as discussed in Section 3.2, we conduct our experiments on 25 datasets encompassing both binary and multi-class classification tasks, curated from popular papers TabLLM (Hegselmann et al. 2023), TabPFN (Hollmann et al. 2023), and large scale datasets in OpenML (Casalicchio et al. 2017). Following the principal goal of TabGLM, we consider heterogeneous datasets that encapsulate both numerical and textual columns like **Bank** ($\sim$45k records with 7 numerical and 9 categorical columns), **Creditg** (1k rows with 7 numerical and 13 categorical columns), **Heart** (918 rows with 6 numerical and 5 categorical columns) and **Income** ($\sim$48k rows with 4 numerical and 8 categorical columns), as shown in TabLLM. In addition, we use 12 datasets from OpenML, containing at least 1 numerical and 1 categorical column, including **balance-scale** (5 numerical and 1 categorical), **tic-tac-toe** (10 numerical and 10 categorical), **dress-sales** (13 numerical and 12 categorical) etc with more details in appendix. We also include datasets containing only numerical columns like **blood** (4 numerical columns), **calhousing** (8 numerical columns), **coil2000** (86 numerical columns) etc. alongside datasets containing only categorical columns like **car**, from both OpenML and TabPFN. We adopted datasets of varying sizes, with number of rows ranging from 500 (in **dress-sales**) to 45,211 (in **bank**) to demonstrate the applicability of our method to real-world large tabular datasets. Note that the multi-modal architecture in TabGLM involves a LLM encoder (Herzig et al. 2020; Liu et al. 2022) that is limited by the number of input tokens, which is 512 (from TAPAS) in our case. More details on each dataset experimented upon in Table 1 is discussed in the supplementary material.

4.2 Experimental Setup

We conduct our experiments on datasets discussed in Section 4.1 and report the average performance (AUC-ROC scores) of each model across the same 5 random seeds (kept constant across datasets) in Section 4.3. For all numerical and heterogeneous datasets, numerical columns are normalized using min-max[1] normalization while the categorical (text) columns are converted into One-Hot

[1]Scikit learn package: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Dataset	Performance (AUC-ROC)								
	TabGLM (ours)	**CatBoost**	**GB**	**LR**	**RF**	**XGBoost**	**Tab Transformer**	**FT-Transformer**	**NODE**
bank	92.07	**93.51**	92.36	86.76	92.46	92.84	90.05	92.07	92.67
blood	**78.48**	74.94	72.24	76.76	70.77	69.51	74.26	74.98	76.21
calhousing	**95.47**	93.55	92.47	90.84	93.45	81.99	83.13	93.62	93.84
car	99.40	**99.97**	99.83	78.46	99.41	99.92	98.57	98.51	99.64
coil2000	74.17	73.97	**74.66**	73.22	69.43	71.19	71.64	65.59	73.09
creditg	79.32	**80.54**	78.36	75.21	79.76	76.81	79.40	56.60	79.83
diabetes	**83.70**	82.55	82.34	82.89	81.65	79.17	82.72	82.34	82.18
heart	**93.29**	92.61	92.00	90.74	91.92	91.16	92.16	91.81	92.61
kr-vs-kp	99.43	**99.95**	99.77	99.15	99.86	99.95	99.30	86.79	99.41
mfeat-fourier	99.94	99.97	99.62	**100.00**	99.99	99.70	99.99	99.92	**100.00**
pc3	**82.82**	82.48	80.80	79.44	80.89	77.76	79.02	76.57	81.00
income	**92.59**	92.44	91.75	79.03	89.19	92.35	89.63	70.57	90.30
texture	**100.0**	99.98	99.93	99.87	99.94	99.96	99.98	99.94	99.94
balance-scale	**99.10**	92.35	98.37	93.11	84.89	98.99	91.60	91.03	94.41
mfeat-karhunen	**99.88**	99.86	99.79	99.52	99.71	98.69	99.56	98.85	**99.88**
mfeat-morphological	**96.99**	96.20	96.01	95.74	95.53	96.12	95.75	96.33	96.34
mfeat-zernike	**98.09**	97.59	97.16	97.74	96.72	97.35	98.02	97.76	97.49
cmc	**74.45**	72.56	72.89	70.41	70.52	73.00	69.96	71.56	73.88
tic-tac-toe	99.85	99.92	99.81	72.00	96.12	**99.98**	70.90	72.76	98.82
vehicle	**94.50**	93.02	92.33	88.79	93.23	92.84	93.19	90.50	91.61
eucalyptus	**91.95**	88.59	89.31	87.45	90.11	90.04	88.27	89.98	89.70
analcatdata_author	**58.96**	55.89	54.61	53.56	53.20	57.43	53.63	53.94	55.50
MiceProtein	99.98	**99.99**	99.97	99.51	99.85	99.98	99.91	99.41	99.97
steel-plates-fault	94.52	96.51	**96.26**	91.35	91.71	96.56	91.91	91.92	94.45
dress-sales	**57.89**	56.96	55.93	55.94	53.72	57.23	53.38	54.41	52.62
Average	**89.47**	88.64	88.34	84.69	86.96	87.62	85.84	83.91	88.22

Table 1: **Comparison of performance (AUC-ROC) of existing approaches in tabular Machine Learning against TabGLM**. Our proposed method TabGLM achieves significant performance gains across 25 classification datasets. The best performing model is in **bold** while the second best is underlined.

encodings (refer ablation in supplementary material) to create a numeric dataset for graph transformation. For the text transformation, each record in the table is converted to serialized text following the tokenizer in TAPAS (Herzig et al. 2020). We chose TAPAS based on ablation experiments on the choice of LLMs in Section 4.4. For datasets that contain only categorical columns, our TabGLM method uses only the text pipeline, utilizing only the semantic information present in such datasets. Models for all datasets are trained on a fixed set of hyperparameters with an initial learning rate of $1e^{-4}$, batch size of 256 and weighting the consistency loss at 20% ($\lambda = 0.2$). All experiments are conducted on 4 NVIDIA V100 GPUs with additional details on the experiment setup in the Appendix and code released at https://github.com/amajee11us/TabGLM.

4.3 Results

At first, we compare the performance of TabGLM with traditional linear and tree based Machine Learning models like CatBoost (Prokhorenkova et al. 2018), XGBoost (Chen and Guestrin 2016), Gradient Boosting (GB) (Ke et al. 2017), Random Forest (RF) (Breiman 2001) and Logistic Regression (LR). Our results in Table 1 show that TabGLM demonstrates significant increase in AUROC of 4.77% over LR, 2.51% over RF etc. outperforming such techniques across 25 downstream tabular classification tasks. However, for simple datasets with lower number of feature columns like **kr-vs-kp**, **pc3** etc., tree based models (CatBoost) continue to show dominance in performance.

Secondly, we compare the performance of TabGLM with SoTA tabular DL models like FT-Transformer (Gorishniy et al. 2021), TabTransformer (Huang et al. 2020) and NODE (Popov, Morozov, and Babenko 2019). TabGLM consistently outperforms tabular DL models like FT-Transformer by 5.56%, TabTransformer by 3.64% and NODE by 1.26% respectively. Finally, we compare the performance of TabGLM against SoTA uni-modal DL architectures like IGNNet (table-to-graph) and TabLLM (table-to-text) on 9 datasets in the benchmark introduced in TabLLM Hegselmann et al. (2023). We observe that TabGLM outperforms TabLLM by 1.35% and IGNNet by 7.96% respectively on the benchmark datasets in (Hegselmann et al. 2023), summarized in Table 2. The above results indicate a strong generalization of the proposed TabGLM architecture to a variety of downstream tasks, establishing TabGLM as a strong choice for Tabular Deep Learning under feature heterogeneity.

Dataset	Tabular DL Methods			
	TabGLM (multi-modal)	**IGNNet** (graph)	**TabLLM** (text)	**TabPFN**
bank	92.07	91.11	91.20	91.19
blood	**78.48**	74.09	74.03	77.01
calhousing	**95.47**	94.79	95.38	95.31
car	99.40	50.16	**99.99**	99.53
creditg	79.32	71.99	70.82	**80.79**
diabetes	**83.70**	77.79	80.40	73.67
heart	93.29	92.06	**94.21**	82.60
jungle	88.98	88.98	93.00	87.36
income	**92.59**	90.76	92.19	90.14
Average	**89.26**	81.30	87.91	86.40

Table 2: **Comparison of performance (AUC-ROC) of TabGLM against benchmark datasets in TabLLM** (Hegselmann et al. 2023). Results from all methods are averaged over five seeds.

Dataset	Methods		
	TabLLM	**TabGLM** (w TAPEX encoder)	**TabGLM** (w TAPAS encoder)
Param. Count	2.9B	336M	129M
blood	71.78	77.57	**78.48**
calhousing	95.00	95.29	**95.47**
creditg	78.56	78.72	**79.32**

Table 3: **Ablation on the Choice of LLM** architecture for the text transformation module of TabGLM.

4.4 Ablation Study

Multi-Modal vs. Uni-Modal training: The core contribution of TabGLM lies in its multi-modal architecture for tabular representation learning. To evaluate its components, we decompose it into two uni-modal architectures: *Graph only* (using only the graph encoder E_{graph}) and *Text only* (using only the text encoder E_{text}), based on the choice of the feature extractor during both training and inference. Their performance is compared against the complete multi-modal TabGLM training recipe, with results summarized in Table 4. The *Graph only* pipeline employs the GNN from (Alkhatib et al. 2024), while the *Text only* pipeline uses the BART-based TAPAS (Herzig et al. 2020) encoder. Both pipelines use the same classifier head (Section 3.2) for downstream tasks. In **Text only**, the encoder is frozen, and only the classifier head is trained, whereas in *Graph only*, both the encoder and classifier head are trained, to ensure fair comparison with TabGLM, where the text encoder remains frozen during training. Experiments on three representative datasets—**pc3** (numerical), **bank** (balanced numerical and categorical), and **creditg** (categorical-heavy)—show that TabGLM's multi-modal design consistently outperforms its uni-modal variants, underscoring the value of modality fusion for learning from heterogeneous tables.

Choice of LLM architecture for Text Transformation: The choice of the pretrained LLM architecture plays a crucial role in improving the model performance of TabGLM. While larger LLMs like (Sun 2023; Hegselmann et al. 2023; Liu et al. 2022) ($\geq$7 billion parameters) can encode superior semantic features in complex text, it also adds a significant computational overhead. Additionally, their benefits may be negligible when dealing with simpler semantic content. To address this trade off, we conducted an ablation experiment by varying the architecture of the text encoder (E_{text}) across three popular LLM models - TAPAS (Herzig et al. 2020), TAPEX (Liu et al. 2022) and TabLLM (Hegselmann et al. 2023). For all three settings we adopt the complete multi-modal training strategy, modifying only the text encoder E_{text}. The results from this experiment, shown in Table 3, highlight that TAPAS [2], a smaller parameter count, BERT (Devlin et al. 2018) based text encoder, outperforms other larger models like TAPEX (Liu et al. 2022). We thus adopt this architecture for the text transformation pipeline in TabGLM.

Dataset	**Graph Trans.** (E_{graph})	**Text Trans.** (E_{text})	**AUCROC**
pc3	✓		77.04
		✓	78.24
	✓	✓	**82.82**
bank	✓		91.11
		✓	90.52
	✓	✓	**92.07**
Creditg	✓		71.99
		✓	77.36
	✓	✓	**79.32**

Table 4: **Ablations on the graph and text components of the proposed TabGLM approach**. Results are averaged over five seeds.

5 Conclusion

In conclusion, TabGLM marks a pivotal advancement in deep learning for tabular data by adeptly handling the inherent heterogeneity of these datasets. By transforming each row into both a fully connected graph and serialized text, and leveraging a graph neural network alongside a pretrained text encoder, TabGLM captures rich structural and semantic information. Its joint multi-modal, semi-supervised learning objective enhances generalization and feature representation. The model's flexible graph-text pipeline efficiently processes diverse feature types, resulting in a streamlined architecture with significantly fewer parameters than state-of-the-art approaches. Evaluations across 25 benchmark datasets reveal substantial performance gains in AUC-ROC scores, with TabGLM surpassing both existing deep learning and traditional machine learning methods. These findings underscore the power of multi-modal architectures for tabular data, opening new horizons for innovative applications across various domains.

[2]We adopt the TAPAS-base model from https://huggingface.co/google/tapas-base

Acknowledgments

We gratefully thank anonymous reviewers for their valuable comments. We would also like to extend our gratitude to the members of the AI Lab at Fujitsu Research of America (FRA) for their valuable feedback and constructive criticism during the development of the project. Additionally, Anay Majee would like to thank the leadership at FRA for providing the opportunity to intern at FRA.

References

Alkhatib, A.; Ennadir, S.; Bostrom, H.; and Vazirgiannis, M. 2024. Interpretable Graph Neural Networks for Tabular Data. In *ICLR 2024 Workshop on Data-centric Machine Learning Research (DMLR): Harnessing Momentum for Science*.

Arik, S. Ö.; and Pfister, T. 2021. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 6679–6687.

Badaro, G.; Saeed, M.; and Papotti, P. 2023. Transformers for Tabular Data Representation: A survey of models and applications. *Transactions of the Association for Computational Linguistics*, 11: 227–249.

Bahri, D.; Jiang, H.; Tay, Y.; and Metzler, D. 2021. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*.

Bentéjac, C.; Csörgő, A.; and Martínez-Muñoz, G. 2021. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54: 1937–1967.

Breiman, L. 2001. Random forests. *Machine learning*, 45: 5–32.

Casalicchio, G.; Bossek, J.; Lang, M.; Kirchhoff, D.; Kerschke, P.; Hofner, B.; Seibold, H.; Vanschoren, J.; and Bischl, B. 2017. OpenML: An R package to connect to the machine learning platform OpenML. *Computational Statistics*, 32(3): 1–15.

Chen, M.; Xing, L.; Wang, Y.; and Zhang, X. 2023a. Enhanced Multimodal Representation Learning with Cross-modal KD. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chen, P.; Sarkar, S.; Lausen, L.; Srinivasan, B.; Zha, S.; Huang, R.; and Karypis, G. 2023b. HyTrel: Hypergraph-enhanced Tabular Data Representation Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, 1597–1607. PMLR.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Galkin, M.; Yuan, X.; Mostafa, H.; Tang, J.; and Zhu, Z. 2024. Towards Foundation Models for Knowledge Graph Reasoning. In *The Twelfth International Conference on Learning Representations*.

Geurts, P.; Ernst, D.; and Wehenkel, L. 2006. Extremely randomized trees. *Machine learning*, 63: 3–42.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943.

Grinsztajn, L.; Oyallon, E.; and Varoquaux, G. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35: 507–520.

Guo, X.; Quan, Y.; Zhao, H.; Yao, Q.; Li, Y.; and Tu, W.-W. 2021. TabGNN: Multiplex Graph Neural Network for Tabular Data Prediction. In *DLP-KDD*.

Hegde, D.; Jose Valanarasu, J. M.; and Patel, V. M. 2023. CLIP goes 3D: Leveraging Prompt Tuning for Language Grounded 3D Recognition. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*.

Hegselmann, S.; Buendia, A.; Lang, H.; Agrawal, M.; Jiang, X.; and Sontag, D. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International Conference on Artificial Intelligence and Statistics*, 5549–5581. PMLR.

Herzig, J.; Nowak, P. K.; Müller, T.; Piccinno, F.; and Eisenschlos, J. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Hollmann, N.; Müller, S.; Eggensperger, K.; and Hutter, F. 2023. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*.

Hosmer Jr, D. W.; Lemeshow, S.; and Sturdivant, R. X. 2013. *Applied logistic regression*, volume 398. John Wiley & Sons.

Huang, X.; Khetan, A.; Cvitkovic, M.; and Karnin, Z. 2020. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. arXiv:2012.06678.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.

Kim, M. J.; Grinsztajn, L.; and Varoquaux, G. 2024. CARTE: Pretraining and Transfer for Tabular Learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *PMLR*, 23843–23866.

Liu, Q.; Chen, B.; Guo, J.; Ziyadi, M.; Lin, Z.; Chen, W.; and Lou, J.-G. 2022. TAPEX: Table Pre-training via Learning a Neural SQL Executor. In *International Conference on Learning Representations*.

Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; and Tang, J. 2021. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, 35(1): 857–876.

Majmundar, K.; Goyal, S.; Netrapalli, P.; and Jain, P. 2022. Met: Masked encoding for tabular data. *arXiv preprint arXiv:2206.08564*.

Margeloiu, A.; Simidjievski, N.; Lio, P.; and Jamnik, M. 2023. GCondNet: A Novel Method for Improving Neural Networks on Small High-Dimensional Tabular Data. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Popov, S.; Morozov, S.; and Babenko, A. 2019. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*.

Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; and Gulin, A. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*.

Ramesh, A.; Pavlov, M.; Goh, G.; Gray, S.; Voss, C.; Radford, A.; Chen, M.; and Sutskever, I. 2021. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*.

Rubachev, I.; Alekberov, A.; Gorishniy, Y.; and Babenko, A. 2022. Revisiting pretraining objectives for tabular deep learning. *arXiv preprint arXiv:2207.03208*.

Sharma, A.; Vans, E.; Shigemizu, D.; Boroevich, K. A.; and Tsunoda, T. 2019. DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific Reports*, 9: 11399.

Shwartz-Ziv, R.; and Armon, A. 2022. Tabular data: Deep learning is not all you need. *Information Fusion*, 81: 84–90.

Somepalli, G.; Goldblum, M.; Schwarzschild, A.; Bruss, C. B.; and Goldstein, T. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pretraining. *arXiv preprint arXiv:2106.01342*.

Sun, e. a. 2023. Graph Propagation Transformer for Graph Representation Learning. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Ucar, T.; Hajiramezanali, E.; and Edwards, L. 2021. Subtab: Subsetting features of tabular data for self-supervised representation learning. *Advances in Neural Information Processing Systems*, 34: 18853–18865.

Wang, P.; Li, K.; Gao, J.; and Zhang, C. 2019. SuperTML: Two-Dimensional Word Embedding for the Precognition on Structured Tabular Data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2327–2335.

Wang, Z.; and Sun, J. 2022. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35: 2902–2915.

Wu, C.; Wu, F.; Qi, T.; Huang, Y.; and Xie, X. 2021. Fastformer: Additive attention can be all you need. *arXiv preprint arXiv:2108.09084*.

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2019. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*.

Yoon, J.; Zhang, Y.; Jordon, J.; and van der Schaar, M. 2020. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33: 11033–11043.

Zhang, e. a. 2024. Graph Neural Network contextual embedding for Deep Learning on tabular data. *Neural Networks (NN)*.

Zhou, K.; Liu, Z.; Chen, R.; Li, L.; Choi, S.-H.; and Hu, X. 2022. Table2Graph: Transforming Tabular Data to Unified Weighted Graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 2420–2426.

Zhu, B.; Shi, X.; Erickson, N.; Li, M.; Karypis, G.; and Shoaran, M. 2023. XTab: Cross-table Pretraining for Tabular Transformers. *arXiv preprint arXiv:2305.06090*.