# Multi-level Diagnosis and Evaluation for Robust Tabular Feature Engineering with Large Language Models

**Yebin Lim**
Computer Science and Engineering
Korea University
yebinuni@korea.ac.kr

**Susik Yoon**
Computer Science and Engineering
Korea University
susik@korea.ac.kr

## Abstract

Recent advancements in large language models (LLMs) have shown promise in feature engineering for tabular data, but concerns about their reliability persist, especially due to variability in generated outputs. We introduce a multi-level diagnosis and evaluation framework to assess the robustness of LLMs in feature engineering across diverse domains, focusing on the three main factors: key variables, relationships, and decision boundary values for predicting target classes. We demonstrate that the robustness of LLMs varies significantly over different datasets, and that high-quality LLM-generated features can improve few-shot prediction performance by up to 10.52%. This work opens a new direction for assessing and enhancing the reliability of LLM-driven feature engineering in various domains. Our source code is available at https://github.com/DohaLim/Robustness-eval.

## 1 Introduction

Recent breakthroughs in large language models (LLMs) have opened new possibilities in tabular learning, such as feature engineering, question answering, and table comprehension (Fang et al., 2024). The extensive pretrained knowledge of LLMs, when equipped with only a few examples, can automate costly data science workflows manually handled by domain experts. Notably, recent studies have shown that LLM-driven feature engineering can help outperform traditional tabular prediction methods, especially in a few- or zero-shot settings (Han et al., 2024; Hegselmann et al., 2023; Hollmann et al., 2023).

Despite these promising results, the open-ended nature of LLM-generated outputs has raised concerns about their robustness (Huang et al., 2023). Existing approaches for feature generation focus on either *feature-feature relationships* with predefined operators (Hollmann et al., 2023) or *feature-target relationships* with unbounded rule conditions (Han et al., 2024). They rely on in-context learning by LLM with arbitrary domain knowledge and a few samples, which inherently entails risks of inconsistency and unreliability in outputs. Accordingly, evaluating the reliability of LLM-generated features remains a significant challenge.

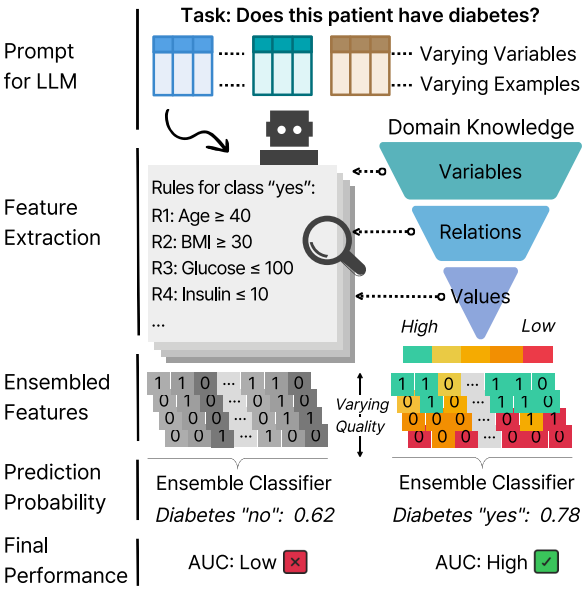


Figure 1: LLMs vulnerable to generating features of varying quality (left). Measures for high-quality features leading to performance improvement (right).

For example, in Figure 1, given the task of predicting diabetes for a new patient, an LLM is asked to generate a set of new features describing feature-target relationships (i.e., between patient information and diabetes). An ensemble classifier then uses these new features to make a final prediction. The LLM-generated features produced by the state-of-the-art approach (Han et al., 2024), however, could be ineffective (e.g., '$\texttt{Glucose} \leq 100$ and $\texttt{Insulin} \leq 10$' for $\texttt{Diabetes} = \text{yes}$) depending on the quality of input samples and the LLM's inherent knowledge. This variability can introduce noise into the resulting prediction probabilities, potentially degrading the overall classifier performance.



*Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4630–4655
November 4-9, 2025 

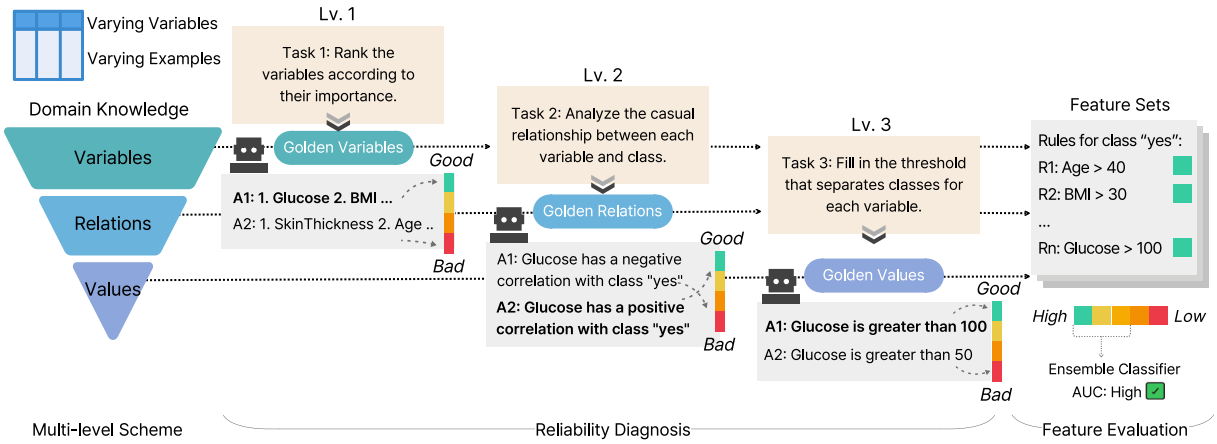


Figure 2: Overall procedure of our framework involves a multi-level scheme of variables, relations, and values to diagnose reliability and evaluate features generated by LLMs in feature engineering on different domains and inputs.

For more practical and reliable LLM-generated features, it is crucial to understand the consistency of their performance on feature engineering under varying contexts. Although significant strides have been made in evaluating the robustness of LLM (Chang et al., 2024; Kenthapadi et al., 2024), there remains insufficient exploration of these aspects in feature engineering, especially in the context of feature-target relationships. A recent work ELF-GYM (Zhang et al., 2024) has attempted to compare LLM-generated features with human-crafted ones, but further investigation is lacking regarding the capabilities and limitations of LLMs with varying domain knowledge and examples.

To address this gap, we propose a framework that systematically diagnoses and evaluates the robustness of LLMs in feature engineering for tabular data. We focus on how consistently LLMs maintain reliability in engineering features for tabular prediction, the most prevalent task in tabular learning. Specifically, drawing inspiration from real-world practices of domain experts, we identify three core elements considered in feature engineering: golden variable, golden relation, and golden value. Our framework incorporates a novel *multi-level scheme* to analyze LLM-generated features, specifically addressing the following research questions.

- **RQ1 (Golden Variable):** *Can LLMs identify key variables highly correlated with target classes given varying domain knowledge?*
- **RQ2 (Golden Relation):** *Can LLMs understand the causal relationship (i.e., correlation polarity) between golden variables and target classes?*
- **RQ3 (Golden Value):** *Can LLMs set the decision boundary values of golden variables that differentiate the target classes?*

Figure 2 shows the overall procedure of the proposed diagnosis and evaluation framework. Based on the multi-level scheme, we first conduct a reliability diagnosis to assess the consistency in LLM responses across varying contexts at each level. This serves as a fine-grained proxy to measure the trustworthiness of an LLM in generating features for a given dataset. The robustness of an LLM given a dataset directly influences the quality of the generated features; less robust models may produce features of varying quality, leading to prediction performance degradation. Thus, we further conduct an evaluation on the generated features to investigate how high-quality features can enhance the effectiveness of LLM-driven feature engineering and ultimately improve the prediction performance.

**Summary** We demonstrated the efficacy of the proposed framework through comprehensive experiments on six LLMs and eight benchmark datasets In brief, the multi-level diagnosis results show that the robustness of LLMs in feature engineering varies significantly across datasets with diverse domains, and the multi-level evaluation contributes to the improvement of few-shot prediction performance. Our key contributions and findings are summarized as follows:

- To the best of our knowledge, this is the first work to address the robustness of LLMs in feature engineering with feature-target relationships for tabular prediction in the multi-level scheme.
- With reliability diagnosis, we confirm the significant variations in the robustness of LLMs in feature engineering across different datasets.
- Our analysis reveals that simply adding more descriptions or examples does not necessarily

lead to performance gains, whereas providing high-quality examples is critical for improving the robustness of LLMs in feature engineering.
- We empirically demonstrate that utilizing high-quality features identified through our evaluation scheme enhances the prediction performance of the state-of-the-art method by up to $10.52\%$.

## 2 Related Work

### 2.1 Few-shot Tabular Learning

Tabular data, consisting of distinct data instances (i.e., rows) and their variables (i.e., columns), is one of the most prevalent data types in real-world. Achieving high and robust predictive performance can provide significant benefits in data science applications (Ruan et al., 2024). A longstanding challenge of learning on tabular data involves reasoning over structured and semantically sparse data, where each variable has a fixed type (e.g., numerical or categorical) with potentially unbounded values and a domain-specific context within the predictive modeling task.

Recently, LLMs, initially trained on extensive textual corpora, have demonstrated significant capabilities in generalizing to unseen tasks (Brown et al., 2020), prompting investigations into their utility for tabular data learning. Early approaches focused on converting tabular data into serialized textual prompts, enabling direct handling by LLMs (Dinh et al., 2022; Hegselmann et al., 2023; Wang et al., 2023, 2024; Zhang et al., 2023). Despite its effectiveness in few-shot settings, the reliance on an expensive LLM for the entire inference process, coupled with limited interpretability, poses practical challenges. Consequently, the research focus has been shifted toward utilizing LLMs primarily as *feature engineers* rather than employing them in an end-to-end, black-box prediction.

### 2.2 LLMs for Feature Engineering

**Feature Selection** Recent studies have explored the use of LLMs for extracting domain-relevant knowledge to aid in feature selection tasks. Choi et al., 2022 proposed leveraging LLMs as a knowledge source to guide feature selection with the induced feature importance. Building on this idea, Jeong et al., 2024 introduced three different prompts that directly utilize the textual outputs generated by LLMs for feature selection tasks. Additionally, Li et al., 2024 demonstrated that this text-based approach is not only more robust than traditional data-driven approaches based on statistical inference from samples but also delivers competitive performance across diverse scenarios, including resource-limited settings.

**Feature Generation** To move beyond the straightforward selection of predefined features, researchers have increasingly leveraged LLMs to generate features. A line of research focuses on *feature-feature relationships*, by utilizing predefined operators (e.g., add or multiply). Hollmann et al., 2023 integrated LLMs into the AutoML process to iteratively generate additional features by leveraging the dataset's semantic and contextual descriptions, enhancing model performance by embedding domain knowledge. Zhang et al., 2024 proposed a framework that assesses the quality of LLM-generated features by comparing them to human-engineered ones. They quantified the gap between the two feature sets in terms of semantic and functional similarity and identified its impact on downstream task performance.

On the other hand, another line of research emphasizes *feature–target relationships*, aiming to generate feature-wise rules directly related to each target class. Han et al., 2024 employed LLMs to create binary features through rule generation and parsing, achieving significant improvements in downstream tabular prediction tasks. However, the core challenge in this approach is that selecting and transferring meaningful rule conditions involves navigating a large combinatorial search space grounded in a given table schema and domain logic. Moreover, due to the limited input sequence lengths in LLM, tabular inference performance remains highly sensitive to subtle variations in prompts and potentially spurious correlations in samples (Wen et al., 2024; Gardner et al., 2024).

However, no studies to date have systematically evaluated the robustness of LLMs in feature engineering, specifically in the scope of feature-target relationships, leaving a critical gap in understanding their consistency and reliability. This gap is especially important given the complex and potentially unbounded search space for feature engineering, which often leads models to produce overly broad or unstable responses. In response, our framework focuses on analyzing the reliability of LLMs in feature–target settings, where prior work has reported strong performance but robustness has not been thoroughly studied.

## 3 Methodology

### 3.1 Preliminary

Given a tabular dataset $D = \{(x^i, y^i)\}_{i=1}^N$ of $N$ labeled samples, each sample $x^i$ includes an original feature set of $d$ dimensional variables, $F = \{f_j\}_{j=1}^d$. We utilize LLMs to transform the original feature set $F$ into a new feature set $F'$ through prompted feature engineering, $F \xrightarrow{\text{LLM}} F'$. The transformed feature set $F'$ is then used as input to a classifier to predict the target class $y$. A general pipeline of relevant works is summarized as:

1. **Prompting for LLM**: The first step involves providing a well-structured input prompt to the LLM. This input typically includes a task description, variable descriptions, and a few samples with original features and true labels.
2. **Feature Selection/Generation via LLM**: Once prompted, LLMs either select relevant features from the dataset (*feature selection*) or generate new feature representations (*feature generation*). For example, an LLM-driven feature rule can transform an original feature such as `Glucose` into a new feature rule `Glucose` $\geq$ `100`.
3. **Featurization**: The next step is to transform the new feature of samples into structured input for classification modeling. For example, the new feature rules {`Age` $\geq$ `40`, `Glucose` $\geq$ `100`, ...} can form a corresponding binary feature set $[0, 1, \ldots]$ of a sample.
4. **Model Training**: The final step involves training a machine learning model using the new feature set. This phase assesses the effectiveness of LLM-generated features by evaluating predictive performance.

### 3.2 Overview

To evaluate the robustness of LLMs in feature engineering, we propose a multi-level diagnosis and evaluation framework built upon three fundamental aspects of domain expertise, which are essential for reliable feature engineering.

- **Level 1 (Identifying Key Variables):** LLMs are tested on their ability to recognize the most important variables for a given task. Domain experts can readily identify important variables that are crucial for prediction, such as `Glucose` in diabetes classification. We introduce perturbations in variable descriptions and samples to examine whether LLMs can consistently rank the correct variables.
- **Level 2 (Understanding Variable-Class Relationships):** This level evaluates whether LLMs can correctly determine the causal relationship between variables and target classes. While experts understand that high `Glucose` levels are positively correlated with diabetes, an LLM might generate incorrect associations depending on input variations. We test robustness by altering sample quality and variable value mixing.
- **Level 3 (Setting Decision Boundaries):** Domain knowledge is often reflected in the ability to determine boundary values that separate classes. For example, experts might set a `Glucose` threshold above 100 to indicate diabetes. We assess whether LLMs can provide stable decision boundaries under different input perturbations.

Based on the multi-level scheme, we first assess the reliability of LLM responses to evaluate their ability to handle variations in input conditions. This assessment helps determine the robustness of LLM-driven feature engineering across different models and datasets. Furthermore, we utilize the multi-level scheme as a framework for feature evaluation, ensuring that LLM-generated features align with domain knowledge and maintain high quality.

In this study, we demonstrate how each factor in the multi-level scheme can be derived from statistical information in datasets. In real-world scenarios, diagnosis and evaluations can be easily performed based on criteria established by domain experts.

### 3.3 Multi-level Reliability Diagnosis

At each level, we introduce variations in the input and measure how LLM-generated responses change. The variations include differences in variable descriptions, ordering, sample quality, and mixing strategies. This setup allows us to categorize LLM outputs into *high-score cases*, where the predictions align with domain knowledge, and *low-score cases*, where inconsistencies emerge. By analyzing the response patterns under different conditions, we gain insights into how robust LLMs are in performing feature engineering tasks.

#### 3.3.1 Level 1: Golden Variable

**Definition** Among the variables in $F$, we define $F_{\text{golden}}$ as the subset of variables most strongly associated with the target class $y$:

$$F_{\text{golden}} = \{f_j \mid |\text{Covariance}(f_j, y)| \geq \gamma\}.$$

Specifically, covariances between each variable and the target class are computed and ranked by their

absolute values (Lazar et al., 2012). The elbow method is then used to determine a threshold $\gamma$ by identifying the largest gaps. Categorical variables are represented by the one-hot encoded feature having the highest absolute covariance.

**Prompt** An LLM is asked to rank the variables in order of importance, provided with a task description, variable descriptions, and examples:

> *..., rank variables according to their importance to solve the task, ..., [Task] [Variables] [Examples]*

The detail of information for variables and example conditions can be varied to measure reliability at level 1. See Appendix C for the complete prompt.

**Reliability Score ($\mathcal{RS}_1$)** Using the rankings obtained from the LLM's responses and the identified golden variables, a rank score is computed to evaluate how well the golden variables are positioned in the higher ranks.

The rank score for each golden variable $f \in F_{\text{golden}}$ is defined as:

$$S_{Rank}(f) = 1 - \frac{\text{Rank}(f) - 1}{|F|},$$

where $\text{Rank}(f)$ represents the rank of variable $f$ in the LLM's response, and $|F|$ is the total number of variables in the dataset. The overall reliability score for Level 1 is calculated as the average rank score of all golden variables.

#### 3.3.2 Level 2: Golden Relation

**Definition** The golden relation between golden variables $F_{\text{golden}}$ and target class $y$ is defined by the direction of their correlation:

$$R_{\text{golden}} = \begin{cases} \text{Positive}, & \text{if Covariance}(f, y) > 0, \\ \text{Negative}, & \text{if Covariance}(f, y) < 0. \end{cases}$$

**Prompt** An LLM is asked to identify the relationship between key variables and target classes, provided with a task description, variable descriptions, and examples.

> *..., analyze the causal relationship or tendency between each variable and class, ..., [Task] [Variables] [Examples]*

The number of examples, sampling methods for examples, and variable corruption can be varied to measure reliability at level 2. See Appendix C for the complete prompt.

**Reliability Score ($\mathcal{RS}_2$)** To measure the accuracy of LLM-generated variable-class relations, we define a correctness function based on the exact match principle. Given a feature $f \in F_{\text{golden}}$, its golden relation $R_{\text{LLM}}$ from LLM, and true golden relation $R_{\text{golden}}$, we define a correctness score as:

$$S_{correct}(f, R_{\text{LLM}}, R_{\text{golden}}) = \mathbb{1}_{(R_{\text{LLM}} = R_{\text{golden}})}.$$

The overall reliability score for Level 2 is computed as the average correctness score across all golden variables.

#### 3.3.3 Level 3: Golden Value

**Definition** Since domain experts typically have insights into distinguishing classes based on key variable values, we define the golden value as the specific variable value that best separates the classes. Specifically, for numerical variables $f$ with range $[f^{\min}, f^{\max}]$, the golden value is the value $v$ that maximizes the AUC score:

$$V_{\text{golden}} = \text{argmax}_{v \in [f^{\min}, f^{\max}]} \text{AUC}.$$

For categorical variables, the golden value is the value most correlated with the target class.

**Prompt** An LLM is asked to fill in the feature condition, provided with a task description, variable descriptions, and examples.

> *..., fill in the variable conditions for each class to solve the task. [Task] [Variables] [Examples]*

The number of examples, sampling methods for examples, and variable corruptions can be varied to measure reliability at level 3. See Appendix C for the complete prompt.

**Reliability Score ($\mathcal{RS}_3$)** Given a variable $f \in F_{\text{golden}}$, the value $V_{\text{LLM}}$ returned from LLM, and true golden value $V_{\text{golden}}$, the correctness of predicted threshold value is evaluated using normalized error as follows:

$$\mathcal{RS}_3 = 1 - |N(V_{\text{LLM}}) - N(V_{\text{golden}})|,$$

where $N()$ is the min-max normalization. The overall reliability score is computed as the average correctness score across all golden variables.

#### 3.3.4 Diagnosis Result Highlights

We preview the diagnosis results before fully discussing in Section 4.1. In Figure 3, the average reliability scores of models in the default setting

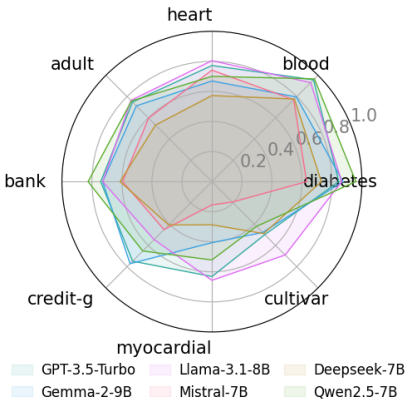


Figure 3: Variation of reliability scores (averaged over three levels) for different LLMs and datasets.

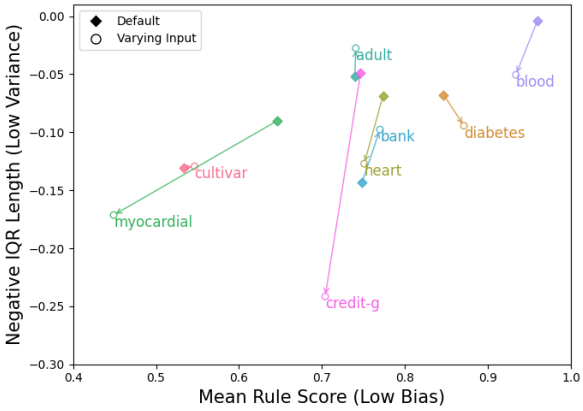


Figure 4: The change of variance and bias of reliability score with varying input for GPT-3.5-Turbo.

vary across datasets. This demonstrates the uncertainty in the reliability of LLMs for feature engineering, which depends on their prior knowledge of the dataset domain. Figure 4 highlights how `bias` (i.e., correct responses) and `variance` (i.e., consistent responses) in an LLM's reliability fluctuate across datasets with varying inputs, emphasizing the necessity of evaluating the quality of LLM's feature engineering results.

### 3.4 Multi-level Feature Evaluation

To address the uncertainty in the robustness of LLMs on different datasets, we introduce a simple yet effective method for verifying the quality of transformed feature set $F'$ through the multi-level evaluation scheme. The corresponding results and analyses, examined using the state-of-the-art feature engineering method (Han et al., 2024), are presented in Section 4.2.

#### 3.4.1 Level 1: Golden Variable

**Feature Score ($\mathcal{FS}_1$)** To evaluate the correctness of feature selection, we measure the F1-score of the variables in the transformed feature set $F_{\text{LLM}}$ against $F_{\text{golden}}$:

$$\mathcal{FS}_1 = \frac{2 \times P \times R}{P + R}, \text{ where}$$

$$P = \frac{\sum_{f \in F_{\text{LLM}}} S_{correct}(f, F_{\text{LLM}}, F_{\text{golden}})}{|F_{\text{LLM}}|}$$

$$R = \frac{\sum_{f \in F_{\text{golden}}} S_{correct}(f, F_{\text{LLM}}, F_{\text{golden}})}{|F_{\text{golden}}|}$$

#### 3.4.2 Level 2: Golden Relation

**Feature Score ($\mathcal{FS}_2$)** Transformed feature sets are evaluated based on their alignment with the class-specific variable relations. Given a variable $f$, LLM-generated relation $R_{\text{LLM}}$, and ground-truth relation $R_{\text{golden}}$, the overall score for golden relation evaluation is defined as:

$$\mathcal{FS}_2 = \frac{1}{|F_{\text{golden}}|} \sum_{f \in F_{\text{golden}}} S_{correct}(f, R_{\text{LLM}}, R_{\text{golden}})$$

#### 3.4.3 Level 3: Golden Value

**Feature Score ($\mathcal{FS}_3$)** The correctness of predicted threshold value is evaluated using normalized error as follows:

$$\mathcal{FS}_3 = 1 - |N(V_{\text{LLM}}) - N(V_{\text{golden}})|,$$

where $N()$ is the min-max normalization. For categorical variables, $\mathcal{FS}_3 = 1$ if value $\in V_{\text{golden}}$ or 0.5 otherwise.

## 4 Experiments

### 4.1 Reliability Diagnosis

#### 4.1.1 Reliability Diagnosis Setting

**LLMs** We employ `GPT-3.5-Turbo` as the base model, following the state-of-the-art feature engineering method (Han et al., 2024). Considering the versatility and usability in various scenarios, we also employ lightweight models, which are `Gemma-2-9B`, `Llama-3.1-8B`, `Mistral-7B`, `Qwen2.5-7B`, `Deepseek-7B` to compare their reliability scores with those of the base model.

**Datasets** We utilized eight binary classification datasets commonly adopted in recent studies on tabular feature engineering and prediction, ensuring consistency with prior research. These datasets were selected based on several criteria, including diverse application domains—such as healthcare (e.g., Blood (Yeh et al., 2009), Diabetes (Smith et al., 1988), Heart (Fedesoriano,

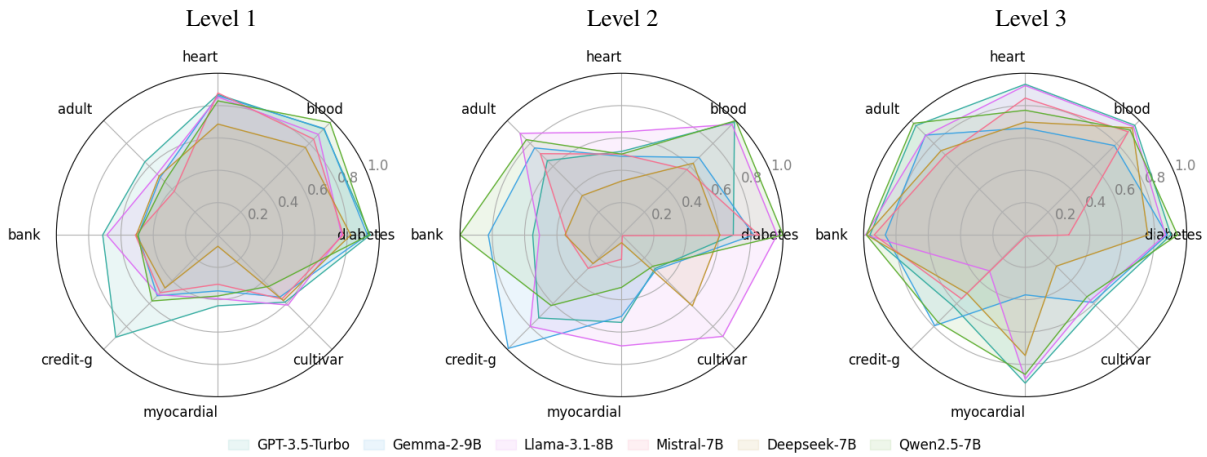


Figure 5: Variation of reliability scores of each level for different models and datasets.

2021)), finance (e.g., Adult (Asuncion and Newman, 2007), Bank (Moro et al., 2014), Credit-g (Kadra et al., 2021)), and agriculture (e.g., Cultivar (Rodrigues de Oliveira and Mario Zuffo, 2023))—as well as varying dataset sizes (e.g., Myocardial (Golovenkin et al., 2020), comprising 111 variables). Additionally, we incorporated an out-of-distribution scenario using Cultivar, representing a domain that LLMs are unlikely to have encountered during pre-training.

#### 4.1.2 Reliability Diagnosis Results

Figure 5 shows the reliability diagnosis results of each level across models and datasets. To further understand the factors that affect reliability scores, we conduct a series of analyses at each level. (Figures 6–8 show representative results; see Appendix B.3 for the full results).

**Which Model Performs Best in the Zero-shot Setting?** We compare LLMs by providing detailed variable descriptions without examples across levels. Even within the same dataset, models exhibit distinct strengths at each level. GPT-3.5-Turbo outperforms in identifying golden variables at Level 1, particularly excelling in Credit-g. However, its performance drops significantly in Level 2, where it must infer relationships between variables and the target class. Conversely, Gemma-2-9B shows strong performance in Credit-g at Levels 2 and 3. Llama-3.1-8B excels in Level 2, especially in Cultivar, but performs worse in Bank compared to other levels. Qwen-2.5-7B consistently performs well across all three tasks in Blood and Diabetes, though it exhibits large performance gaps across levels in Bank and Adult. Deepseek-7B performs the worst, often responding with "neutral" when asked about correlations in Level 2. Mistral-7B also struggles in Level 2 and 3, producing hallucinated responses when faced with datasets containing many variables (e.g., Myocardial) or out-of-distribution scenarios (e.g., Cultivar).

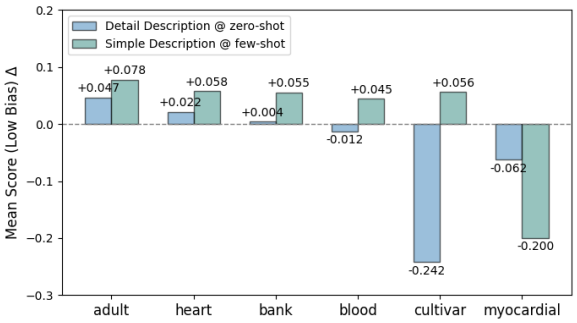


Figure 6: Impact of additional descriptions and examples on reliability scores at Level 1.

To analyze how different input conditions affect LLM responses, we conduct experiments using GPT-3.5-Turbo under controlled variations in the following robustness analyses. Specifically, we further adjusted the number of shots, sampling methods, and variable corruption strategies.

**Does Adding More Descriptions and Examples Improve Robustness?** Figure 6 compares reliability scores in terms of `bias` when either descriptions or examples are added under the simplest description setting (i.e., variable name only) in zero-shot. Across most datasets (e.g., Adult, Bank, Blood, Cultivar, Heart), adding examples improves the reliability score more effectively than descriptions. In some cases (e.g., Blood and Cultivar), additional descriptions even degrade performance. For datasets with a large number of variables, such as Myocardial, both descriptions and examples negatively impact the reliability score. These findings indicate that additional information does not al-

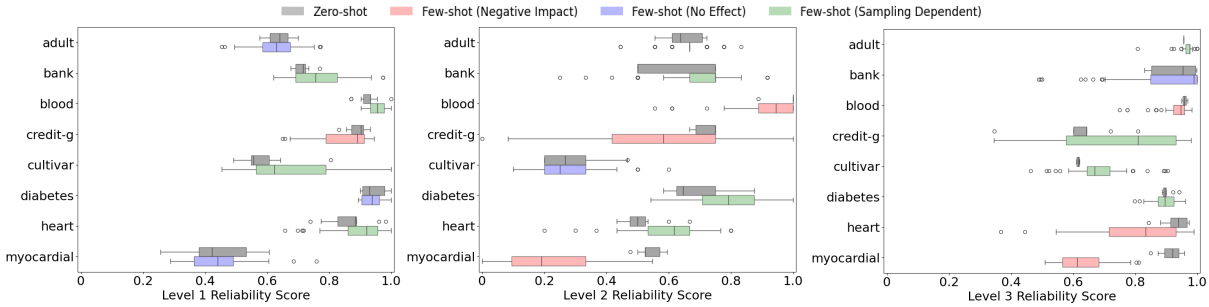


Figure 7: Effects of varying the number of examples on reliability scores at each level for GPT-3.5-Turbo.

ways lead to better outcomes. Figure 7 shows the different types of robustness change patterns by dataset, when few-shot samples are given with detailed variable descriptions.

- Negative Impact of Few-Shot: In some cases, the few-shot approach produced lower scores than zero-shot. For example, in the credit-g dataset, few-shot resulted in lower scores and greater variability (Level 1). Similar patterns were observed in the myocardial, credit-g, and blood datasets (Level 2). In myocardial, heart, and blood, adding sample data did not help and sometimes led to decreased scores (Level 3). This finding implies that the provided samples sometimes functioned as noise.
- No Effect: Several datasets showed little to no difference between zero-shot and few-shot performance. For instance, adult, diabetes, and myocardial showed no significant change (Level 1). Adult and cultivar maintained stable performance across both methods (Level 2). Bank did not benefit from the sampling approach, showing no notable score changes (Level 3). This indicates that methods other than few-shot prompting might be required to help the model learn meaningful variable–target relations.
- Dependence on Sampling: Some datasets benefited from few-shot prompting, but the effectiveness depended heavily on the sample quality. For example, bank, blood, cultivar, and heart showed performance improvements (Level 1). For bank, diabetes, and heart, the presence of samples improved scores, albeit inconsistently depending on the sampling method (Level 2). Datasets such as diabetes, adult, credit-g, and cultivar exhibited improved scores when samples were provided, although credit-g was particularly sensitive to the quality of those samples (Level 3). This underscores the importance of sample selection that aids the model's reasoning.

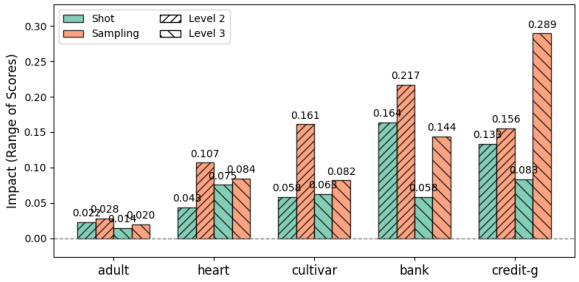


Figure 8: Impact of the number of shots and example quality on reliability scores at Levels 2 and 3.

**Does Sampling Matter More than the Number of Shots?** Figure 8 shows the impact of the number of shots and the sampling method at Levels 2 and 3. Across most datasets, sampling quality exerts a stronger influence on robustness than the number of shots, with larger effects observed in Heart and Cultivar at Level 2 and in Credit-g and Bank at Level 3. Besides, in-depth analysis between worst/best sampling and random sampling reveals additional dataset-specific patterns. At Level 1, low-quality examples in Bank substantially degraded performance when identifying key variables. At Level 2, high-quality samples in Credit-g led to significant gains when inferring variable–class relations. At Level 3 the gap between high- and low-quality examples is substantial when identifying golden values. In Bank and Blood, however, this gap decreases as the number of shots increases, indicating that more examples can mitigate the negative effect of poor-quality examples.

## 4.2 Feature Evaluation

### 4.2.1 Feature Evaluation Setting

We evaluate the performance of binary classification in the eight datasets. We also used Communities (Redmond, 2009) to show extendability of our framework in a multi-class classification task (see Appendix B.2). We compare three conventional classifiers, (1) Logistic regression (LogReg), (2) XGBoost (Chen and Guestrin, 2016), and (3)

Table 1: Few-shot classification performance evaluation results. We used only the top 3 feature sets or excluded the bottom 3 feature sets based on the average evaluation scores of three levels.

| Data | Shot | LogReg | RandomForest | XGBoost | FeatLLM | Ours (Top 3) | Ours (w/o Bottom 3) | Improvement (%) |
|---|---|---|---|---|---|---|---|---|
| Credit-g | 4 | 56.77 ± 11.93 | 51.35 ± 8.5 | 50.0 ± 0.0 | 52.27 ± 8.38 | **57.77 ± 5.37** | 55.65 ± 7.08 | ▲10.52 |
| | 8 | 49.7 ± 12.84 | 57.06 ± 8.59 | 49.84 ± 6.37 | 58.87 ± 4.69 | **62.89 ± 5.72** | 61.49 ± 7.65 | ▲6.83 |
| | 16 | **64.48 ± 9.71** | 64.27 ± 11.32 | 59.49 ± 10.36 | 56.47 ± 4.51 | 57.51 ± 1.87 | 58.74 ± 8.04 | ▲4.02 |
| Myocardial | 4 | 54.28 ± 5.09 | **57.93 ± 2.64** | 50.0 ± 0.0 | 54.08 ± 3.28 | 56.35 ± 12.34 | 55.46 ± 4.77 | ▲4.20 |
| | 8 | 54.25 ± 8.33 | 52.78 ± 2.67 | **55.44 ± 5.34** | 51.6 ± 7.06 | 54.04 ± 6.16 | 52.26 ± 7.69 | ▲4.73 |
| | 16 | 56.39 ± 5.57 | 50.96 ± 5.98 | 55.21 ± 5.96 | 58.54 ± 1.84 | **61.96 ± 3.64** | 60.92 ± 2.09 | ▲5.84 |
| Cultivar | 4 | 41.93 ± 9.19 | 44.14 ± 4.23 | 50.0 ± 0.0 | **55.84 ± 4.99** | 55.14 ± 6.45 | 55.63 ± 8.79 | ▼0.38 |
| | 8 | 48.67 ± 7.27 | 49.2 ± 4.68 | 48.44 ± 1.56 | 56.95 ± 3.52 | **60.43 ± 6.79** | 57.45 ± 5.37 | ▲6.11 |
| | 16 | 53.86 ± 8.89 | 50.28 ± 5.77 | 57.08 ± 5.59 | 57.57 ± 2.67 | 57.49 ± 3.22 | **58.3 ± 2.46** | ▲1.27 |
| Bank | 4 | 67.65 ± 16.53 | 64.28 ± 5.0 | 50.0 ± 0.0 | 74.34 ± 1.71 | 75.17 ± 1.6 | **76.07 ± 2.87** | ▲2.33 |
| | 8 | 75.05 ± 1.57 | 63.36 ± 7.13 | 58.52 ± 10.73 | 76.09 ± 2.57 | 77.87 ± 0.38 | **78.03 ± 1.66** | ▲2.55 |
| | 16 | 77.6 ± 2.18 | 77.69 ± 2.51 | 68.75 ± 10.87 | 79.57 ± 1.01 | **79.59 ± 2.72** | 79.5 ± 2.96 | ▲0.03 |
| Heart | 4 | 52.19 ± 1.59 | **79.92 ± 7.71** | 50.0 ± 0.0 | 73.82 ± 6.06 | 77.69 ± 2.7 | 77.18 ± 3.53 | ▲5.24 |
| | 8 | 60.86 ± 8.74 | **81.84 ± 2.88** | 53.76 ± 11.81 | 70.88 ± 13.15 | 76.9 ± 7.8 | 70.99 ± 10.31 | ▲8.49 |
| | 16 | 65.45 ± 13.36 | **85.5 ± 2.39** | 82.99 ± 1.69 | 80.31 ± 7.69 | 83.57 ± 9.29 | 81.08 ± 5.33 | ▲4.06 |
| Diabetes | 4 | 47.04 ± 12.37 | 56.67 ± 11.65 | 50.0 ± 0.0 | 79.55 ± 0.35 | 79.65 ± 0.97 | **79.74 ± 0.5** | ▲0.24 |
| | 8 | 52.73 ± 5.8 | 64.19 ± 6.21 | 39.2 ± 14.42 | **80.48 ± 0.21** | 79.71 ± 0.24 | 80.41 ± 0.76 | ▼0.09 |
| | 16 | 64.78 ± 14.34 | 67.3 ± 6.02 | 72.69 ± 2.33 | 79.85 ± 0.83 | **80.94 ± 2.11** | 80.25 ± 1.52 | ▲1.37 |
| Blood | 4 | 42.75 ± 16.56 | 48.66 ± 12.56 | 50.0 ± 0.0 | **56.34 ± 6.66** | 54.57 ± 10.59 | 55.89 ± 6.51 | ▼0.80 |
| | 8 | 60.27 ± 8.9 | 57.67 ± 8.98 | 55.87 ± 5.1 | 66.63 ± 0.69 | 62.28 ± 7.24 | **66.71 ± 0.84** | ▲0.12 |
| | 16 | **68.59 ± 3.81** | 51.9 ± 8.84 | 63.43 ± 8.09 | 67.61 ± 1.9 | 67.98 ± 0.31 | 67.08 ± 1.71 | ▲0.55 |
| Adult | 4 | 58.3 ± 7.89 | 70.28 ± 5.32 | 50.0 ± 0.0 | **87.58 ± 0.29** | 86.48 ± 1.21 | 87.55 ± 0.83 | ▼0.03 |
| | 8 | 58.97 ± 8.93 | 57.27 ± 21.03 | 59.19 ± 7.96 | **87.29 ± 0.31** | 86.35 ± 0.3 | 86.95 ± 0.15 | ▼0.39 |
| | 16 | 67.61 ± 10.76 | 77.93 ± 2.79 | 68.17 ± 9.31 | 87.59 ± 0.9 | 85.53 ± 1.74 | **87.61 ± 0.97** | ▲0.02 |

RandomForest (Ho, 1995), and the state-of-the-art feature engineering method FeatLLM (Han et al., 2024) ensembling with ten feature sets. The primary aim of our study is to evaluate the robustness and reliability of LLM-generated features, particularly in the SOTA setting proposed by FeatLLM. Therefore, we employed our framework built upon FeatLLM, which has demonstrated superior performance over other baselines (e.g., TabLLM (Hegselmann et al., 2023)). In our framework, we simply averaged the Levels 1–3 evaluation scores of each feature set and selected the top 3 or excluded the bottom 3 feature sets out of the ten feature sets. As in the prior work, these feature sets were ensembled with equal weights for the final prediction.

### 4.2.2 Feature Evaluation Result

Table 1 summarizes the average AUC scores and standard deviations obtained from three runs—each with a different random seed for the training set selection—across all datasets. We observe that our framework ranks as the top performer on non-robust datasets when compared with FeatLLM. These results highlight the efficacy of our approach in scenarios where the model is particularly sensitive to input variations. In Appendix B.1, we further discuss the relationship between reliability diagnosis results and feature evaluation results.

## 5 Conclusion

We present a multi-level framework for evaluating the robustness of LLMs in tabular feature engineering. Our analysis reveals that the few-shot prediction performance of LLMs varies significantly across different datasets, highlighting the need for consistent and reliable methods in real-world applications. By focusing on golden variables, relations, and values, we demonstrate that high-quality features generated by LLMs can lead to substantial performance improvements. Our findings emphasize the importance of robustness in LLM-driven feature engineering and provide valuable insights for enhancing its reliability and effectiveness.


## Acknowledgments

This work was partly supported by Korea University - KT (Korea Telecom) R&D Center, the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ICT Creative Consilience Program (IITP-2025-RS-2020-II201819), IITP-ITRC (Information Technology Research Center) (IITP-2025-RS-2024-00436857), Artificial Intelligence Star Fellowship Program (IITP-2025-RS-2025-02304828), and the National Research Foundation of Korea (NRF) (RS-2024-00406320) funded by the Korea government (MSIT).

## Limitations

Despite the promising findings of this work, we acknowledge several limitations that can guide future research. First, the robustness of LLMs in feature engineering remains highly dependent on the characteristics of the underlying dataset. The variability observed across different domains suggests that LLMs may struggle with datasets that deviate significantly from their pre-trained knowledge.

Additionally, while our multi-level evaluation framework with various sampling and corruption strategies provides insights into model-dataset reliability, it does not fully mitigate the risks associated with variation in quality and the selection of few-shot samples for inference. LLMs are still susceptible to generating features with incorrect relationships or suboptimal decision boundaries, depending on the given samples, which can negatively impact prediction performance.

Second, our study primarily evaluates LLM-driven feature engineering using binary features adopted by the state-of-the-art LLM-driven feature engineering method. While this approach is simple and effective, real-world applications often require more complex and intricate feature representations, where LLMs may exhibit even greater instability. Future research should explore strategies to address these challenges, such as focusing on zero-shot feature engineering or incorporating generated features of varying forms, allowing the proposed multi-level scheme to be further generalized and expanded.

Lastly, although our multi-level scheme is specifically designed to reveal the general-purpose model's ability to handle variations in input conditions across diverse datasets, we intentionally kept the framework domain-agnostic to maximize its applicability. Nevertheless, we recognize that evaluating domain-specific LLMs could provide valuable insights into whether domain internalization enhances robustness in feature engineering. Such evaluations could further inform best practices and strengthen the reliability of LLM-driven methodologies across specialized application areas.

## References


A. Asuncion and D. J. Newman. 2007. UCI machine learning repository.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *arXiv*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

T. Chen and C. Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Kristy Choi, Chris Cundy, Sanjari Srivastava, and Stefano Ermon. 2022. LMPriors: Pre-trained language models as task-specific priors. *arXiv*.

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. *Advances in Neural Information Processing Systems*, 35:11763–11784.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large language models on tabular data–a survey. *arXiv*.

Fedesoriano. 2021. Heart failure prediction dataset. Accessed: Feb. 16, 2025.

Joshua P Gardner, Juan Carlos Perdomo, and Ludwig Schmidt. 2024. Large scale transfer learning for tabular data via language modeling. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

S. E. Golovenkin, Jonathan Bac, A. Chervov, E. M. Mirkes, Y. Orlova, E. Barillot, A. Gorban, and A. Zinovyev. 2020. Myocardial infarction complications. UCI Machine Learning Repository.

Sungwon Han, Jinsung Yoon, Sercan O Arik, and Tomas Pfister. 2024. Large language models can automatically engineer features for few-shot tabular learning. In *Forty-first International Conference on Machine Learning*.

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 *International Conference on Artificial Intelligence and Statistics*, pages 5549–5581. PMLR.

T. K. Ho. 1995. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE.

Noah Hollmann, Samuel Müller, and Frank Hutter. 2023. Large language models for automated data science: Introducing CAAFE for context-aware automated feature engineering. *arXiv*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv*.

Daniel P. Jeong, Zachary C. Lipton, and Pradeep Ravikumar. 2024. LLM-Select: Feature selection with large language models. *arXiv*.

A. Kadra, M. Lindauer, F. Hutter, and J. Grabocka. 2021. Well-tuned simple nets excel on tabular datasets. In *Advances in Neural Information Processing Systems*, volume 34, pages 23928–23941.

Krishnaram Kenthapadi, Mehrnoosh Sameki, and Ankur Taly. 2024. Grounding and evaluation for large language models: Practical challenges and lessons learned (survey). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6523–6533.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Cosmin Lazar, Jonatan Taminau, Stijn Meganck, David Steenhoff, Alain Coletta, Colin Molter, Virginie de Schaetzen, Robin Duque, Hugues Bersini, and Ann Nowe. 2012. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 9(4):1106–1119.

Dawei Li, Zhen Tan, and Huan Liu. 2024. Exploring large language models for feature selection: A data-centric perspective. *arXiv*.

S. Moro, P. Cortez, and P. Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31.

Michael Redmond. 2009. Communities and crime. UCI machine learning repository.

Bruno Rodrigues de Oliveira and Alan Mario Zuffo. 2023. Forty Soybean Cultivars from Subsequent Harvests. UCI Machine Learning Repository.

Yucheng Ruan, Xiang Lan, Jingying Ma, Yizhi Dong, Kai He, and Mengling Feng. 2024. Language modeling on tabular data: A survey of foundations, techniques and evolution. *arXiv*.

Jack W Smith, James E Everhart, WC Dickson, William C Knowler, and Robert Scott Johannes. 1988. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the annual symposium on computer application in medical care*, page 261. American Medical Informatics Association.

Ruiyu Wang, Zifeng Wang, and Jimeng Sun. 2023. Unipredict: Large language models are universal tabular classifiers. *arXiv*.

Zifeng Wang, Chufan Gao, Cao Xiao, and Jimeng Sun. 2024. Meditab: scaling medical tabular data predictors via data consolidation, enrichment, and refinement. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 6062–6070.

Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. 2024. From supervised to generative: A novel paradigm for tabular deep learning with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3323–3333.

I.-C. Yeh, K.-J. Yang, and T.-M. Ting. 2009. Knowledge discovery on rfm model using bernoulli sequence. *Expert Systems with Applications*, 36(3):5866–5871.

Han Zhang, Xumeng Wen, Shun Zheng, Wei Xu, and Jiang Bian. 2023. Towards foundation models for learning on tabular data.

Yanlin Zhang, Ning Li, Quan Gan, Weinan Zhang, David Wipf, and Minjie Wang. 2024. ELF-GYM: Evaluating large language models generated features for tabular prediction. *arXiv*.

# A Implementation Details

## A.1 Datasets

Table 2 shows the basic information of each dataset used in our experiments. The numbers in parentheses under # of features represent the number of categorical and numerical features, respectively. Similarly, the numbers in parentheses under # of golden features represent the number of categorical and numerical golden features.

Table 2: Dataset statistics.

| Data | # of samples | # of features | Label ratio (%) | # of golden features |
|---|---|---|---|---|
| Adult | 48842 | 14 (7/7) | 76:24 | 3 (2/1) |
| Bank | 45211 | 16 (8/8) | 88:12 | 2 (1/1) |
| Blood | 748 | 4 (0/4) | 76:24 | 3 (0/3) |
| Communities | 1994 | 103 (1/102) | 34:33:33 | 19 (19/0) |
| Credit-g | 1000 | 20 (12/8) | 70:30 | 2 (1/1) |
| Cultivar | 320 | 10 (3/7) | 50:50 | 2 (1/1) |
| Diabetes | 768 | 8 (0/8) | 65:35 | 4 (0/4) |
| Heart | 918 | 11 (4/7) | 45:55 | 5 (3/2) |
| Myocardial | 1700 | 111 (94/17) | 22:78 | 7 (1/6) |

## A.2 LLMs and Baselines

For various LLM backbones, the temperature for LLM inference is set to nonzero (i.e., 0.5). For experiments involving open-source models, we use vLLM 0.9.2 (Kwon et al., 2023) with two A6000 GPUs. We vary the data availability to conduct evaluations with 4-shot, 8-shot, and 16-shot configurations. The test performance is measured using a logistic/linear regression model, selected via grid search with 5-fold cross-validation. To evaluate classification tasks, we use the area under the ROC curve (AUROC) as the primary metric.

The number of conditions included in the feature rule is determined as:

$$\max(\text{golden variables}, \text{variables} \times 0.5).$$

This ensures a balance between model interpretability and robustness.

## A.3 Sampling

Examples with varying levels of quality are used to evaluate the robustness of LLM responses. The examples provided to the LLM can act as either informative signals or noise. To evaluate how robust the LLM's prior knowledge is, we modify the quality of the provided examples and conduct experiments. Sampling is divided into best-case and worst-case scenarios based on the distance between each sample's feature and the golden value $V_{\text{golden}}$.

$$N(V_f) = \frac{V_f - f^{\min}}{f^{\max} - f^{\min}}.$$

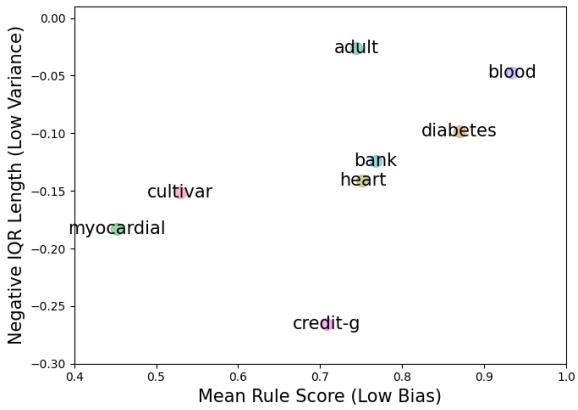


Figure 9: Variance and bias of average reliability score of GPT-3.5-Turbo.

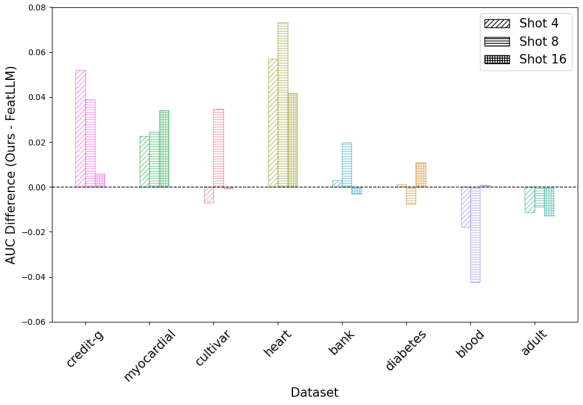


Figure 10: Performance improvement of ours over FeatLLM with different numbers of examples.

When $R_{\text{golden}}$ is positive and $N(V_f) > N(V_{\text{golden}})$, the distance is defined as:

$$\text{distance} = |N(V_f) - N(V_{\text{golden}})|.$$

When $N(V_f) \leq N(V_{\text{golden}})$, the distance incorporates a penalty:

$$\text{distance} = |N(V_f) - N(V_{\text{golden}})| + \text{penalty}_{\text{pos}}.$$

The penalty is defined as:

$$\text{penalty}_{\text{pos}} = |N(f^{\max}) - N(V_{\text{golden}})|.$$

# B Additional Results

## B.1 Feature Evaluation Results

Figures 9 and 10 indicate the correlation between robustness and overall performance. For datasets that exhibit large performance fluctuations depending on the input, adopting well-designed ensemble rules can lead to notable improvements. This underscores the importance of diagnosing which inputs serve as genuine "information" as opposed to mere "data." Our findings demonstrate that domain

knowledge serves as a valuable guide for pinpointing critical variables and mitigating irrelevant complexity. In practice, diagnosing before fully trusting a model—by uncovering its weaknesses and evaluating its robustness—offers a principled approach to optimizing performance. By strategically combining diagnostic insights with expert knowledge, practitioners can effectively enhance LLM reliability and achieve more consistent results across a range of datasets. In Figure 11, we further demonstrate the relationships between the multi-level evaluation scores and AUC scores across datasets.

### B.2 Results on the Communities Dataset

For a multi-class classification setting, we report additional results of GPT-3.5-Turbo on the Communities dataset (Table 3), reliability diagnosis results (Figures 12 and 13), and performance improvement results with varying shots (Figure 14).

### B.3 Box Plots of Reliability Diagnosis Score

We include additional reliability diagnosis results of GPT-3.5-Turbo (Figures 15–22) across datasets and levels.

## C Prompt Examples

In Figures 23–25, we include example prompts designed for our multi-level reliability diagnosis in Heart dataset. In Figure 26, we also include an example prompt for rule evaluation proposed by FeatLLM (Han et al., 2024).

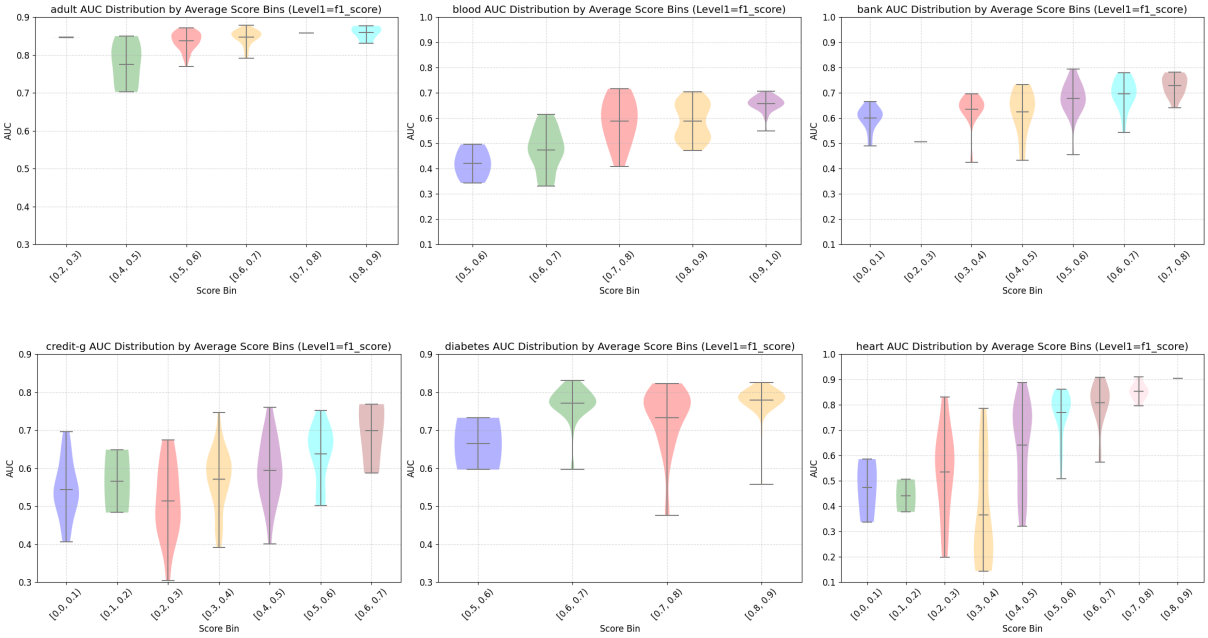


Figure 11: Relationship between feature evaluation score and AUC score.


Figure 12: Effects of varying the number of examples on reliability scores at each level (with Communities).

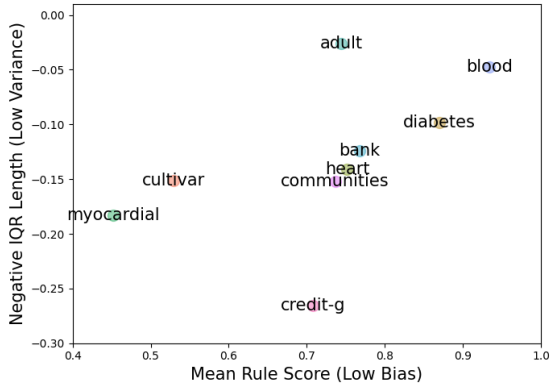


Figure 13: Variance and bias of average reliability score (with Communities).

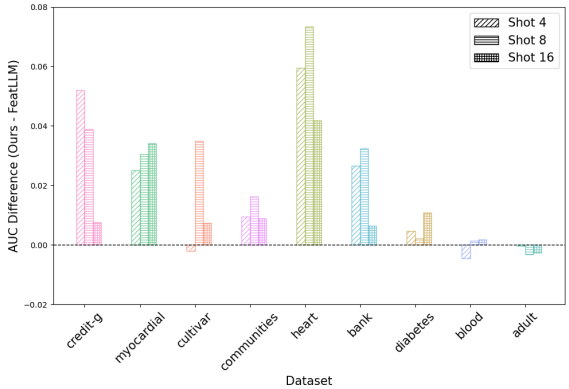


Figure 14: Performance improvement of ours over FeatLLM with different shots (with Communities).

Table 3: Few-shot classification performance evaluation results on the Communities dataset.

| Data | Shot | LogReg | RandomForest | XGBoost | FeatLLM | Ours (Top 3) | Ours (Top 5) | Improvement (%) |
|---|---|---|---|---|---|---|---|---|
| Communities | 4 | $47.95 \pm 2.07$ | $54.94 \pm 7.06$ | $50.0 \pm 0.0$ | $\underline{73.82 \pm 2.93}$ | $72.26 \pm 8.69$ | $\mathbf{74.77 \pm 5.88}$ | ▼0.99 |
| | 8 | $55.81 \pm 10.46$ | $64.09 \pm 4.39$ | $68.82 \pm 3.26$ | $\underline{74.88 \pm 3.73}$ | $74.64 \pm 5.46$ | $\mathbf{76.5 \pm 3.25}$ | ▲1.62 |
| | 16 | $59.23 \pm 13.75$ | $68.82 \pm 0.69$ | $66.03 \pm 0.96$ | $73.88 \pm 2.87$ | $\underline{73.91 \pm 3.79}$ | $\mathbf{74.77 \pm 2.64}$ | ▲0.89 |

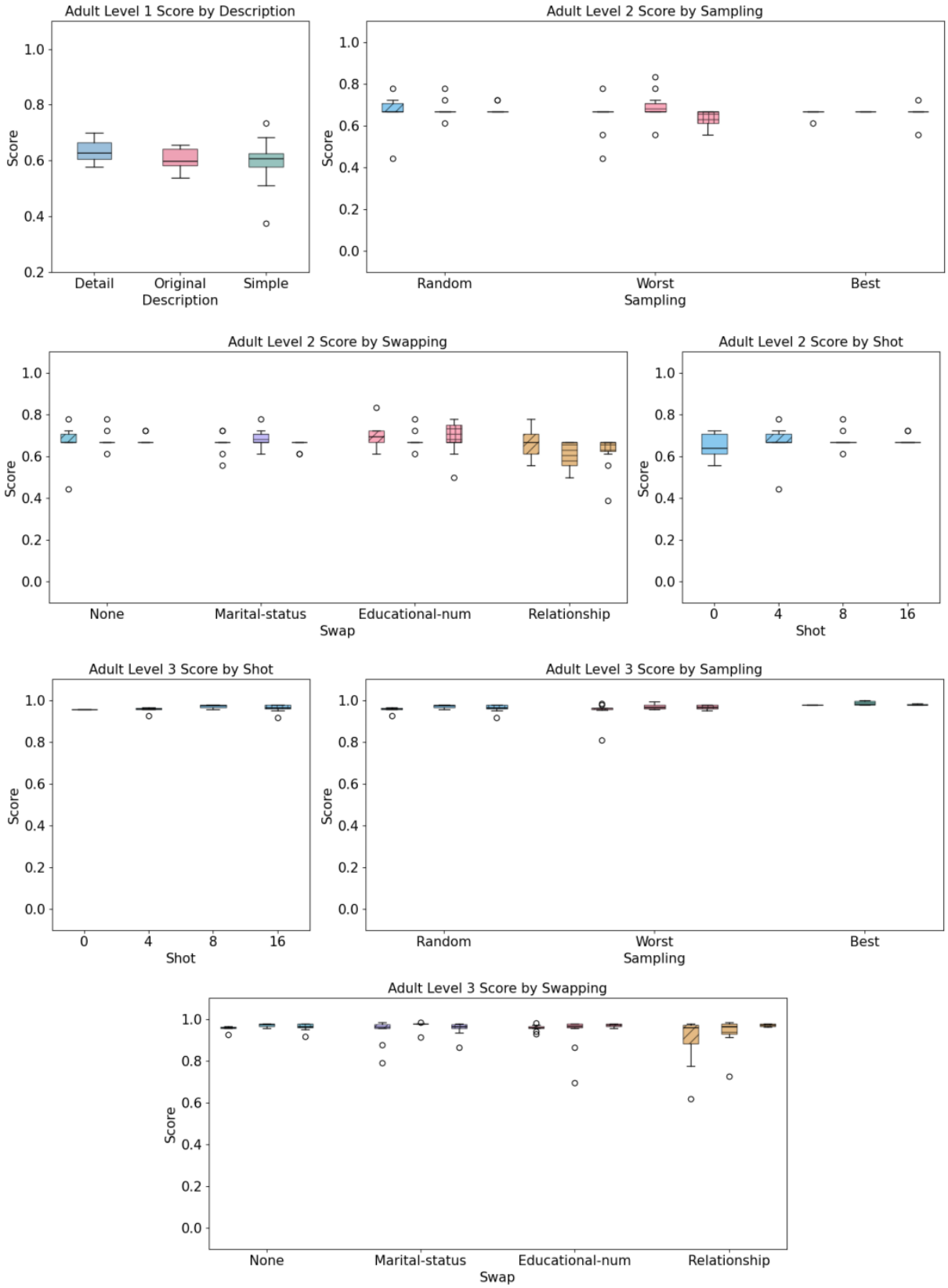


Figure 15: Full results of reliability diagnosis on Adult.

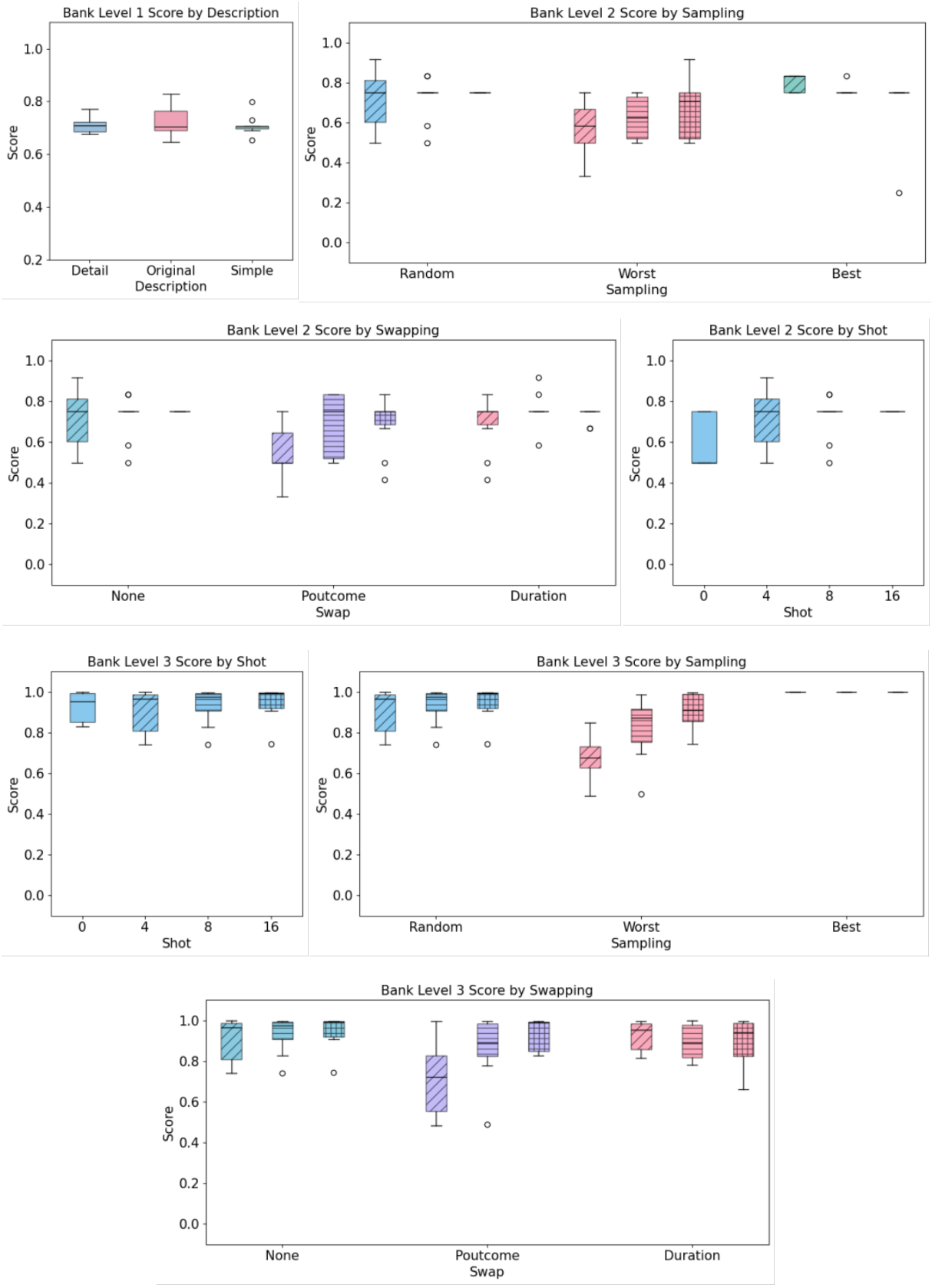


Figure 16: Full results of reliability diagnosis on Bank.

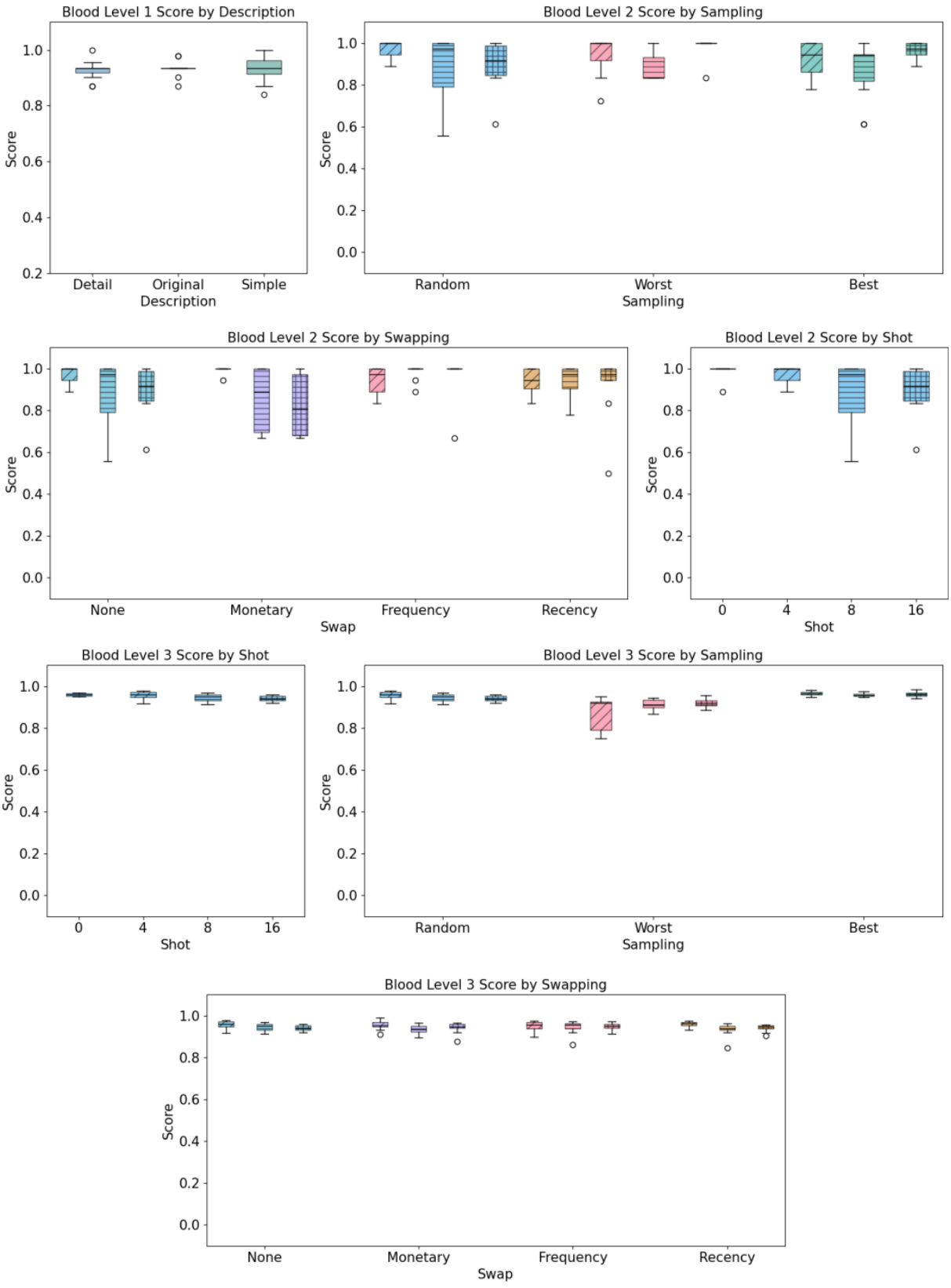


Figure 17: Full results of reliability diagnosis on Blood.

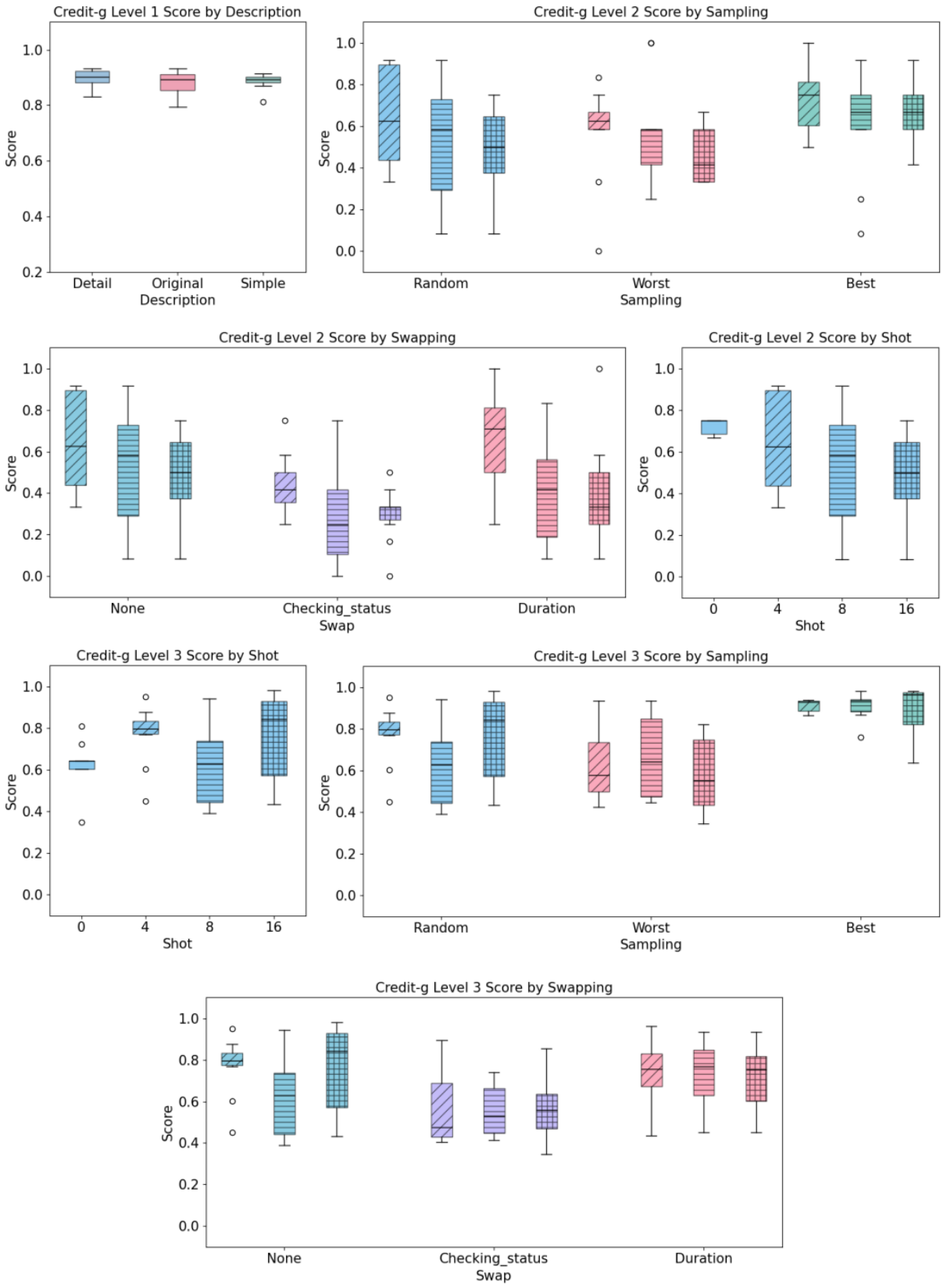


Figure 18: Full results of reliability diagnosis on Credit-g.

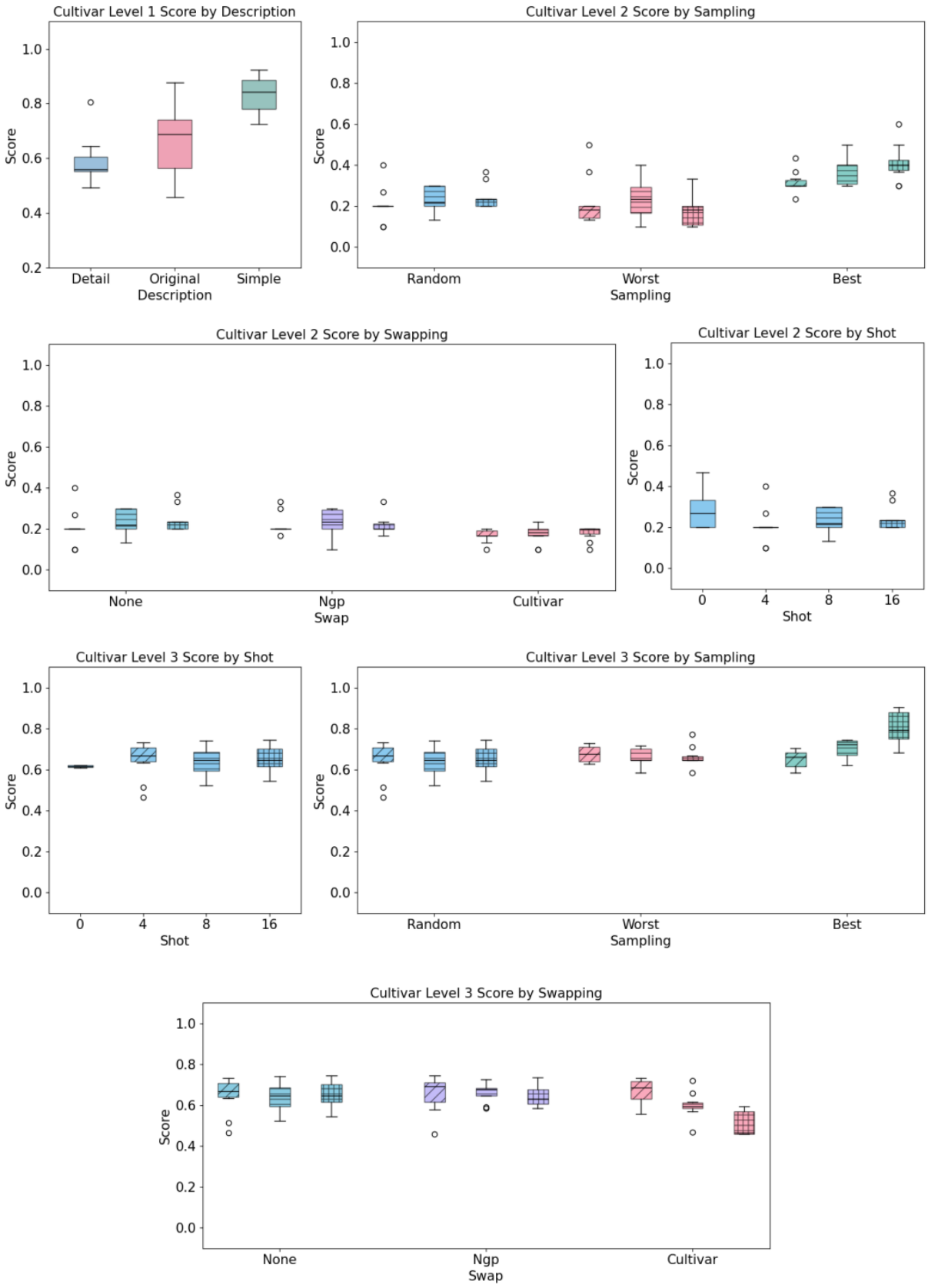


Figure 19: Full results of reliability diagnosis on Cultivar.

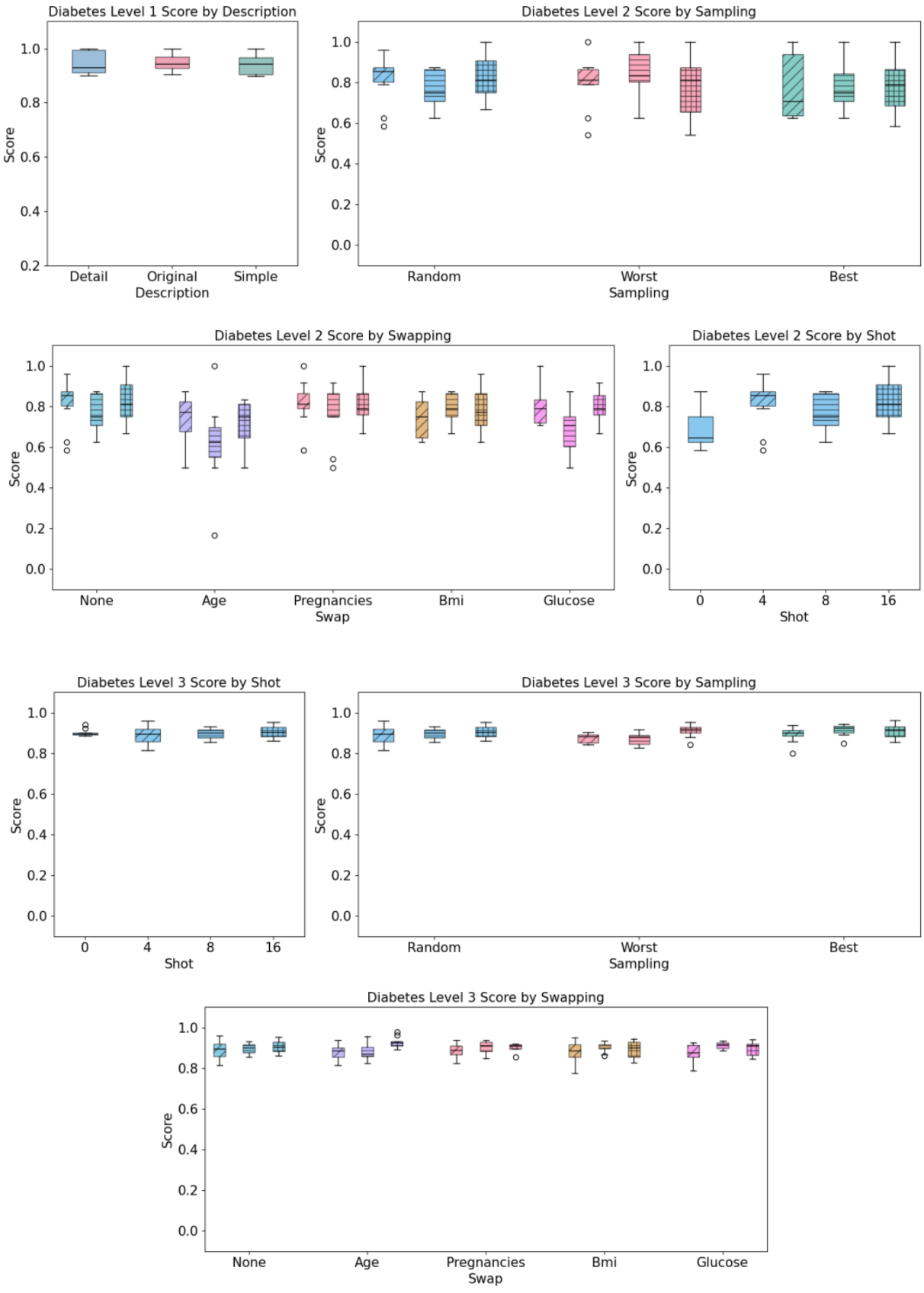


Figure 20: Full results of reliability diagnosis on Diabetes.

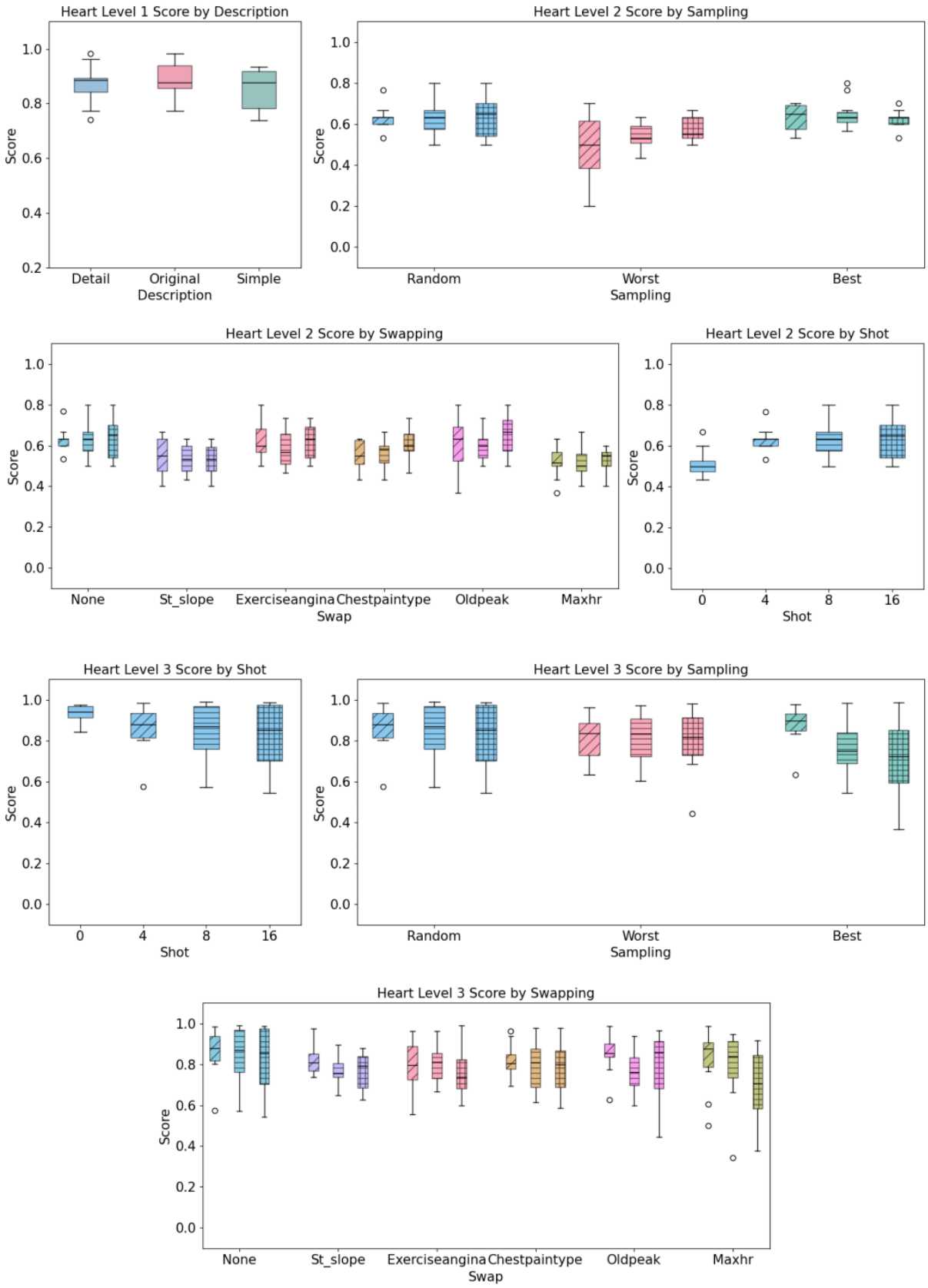


Figure 21: Full results of reliability diagnosis on Heart.

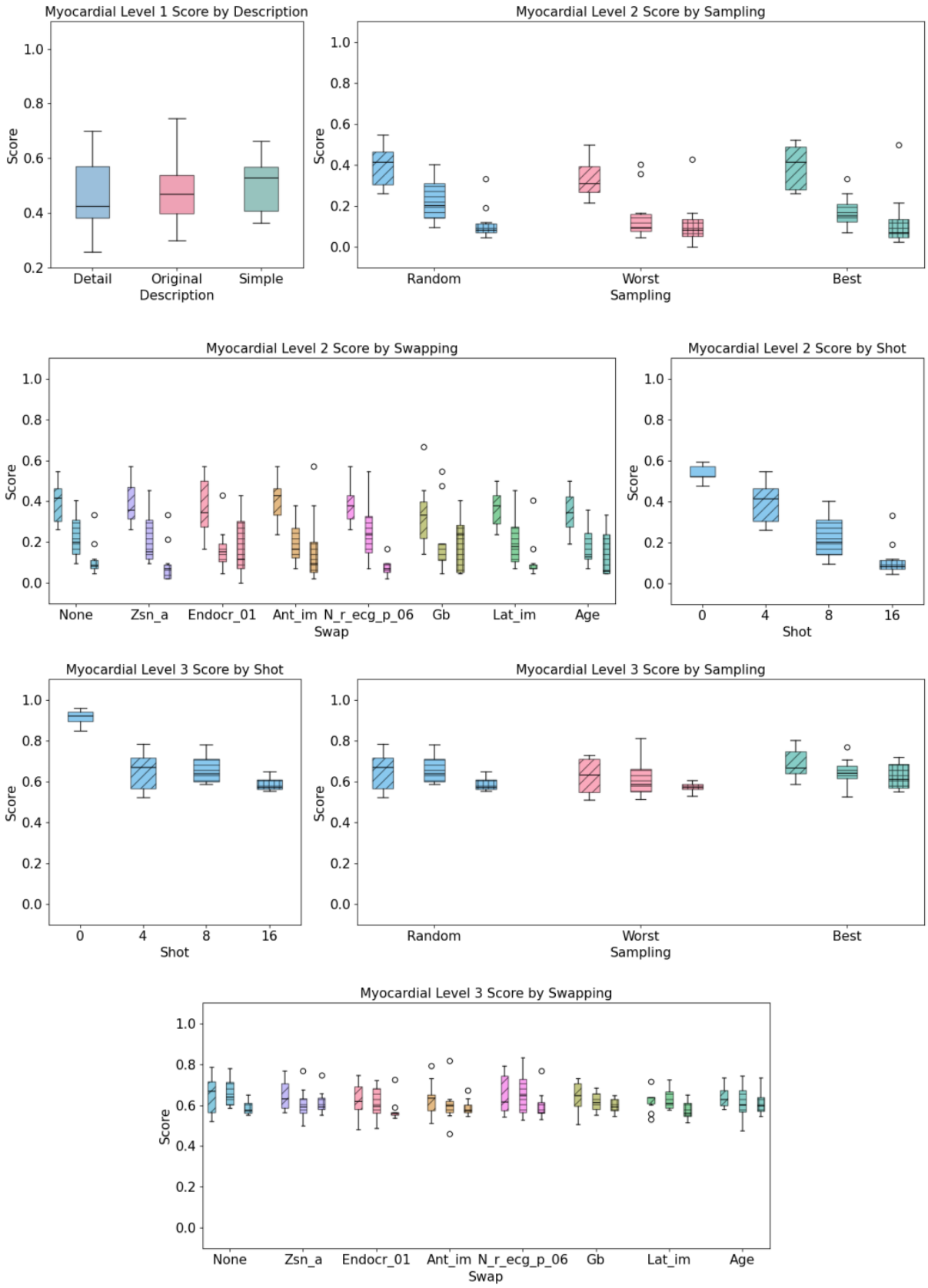


Figure 22: Full results of reliability diagnosis on Myocardial.

You are an expert. Given the task description and the list of features, you are ranking them according to their importances to solve the task. The ranking should be in descending order, starting with the most important feature.

Task: Does the coronary angiography of this patient show a heart disease? Yes or no?

Features:

- Age: age of the patient (numerical variable)
- Sex: sex of the patient (categorical variable with categories [M, F])
- ChestPainType: chest pain type (categorical variable with categories [ATA, NAP, ASY, TA])
- RestingBP: resting blood pressure [mm Hg] (numerical variable)
- Cholesterol: serum cholesterol [mm/dl] (numerical variable)
- FastingBS: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise] (numerical variable)
- RestingECG: resting electrocardiogram results (categorical variable with categories [Normal, ST, LVH])
- MaxHR: maximum heart rate achieved (numerical variable)
- ExerciseAngina: exercise-induced angina (categorical variable with categories [N, Y])
- Oldpeak: oldpeak = ST [Numeric value measured in depression] (numerical variable)
- ST_Slope: the slope of the peak exercise ST segment (categorical variable with categories [Up, Flat, Down])

Your response should be a numbered list with each item on a new line.

Format for Response:

```
Rank:
FeatA
FeatB....
```

Answer:

Figure 23: Example prompt for reliability diagnosis level 1 for Heart dataset.

You are an expert. Given the task description and the list of features and data examples, analyze the causal relationship or tendency between each feature and class based on general knowledge and common sense within a short sentence.

Task: Does the coronary angiography of this patient show a heart disease? Yes or no?

Features:

- ChestPainType: chest pain type (categorical variable with categories [ATA, NAP, ASY, TA])
- MaxHR: maximum heart rate achieved (numerical variable)
- Oldpeak: oldpeak = ST [Numeric value measured in depression] (numerical variable)
- ST_Slope: the slope of the peak exercise ST segment (categorical variable with categories [Up, Flat, Down])
- ExerciseAngina: exercise-induced angina (categorical variable with categories [N, Y])

Examples:

```
Age is 63. Sex is M. ChestPainType is NAP. RestingBP is 130. Cholesterol is 0.
FastingBS is 1. RestingECG is ST. MaxHR is 160. ExerciseAngina is Y.
Oldpeak is 3.0. ST_Slope is Flat.
Answer: no
Age is 39. Sex is M. ChestPainType is ATA. RestingBP is 120. Cholesterol is 204.
FastingBS is 0. RestingECG is Normal. MaxHR is 145. ExerciseAngina is N.
Oldpeak is 0.0. ST_Slope is Up.
Answer: no
Age is 55. Sex is M. ChestPainType is ASY. RestingBP is 160. Cholesterol is 289.
FastingBS is 0. RestingECG is LVH. MaxHR is 145. ExerciseAngina is N.
Oldpeak is 0.8. ST_Slope is Flat.
Answer: yes
Age is 58. Sex is M. ChestPainType is NAP. RestingBP is 160. Cholesterol is 211.
FastingBS is 1. RestingECG is ST. MaxHR is 92. ExerciseAngina is N.
Oldpeak is 0.0. ST_Slope is Flat.
Answer: yes
```

Format for Response:

```
Causal Relationship for class "no":
ChestPainType in ATA has a [positive/negative] correlation with class no
MaxHR has a [positive/negative] correlation with class no
Oldpeak has a [positive/negative] correlation with class no
ST_Slope in Up has a [positive/negative] correlation with class no
ExerciseAngina in N has a [positive/negative] correlation with class no

Causal Relationship for class "yes":
ChestPainType in ASY has a [positive/negative] correlation with class yes
MaxHR has a [positive/negative] correlation with class yes
Oldpeak has a [positive/negative] correlation with class yes
ST_Slope in Flat has a [positive/negative] correlation with class yes
ExerciseAngina in Y has a [positive/negative] correlation with class yes
```

Answer:

Figure 24: Example prompt for reliability diagnosis level 2 for Heart dataset.

You are an expert. Given the task description and the list of features and data examples, you are filling in the feature conditions for each class to solve the task.

Task: Does the coronary angiography of this patient show a heart disease? Yes or no?

Features:

- ChestPainType: chest pain type (categorical variable with categories [ATA, NAP, ASY, TA])
- MaxHR: maximum heart rate achieved (numerical variable)
- Oldpeak: oldpeak = ST [Numeric value measured in depression] (numerical variable)
- ST_Slope: the slope of the peak exercise ST segment (categorical variable with categories [Up, Flat, Down])
- ExerciseAngina: exercise-induced angina (categorical variable with categories [N, Y])

Examples:

```
Age is 39. Sex is M. ChestPainType is ATA. RestingBP is 120. Cholesterol is 204.
FastingBS is 0. RestingECG is Normal. MaxHR is 145. ExerciseAngina is N.
Oldpeak is 0.0. ST_Slope is Up. Answer: no
Age is 63. Sex is M. ChestPainType is NAP. RestingBP is 130. Cholesterol is 0.
FastingBS is 1. RestingECG is ST. MaxHR is 160. ExerciseAngina is Y.
Oldpeak is 3.0. ST_Slope is Flat. Answer: no
Age is 55. Sex is M. ChestPainType is ASY. RestingBP is 160. Cholesterol is 289.
FastingBS is 0. RestingECG is LVH. MaxHR is 145. ExerciseAngina is N.
Oldpeak is 0.8. ST_Slope is Flat. Answer: yes
Age is 58. Sex is M. ChestPainType is NAP. RestingBP is 160. Cholesterol is 211.
FastingBS is 1. RestingECG is ST. MaxHR is 92. ExerciseAngina is N.
Oldpeak is 0.0. ST_Slope is Flat. Answer: yes
```

Format for Response:

```
Condition for class "no":
ChestPainType is in [Value]
MaxHR is greater than [Value]
Oldpeak is less than [Value]
ST_Slope is in [Value]
ExerciseAngina is in [Value]

Condition for class "yes":
ChestPainType is in [Value]
MaxHR is less than [Value]
Oldpeak is greater than [Value]
ST_Slope is in [Value]
ExerciseAngina is in [Value]
```

Format for `[Value]`:

- For the categorical variable only: `[List of Categories]`
- For the numerical variable only: `[Value]`

Answer:

Figure 25: Example prompt for reliability diagnosis level 3 for Heart dataset.

You are an expert. Given the task description and the list of features and data examples, you are extracting conditions for each answer class to solve the task.

Task: `[TASK]`

Features: `[FEATURES]`

Examples: `[EXAMPLES]`

Let's first understand the problem and solve the problem step by step.
Step 1. Analyze the causal relationship or tendency between each feature and task description based on general knowledge and common sense within a short sentence.
Step 2. Based on the above examples and Step 1 results, infer 10 different conditions per answer, following the format below. The condition should make sense, well match examples, and must match the format for `[Condition]` according to value type.

Format for Response:
10 different conditions for class "no":
- `[Condition]`
...
10 different conditions for class "yes":
- `[Condition]`
...

Format for `[Condition]`:
- For the categorical variable only:
  `[Feature] is in [list of Categories]`
- For the numerical variable only:
  `[Feature] (> or >= or < or <=) [Value]`
  `[Feature] is within range of [Value start, Value end]`

Answer: Step 1.

Figure 26: Prompt for default feature engineering in FeatLLM.