

Zero-Shot Tabular Learning via Hypergraph Reasoning with LLMs

Geon Lee Kyungho Kim Dongwon Choi
Juyeon Kim Shinhwan Kang Kijung Shin
KAIST

{geonlee0325, kkyungho, cookie000215, juyeonkim,
shinhwan.kang, kijungs}@kaist.ac.kr

Abstract

Tabular data are prevalent across diverse domains such as healthcare and finance. Zero-shot tabular learning aims to make predictions without labeled data. However, existing LLM-based approaches rely on row-wise prompting, resulting in high inference cost and limited use of inter-row dependencies. In this work, we propose TABHGR⁺, an efficient relation-aware zero-shot framework for tabular data that leverages LLM-generated rules to label and group rows. Specifically, based on column and task descriptions, we use LLMs to generate rules that assign weak labels to all rows. To refine these noisy weak labels, we construct a hypergraph by grouping semantically related rows using LLM-generated rules and perform hypergraph-based label propagation. Lastly, we further enhance the LLM-generated hypergraph for propagation, which is potentially sub-optimal, by assigning label-adaptive weights to hyperedges. By requiring only a constant number of LLM calls independent of dataset size, TABHGR⁺ substantially reduces inference cost while achieving higher classification accuracy than zero-shot and strong few-shot baselines.

1 Introduction

Tabular data are ubiquitous across real-world domains such as healthcare, finance, and material science. Despite their prevalence, effective learning from tabular data often relies on labeled supervision, which is expensive, unavailable, or impractical to obtain. These challenges motivate *zero-shot tabular learning*, which aims to perform predictions on a new task without labeled examples or task-specific fine-tuning.

Large language models (LLMs), pretrained on broad world knowledge, offer strong potential for zero-shot reasoning. Recent work applies LLMs to tabular data by prompting the model with individual rows (tuples) (Slack and Singh, 2023; Hegselmann et al., 2023). However, such *row-wise* reason-

ing relies on per-row prompts and incurs substantial inference cost that grows with the dataset size. In addition, it overlooks *inter-row* dependencies, which are prevalent in real-world tabular data. In practice, rows are often semantically related, such as patients with similar clinical profiles or financial records sharing risk patterns, which are not fully captured by row-wise predictions.

In this work, we explore a new way of leveraging LLMs in zero-shot tabular learning to address these limitations. Rather than prompting LLMs on a per-row basis, we use them to generate attribute (column)-based rules not just for weak labeling but also for grouping related rows, which together enable efficient, relation-aware reasoning.

Specifically, we first use an LLM to produce a small set of attribute-based rules that characterize each class. These rules are applied to all rows to produce initial *weak labels*. This requires only a constant number of LLM prompts based on attribute and class descriptions, independent of dataset size. However, the resulting weak labels are inevitably noisy, as they rely on coarse rules.

To refine these noisy labels, we then exploit inter-row relations that are not explicitly given but are inferred by LLMs. Specifically, we use LLMs to produce attribute-based rules that group related rows based on their semantic understanding of attributes. These groups, which correspond to *hyperedges*, together form a *hypergraph* that models higher-order relations among rows. Weak labels are then propagated over this hypergraph, refining them by aggregating information while accounting for these higher-order relations. This propagation requires only parameter-free forward computations on the hypergraph, without gradient-based optimization.

Lastly, we further enhance the LLM-generated hypergraph for propagation, which may be sub-optimal, by assigning hyperedge weights. These weights are label-adaptive, motivated by our observation that the effectiveness of a hyperedge varies

across weak labels. Specifically, we observe that some hyperedges facilitate the propagation of certain weak labels while degrading others. By selectively emphasizing hyperedges that are beneficial for each weak label, we achieve more effective weak label aggregation, leading to improved zero-shot prediction accuracy.

These ideas form TABHGR⁺, our proposed method for zero-shot tabular learning. Experiments on 9 benchmarks show that TABHGR⁺ outperforms recent zero-shot methods and strong few-shot baselines, while requiring substantially lower inference cost than row-wise LLM prediction.

Our contributions are summarized as follows:

- **Novel usage of LLMs.** For zero-shot tabular learning, we use LLMs to produce attribute-based rules not just for labeling but also for grouping, with constant inference cost independent of dataset size.
- **Method.** We propose TABHGR⁺, a zero-shot tabular learning framework that integrates LLM-generated weak labels and higher-order inter-row relations through hypergraph-based propagation with adaptive weights.
- **Experiments.** We conduct extensive experiments on 9 benchmark datasets and demonstrate that TABHGR⁺ outperforms zero-shot LLM-based methods and strong few-shot baselines.

Our code and datasets are available at <https://github.com/geon0325/TabHGR>. All code is hosted openly for academic use.

2 Related Work

We review prior work relevant to this study.

LLMs for Tabular Learning. LLMs enable rich semantic reasoning over attributes, opening new opportunities for tabular learning. Most existing approaches apply LLMs by serializing each row into natural language prompts for direct prediction or fine-tuning (Hegselmann et al., 2023; Dinh et al., 2022; Nam et al., 2023a), or by deriving interpretable class-level rules and rationales applicable to unseen rows (Han et al., 2024; Slack and Singh, 2023). Notably, prior work primarily focuses on few-shot settings with a small number of labeled examples (Slack and Singh, 2023; Nam et al., 2023a; Dinh et al., 2022; Hegselmann et al., 2023). Recent efforts have begun exploring zero-shot tabular learning by independently prompting LLMs on individual rows (Slack and Singh, 2023; Hegselmann et al., 2023), but such approaches overlook im-

plicit relationships between rows. To address this limitation, we use LLMs to extract higher-order relationships among rows, even without requiring row-wise inputs.

LLMs for Hypergraphs. Recent studies have explored integrating LLMs with graphs (Li et al., 2023; Jin et al., 2024; Tang et al., 2024; Zhang et al., 2024) and extended these efforts to hypergraphs. Early approaches focus on latent-level fusion, combining representations from LLMs and hypergraph encoders to capture semantic and structural information (Bazaga et al., 2024; Li et al., 2025a). More recent studies use LLMs to reason over small hypergraphs via textualized higher-order relations (Feng et al., 2025), or to generate hypergraphs that preserve structural patterns in real-world data (Gu et al., 2025). In contrast, in our work, LLMs generate hypergraphs by grouping semantically related rows, grounded in reasoning over tabular attributes rather than preserving structural patterns.

Hypergraphs for Tabular Data. Prior work has explored modeling tabular data using hypergraphs. For instance, HyTrel (Chen et al., 2023) represents each cell as a node and forms hyperedges over rows, columns, or the entire table to encode explicit structural dependencies. Other approaches construct row-level hypergraphs by linking each entity to its top- K related rows using BM25-style relevance (Du et al., 2022) or to nearest neighbors in feature space (Li et al., 2025b). However, these methods rely on heuristic similarity measures and lack semantic reasoning over attributes, which our approach leverages through LLMs.

Weakly-Supervised Learning. In the broader machine learning context beyond tabular learning, weakly supervised learning aims to train models by leveraging and combining weak, noisy labels that are often generated by multiple rules, programs, or even LLMs (Ratner et al., 2016, 2017; Zhou, 2018; Smith et al., 2024). Prior work refines labels through various strategies, including modeling inter-label dependencies (Ratner et al., 2017), accounting for source reliability (Rühling Cachay et al., 2021), and integrating foundation model embeddings (Chen et al., 2022). The most closely related work is (Pukdee et al., 2023), which refines given weak labels via label propagation on a pairwise graph constructed using simple mechanisms, such as thresholding Euclidean distances, for datasets composed of purely numerical feature vectors. In contrast, our approach exploits LLMs

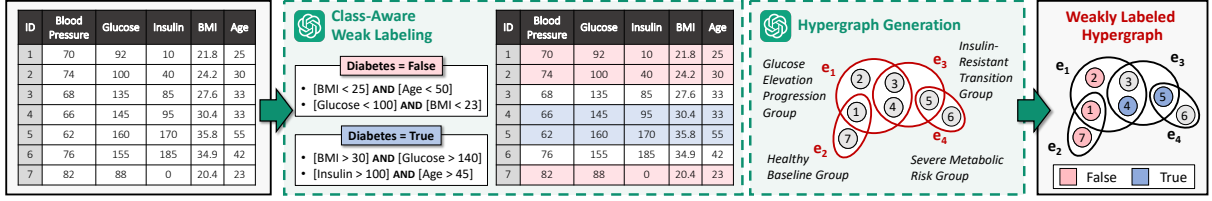


Figure 1: Given an unlabeled tabular dataset, TABHGR employs an LLM (1) to derive attribute-based condition rules for weak label assignment and (2) to identify semantically related row groups for hypergraph construction, yielding a weakly labeled hypergraph.

to generate both weak labels and a hypergraph through semantic reasoning over tabular attributes.

3 Proposed Method: TABHGR⁺

In this section, we first define the problem (Section 3.1). Then, we present TABHGR⁺, consisting of three steps: LLM-based weak labeling (Section 3.2), label refinement via hypergraph construction and propagation (Section 3.3), and hypergraph enhancement through label-adaptive hyperedge weighting (Section 3.4).

3.1 Problem Statement

Let $\mathcal{D} = \{\mathbf{s}_i\}_{i=1}^N$ be a tabular dataset with N rows and heterogeneous attributes (e.g., categorical, numerical, or ordinal), where each row $\mathbf{s}_i \in \mathcal{X}$ represents an entity. Let d_{task} and d_{attr} denote text descriptions of the target task and attributes, and let $\mathcal{Y} = \{1, \dots, C\}$ be the label space. We consider a *transductive zero-shot* setting: no ground-truth labels are available, while the unlabeled rows in $\mathcal{D}$ are jointly accessible during inference. The goal of *zero-shot tabular learning* is to learn a classifier $f: \mathcal{X} \rightarrow \mathcal{Y}$ that predicts $\hat{y}_i = f(\mathbf{s}_i)$ without labeled training data. Supervision is replaced by external semantic knowledge from off-the-shelf LLMs, which bridge the input space $\mathcal{X}$ and label space $\mathcal{Y}$.

3.2 LLM-Based Weak Labeling

In the absence of labeled data, we prompt an LLM to act as a *domain expert* that characterizes each class using attribute-based conditions. The prompt includes the task description, attribute descriptions, and a small random set of unlabeled rows to ground its reasoning (see Appendix B).

For each class $c \in \{1, \dots, C\}$, the LLM generates L interpretable *condition rules* over attributes:

$$\begin{aligned} \mathcal{R}^{(c)} &= \{r_1^{(c)}, r_2^{(c)}, \dots, r_L^{(c)}\} \\ &= \text{LLM}(\mathcal{P}_{\text{class}}(c, L, d_{\text{task}}, d_{\text{attr}})). \end{aligned}$$

where $\mathcal{P}_{\text{class}}$ denotes a prompt template that instructs the LLM to describe class c by producing L

condition rules based on the task and attribute information. As illustrated in Figure 1, each rule $r_j^{(c)} \in \mathcal{R}^{(c)}$ is a conjunction of attribute-value conditions that characterize class c (e.g., [insulin > 150] and [BMI > 30] for the class diabetes = true). We define the complete set of rules across all classes as $\mathcal{R} = \bigcup_{c=1}^C \mathcal{R}^{(c)}$ where $|\mathcal{R}| = C \cdot L$.

Then, each row $\mathbf{s}_i \in \mathcal{D}$ is evaluated against every generated rule, yielding a binary *weak label*:

$$x_{i,j}^{(c)} = \begin{cases} 1, & \text{if } \mathbf{s}_i \text{ satisfies } r_j^{(c)}, \\ 0, & \text{otherwise.} \end{cases}$$

This induces a weak label assignment vector associated with rule $r_j^{(c)}$:

$$\mathbf{x}_j^{(c)} = [x_{1,j}^{(c)}, \dots, x_{N,j}^{(c)}]^\top \in \{0, 1\}^N. \quad (1)$$

We obtain class-level predictions by aggregating the weak label assignments across rules. Specifically, the compatibility score of class c for row $\mathbf{s}_i$ is computed by summing the weak labels over the L rules associated with class c :

$$z_i^{(c)} = \sum_{j=1}^L [\mathbf{x}_j^{(c)}]_i. \quad (2)$$

If necessary, we can obtain predictions by normalizing these scores across all classes:

$$\hat{y}_i^{(c)} = \frac{z_i^{(c)}}{\sum_{c'=1}^C z_i^{(c')}}. \quad (3)$$

Importantly, weak labeling requires only a constant number of LLM prompts without feeding individual rows, making it independent of the row count and more efficient than row-wise prompting. However, the resulting weak labels are derived from coarse rules and are therefore inherently noisy. Moreover, they are assigned independently to rows and thus fail to capture inter-row dependencies.

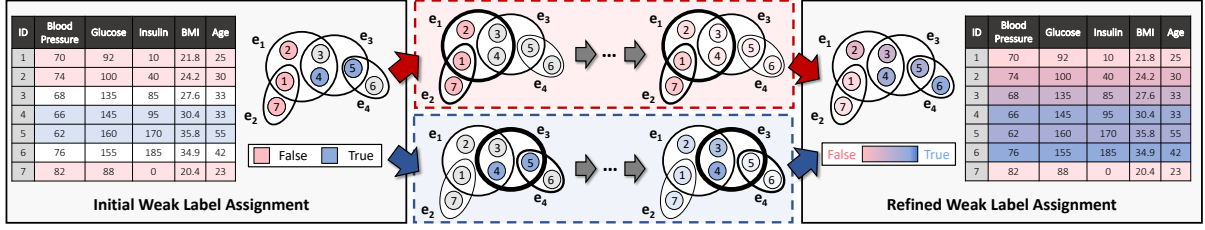


Figure 2: Given the initial weakly labeled hypergraph, TABHGR⁺ propagates each weak label independently over a label-adaptive hypergraph, where hyperedge weights (indicated by edge thickness) reflect their effectiveness for the target label. The resulting label-wise propagation outputs are then aggregated into refined weak label assignment probabilities, capturing higher-order inter-row dependencies.

3.3 Hypergraph-Based Label Refinement

To address the limitations of weak labels, we refine them by leveraging inter-row relations. We first use an LLM to identify semantically related groups of rows, which induces a hypergraph. Weak labels are then propagated over this hypergraph to refine their assignment probabilities.

Hypergraph Construction using LLMs. In tabular data, rows often share attribute patterns, and jointly considering their weak labels can facilitate refinement. To capture such dependencies, we prompt the LLM to construct attribute-based condition rules that define *semantic groups*, which are subsets of related rows, as follows:

$$\begin{aligned} \mathcal{R}_{\text{group}} &= \{\tilde{r}_1, \tilde{r}_2, \dots, \tilde{r}_K\} \\ &= \text{LLM}(\mathcal{P}_{\text{group}}(K, d_{\text{task}}, d_{\text{attr}})). \end{aligned}$$

where $\mathcal{P}_{\text{group}}$ denotes a prompt template that instructs the LLM to generate K semantic grouping rules (see Appendix B for the prompt template). Each rule $\tilde{r}_k \in \mathcal{R}_{\text{group}}$ is a conjunction of attribute-value conditions that defines a row group e_k . For example, in Figure 1, $\tilde{r}_3$ is characterized by $[70 < \text{insulin} < 175]$ and $[\text{BMI} > 27]$. The rows satisfying this rule form the semantic group e_3 , referred to as the *insulin-resistant transition group*. Note that row grouping requires only a few examples in LLM prompts, rather than all rows.

These groups define a hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes are rows ($|\mathcal{V}| = N$) and hyperedges are LLM-induced semantic groups.

Hypergraph-based Label Propagation. Given the constructed hypergraph, TABHGR refines weak label assignments by propagating them over the hypergraph. Let $H \in \{0, 1\}^{|\mathcal{V}| \times |\mathcal{E}|}$ denote the *incidence matrix* of $\mathcal{G}$, where $H_{i,k} = 1$ if row s_i belongs to group e_k . The node and hyperedge degree matrices are defined as $D_V = \text{diag}(H\mathbf{1}) \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ and $D_E = \text{diag}(H^\top \mathbf{1}) \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$, respectively.

For each rule $r_j^{(c)} \in \mathcal{R}$, we use its initial weak label assignment vector $\mathbf{x}_j^{(c,0)} = \mathbf{x}_j^{(c)}$ (Eq. (1)) as the initialization for propagation. These weak labels are iteratively propagated over the hypergraph as:

$$\mathbf{x}_j^{(c,t+1)} = \alpha D_V^{-1} H D_E^{-1} H^\top \mathbf{x}_j^{(c,t)} + (1 - \alpha) \mathbf{x}_j^{(c,0)},$$

where $\mathbf{x}_j^{(c,t)} \in \mathbb{R}^{|\mathcal{V}|}$ denotes the propagated weak label probability vector at the t^{th} iteration, and $\alpha \in [0, 1]$ controls the propagation strength. After T iterations, we obtain the refined vector $\mathbf{x}_j^{(c,T)} \in \mathbb{R}^{|\mathcal{V}|}$. Note that this process requires only a forward pass on the hypergraph, without any training.

For prediction, we compute class-level compatibility scores using the refined labels by replacing the initial assignments $\mathbf{x}_j^{(c,0)} (= \mathbf{x}_j^{(c)})$ with the propagated vectors $\mathbf{x}_j^{(c,T)}$ in Eq. (2).

In essence, label propagation refines each row’s weak labels by aggregating labels from related rows while accounting for higher-order relationships captured by the hypergraph. This improves zero-shot performance, as shown in Section 4.

3.4 Label-Adaptive Hyperedge Weighting

Although LLM-generated hypergraphs use semantic attribute reasoning, they may be suboptimal for propagation. We therefore enhance them with label-adaptive hyperedge weights.

Analysis of Label-Structure Dependency. We first analyze how each hyperedge contributes to weak label propagation. Intuitively, a hyperedge may help propagate some weak labels while being ineffective or harmful for others.

To quantify this effect, we define *hyperedge effectiveness* by measuring the change in prediction quality when a hyperedge is removed. Specifically, for a hyperedge $e_k \in \mathcal{E}$ and a weak-label

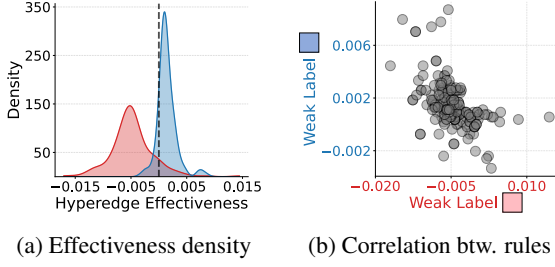


Figure 3: The distribution of hyperedge effectiveness varies across weak labels (■ and ■). Moreover, hyperedge effectiveness with respect to weak labels is not necessarily positively correlated and may be weakly correlated or even negatively correlated.

rule $r_j^{(c)} \in \mathcal{R}$, we define:

$$\Delta(e_k | r_j^{(c)}) = \mathcal{M}(\mathbf{x}_j^{(c,T)}(\mathcal{G}), Y) - \mathcal{M}(\mathbf{x}_j^{(c,T)}(\mathcal{G}_{\setminus e_k}), Y), \quad (4)$$

where $\mathbf{x}_j^{(c,T)}(\mathcal{G})$ denotes the propagated weak label scores obtained on hypergraph $\mathcal{G}$ under rule $r_j^{(c)}$, and $\mathcal{G}_{\setminus e_k}$ is the hypergraph with hyperedge e_k removed. Here, $\mathcal{M}(\cdot, \cdot)$ is a scalar performance metric instantiated as AUROC and evaluated against ground-truth labels Y . This definition is used only for analysis to study how weak label semantics depend on hypergraph structure.¹ A positive $\Delta(e_k | r_j^{(c)})$ indicates that e_k supports propagation for rule $r_j^{(c)}$, while a negative value indicates an adverse effect.

As shown in Figure 3a, hyperedge effectiveness varies across weak labels, and Figure 3b shows that these patterns can be weakly or even negatively correlated. Thus, hyperedges useful for one weak label may be ineffective or harmful for others, making the constructed hypergraph suboptimal for universal weak-label propagation.

Motivated by this analysis, we introduce label-adaptive hyperedge weights, enabling propagation to focus on effective hyperedges. These weights are estimated without ground-truth labels.

Hyperedge Effectiveness Estimation. In the zero-shot setting, ground-truth labels Y are unavailable, making direct evaluation of hyperedge effectiveness (Eq. (4)) infeasible. To address this, we first estimate hyperedge effectiveness in an *unsupervised* manner by using the predicted labels:

$$\hat{Y} = \{\hat{c}_i\}_{i=1}^N, \quad \hat{c}_i = \arg \max_{c \in \{1, \dots, C\}} \hat{y}_i^{(c)},$$

using Eq. (3) as a proxy for ground-truth labels.

¹Since ground-truth labels are unavailable in the zero-shot setting, this score is later estimated in an unsupervised manner.

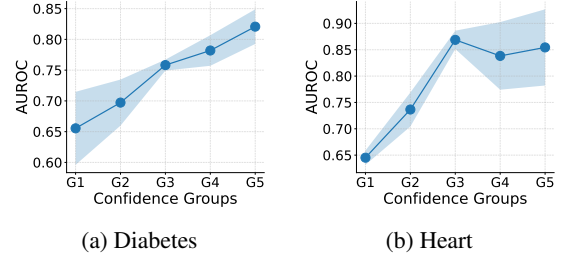


Figure 4: Rows are grouped from G1 to G5 by confidence, defined as the margin between the top two compatibility scores (Eq. (2)). Since propagation is more accurate on high-confidence rows, we use them to estimate hyperedge effectiveness.

To ensure reliability, we define each row’s confidence as the margin between its top two compatibility scores (Eq. (2)), i.e., $\max_c z_i^{(c)} - \max_{c' \neq c} z_i^{(c')}$. A larger margin indicates stronger class separation. Then, we retain the top- $q\%$ most confident rows, forming $\mathcal{D}_q \subseteq \mathcal{D}$ with weak labels $\hat{Y}_q$.

As shown in Figure 4, hypergraph-based propagation (Section 3.3) is more accurate on high-confidence rows. We therefore use them as reliable proxies for estimating hyperedge effectiveness:

$$\tilde{\Delta}(e_k | r_j^{(c)}) = \mathcal{M}(\mathbf{x}_j^{(c,T)}(\mathcal{G}), \hat{Y}_q) - \mathcal{M}(\mathbf{x}_j^{(c,T)}(\mathcal{G}_{\setminus e_k}), \hat{Y}_q), \quad (5)$$

which replaces the ground-truth labels Y in Eq. (4) with the proxy labels $\hat{Y}_q$ from confident rows $\mathcal{D}_q$.

Label-adaptive Hyperedge Weighting. The estimated effectiveness of each hyperedge for weak label rule $r_j^{(c)}$ (Eq. (5)) is used as its label-specific weight, $w_{k,j}^{(c)}$.² Then, we define the hyperedge weight vector as:

$$\mathbf{w}_j^{(c)} = [w_{1,j}^{(c)}, w_{2,j}^{(c)}, \dots, w_{|\mathcal{E}|,j}^{(c)}]^\top \in \mathbb{R}^{|\mathcal{E}|}.$$

By Eq. (5), $w_{k,j}^{(c)}$ measures how much hyperedge e_k contributes to propagating weak labels from $r_j^{(c)}$.

Using these label-specific weights, propagation for each weak label is performed as:

$$\mathbf{x}_j^{(c,t+1)} = \alpha D_V^{-1} H \mathbf{W}_j^{(c)} D_E^{-1} H^\top \mathbf{x}_j^{(c,t)} + (1 - \alpha) \mathbf{x}_j^{(c,0)},$$

where $\mathbf{W}_j^{(c)} = \text{diag}(\mathbf{w}_j^{(c)}) \in \mathbb{R}^{|\mathcal{E}| \times |\mathcal{E}|}$ encodes the weak label-specific relevance of hyperedges. As

²To obtain non-negative weights, we apply a softmax:

$$w_{k,j}^{(c)} = |\mathcal{E}| \frac{\exp(\tilde{\Delta}(e_k | r_j^{(c)})/\tau)}{\sum_{\ell=1}^{|\mathcal{E}|} \exp(\tilde{\Delta}(e_\ell | r_j^{(c)})/\tau)},$$

where τ controls the concentration.

Table 1: Statistics of the datasets. Cat. and Num. denote the numbers of categorical and numerical attributes.

Data	# of Rows	Cat.	Num.	Label ratio (%)
Adult	48,842	7	7	76:24
Bank	45,211	8	8	88:12
Blood	748	0	4	76:24
Car	1,728	5	1	70:22:4:4
Credit-G	1,000	12	8	70:30
Diabetes	768	0	8	65:35
Heart	918	4	7	45:55
Communities	1,994	1	102	34:33:33
Myocardial	1,700	94	17	22:78

before, this process requires only a forward pass and involves no training.

For prediction, class-level compatibility scores are computed from the weak labels refined by label-adaptive propagation by substituting the initial assignments $\mathbf{x}_j^{(c,0)}$ ($= \mathbf{x}_j^{(c)}$) in Eq. (2) with the propagated vectors $\mathbf{x}_j^{(c,T)}$.

This enables hypergraph propagation to prioritize hyperedges that are more effective for each weak label. We empirically validate this label-adaptive propagation in Section 4.

Complexity. TABHGR⁺ requires $\lceil |\mathcal{R}|/B \rceil + \lceil |\mathcal{E}|/B \rceil$ LLM calls, independent of the number of rows, and has time complexity $O(|\mathcal{R}|T|\mathcal{E}|\sum_{e \in \mathcal{E}} |e|)$ due to hyperedge effectiveness estimation. Refer to Appendix C for further details.

4 Experimental Results

We evaluate the accuracy and effectiveness of TABHGR⁺ by answering the research questions:

- **Q1. Accuracy.** How accurate is TABHGR⁺ compared to existing zero/few-shot baselines?
- **Q2. Efficiency.** How efficient is TABHGR⁺ in terms of LLM usage and inference cost?
- **Q3. Effectiveness.** Do LLM-generated hypergraphs improve zero-shot prediction?
- **Q4. Inductive Applicability.** Can TABHGR⁺ generalize to unseen rows?
- **Q5. Parameter Analysis.** How do the key parameters of TABHGR⁺ affect performance?

4.1 Experimental Settings

We first describe the experimental settings.

Datasets. We evaluate our methods on nine publicly available tabular datasets covering diverse domains, including income prediction (Adult (Bache and Lichman, 2013)), marketing (Bank (Moro et al., 2014)), healthcare (Blood (Yeh et al., 2009)),

Diabetes ³, Heart ⁴, Myocardial (Golovenkin et al., 2020)), credit assessment (Credit-G (Kadra et al., 2021)), automobile evaluation (Car (Kadra et al., 2021)), and community-level crime modeling (Communities (Redmond, 2009)). Refer to Table 1 for the statistics and Appendix A for further details.

Baselines. We compare our framework with nine representative baselines spanning traditional and recent LLM-based methods: Logistic Regression (LogReg), XGBoost (Chen and Guestrin, 2016), SCARF (Bahri et al., 2022), TabPFN (Hollmann et al., 2023), STUNT (Nam et al., 2023b), In-Context Learning (Wei et al., 2022), TABLET (Slack and Singh, 2023), TabLLM (Hegselmann et al., 2023), and FeatLLM (Han et al., 2024). Most require few-shot supervision or labeled data, while TABLET and TabLLM support row-wise zero-shot prediction. We also evaluate two variants of TABHGR⁺: Weak Labels Only, which uses weak labels without propagation, and TABHGR, which removes label-adaptive hyperedge weighting.

Implementation. Note that hyperparameter tuning is prohibited in our zero-shot setting; thus, all parameters are fixed across datasets. We use GPT-4o (OpenAI, 2024) as the default LLM and generate 20 condition rules per class for weak labeling. By default, we generate 200 semantic groups (hyperedges) per dataset, and 1000 for larger datasets such as Adult and Bank to improve row coverage. Due to LLM output-length limits, we generate semantic groups through multiple independent calls, each producing 20 groups from a newly sampled set of five unlabeled rows. We aggregate the groups generated across all calls to construct the final hypergraph, repeating this process until reaching the target number of groups. We set the propagation coefficient to $\alpha = 0.7$, propagation steps to 2, and confidence ratio to $q = 50$ by default. We conduct five independent runs and report the average performance of TABHGR⁺ and its baselines.

4.2 Q1. Overall Accuracy

We first evaluate the prediction accuracy of TABHGR⁺. As shown in Table 2, our zero-shot approaches, including weak labeling, TABHGR, and TABHGR⁺, substantially outperform row-wise zero-shot LLM baselines, TABLET and TabLLM. This suggests that class-aware LLM-generated

³kaggle.com/datasets/uciml/pima-indians-diabetes-database

⁴kaggle.com/datasets/fedesoriano/heart-failure-prediction

Table 2: Overall performance comparison. TABHGR⁺ outperforms existing zero-shot and 4-, 8-, and 16-shot tabular learning methods in AUROC across nine benchmarks, achieving the best average rank. Best results are in **bold**, second-best are underlined, and ranks are computed within each shot setting. For methods with missing results, the average rank is computed only over datasets with valid results.

Method	Shot	Adult	Bank	Blood	Car	Credit-G	Diabetes	Heart	Comm-unities	Myoca-rdial	Avg. Rank
LogReg	4	0.7210	0.6370	0.5679	0.6238	0.5268	0.5709	0.7054	0.6745	0.5125	7.00
	8	0.7602	0.7252	0.6851	0.7205	0.5552	0.6552	0.7812	0.7373	0.5534	5.89
	16	0.7520	<u>0.7751</u>	0.6830	0.8242	0.5826	0.7344	0.8302	0.7255	0.6000	5.22
XGBoost (Chen and Guestrin, 2016)	4	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5394	10.33
	8	0.5919	0.5878	0.5997	0.6400	0.5222	0.5086	0.5588	0.6665	0.5563	10.33
	16	0.6068	0.7034	0.6328	0.7226	0.5623	0.6559	0.7862	0.6801	0.5655	9.00
SCARF (Bahri et al., 2022)	4	0.5834	0.5853	0.5622	0.6252	0.4892	0.6235	0.5938	0.6618	0.4770	9.00
	8	0.7242	0.5528	0.6577	0.7223	0.5526	0.6469	0.7435	0.7269	0.4937	8.11
	16	0.7563	0.6581	0.6627	0.7577	<u>0.5922</u>	0.7186	0.8366	0.7309	0.5431	7.11
TabPFN (Hollmann et al., 2023)	4	0.6089	0.6319	0.5872	0.5814	0.5400	0.5628	0.6733	-	-	8.00
	8	0.7042	0.6281	0.6630	0.6395	0.5258	0.6908	0.7789	-	-	8.86
	16	0.7034	0.7379	0.6414	0.7135	0.5891	0.7369	0.8145	-	-	7.43
STUNT (Nam et al., 2023b)	4	0.6743	0.5634	0.4857	0.6132	0.4880	0.6422	0.8827	0.6687	0.5277	8.33
	8	0.8216	0.6301	0.6000	0.6786	0.5450	0.6739	0.8878	0.7636	0.5540	6.67
	16	0.8057	0.6985	0.5476	0.7556	0.5763	0.7379	0.8913	<u>0.7729</u>	<u>0.6122</u>	5.78
In-Context (Wei et al., 2022)	4	0.7751	0.6138	0.5630	0.6247	0.5299	0.7171	0.6076	-	-	7.00
	8	0.7930	0.6957	0.5899	0.6757	0.5243	0.7221	0.6546	-	-	8.57
	16	0.7950	0.6976	0.5659	0.7694	0.5529	0.7164	0.6700	-	-	9.14
TABLET (Slack and Singh, 2023)	0	0.7723	0.6910	0.5139	0.7785	0.4681	0.6946	0.7916	0.6604	0.6103	4.33
	4	0.7529	0.5811	0.5645	0.6021	0.5433	0.6396	0.6819	-	-	7.57
	8	0.7756	0.6908	0.5637	0.6553	0.5290	0.6547	0.6985	-	-	9.14
	16	0.7974	0.6940	0.6062	0.7402	0.5462	0.6671	0.6839	-	-	10.00
TabLLM (Hegselmann et al., 2023)	0	0.7969	0.6220	0.5392	0.7812	0.5236	0.6940	0.7824	0.6809	0.5818	4.33
	4	0.8357	0.6251	0.5587	0.8582	0.5190	0.7042	0.5974	-	-	6.86
	8	0.8352	0.6319	0.6601	<u>0.8743</u>	0.5642	0.6430	0.7014	-	-	6.14
	16	0.8323	0.6373	0.6514	0.8865	0.6038	0.6734	0.8172	-	-	6.29
FeatLLM (Han et al., 2024)	4	0.8668	0.7045	0.6834	0.7269	0.5594	0.8028	0.7566	0.7539	0.5287	3.56
	8	0.8789	0.7585	0.7037	0.7326	0.5742	0.7938	0.7946	0.7659	0.5622	3.11
	16	0.8754	0.7841	0.7007	0.7943	0.5660	<u>0.8015</u>	0.8371	0.7625	0.5532	<u>4.00</u>
Weak Labels Only	0	0.8722	0.6998	0.7447	0.7799	0.5524	0.7406	0.8782	0.7700	0.5856	3.00 (vs. 0-shot)
		±0.0069	±0.0173	±0.0335	±0.0313	±0.0222	±0.0193	±0.0210	±0.0020	±0.0151	3.33 (vs. 4-shot)
											3.89 (vs. 8-shot)
											5.11 (vs. 16-shot)
TABHGR	0	<u>0.8817</u>	0.7092	<u>0.7471</u>	0.7900	0.5464	0.7511	<u>0.9002</u>	0.7727	0.5867	<u>2.11</u> (vs. 0-shot)
		±0.0046	±0.0157	±0.0286	±0.0295	±0.0195	±0.0218	±0.0131	±0.0015	±0.0093	<u>2.33</u> (vs. 4-shot)
											<u>2.89</u> (vs. 8-shot)
											<u>4.00</u> (vs. 16-shot)
TABHGR⁺	0	0.8820	0.7202	0.7572	0.7966	0.5285	0.7716	0.9114	0.7778	0.6131	1.22 (vs. 0-shot)
		±0.0044	±0.0112	±0.0440	±0.0297	±0.0182	±0.0174	±0.0101	±0.0018	±0.0201	1.89 (vs. 4-shot)
											2.33 (vs. 8-shot)
											2.89 (vs. 16-shot)

rules provide more reliable semantic supervision than row-wise prompting. Moreover, despite being fully zero-shot, TABHGR and TABHGR⁺ outperform strong few-shot baselines; for instance, TABHGR⁺ exceeds 16-shot baselines on Adult, Blood, and Heart, and achieves the best average rank across 4-, 8-, and 16-shot settings. These results show the effectiveness of leveraging inter-row relationships via hypergraph generation and label

propagation for zero-shot tabular learning.

4.3 Q2. Efficiency Analysis

We compare the efficiency of TABHGR⁺ and TABHGR with row-wise LLM baselines in terms of tokens, API calls, and runtime. As shown in Figure 5, row-wise methods scale with the number of rows since each row is queried by the LLM. In contrast, TABHGR⁺ and TABHGR require only a constant number of LLM calls, independent of

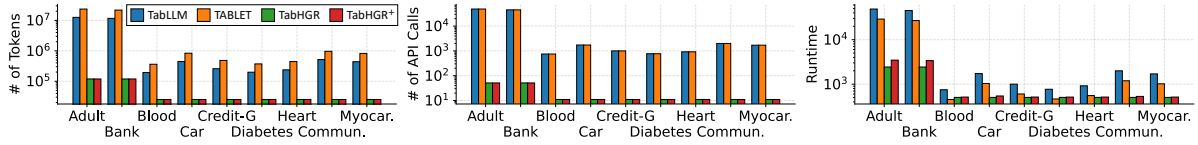


Figure 5: Efficiency analysis. TABHGR⁺ and its simple variant (TABHGR) are substantially more efficient than existing row-wise zero-shot LLM methods. While row-wise approaches incur inference costs that scale with the row count, TABHGR⁺ and TABHGR require only a constant number of input/output tokens and LLM calls, independent of dataset size. Hypergraph propagation and hyperedge effectiveness estimation are performed without extra LLM queries and incur negligible overhead compared to LLM API calls.

Table 3: Effectiveness of LLM-generated hypergraphs. TABHGR⁺ and its simple variant (TABHGR) with hypergraphs generated by LLMs outperform their kNN-based and randomized counterparts, showing the effectiveness of LLMs in inducing meaningful inter-row relations from tabular attributes in a zero-shot setting.

Dataset	Raw	TABHGR			TABHGR ⁺		
		LLM	KNN	Rand.	LLM	KNN	Rand.
Adult	0.8722	0.8817	0.8429	0.8627	0.8820	0.8530	0.8650
Bank	0.6998	0.7092	0.7088	0.6906	0.7202	0.7155	0.6895
Blood	0.7447	0.7471	0.7382	0.7342	0.7572	0.7408	0.7329
Car	0.7799	0.7900	0.7882	0.7830	0.7966	0.7815	0.7814
Credit-G	0.5524	0.5464	0.5658	0.5513	0.5285	0.5587	0.5644
Diabetes	0.7406	0.7511	0.7125	0.7459	0.7716	0.7086	0.7351
Heart	0.8782	0.9002	0.8238	0.8657	0.9114	0.8141	0.8690
Comm.	0.7700	0.7727	0.7809	0.7645	0.7778	0.7793	0.7651
Myocar.	0.5856	0.5867	0.5839	0.5770	0.6131	0.5579	0.5489
Avg.	0.7359 (4.11)	0.7428 (2.67)	0.7272 (4.11)	0.7305 (5.22)	0.7509 (1.89)	0.7256 (4.56)	0.7279 (5.33)

dataset size. Subsequent hypergraph propagation and effectiveness estimation require no additional LLM queries and add negligible overhead. Thus, our methods achieve much lower inference cost while improving predictive performance.

4.4 Q3. Effectiveness Analysis

We evaluate the effectiveness of hypergraphs generated by LLMs in TABHGR⁺ and TABHGR.

Alternative Hypergraphs. We compare LLM-generated hypergraphs with kNN hypergraphs based on raw attribute similarity and randomized hypergraphs preserving node degrees and hyper-edge sizes (Lee et al., 2021). Table 3 shows that TABHGR⁺ and TABHGR perform best with LLM-generated hypergraphs, confirming their effectiveness in zero-shot tabular learning.

Case Study. We present a case study illustrating how LLM-generated hypergraphs support zero-shot reasoning. As shown in Figure 6, for a diabetic patient, the LLM constructs a hypergraph where the patient belongs to multiple semantic

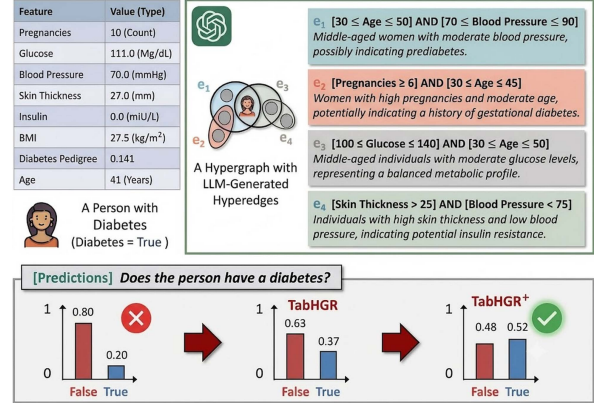


Figure 6: Case study of LLM-generated hypergraphs. Given a diabetic patient, an LLM builds a hypergraph over semantically related rows. Although the initial weak label is incorrect, TABHGR partially corrects it via propagation, while TABHGR⁺ emphasizes informative hyperedges and yields the correct prediction.

groups. Although the initial weak labels are incorrect, TABHGR partially refines the prediction via hypergraph propagation, while TABHGR⁺ further improves it by weighting hyperedges according to their weak-label-specific effectiveness.

4.5 Q4. Inductive Applicability

While TABHGR and TABHGR⁺ primarily operate transductively, their generated weak-labeling and semantic-group rules can also be applied to unseen rows. We construct a reference hypergraph from available unlabeled rows and independently insert each target row based on its rule assignments and semantic-group memberships. Weak labels are then propagated on the augmented hypergraph without accessing other target rows (Appendix D). As shown in Table 4, both inductive variants outperform TABLET and weak labeling only (WLO) on average. Inductive TABHGR⁺ achieves an average AUROC of 0.7431, close to its transductive counterpart (0.7509), demonstrating effectiveness beyond the transductive setting.

Table 4: Inductive performance. The inductive variants of TABHGR and TABHGR⁺ process target rows independently, outperform inductive baselines on average, and remain competitive with their transductive counterparts. WLO denotes weak labeling only.

Dataset	Inductive				Transductive	
	TABLET	WLO	TABHGR	TABHGR ⁺	TABHGR	TABHGR ⁺
Adult	0.7723	0.8722	0.8747	0.8744	<u>0.8817</u>	0.8820
Bank	0.6910	0.6998	0.7026	0.7031	<u>0.7092</u>	0.7202
Blood	0.5139	0.7447	0.7470	<u>0.7545</u>	0.7472	0.7572
Car	0.7785	0.7799	0.7814	0.7843	<u>0.7900</u>	0.7966
Commun.	0.6604	0.7700	0.7725	<u>0.7727</u>	<u>0.7727</u>	0.7778
Credit-G	0.4681	0.5524	<u>0.5510</u>	0.5504	0.5464	0.5285
Diabetes	0.6946	0.7406	0.7413	<u>0.7520</u>	0.7511	0.7716
Heart	0.7916	0.8782	0.8828	0.8943	<u>0.9002</u>	0.9114
Myocar.	<u>0.6103</u>	0.5856	0.5897	0.6024	0.5867	0.6131
Avg.	0.6645	0.7359	0.7381	<u>0.7431</u>	0.7428	0.7509

4.6 Q5. Parameter Analysis

Lastly, we analyze the sensitivity of TABHGR⁺ to key parameters: (a) the number of hyperedges, (b) the propagation strength α , and (c) the number of propagation steps (Figure 7). All parameters are fixed across datasets without task-specific tuning. Increasing hyperedges generally boosts performance, yielding larger gains for TABHGR⁺ thanks to more reliable effectiveness estimation. Very weak propagation (e.g., $\alpha = 0.1$) degrades results. Because the optimal step count depends on hypergraph density, we leave unsupervised step adaptation to future work.

5 Conclusions and Future Directions

We propose TABHGR⁺, an efficient, relation-aware zero-shot tabular learning framework. Instead of costly row-wise prompting, TABHGR⁺ uses LLMs to generate attribute-based rules for weak labeling and semantic grouping, keeping inference costs constant regardless of dataset size. To refine noisy weak labels, we perform label propagation over hypergraphs formed by these LLM-inferred groups. We also introduce label-adaptive hyperedge weighting to account for varying hyperedge contributions across labels. Empirically, TABHGR⁺ outperforms recent zero-shot and strong few-shot baselines while substantially reducing inference costs.

A promising future direction is relational databases (RDBs) (Robinson et al., 2024; Kim et al., 2026; Choi et al., 2026), where LLM-generated semantic hyperedges can be combined with explicit cross-table relations to support zero-shot prediction over multi-table data. In addition, future work could explicitly account for higher-

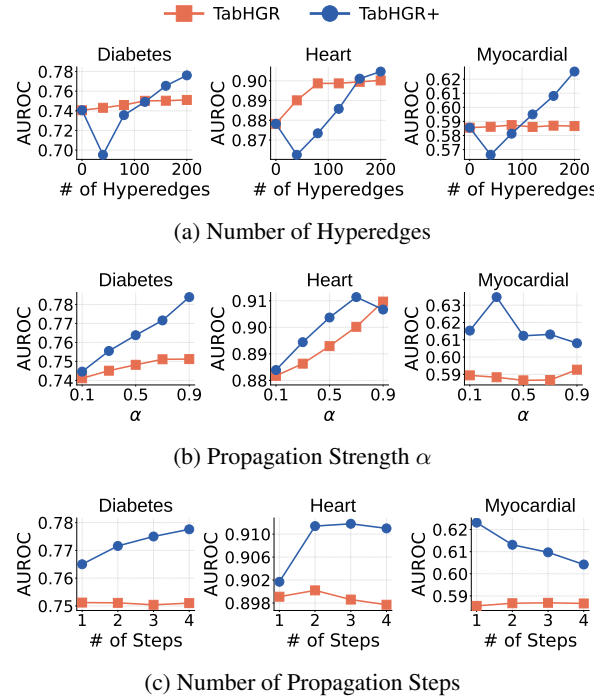


Figure 7: Parameter analysis. The performance of TABHGR⁺ and TABHGR varies with respect to (a) the number of hyperedges, (b) the propagation strength α , and (c) the number of propagation steps. All parameters are fixed across datasets in the zero-shot setting.

order properties such as hyperedge overlap, degree and size distributions, and local motifs when generating and refining hypergraphs (Lee et al., 2025).

Limitations

While TABHGR⁺ achieves strong zero-shot performance with lower LLM inference cost, it has several limitations and potential risks. Its effectiveness depends on the quality of LLM-generated rules and semantic groups, which may vary across domains and prompts; biased or spurious rules could propagate to downstream predictions, especially in sensitive domains. Although LLM calls are constant with dataset size, complex datasets may require multiple prompts to generate diverse hyperedges. Fixed zero-shot hyperparameters avoid task-specific tuning but may be suboptimal under different hypergraph densities, label distributions, or distribution shifts. Finally, the framework currently focuses on classification with meaningful attribute descriptions; applying it to regression, noisy metadata, or high-stakes settings requires careful validation and human oversight.

Ethical Considerations

TABHGR⁺ relies on LLM-generated rules and semantic groups, which may contain biases or spurious correlations. These errors can be amplified through label propagation, especially in sensitive domains such as healthcare, finance, or crime prediction. Although we use public benchmark datasets, real-world deployment requires privacy protection, fairness auditing, and compliance with data-use policies. TABHGR⁺ should thus be used as a decision-support tool with human oversight.

Acknowledgements.

This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024-00406985, 30%). This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2024-00438638, EntireDB2AI: Foundations and Software for Comprehensive Deep Representation Learning and Prediction on Entire Relational Databases, 50%) (No. RS-2019-II190075, Artificial Intelligence Graduate School Program (KAIST), 10%). This research was partly supported by the “Advanced GPU Utilization Support Program” funded by the Government of the Republic of Korea (Ministry of Science and ICT) (05-26-04-0100, 10%).

References

- K. Bache and M. Lichman. 2013. [UCI machine learning repository](#).
- Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. 2022. Scarf: Self-supervised contrastive learning using random feature corruption. In *ICLR*.
- Adrián Bazaga, Pietro Liò, and Gos Micklem. 2024. Hyperbert: Mixing hypergraph-aware layers with language models for node classification on text-attributed hypergraphs. In *Findings of EMNLP 2024*.
- Mayee F Chen, Daniel Y Fu, Dyah Adila, Michael Zhang, Frederic Sala, Kayvon Fatahalian, and Christopher Ré. 2022. Shoring up the foundations: Fusing model embeddings and weak supervision. In *UAI*.
- Pei Chen, Soumajyoti Sarkar, Leonard Lausen, Balasubramaniam Srinivasan, Sheng Zha, Ruihong Huang, and George Karypis. 2023. Hytrel: Hypergraph-enhanced tabular data representation learning. In *NeurIPS*.
- Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *KDD*.
- Dongwon Choi, Sunwoo Kim, Juyeon Kim, Kyungho Kim, Geon Lee, Shinhwan Kang, Myunghwan Kim, and Kijung Shin. 2026. Rdb2g-bench: A comprehensive benchmark for automatic graph modeling of relational databases. *NeurIPS*.
- Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. 2022. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. In *NeurIPS*.
- Kounianhua Du, Weinan Zhang, Ruiwen Zhou, Yangkun Wang, Xilong Zhao, Jiarui Jin, Quan Gan, Zheng Zhang, and David P Wipf. 2022. Learning enhanced representation for tabular data via neighborhood propagation. In *NeurIPS*.
- Yifan Feng, Chengwu Yang, Xingliang Hou, Shaoyi Du, Shihui Ying, Zongze Wu, and Yue Gao. 2025. Beyond graphs: Can large language models comprehend hypergraphs? In *ICLR*.
- SE Golovenkin, VA Shulman, DA Rossiev, PA Shesternya, S Yu Nikulina, Yu V Orlova, and VF Voino-Yasenetsky. 2020. Myocardial infarction complications. uci machine learning repository.
- Google DeepMind. 2026. [Gemini 3.5 flash model card](#). Technical report, Google DeepMind.
- Bingqiao Gu, Jiale Zeng, Xingqin Qi, and Dong Li. 2025. Modeling hypergraph using large language models. *arXiv preprint arXiv:2510.11728*.
- Sungwon Han, Jinsung Yoon, Sercan O Arik, and Tomas Pfister. 2024. Large language models can automatically engineer features for few-shot tabular learning. In *ICML*.
- Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *AISTATS*.
- Noah Hollmann, Samuel Müller, Katharina Eggenberger, and Frank Hutter. 2023. TabPFN: A transformer that solves small tabular classification problems in a second. In *ICLR*.
- Liusheng Huang, Huaping Chen, Xun Wang, and Guoliang Chen. 2000. A fast algorithm for mining association rules. *Journal of Computer Science and Technology*, 15(6):619–624.
- Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. 2024. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):8622–8642.

- Arlind Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. 2021. Well-tuned simple nets excel on tabular datasets. In *NeurIPS*.
- Kyungho Kim, Geon Lee, Juyeon Kim, Dongwon Choi, Shinhwan Kang, and Kijung Shin. 2026. Refuge: Feature generation for prediction tasks on relational databases with llm agents. In *WWW*.
- Geon Lee, Fanchen Bu, Tina Eliassi-Rad, and Kijung Shin. 2025. A survey on hypergraph mining: Patterns, tools, and generators. *ACM Computing Surveys*, 57(8):1–36.
- Geon Lee, Minyoung Choe, and Kijung Shin. 2021. How do hyperedges overlap in real-world hypergraphs?-patterns, measures, and generators. In *WWW*.
- Ming Li, Yukang Cheng, Lu Bai, Feilong Cao, Ke Lv, Jiye Liang, and Pietro Lio. 2025a. Edullm: Leveraging large language models and framelet-based signed hypergraph neural networks for student performance prediction. In *ICML*.
- Shu Li, Yi Lu, Shicheng Jiu, Haoxiang Huang, Guangqi Yang, and Jiong Yu. 2025b. Prototype-oriented hypergraph representation learning for anomaly detection in tabular data. *Information Processing & Management*, 62(1):103877.
- Yuhan Li, Zhixun Li, Peisong Wang, Jia Li, Xiangguo Sun, Hong Cheng, and Jeffrey Xu Yu. 2023. A survey of graph meets large language model: Progress and future directions. *arXiv preprint arXiv:2311.12399*.
- Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.
- Jaehyun Nam, Woomin Song, Seong Hyeon Park, Jihoon Tack, Sukmin Yun, Jaehyung Kim, and Jinwoo Shin. 2023a. Semi-supervised tabular classification via in-context learning of large language models. In *Workshop on Efficient Systems for Foundation Models@ ICML2023*.
- Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. 2023b. Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables. In *ICLR*.
- OpenAI. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Rattana Pukdee, Dylan Sam, Maria-Florina Balcan, and Pradeep Ravikumar. 2023. Label propagation with weak supervision. In *ICLR*.
- Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB endowment. International conference on very large data bases*, volume 11, page 269.
- Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *NeurIPS*.
- Michael Redmond. 2009. Communities and crime. uci machine learning repository.
- Joshua Robinson, Rishabh Ranjan, Weihua Hu, Kexin Huang, Jiaqi Han, Alejandro Dobles, Matthias Fey, Jan E. Lenssen, Yiwen Yuan, Zecheng Zhang, Xinwei He, and Jure Leskovec. 2024. Relbench: A benchmark for deep learning on relational databases. *NeurIPS*.
- Salva Rühling Cachay, Benedikt Boecking, and Artur Dubrawski. 2021. End-to-end weak supervision. In *NeurIPS*.
- Dylan Slack and Sameer Singh. 2023. Tablet: Learning from instructions for tabular data. *arXiv preprint arXiv:2304.13188*.
- Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. 2024. Language models in the loop: Incorporating prompting into weak supervision. *ACM/JMS Journal of Data Science*, 1(2):1–30.
- Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. 2024. Graphgpt: Graph instruction tuning for large language models. In *SIGIR*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- I-Cheng Yeh, King-Jang Yang, and Tao-Ming Ting. 2009. Knowledge discovery on rfm model using bernoulli sequence. *Expert Systems with applications*, 36(3):5866–5871.
- Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. Graphtranslator: Aligning graph model to large language model for open-ended tasks. In *WWW*.
- Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53.

A Details on Experimental Settings

We provide details on the experimental settings.

Datasets. We use nine public datasets: (1) Adult (Bache and Lichman, 2013), aimed at predicting whether a person’s annual income exceeds \$50K; (2) Bank (Moro et al., 2014), aimed at predicting a client’s likelihood of subscribing to a term deposit; (3) Blood (Yeh et al., 2009), aimed at predicting whether blood donors will return for future donations; (4) Car (Kadra et al., 2021), aimed at classifying automobile quality levels; (5) Credit-G (Kadra et al., 2021), aimed at classifying an individual’s credit risk as good or bad; (6) Diabetes⁵, aimed at predicting the presence of diabetes based on clinical indicators; (7) Heart⁶, aimed at predicting the likelihood of coronary artery disease; (8) Communities (Redmond, 2009), aimed at predicting the crime rate of a given community; and (9) Myocardial (Golovenkin et al., 2020), aimed at predicting whether a patient develops chronic heart failure following myocardial infarction.

Preprocessing Datasets with Many Attributes.

For Communities and Myocardial, whose number of attributes exceeds 100, we perform an attribute selection step to reduce prompt complexity. Specifically, we prompt the LLM to identify the 20 most relevant attributes based on the task description and column semantics, and use only these attributes for subsequent rule generation. The prompt template used for attribute selection is shown in Figure 11.

Documentation of Artifacts. We provide the code and datasets used in our experiments in the anonymous repository, including the implementation of TABHGR⁺, LLM prompt scripts, hypergraph construction and propagation code, and scripts for reproducing the results. Our experiments use nine public tabular classification datasets: Adult, Bank, Blood, Car, Credit-G, Diabetes, Heart, Communities, and Myocardial, covering domains such as income prediction, marketing, healthcare, credit assessment, automobile evaluation, and community-level crime modeling. The artifacts are intended for academic research and reproducibility; since some datasets involve sensitive domains, users should consider potential biases and demographic coverage before applying them beyond benchmarking.

⁵ kaggle.com/datasets/uciml/pima-indians-diabetes-database

⁶ kaggle.com/datasets/fedesoriano/heart-failure-prediction

Data Consent. We use only publicly available benchmark datasets and collect no new human-subject data. We therefore rely on the original providers for consent, data-collection procedures, and usage terms, and use the datasets solely for academic benchmarking.

Offensive Content. We use only publicly available benchmark datasets and do not collect new data. We did not add personally identifying information or offensive content during artifact creation. To the best of our knowledge, the datasets do not contain direct identifiers such as names or contact information; however, some may include sensitive demographic or health-related attributes. We therefore use them only for academic benchmarking and do not attempt to identify individuals or make real-world decisions based on the data.

Parameters for Packages. We report the main implementation and parameter settings used in our experiments. For TABHGR⁺, all hyperparameters are fixed across datasets in the zero-shot setting, including the number of weak-labeling rules, number of semantic groups, propagation coefficient, propagation steps, and confidence ratio. For baseline methods and external packages, we follow the official implementations and recommended settings whenever available, and use consistent preprocessing and evaluation protocols across datasets.

B Prompt Templates

We provide the prompt templates used in TABHGR⁺, specifically (1) weak label generation (Figure 12) and (2) group generation (Figure 13).

C Complexity Analysis

We analyze the time complexity of TABHGR⁺.

Weak Label & Hypergraph Generation. TABHGR⁺ employs LLMs to generate $|\mathcal{R}|$ rules for defining weak labels and $|\mathcal{E}|$ rules for constructing hyperedges. In practice, we generate rules and hyperedges in batches of size B , requiring $\lceil |\mathcal{R}|/B \rceil$ and $\lceil |\mathcal{E}|/B \rceil$ LLM prompts, respectively. Thus, the total number of LLM API calls is $\lceil |\mathcal{R}|/B \rceil + \lceil |\mathcal{E}|/B \rceil$, which is typically substantially smaller than row-wise prompting that requires $|\mathcal{D}|$ calls.

Hypergraph Propagation. The time complexity of a single hypergraph propagation step is $O(\sum_{e \in \mathcal{E}} |e|)$ using sparse matrix multiplications. Propagating for T steps takes $O(T \sum_{e \in \mathcal{E}} |e|)$ time. Since TABHGR propagates $|\mathcal{R}|$ weak labels, its

Table 5: Effectiveness of hyperedge generation in AU-ROC. Weak-HG constructs hyperedges from weak-labeling-rule memberships, whereas TABHGR constructs them from LLM-generated semantic groups. Semantic hyperedges lead to better average performance.

Dataset	TabHGR	Weak-HG
Adult	0.8817 $\pm$ 0.0045	0.8774 $\pm$ 0.0066
Bank	0.7092 $\pm$ 0.0157	0.7013 $\pm$ 0.0160
Blood	0.7472 $\pm$ 0.0286	0.7346 $\pm$ 0.0367
Car	0.7900 $\pm$ 0.0295	0.7857 $\pm$ 0.0312
Commun.	0.7727 $\pm$ 0.0195	0.7759 $\pm$ 0.0210
Credit-G	0.5464 $\pm$ 0.0218	0.5513 $\pm$ 0.0203
Diabetes	0.7511 $\pm$ 0.0131	0.7450 $\pm$ 0.0212
Heart	0.9002 $\pm$ 0.0015	0.8843 $\pm$ 0.0025
Myocar.	0.5867 $\pm$ 0.0093	0.5812 $\pm$ 0.0149
Average	0.7428	0.7374

time complexity is $O(|\mathcal{R}|T \sum_{e \in \mathcal{E}} |e|)$. A hypergraph propagation consists only of forward passes and does not require gradient-based optimization.

Hyperedge Effectiveness Estimation. TABHGR⁺ estimates hyperedge effectiveness by evaluating the impact of removing each hyperedge. Specifically, for each weak label, propagation is performed once on the full hypergraph and once for each hyperedge removal. This results in $|\mathcal{E}| + 1$ propagations per weak label. Therefore, the total time complexity of TABHGR⁺ is $O(|\mathcal{R}|T|\mathcal{E}|\sum_{e \in \mathcal{E}} |e|)$. In practice, this step incurs negligible overhead to LLM API calls, as empirically shown in Section 4.3.

D Inductive Extension

The main formulation of TABHGR and TABHGR⁺ is transductive, jointly propagating weak labels over all unlabeled target rows. We additionally consider an inductive setting in which each target row is predicted without accessing the other target rows. Let $\mathcal{D}_{\text{ref}} = \{\mathbf{s}_i^{\text{ref}}\}_{i=1}^{N_{\text{ref}}}$ and $\mathcal{D}_{\text{tar}} = \{\mathbf{s}_i^{\text{tar}}\}_{i=1}^{N_{\text{tar}}}$ denote the unlabeled reference and target sets, respectively, where each row is represented by its attribute values.

Reference Hypergraph. We generate the weak-labeling rules $\mathcal{R}$ and semantic-group rules $\mathcal{R}_{\text{group}}$ using only the task and attribute descriptions and examples sampled from $\mathcal{D}_{\text{ref}}$. Applying the semantic-group rules to the reference rows produces a hypergraph $\mathcal{G}_{\text{ref}} = (\mathcal{V}_{\text{ref}}, \mathcal{E})$ with incidence matrix $H_{\text{ref}} \in \{0, 1\}^{N_{\text{ref}} \times |\mathcal{E}|}$. For each weak-labeling rule $r_j^{(c)}$, we also obtain its assignment vector over the reference rows, $\mathbf{x}_{j,\text{ref}}^{(c)} \in \{0, 1\}^{N_{\text{ref}}}$.

Table 6: Structural comparison of weak-labeling and semantic-group rules. Cov. denotes the average percentage of rows covered by each rule, while Jaccard denotes the average pairwise Jaccard similarity between weak-labeling and semantic-group rules.

Dataset	Weak Cov.	Group Cov.	Jaccard
Adult	31.3%	9.3%	0.0581 $\pm$ 0.0046
Bank	37.6%	8.8%	0.0631 $\pm$ 0.0010
Blood	29.8%	11.3%	0.0935 $\pm$ 0.0030
Car	11.2%	11.1%	0.0610 $\pm$ 0.0034
Commun.	32.0%	22.5%	0.1405 $\pm$ 0.0026
Credit-G	44.7%	14.0%	0.0925 $\pm$ 0.0022
Diabetes	32.9%	10.5%	0.0828 $\pm$ 0.0018
Heart	35.0%	9.3%	0.0741 $\pm$ 0.0018
Myocar.	41.3%	12.4%	0.0797 $\pm$ 0.0028
Average	32.9%	12.1%	0.0828

Target-Row Inference. For each target row $\mathbf{s}_i^{\text{tar}} \in \mathcal{D}_{\text{tar}}$, we apply the generated rules to determine its initial weak labels and semantic-group memberships. Let $x_{i,j}^{(c),\text{tar}} \in \{0, 1\}$ denote its assignment under weak-labeling rule $r_j^{(c)}$, and let $\mathbf{h}_i^{\text{tar}} \in \{0, 1\}^{|\mathcal{E}|}$ denote its hyperedge-membership vector. We insert the target row into the reference hypergraph by constructing

$$H_i = \begin{bmatrix} H_{\text{ref}} \\ (\mathbf{h}_i^{\text{tar}})^\top \end{bmatrix}, \quad \mathbf{x}_{i,j,\text{aug}}^{(c,0)} = \begin{bmatrix} \mathbf{x}_{j,\text{ref}}^{(c)} \\ x_{i,j}^{(c),\text{tar}} \end{bmatrix}.$$

We perform the propagation described in Section 3.3 on this augmented hypergraph and retain only the propagated score corresponding to $\mathbf{s}_i^{\text{tar}}$. Its class-level scores are then obtained by aggregating the propagated weak labels as in Eq. (2).

Each target row is inserted and processed independently and is removed before the next target row is considered. Therefore, its prediction can use the structure and weak labels of the reference rows but cannot depend on any other target row. The reference hypergraph and generated rules are reused across all target rows.

Inductive Variants. Inductive TABHGR performs unweighted propagation separately on each augmented hypergraph H_i . For inductive TABHGR⁺, we first estimate the effectiveness of each hyperedge using only $\mathcal{G}_{\text{ref}}$. Specifically, confident reference rows provide proxy labels, and the effectiveness of a hyperedge is measured by the change in propagation quality when it is removed, following Section 3.4. The resulting label-specific weights are fixed and reused when propagating on each augmented hypergraph H_i . Thus, a target row affects

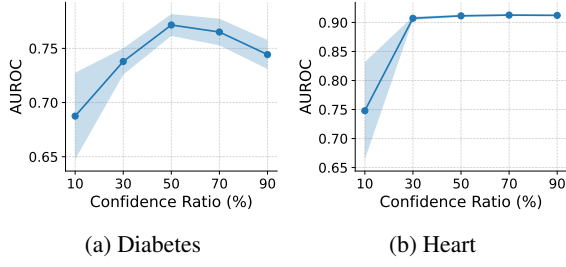


Figure 8: TABHGR⁺ with varying confidence ratio q . Performance generally improves as q increases up to 0.5 and remains stable thereafter.

only its own hyperedge memberships and propagation result, without influencing weight estimation or the predictions of other target rows.

Evaluation Protocol. We evaluate predictions only on $\mathcal{D}_{\text{tar}}$, without using its ground-truth labels or jointly accessing its rows during inference. We repeat the experiment over five independent weak-rule and hypergraph generations and report the mean performance.

E Additional Experiments

We provide additional experiments on the effectiveness and structural properties of LLM-generated hyperedges and the effect of the pseudo-label ratio.

LLM-Generated Hypergraphs. Table 5 compares TabHGR with Weak-HG, a variant that replaces LLM-generated semantic-group hyperedges with hyperedges formed by samples selected by the same weak-labeling rule. TabHGR achieves a higher average AUROC (0.7428 vs. 0.7374) and outperforms Weak-HG on seven of nine datasets, demonstrating the effectiveness of the LLM-generated hypergraph. Table 6 provides further structural insights. Weak-labeling rules generally have higher coverage, whereas semantic-group rules capture more fine-grained row relationships. Their low average pairwise Jaccard similarity further indicates that they capture largely distinct structures.

Effect of the Confidence Ratio. Figure 8 shows the effect of the confidence ratio q . Performance improves as q increases from 10 to 50 on both datasets. It slightly decreases beyond 50 on Diabetes, while remaining stable on Heart.

Alternative Hypergraph Constructions. In addition to the results in Table 3, we compare our LLM-generated semantic hypergraph with frequent-itemset- and embedding-based constructions. Apri-

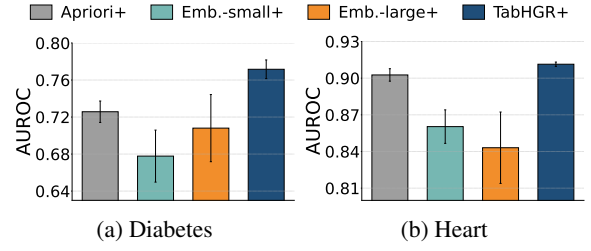


Figure 9: Evaluation with alternative hypergraph construction methods. Apriori+ uses frequent itemsets as hyperedges, while Emb.-small+ and Emb.-large+ cluster row embeddings from text-embedding-3-small and text-embedding-3-large, respectively. TABHGR⁺ achieves the best AUROC on both datasets.

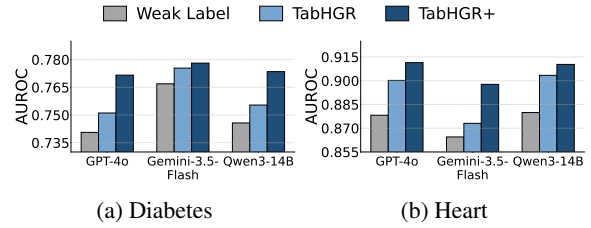


Figure 10: Performance across different LLM backbones. TABHGR and TABHGR⁺ consistently improve upon weak labeling across all backbones, with TABHGR⁺ achieving the best AUROC in every setting.

ori+ (Huang et al., 2000) discretizes continuous attributes and uses frequent itemsets as hyperedges, while Emb.-small+ and Emb.-large+ cluster row embeddings from text-embedding-3-small and text-embedding-3-large, respectively. As shown in Figure 9, TABHGR⁺ consistently outperforms these alternatives on Diabetes and Heart.

Robustness across LLM Backbones. We evaluate our methods using GPT-4o (OpenAI, 2024), Gemini-3.5-Flash (Google DeepMind, 2026), and Qwen3-14B (Yang et al., 2025) on Diabetes and Heart. As shown in Figure 10, TABHGR consistently improves upon weak labeling, while TABHGR⁺ yields further gains across all backbones. These results indicate that the benefits of hypergraph propagation and label-adaptive weighting are not specific to a particular LLM.

F Use of AI Assistants

We used AI assistants only for limited writing support, such as improving grammar, clarity, and conciseness of the manuscript. All research ideas, method design, experiments, analyses, and final claims were developed and verified by the authors.

Feature Selection

You are a domain expert in **tabular classification**. Your objective is to **select the most useful features (columns) from the given table**.

Task: [TASK]

Features: [FEATURES]

Examples: [EXAMPLES]

Instructions: Select [NUM_FEATURE] features that are most relevant for the classification task. Consider discriminative power, coverage, and generalizability. Think step by step.

Response Format (feature indices only, no explanation): A list of 20 feature indices separated by commas.

Figure 11: LLM prompt template for feature selection.

Class-Aware Rule Generation

You are an expert in **tabular data classification**. Your objective is to derive **interpretable and generalizable rules** that accurately characterize entities belonging to each class label.

Task: [TASK]

Features: [FEATURES]

Examples: [EXAMPLES]

Instructions: Generate [NUM_CONDITION] interpretable rule conditions for each label listed in [LABEL_LIST]. Each condition should be general enough to apply to unseen rows, not overly specific or narrowly tailored. Refer to the provided examples to identify useful patterns, but do not simply memorize or repeat them. Each condition should be general enough to apply to unseen rows, not overly specific to particular values in the examples. Think step-by-step.

For each label in [LABEL_LIST] (order strictly enforced), format for its response:

Label: [Label name or value]

[Condition 1]; [A concise yet comprehensive natural-language description]

[Condition 2]; [A concise yet comprehensive natural-language description]

...

Format for [Condition]:

For the categorical variable only,

- [Feature_name] is in [list of Categorical_values]

For the numerical variable only,

- [Feature_name] (> or >= or < or <=) [Numerical_value]

- [Feature_name] is within range of [Numerical_range_start, Numerical_range_end]

Figure 12: LLM prompt template for interpretable rule generation.

Group Generation

You are an expert in **tabular data classification**. Your objective is to define **meaningful and interpretable groups of entities (rows)**.

Features: [FEATURES]

Examples: [EXAMPLES]

Instructions: Generate [NUM_GROUP] interpretable and meaningful groups. Ensure groups are non-trivial (avoid overly generic or overly specific conditions). Refer to the provided examples to identify useful patterns, but do not simply memorize or repeat them. Each group should be general enough to apply to unseen rows, not overly specific to particular values in the examples. Think step-by-step.

Each group must:

- Consist of 1 to 4 simple, interpretable conditions
- Be distinct from other groups (avoid overlap in definition)
- Optionally overlap in coverage (i.e., entities can belong to multiple groups)

Include a concise explanation that describes:

- What kind of entities the group represents
- Their distinctive characteristics or behaviors
- How these characteristics relate to the task and target labels (e.g., why this group tends to belong to or differ from the target class).
- Why this group is meaningful or relevant based on the features
- Avoid any repetition, commentary, or meta-explanation

Format for response:

Group 1: [A concise yet comprehensive natural-language description]

[Condition 1]

[Condition 2]

...

Group 2: ...

...

Group [NUM_GROUP]: ...

...

Format for [Condition]:

For the categorical variable only,

- [Feature_name] is in [list of Categorical_values]

For the numerical variable only,

- [Feature_name] (> or >= or < or <=) [Numerical_value]
- [Feature_name] is within range of [Numerical_range_start, Numerical_range_end]

Figure 13: LLM prompt template for group generation.