Harnessing LLMs Explanations to Boost Surrogate Models in Tabular Data Classification

Ruxue Shi, Hengrui Gu, Xu Shen, and Xin Wang (✉)

Jilin University, Changchun, China
{shirx24, guhr22, shenxu23}@mails.jlu.edu.cn
xinwang@jlu.edu.cn

Abstract. Large Language Models (LLMs) have shown remarkable ability in solving complex tasks, making them a promising tool for enhancing tabular learning. However, existing LLM-based methods suffer from high resource requirements, suboptimal demonstration selection, and limited interpretability, which largely hinder their prediction performance and application in the real world. To overcome these problems, we propose a novel in-context learning framework for tabular prediction. The core idea is to leverage the explanations generated by LLMs to guide a smaller, locally deployable Surrogate Language Model (SLM) to make interpretable tabular predictions. Specifically, our framework mainly involves three stages: (i) Post Hoc Explanation Generation, where LLMs are utilized to generate explanations for question-answer pairs in candidate demonstrations, providing insights into the reasoning behind the answer. (ii) Post Hoc Explanation-Guided Demonstrations Selection, which utilizes explanations generated by LLMs to guide the process of demonstration selection from candidate demonstrations. (iii) Post Hoc Explanation-Guided Interpretable SLM Prediction, which utilizes the demonstrations obtained in step (ii) as in-context and merges corresponding explanations as rationales to improve the performance of SLM and guide the model to generate interpretable outputs. Experimental results highlight the framework's effectiveness, with an average accuracy improvement of 5.31% across various tabular datasets in diverse domains.

Keywords: Large Language Models (LLMs) · Tabular data · Surrogate Language Model (SLM).

1 Introduction

Recent advancements in model architecture and training methodologies have driven the remarkable progress and widespread adoption of Large Language Models (LLMs). By leveraging large-scale pre-training and fine-tuned instruction-based learning, LLMs have encoded extensive prior knowledge in their parameters. This has made them highly effective for various tasks, including dialogue generation [11] and code debugging [10], without requiring task-specific training. Inspired by these successes, researchers have investigated applying LLMs to analyze tabular data, as depicted in Figure 1 (top). Existing methods analyze

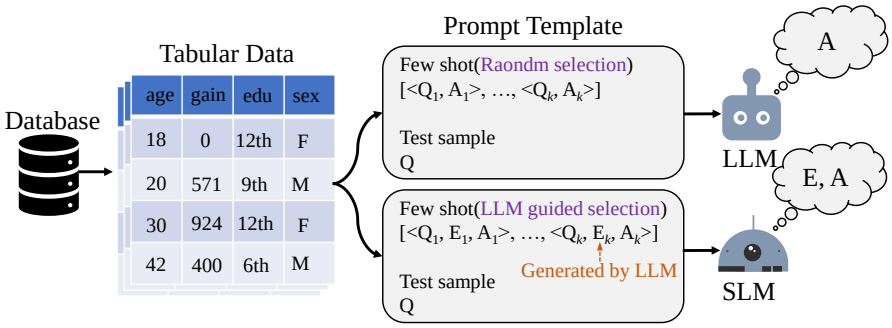

Fig. 1. Previous approaches to context learning with LLMs, as depicted on the top side of the figure, directly utilize tabular data to prompt the LLMs for predictions. This process necessitates frequent and extensive access to large language models (LLMs), resulting in substantial computational and efficiency demands. In contrast, our method depict at the bottom, presents a more efficient approach, which uses LLM generated explanations to guide the few shot demonstrations selection and the interpretable outputs of SLM

tabular data sample-by-sample to uncover causally meaningful relationships between tabular features and their corresponding labels [9]. The ability of LLMs to generalize across different domains through such methods has set new benchmarks in tabular data analysis, particularly in data-scarce scenarios. Despite these advancements, LLM-based approaches for tabular data face significant challenges: ❶ The performance of LLMs in few-shot settings has shown to be sensitive to changes in prompts [38]. To optimize demonstrations for a given test sample, existing methods typically use representation similarity within the textual embedding space spanned by a pre-trained language model as the selection criterion. Due to the redundant nature of features in tabular data, this approach inadvertently incorporates irrelevant or spurious features into the calculated embeddings. This limits the representation of semantics introduced by causal tabular features, ultimately leading to a suboptimal selection scheme. ❷ Ensuring high-quality predictions often necessitates memory-intensive models or frequent API calls, both of which are computationally expensive for large-scale industrial applications. Additionally, most top-performing LLMs are accessible only through response-only APIs, which restrict downstream fine-tuning and limit their potential ability to be further aligned with specific domain requirements. ❸ Another critical limitation is the lack of interpretability in the predictions made by LLMs. While these black-box models excel at generating accurate results, their internal decision-making rule are often non-transparent, raising concerns about their reliability especially in critical applications such as medical diagnosis and autonomous driving.

To address these limitations, we propose a novel tabular learning framework that achieves significant improvements in both task performance and inference

efficiency, offering a robust and effective solution to advance tabular learning tasks. Specifically, the framework introduces an additional warm-up stage prior to inference, where top-performing LLMs (e.g., ChatGPT) are tasked with generating feature attribution-based explanations on minimal task samples (candidate demon- strations) that reflect their behavior patterns. This process identifies a subset of features that are causally meaningful to the decision-making process of these advanced LLMs. During the inference stage, the selected features are utilized to enable a more robust demonstration selection, focusing on retaining only relevant features and mitigating the impact of noisy ones. This approach facilitates precise and personalized demonstration choices that align closely with similar data patterns, thereby improving the quality of input prompts and enhancing overall performance (*Challenge* ❶). These high-quality demonstrations, combined with their corresponding predictive explanations, constitute an explanation-guided inference prompt. This approach enables the use of a computationally efficient small surrogate language model (SLM) as a substitute for the resource-intensive large-scale LLMs or costly commercial API calls. By leveraging the meticulously crafted prompt, the SLM can effectively learn the prediction patterns of top-performing LLMs and produce predictions economically without compromising task performance (*Challenge* ❷). Moreover, the generated explanations accompanying each prediction enhance interpretability by providing clear and transparent rationales for the model's outputs, fostering greater trust and understanding of its decision-making process (*Challenge* ❸).

The main contributions of this work are as follows:

- We use LLMs as post hoc explanation generators to reduce the need for frequent API calls while still capturing key insights from the model.
- We introduce a novel framework that leverages LLMs-generated explanations to guide demonstration selection, and integrating these explanations as rationales to improve both classification performance and interpretability in surrogate language models.
- Our experiments show that the proposed framework leads to significant improvements in classification performance under few-shot learning conditions across multiple benchmark tabular datasets.

2 Related Works

2.1 Classical Tabular Data

Numerous machine learning methods have been developed for classification tasks on tabular data. For modeling linear relationships, logistic regression (LR) [22] and generalized linear models (GLM) [13] are commonly used. For non-linear relationship modeling, tree-based models such as decision trees (DT) [24] and ensemble methods like XGBoost [8], random forests [6], CatBoost [29], and LightGBM [18] are widely applied. With the advancement of deep learning, there has been increasing interest in applying neural networks to tabular data. These methods can be categorized into four main types: 1) Standard Neural

Networks: Examples include SNN [21], AutoInt [32], and DCN V2 [34]. 2) Hybrid Methods: These integrate decision trees with neural networks for end-to-end training, including NODE [28], GrowNet [3], TabNN [20], and DeepGBM [19]. 3) Transformer-Based Methods: These models leverage attention mechanisms to learn from features and data samples, as seen in TabNet [1], TabTransformer [16], and FT Transformer [12]. 4) Representation Learning Methods: These methods use self-supervised and semi-supervised learning to extract meaningful information, with notable examples including VIME [37], SCARF [4], SAINT [31], and Recontab [7].

2.2 In-Context Learning with Tabular Data

In-context learning (ICL) offers an innovative approach by enabling language models to perform tasks based solely on input-output examples, without parameter updates or fine-tuning. This capability, first observed in GPT-3 and subsequent large language models (LLMs) [23], allows models to generalize to new tasks by embedding examples directly into the input prompt. Often referred to as an "emergent ability" [35], this phenomenon has spurred significant research into understanding and enhancing ICL in models exceeding 100 billion parameters. To improve ICL, researchers have explored various strategies for enriching prompt content. For instance, the Chain-of-Thought (CoT) technique [36] incorporates human-annotated rationales, such as step-by-step task instructions, into prompts to enhance reasoning. In the context of tabular data, TABLET [30] integrates rule sets and prototypes from external classifiers to improve inference quality, while SPROUT [26] combines unlabeled data with LLMs to extract transferable knowledge from tabular samples, reducing dependence on labeled data. Despite their success, these approaches often rely on passing all inference tasks through LLMs, which is computationally expensive and impractical for industrial-scale applications. Furthermore, they frequently overlook the critical importance of selecting optimal contextual demonstrations. This study aims to address these limitations by proposing a novel framework that moves away from a purely end-to-end reliance on LLMs inference. Instead, it leverages LLMs to infer the relationships between questions and answers directly. This relational insight is then used to guide the selection of demonstrations and generate rationales, enhancing both the in-context learning performance and the interpretability of surrogate language models.

3 Method

Our framework enhances the performance and interpretability of the surrogate language model (SLM) by leveraging post hoc explanations derived from candidate demonstrations. These explanations guide the SLM in selecting the most relevant demonstrations, achieving superior downstream performance compared to directly prompt LLMs. As shown in Figure 2, the process begins by transforming tabular data into a textual format suitable for processing by the language

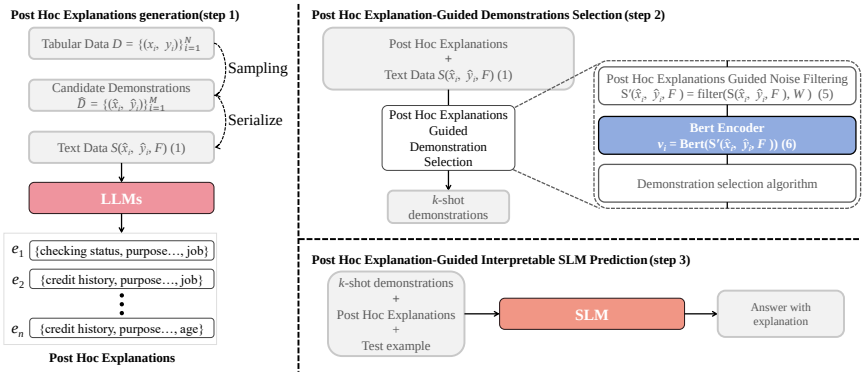

Fig. 2. Overview of our method. We begin by randomly selecting a small number of samples from the tabular dataset to form candidate demonstration sets. These samples are then converted into text and fed into an LLM, which generates post hoc explanations. These explanations are used to guide the selection of relevant demonstrations and to facilitate the generation of interpretable predictions by the SLM.

model. The LLMs then generate input-specific explanations, which are used to inform the selection of relevant demonstrations. Additionally, these explanations are integrated into the selected demonstrations as rationales, serving as a form of pre-feature engineering. By filtering out less significant features, this approach improves the accuracy and relevance of the selection process. Details of the problem formulation and each stage of the framework are discussed in the following sections.

3.1 Preliminaries

Problem Formulation. Consider a tabular dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where N represents the total number of samples. Each sample $\mathbf{x}_i \in \mathbf{X}$ is a d-dimensional vector, corresponding to d input features, with textual feature names denoted as $F = \{f_j\}_{j=1}^{d}$. In the classification setting, the label $y_i \in \mathbf{Y}$ is drawn from a predefined set of classes. For k-shot learning experiments, a subset of k labeled samples, where $k < N$, is randomly selected to train the model. The primary objective is to develop a model that can accurately predict the class label y_i for each $\mathbf{x}_i$ based on its feature representation, even with limited training samples.
Tabular Data Serialization. Tabular data serialization is a commonly used technique in tabular learning, converting tabular data into its corresponding textual form. Given a sample $(\hat{\mathbf{x}}_i, \hat{y}_i) \in \hat{D}$, where candidate demonstrations set $\hat{D}$ is obtained by randomly sampling M samples from D, we define a transformation function, S, which converts each tabular sample into a question-answer format:

$$\mathrm{S}(\hat{\mathbf{x}}_i,\ \hat{y}_i,\ F) = \text{“Q: } f_1 \ \text{ is } \ \hat{x}_{i1}.\ \ldots f_d \ \text{ is } \ \hat{x}_{id}.\ \text{A: } \hat{y}_i\text{”}, \tag{1}$$

Task
<Task descroption>

Candidate Demonstration
<Q, A>

Post Hoc Explanation Generation Instruction
Please provide n words in the question that are most important for obtaining the given answer.

Response Instruction
Note: The output format is as follows: {'word', 'word', ... , 'word', 'word'}

Table 1. Prompt for post hoc explanation generation. Text in orange provides the task description; Blue and green text represent the questions and answers in the candidate demonstration, respectively; purple text outlines post hoc explanation generation instruction; and yellow text details response instruction.

where the vector $\hat{\mathbf{x}}_i = (\hat{x}_{i1}, \ldots, \hat{x}_{ij}, \ldots, \hat{x}_{id})$ represents the feature values for the i-th sample, and $\hat{x}_{ij}$ represents the value of the j-th column feature of the i-th sample.

3.2 Post Hoc Explanation Generation (step 1)

We seek to leverage the internal knowledge of LLMs to generate post hoc explanations for candidate demonstrations by using carefully designed instructions, as shown in the purple text in Table 1. These instructions are crafted to clarify the reasoning behind each question-answer (Q-A) pair, enhancing the interpretability and performance of the surrogate language model (SLM). Specifically, the LLMs identify and select the n most salient features from Q that are most relevant to A. These features serve as the post hoc explanations for the candidate demonstration, highlighting the critical aspects of the input-output relationship. This approach not only improves transparency but also ensures that the model focuses on meaningful and causal correlations, avoiding irrelevant or spurious associations.

To ensure reply consistency and usability, we guide the LLMs to structure their responses according to predefined instructions, illustrated in the yellow text in Table 1. These prompts specify the format and required number of post hoc explanations, enabling straightforward parsing and application of the generated outputs. The formal process for generating post hoc explanations for the i-th candidate demonstration is defined as follows:

$$e_i = \{w_1, w_2, \ldots, w_n\} = \text{LLMs}(\text{S}(\hat{\mathbf{x}}_i,\ \hat{y}_i,\ F),\ I), \tag{2}$$

where e_i represents the set of salient features, $\hat{\mathbf{x}}_i$ and $\hat{y}_i$ are the input and label of the i-th candidate demonstration, F is the set of feature names, and I represents

the instructional prompts used to guide the LLMs. This structured approach ensures clarity, task alignment, and improved model performance.

3.3 Post Hoc Explanation-Guided Demonstrations Selection (step 2)

We propose an innovative approach for demonstration selection, which leverages post hoc explanations generated by LLMs to guide the identification of the most relevant and informative features, enabling improved in-context learning performance. The objective is to prioritize features that contribute most significantly to the quality of demonstrations. To achieve this, we compute the importance of each feature, $g(w_j)$, using the following function:

$$g(w_j) = \frac{1}{n \times M} \sum_{i=1}^{M} \mathbf{1}_{\{w_j \in e_i\}}, \tag{3}$$

where M is the total number of candidate demonstrations, and $\mathbf{1}_{\{w_j \in e_i\}}$ is an indicator function that equals 1 if the feature w_j appears in the explanation e_i. Based on this, we define the set W, which consists of the top q most important features ranked by their importance:

$$W = \{w_i \mid w_i \in \mathrm{Top}_q((w_i, g(w_i), p))\}, \tag{4}$$

where p serves as the feature importance threshold, guiding the selection of significant features. The function $\mathrm{Top}_q((w_i, g(w_i), p))$ identifies the top q most important features by evaluating their importance levels against this threshold. These selected features are then used to filter the candidate demonstrations, this process is mathematically represented as:

$$\mathrm{S}'(\hat{\mathbf{x}}_i, \hat{y}_i, F) = \mathrm{filter}(\mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F), W), \tag{5}$$

where the function $\mathrm{filter}(\cdot)$ removes any sentences in $\mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F)$ that do not contain features from W. To encode the filtered demonstrations, we use Bert:

$$v_i = \mathrm{Bert}(\mathrm{S}'(\hat{\mathbf{x}}_i, \hat{y}_i, F)), \tag{6}$$

where $v_i \in \mathbb{R}^u$ represent the vectorized representation of the i-th filtered demonstration and u is the hidden size of Bert. To evaluate the importance score s_i for i-th filtered candidate demonstration, we employed diversity-based algorithms (Cluster-Based Methods) and similarity-based algorithms, including Cosine Similarity, Euclidean Distance, and Manhattan Distance. The impact of these methods across various datasets is analyzed detailedly in Section 4.7.

- **Cluster-Based Methods.** To obtain demonstrations of diversity, based on the candidate demonstrations, we run a K-Means clustering to generate the importance score s_i.

$$s_i = -\min_{\mathbf{C} \in \mathbb{R}^{d \times m}} \|v_i - \mathbf{C}\|_2^2, \tag{7}$$

  where m is the number of centroids, and $\mathbf{C}$ is the centroid matrix.

– **Cosine Similarity-Based Methods.** We use cosine similarity to obtain a demonstration with higher semantic similarity to the test sample x_t, the semantic similarity score between the i-th candidate demonstration v_i and the embedded test sample v_t is calculated using the following formula:

$$s_i = v_t \odot v_i, \tag{8}$$

where $\odot$ represent element multiplication.

– **Euclidean Distance-Based Methods.** We also considered the influence of physical distance on similarity scores, and the calculation formula is as follows:

$$s_i = - \left\| v_t - v_i \right\|_2^2 \tag{9}$$

– **Manhattan Distance-Based Methods.** In order to reduce the influence of outliers in physical distance, we also use Manhattan distance to calculate the importance score s_i:

$$s_i = - \left\| v_t - v_i \right\| \tag{10}$$

The final k-shot demonstrations are selected based on the score s_i:

$$k\text{-shot} = \{\mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F) \mid \mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F) \in \mathrm{Top}_k(\mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F), s_i)\} \tag{11}$$

Our framework systematically combines feature importance, explanation-guided filtering, and scoring algorithms to optimize the selection of demonstrations for in-context learning, improving both performance and interpretability.

3.4 Post Hoc Explanation-Guided Interpretable SLM Prediction (step 3)

To integrate post hoc explanations into the learning process, the k-shot demonstrations are enhanced by appending their corresponding explanations, as defined below:

$$k\text{-shot}' = \{\mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F) + e_i \mid \mathrm{S}(\hat{\mathbf{x}}_i, \hat{y}_i, F) \in k\text{-shot}\} \tag{12}$$

Using these enriched demonstrations, the model generates predictions for a test sample by applying the SLM with the updated k-shot demonstrations:

$$reply = \mathrm{SLM}(k\text{-shot}', S(x_t, \text{-}, F)), \tag{13}$$

where the model's reply consists of both an explanation (e) and a predicted answer (A). A parser is then employed to extract these components:

$$e, A = \mathrm{Parser}(reply) \tag{14}$$

This process ensures that the model not only provides accurate predictions but also includes clear, interpretable explanations, enhancing the transparency and reliability of the predictions.

4 Experimental Evaluation

In this section, we seek to address the following research questions:

- (**RQ1.**) How does our proposed method compare to baseline approaches in a few-shot learning setting?
- (**RQ2.**) How well does our method balance performance with reduced API usage compared to ChatGPT?
- (**RQ3.**) What is the contribution of each individual component of our method to the overall performance?
- (**RQ4.**) How do different selection algorithms impact model performance? Specifically, does the Post Hoc Explanation Guided Demonstrations Selection approach improve the model's ability to choose the most appropriate examples?

4.1 Datasets

We use four datasets for binary classification tasks in our experiments: The Bank dataset [25] for predicting whether a customer will subscribe to a term deposit is a dataset about marketing activities of Portuguese banking institutions, containing 45211 records and 16 features; The creditg dataset [17] contains 1000 records and 20 features, suitable for predicting credit ratings; The Income dataset [2] has 48842 records and 14 features, focusing on the classification of individual income levels; The Heart dataset[1] has 918 records and 11 features, focusing on identifying and predicting patients with heart disease.

Dataset	bank	creditg	heart	income
# of train samples	40689	900	826	43958
# of test samples	4522	100	92	4884
# of candidate demonstrations	100	100	100	100
# of features	16	20	11	12

Table 2. Basic information of each dataset used in our experiments.

Each dataset was partitioned into training and testing sets with a 9:1 split. To reduce computational complexity and minimize the number of queries to the LLMs, we randomly selected 100 samples from the training set as candidate demonstrations. Table 3 summarizes the key details of these datasets, including the number of training and test samples, as well as the number of features.

4.2 Experimental Setting

Our framework employs GPT-3.5[2] as the backbone for the LLMs. For the LLMs inference, the temperature is set to 0.3 to balance creativity and consistency, while the top-p value remains at the default of 1, as provided by the API settings. We configure the number of post hoc explanations n to 5 and set the filtering

[1] kaggle.com/datasets/fedesoriano/heart-failure-prediction
[2] https://openai.com/blog/chatgpt

threshold p for filtering noisy sentences from the text data based on feature importance at 0.85. For encoding text data, we use SentenceBERT with a hidden dimension of 128. Additionally, the Llama2-7B model [33] is used as SLM for tabular classification, leveraging in-context learning with 4-shot demonstrations for better generalization across diverse scenarios. In replicating baseline models, we include traditional machine learning methods such as Logistic Regression (LR), LightGBM, and Random Forest. The hyperparameters for these models were optimized using Grid Search combined with K-fold cross-validation. The value of K was set to either 2 or 4, ensuring that the training set contained at least one example from each class.

4.3 Baseline Methods

We compare our framework against eight baseline models, categorized into three groups. The first set includes conventional supervised learning methods for tabular data, specifically: (1) Logistic Regression (LR), (2) LightGBM [18], and (3) Random Forest (RF) [14]. These models represent widely used, classical approaches for structured data classification. The second group comprises models that use pretraining followed by fine-tuning on specific tasks, including: (4) SCARF [5], (5) STUNT [27], and (6) TabPFN [15]. The final group consists of LLMs-based methods, namely: (7) In-context Learning(AO) [35] and (8) TABLET [30]. In-context learning (AO) embeds few-shot training examples directly within the input prompt, without any parameter tuning. TABLET enhances in-context learning by incorporating additional information, such as rule sets and prototypes from an external classifier, into the prompt to improve inference quality. It is worth noting that, for a fair comparison, we replicated both AO and TABLET using the same Llama2-7B model as the backbone to ensure consistency across the experiments.

4.4 Results and Analysis(RQ1)

Table 3 presents the 4-shot test accuracy results across four datasets. Our proposed framework consistently outperforms all baselines across the datasets, achieving a more than 5% performance improvement for all settings. Notably, conventional supervised learning approaches (Logistic Regression (LR), Random Forest (RF), and LightGBM) demonstrate mixed results, such as Logistic Regression performs well on the income and heart dataset but fails to maintain competitiveness across other datasets. Random Forest performs similarly, excelling on the heart dataset but showing lower accuracy on the bank and creditg datasets. Pre-trained models (SCARF, STUNT, and TabPFN) have more competitive performance. Compared to conventional supervised learning approaches, pre-trained models benefit from incorporating prior knowledge related to downstream tasks during pre-train, leading to more stable performance across different datasets with minimal fluctuations. In contrast, the in-context learning models AO and TABLET struggle to match the performance of the other approaches. This outcome is reasonable, as the utilization of the less powerful Llama2-7B model

Method	Dataset				
	bank	creditg	heart	income	All Tasks (avg)
LR	$53.00_{22.00}$	$51.00_{13.00}$	$68.00_{6.00}$	$67.00_{4.00}$	59.75
RF	$48.00_{10.00}$	$55.00_{4.00}$	$73.00_{3.00}$	$61.00_{6.00}$	59.25
LightGBM	$51.00_{3.00}$	$42.00_{17.00}$	$59.00_{1.00}$	$64.00_{6.00}$	54.00
SCARF	$55.00_{7.00}$	$54.00_{13.00}$	$68.00_{4.00}$	$61.00_{9.00}$	59.50
STUNT	$54.67_{6.65}$	$\mathbf{64.33}_{0.94}$	$53.26_{4.94}$	$66.33_{6.94}$	59.65
TabPFN	$49.00_{6.00}$	$58.00_{4.00}$	$59.00_{4.00}$	$67.00_{2.00}$	58.25
AO	$35.23_{0.88}$	$62.79_{1.26}$	$60.59_{3.29}$	$57.13_{3.49}$	53.94
TABLET	$21.00_{3.55}$	$38.40_{3.45}$	$60.30_{2.95}$	$61.60_{3.95}$	45.33
ours	$\mathbf{55.26}_{0.88}$	$\mathbf{64.33}_{2.19}$	$\mathbf{73.06}_{1.81}$	$\mathbf{67.58}_{2.00}$	**65.06**

Table 3. 4-shot test accuracy results across 4 datasets. All the remaining tables in this paper follow these setups to avoid clutter: the metric values are averaged over 3 random seeds and standard deviations are given as subscripts; the All Tasks (avg) column reports the average accuracy across all datasets; Top results for each dataset are in bold.

Method	ACC (avg)	API Usage (avg)
AO(ChatGPT)	59.90	2400
ours(Llama2-7B)	65.06	100
ours(ChatGPT)	68.79	2500

Table 4. Comparative analysis of trade-off between model performance and API usage

significantly constrains the overall effectiveness of these methods. In summary, our framework demonstrates a clear advantage over both conventional supervised learning models and advanced methods utilizing pretraining or in-context learning. our framework utilizes the explanations of LLMs to guide SLM in inference, contributing to its superior performance across all datasets. These results validate the effectiveness of leveraging our method in few-shot tabular classification.

4.5 Backbone Model Result(RQ2)

Table 4 presents the average accuracy and corresponding average API usage for ChatGPT across all tasks, comparing our proposed methods with AO(ChatGPT). The results demonstrate that our approach utilizing the locally deployable Llama2-7B model significantly reduces API usage while achieving higher average accuracy (6.18% improvement over AO(ChatGPT)). Furthermore, when directly using ChatGPT as the proxy model, our method achieves a substantial performance gain (8.89% improvement) with only a modest increase in API usage (2400->2500). This highlights the flexibility of our approach in balancing model performance and resource efficiency based on task requirements.

Method	bank	creditg	heart	income	All Tasks (avg)
ours	$\mathbf{55.26}_{0.88}$	$\mathbf{64.33}_{2.19}$	$\mathbf{73.06}_{1.81}$	$\mathbf{67.58}_{2.00}$	**65.06**
-post hoc	$40.31_{2.68}$	$54.36_{1.61}$	$67.83_{0.85}$	$66.04_{1.34}$	57.14
-select	$45.15_{2.61}$	$63.53_{2.76}$	$56.60_{3.38}$	$62.45_{2.05}$	56.93
-both(AO)	$35.23_{0.88}$	$62.79_{1.26}$	$60.59_{3.29}$	$57.13_{3.49}$	53.94

Table 5. Ablation of model performance

4.6 Ablation Study(RQ3)

We evaluate the contributions of key components to the overall performance through one-by-one ablation studies, focusing on the following: (1) −post hoc: excluding the process of adding the post hoc explanations to the k-shot demonstrations; (2) −select: removing the process of Post Hoc Explanation-Guided Demonstrations Selection, and (3) −both (AO): omitting all components.

Table 5 summarizes the change in the acc from ablations as well as the average performance across all datasets. The results show that removing any of the key components leads to a noticeable drop in performance. When post hoc explanations are removed (-post hoc), the performance decreases across all datasets, with the bank dataset experiencing the most significant drop in accuracy (from 55.26% to 40.31%). This decline highlights the contribution of post hoc explanations in providing context and improving interpretability for the surrogate language model, thus aiding in accurate classification. Similarly, excluding the demonstration selection component (-select) results in a noticeable decrease in performance. This suggests that the selection of relevant demonstrations is critical for guiding the model effectively, as it enables the model to leverage representative examples that are closely aligned with the target samples. The combined ablation (-both (AO)) yields the lowest average performance (53.94%), demonstrating the compounded effect of removing both components. The combined ablation performs significantly worse than the full model, indicating that both post hoc explanations and guided demonstration selection are essential for maximizing model accuracy on tabular data classification tasks. In summary, these ablation results confirm that each component contributes uniquely to the model's success, and their combination is essential to achieving robust, high-performing results across varied datasets. The findings further validate the efficacy of our proposed approach in leveraging explanations and optimized demonstration selection to enhance few-shot learning for tabular data classification.

4.7 Impact of Few-shot demonstrations Selection algorithm(RQ4)

Figure 3 compares the performance of our framework under different few-shot demonstration selection algorithms across four tasks on tabular datasets, using the random demonstration selection algorithm as the baseline. The results reveal that for diversity-based selection algorithms (clustering) and distance-based

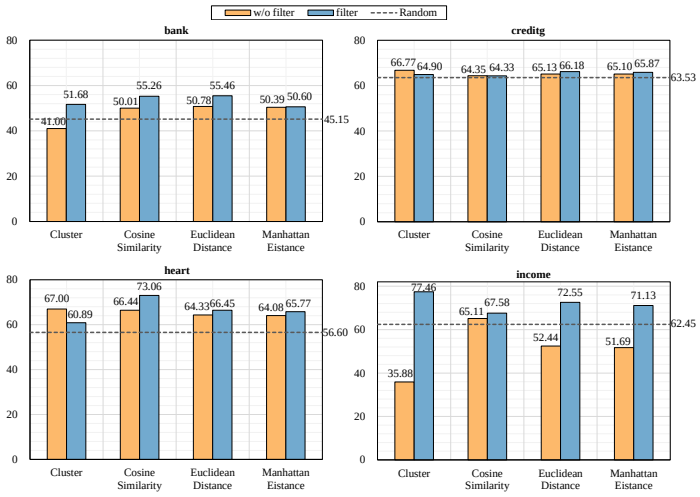

Fig. 3. Comparison of different demonstration selection algorithms and baselines (random selection) on four datasets.

similarity algorithms (Euclidean distance and Manhattan distance), the performance of the unfiltered(w/o filter) approach is occasionally inferior to random selection on certain datasets. However, with the integration of filtering(filter), all these algorithms outperform random selection consistently across all datasets. This demonstrates the critical role of filtering in mitigating noise during the demonstration selection process, enabling the selection of optimal demonstrates to guide the SLM. Additionally, the cosine similarity-based selection algorithm outperforms random selection across all datasets, regardless of the presence of filtering. This consistent superiority underscores the algorithm's robustness and suitability for tabular data, making it a reliable choice for demonstration selection in few-shot learning settings. These findings highlight the benefits of filtering and the effectiveness of cosine similarity in enhancing the quality of few-shot demonstrations for downstream performance.

4.8 Hyper-parameter analysis

Figure 4 illustrates the performance of our model across four datasets, under two different hyper-parameters: p (left) and n (right). In the left Figure, acc values are displayed as a function of the parameter p, which ranges from 0 to 1. The results indicate a relatively stable acc across all datasets, with minor fluctuations as p varies. Notably, the income and creditg datasets maintain a higher and more consistent acc performance compared to the bank and heart datasets. The heart dataset shows greater variability, with peaks and troughs as p increases, suggesting sensitivity to changes in this parameter.

The right plot presents acc values as a function of parameter n, which takes integer values from 1 to 10. where the income and creditg datasets again demon-

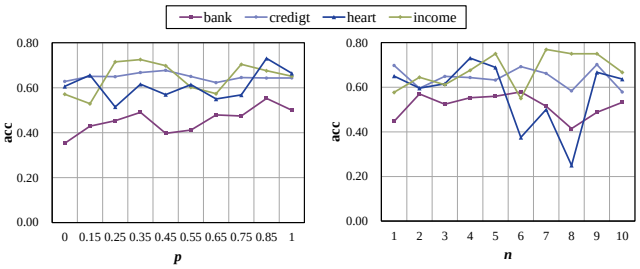

Fig. 4. Effect of filter threshold p and the number of post hoc explanations n. The accuracy (acc) across all datasets is reported.

strate relatively stable acc scores. Conversely, the heart dataset exhibits substantial variability, particularly a marked drop in acc at $n = 6$, before recovering at higher values. This variability may indicate that the heart dataset is more sensitive to the selection or quantity of examples (parameter n) than the other datasets.

Overall, these results suggest that the choice of parameter n may have a significant impact on model performance, particularly for datasets like heart. In contrast, the parameter p appears to have a more modest effect on acc across datasets, though some variability is noted. The income and creditg datasets appear to be more robust to changes in both p and n, consistently achieving higher acc values, while the bank and heart datasets show more sensitivity, highlighting the importance of parameter tuning for optimal performance on these datasets.

5 Conclusion

This paper proposes an in-context learning framework leveraging the priori knowledge of Large Language Models (LLMs) to guide a SLM for tabular learning. By using LLMs as post hoc explanation generators, we address challenges like efficient demonstration selection and reduce dependency on resource-intensive models, achieving a practical, interpretable, and cost-efficient learning process. Experimental results showed a 5.31% accuracy improvement across tabular datasets, demonstrating the framework's effectiveness, especially in data-limited environments. The framework provides a scalable balance between performance and operational cost. Future work will focus on extending these methods to other data types and refining demonstration selection for broader applications.

Acknowledgments. This work was supported by a grant from the National Natural Science Foundation of China under grants (No.62372211, 62272191), and the International Science and Technology Cooperation Program of Jilin Province (No.20230402076GH, No. 20240402067GH), and the Science and Technology Development Program of Jilin Province (No. 20220201153GX).

References

1. Arik, S.Ö., Pfister, T.: Tabnet: Attentive interpretable tabular learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 6679–6687 (2021)
2. Asuncion, A., Newman, D.: Uci machine learning repository (2007)
3. Badirli, S., Liu, X., Xing, Z., Bhowmik, A., Doan, K., Keerthi, S.S.: Gradient boosting neural networks: Grownet. arXiv preprint arXiv:2002.07971 (2020)
4. Bahri, D., Jiang, H., Tay, Y., Metzler, D.: Scarf: Self-supervised contrastive learning using random feature corruption. arXiv preprint arXiv:2106.15147 (2021)
5. Bahri, D., Jiang, H., Tay, Y., Metzler, D.: Scarf: Self-supervised contrastive learning using random feature corruption. In: International Conference on Learning Representations (2022)
6. Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)
7. Chen, S., Wu, J., Hovakimyan, N., Yao, H.: Recontab: Regularized contrastive representation learning for tabular data. arXiv preprint arXiv:2310.18541 (2023)
8. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794 (2016)
9. Dinh, T., Zeng, Y., Zhang, R., Lin, Z., Gira, M., Rajput, S., Sohn, J.y., Papailiopoulos, D., Lee, K.: Lift: Language-interfaced fine-tuning for non-language machine learning tasks. Advances in Neural Information Processing Systems **35**, 11763–11784 (2022)
10. Fan, Z., Gao, X., Mirchev, M., Roychoudhury, A., Tan, S.H.: Automated repair of programs from large language models. In: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE). pp. 1469–1481. IEEE (2023)
11. Finch, S.E., Paek, E.S., Choi, J.D.: Leveraging large language models for automated dialogue analysis. In: Proceedings of the 24th Meeting of the Special Interest Group on Discourse and Dialogue. pp. 202–215 (2023)
12. Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A.: Revisiting deep learning models for tabular data. Advances in Neural Information Processing Systems **34**, 18932–18943 (2021)
13. Hastie, T.J., Pregibon, D.: Generalized linear models. In: Statistical models in S, pp. 195–247. Routledge (2017)
14. Ho, T.K.: Random decision forests. In: Proceedings of 3rd international conference on document analysis and recognition. vol. 1, pp. 278–282. IEEE (1995)
15. Hollmann, N., Müller, S., Eggensperger, K., Hutter, F.: Tabpfn: A transformer that solves small tabular classification problems in a second. In: The Eleventh International Conference on Learning Representations (2023)
16. Huang, X., Khetan, A., Cvitkovic, M., Karnin, Z.: Tabtransformer: Tabular data modeling using contextual embeddings. arXiv preprint arXiv:2012.06678 (2020)
17. Kadra, A., Lindauer, M., Hutter, F., Grabocka, J.: Well-tuned simple nets excel on tabular datasets. Advances in neural information processing systems **34**, 23928–23941 (2021)
18. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)
19. Ke, G., Xu, Z., Zhang, J., Bian, J., Liu, T.Y.: Deepgbm: A deep learning framework distilled by gbdt for online prediction tasks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 384–394 (2019)

20. Ke, G., Zhang, J., Xu, Z., Bian, J., Liu, T.Y.: Tabnn: A universal neural network solution for tabular data (2018)
21. Klambauer, G., Unterthiner, T., Mayr, A., Hochreiter, S.: Self-normalizing neural networks. Advances in neural information processing systems **30** (2017)
22. LaValley, M.P.: Logistic regression. Circulation **117**(18), 2395–2399 (2008)
23. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)
24. Loh, W.Y.: Classification and regression trees. Wiley interdisciplinary reviews: data mining and knowledge discovery **1**(1), 14–23 (2011)
25. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. Decision Support Systems **62**, 22–31 (2014)
26. Nam, J., Song, W., Park, S.H., Tack, J., Yun, S., Kim, J., Shin, J.: Semi-supervised tabular classification via in-context learning of large language models. In: Workshop on Efficient Systems for Foundation Models@ ICML2023 (2023)
27. Nam, J., Tack, J., Lee, K., Lee, H., Shin, J.: Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables. In: The Eleventh International Conference on Learning Representations (2023)
28. Popov, S., Morozov, S., Babenko, A.: Neural oblivious decision ensembles for deep learning on tabular data. arXiv preprint arXiv:1909.06312 (2019)
29. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: Catboost: unbiased boosting with categorical features. Advances in neural information processing systems **31** (2018)
30. Slack, D., Singh, S.: Tablet: Learning from instructions for tabular data. arXiv preprint arXiv:2304.13188 (2023)
31. Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C.B., Goldstein, T.: Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. arXiv preprint arXiv:2106.01342 (2021)
32. Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., Tang, J.: Autoint: Automatic feature interaction learning via self-attentive neural networks. In: Proceedings of the 28th ACM international conference on information and knowledge management. pp. 1161–1170 (2019)
33. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
34. Wang, R., Shivanna, R., Cheng, D., Jain, S., Lin, D., Hong, L., Chi, E.: Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In: Proceedings of the web conference 2021. pp. 1785–1797 (2021)
35. Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022)
36. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems **35**, 24824–24837 (2022)
37. Yoon, J., Zhang, Y., Jordon, J., van der Schaar, M.: Vime: Extending the success of self-and semi-supervised learning to tabular domain. Advances in Neural Information Processing Systems **33**, 11033–11043 (2020)
38. Zhang, Z., Zhang, A., Li, M., Smola, A.: Automatic chain of thought prompting in large language models. In: The Eleventh International Conference on Learning Representations