Teaching LLMs to Reason over Tabular Data with Tri-Level Rationale Distillation

Anonymous ACL submission

Abstract

Tabular data is a primary medium for storing real-world information, driving many industrial applications of machine learning. Traditional predictors achieve strong predictive performance but do not provide readable, case-specific explanations essential for decision-making. Large Language Models (LLMs) can naturally bridge this gap by generating predictions alongside explanations. However, dataset-specific patterns, such as feature distributions and interactions, make tabular data difficult for LLMs to understand and reason over, while label-only fine-tuning improves performance at the cost of catastrophic forgetting. To address this problem, we propose Tri-Level Rationale Distillation (TLRD), a framework that converts label-only tabular datasets into structured rationale supervision for LLMs. TLRD uses a high-capacity teacher to synthesize a rationale corpus grounded in three complementary levels of evidence: instance-level feature, dataset-level distributional context, and comparison-level retrieved neighbors, then distills the rationale into student LLMs, enabling zero-overhead prediction and grounded explanation from raw features only. Experiments on multiple domain datasets show that TLRD significantly closes the performance gap between LLMs and state-of-the-art tree ensembles while producing grounded and readable explanations, offering a valuable reference for high-stakes decision-making. Our code is available at https://anonymous.4open.science/r/TLRD-7E9E.

1 Introduction

Many industrial decision-making tasks rely on tabular data, including credit scoring, healthcare, supply-chain optimization, and user profiling. In these decision-critical settings, tree-based ensembles such as XGBoost (Chen and Guestrin, 2016) and CatBoost (Prokhorenkova et al., 2018) remain the dominant methods, often outperforming deep learning methods under realistic tuning and data-size regimes (Shwartz-Ziv and Armon, 2022; Grinsztajn et al., 2022). Even as foundation models emerge (Hollmann et al., 2023), practitioners still favor tree-based ensembles for their efficiency and strong performance. However, model predictions cannot directly translate into practical decisions, since real-world decisions are made by stakeholders under task-specific scenarios, with models serving as assistive tools. In practice, stakeholders need concise and readable rationales (*Figure 1 (B)*) that explain (i) which specific feature values in this case are concerning, and (ii) how unusual they are relative to the broader population. However, tree-based ensembles provide only coarse interpretability signals (e.g., *feature attribution scores in Figure 1 (A)*) that do not align with operational decision needs. Recent LLM-based methods even claim to outperform strong baselines such as XGBoost (Gardner et al., 2024; Wang et al., 2026), but they are established under few-shot settings, which are less prac-

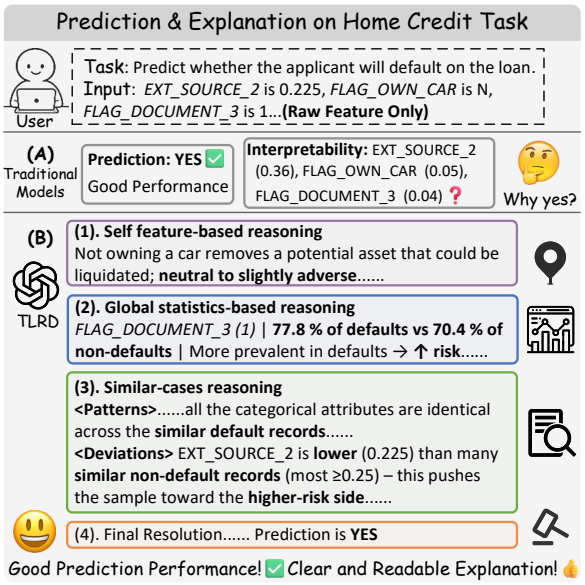

Figure 1: Illustration of a TLRD deployment. Given raw features, the student model generates tri-level rationales.

tical, while optimizing explanation quality remains largely underexplored. Accordingly, our work aims to **provide complementary decision support for tabular data, offering a practical trade-off between performance and interpretability.**

LLMs naturally bridge the gap between predictive capability and human-readable reasoning. However, in zero-shot or few-shot prediction tasks, LLMs often exhibit weak performance on certain tasks (Xie et al., 2024; Hegselmann et al., 2023; Wolff and Hulsebos, 2025), especially those requiring precise numerical comparisons or domain-specific feature interactions not encountered during pre-training. A natural response is to fine-tune an LLM directly on labeled tabular examples. However, this introduces a new technical hurdle: the catastrophic forgetting (Luo et al., 2025) of generative behavior. When supervised solely by categorical labels, the model optimizes strictly for the discriminative mapping, implicitly achieving competitive accuracy while minimizing token length. Consequently, the model becomes biased toward terse label-only outputs and no longer reliably generates free-form explanations. This behavior is also denoted as the explanation collapse (Yang et al., 2025b). This motivates the central question of our work: *How can we fine-tune a deployable (small) LLM on a label-only tabular dataset to achieve high predictive performance while preserving high-quality, evidence-anchored explanations?*

To address this, we propose **Tri-Level Rationale Distillation (TLRD)**, a general pipeline that converts label-only tabular datasets into a supervision-rich, structured reasoning corpus. By distilling from a high-capacity teacher model, TLRD synthesizes a structured tri-level rationale report that closely mirrors decision-making workflows:

- **Instance-level reasoning**: Identifies and isolates the primary feature triggers and local patterns specific to the individual data row, guided by *domain knowledge and commonsense constraints*.
- **Dataset-level reasoning**: Contextualizes specific values against the global feature distribution, anchoring the explanation in *macro-level statistics*.
- **Comparison-level reasoning**: Leverages a few similar cases to distill contrastive insights into *pattern summaries*.

During rationale synthesis, TLRD provides the teacher model with a **Tri-Level Augmented Input** as complementary evidence by extending current row with dataset-level statistics and comparison-level retrieved neighbors, helping the teacher understand feature values not only from the current row, but also through global distributions and local contrastive patterns. Guided by **Tri-Level Rationale Schema**, the teacher converts this evidence into structured rationales. A compact student model is then fine-tuned on the distilled corpus to predict the label and generate the rationale given only raw features. In this way, dataset context and case-based signals are *indirectly injected* into the student, while the student remains deployable without external statistics or retrieval, avoiding added latency. Our contributions are summarized as follows:

- **Tri-Level Evidence to Rationales.** We introduce dataset statistics and retrieved neighbors as extra evidence, and verify that they improve base LLMs' prediction. We then formalize a structured schema that guides a teacher to analyze these evidence sources under the ground-truth label and transform them into tri-level rationales.
- **Context-to-Parameter Tabular Rationale Distillation:** We introduce a distillation method that uses dataset statistics and retrieved neighbors only during corpus construction, indirectly injecting dataset-level and case-based knowledge into the student and enabling zero-overhead inference from raw features.
- **Comprehensive Empirical Evaluation:** We evaluate TLRD on multiple decision-critical tabular tasks, showing that TLRD achieves strong predictive performance against strong baselines while producing grounded and readable rationales. We conduct a study of diverse teacher models, student backbones, and self-distillation settings, hoping to bring useful insights.

2 Related Work

2.1 LLMs on Tabular Data

The adaptation of LLMs for tabular data traditionally relies on serializing structured rows into natural text to bridge the modality gap. Foundational works like TabLLM (Hegselmann et al., 2023) and UniPredict (Wang et al., 2023) demonstrated that serialization enables zero-shot and few-shot classification across diverse tabular inputs. However, these methods do not solve LLMs' difficulty in capturing dataset-specific patterns. There are also some methods working on scaling transfer learning on diverse datasets (Gardner et al., 2024) and optimizing In-Context Learning (ICL) via advanced prompt engineering (Gao et al., 2025). To address numeric instability and context limits, newer archi-

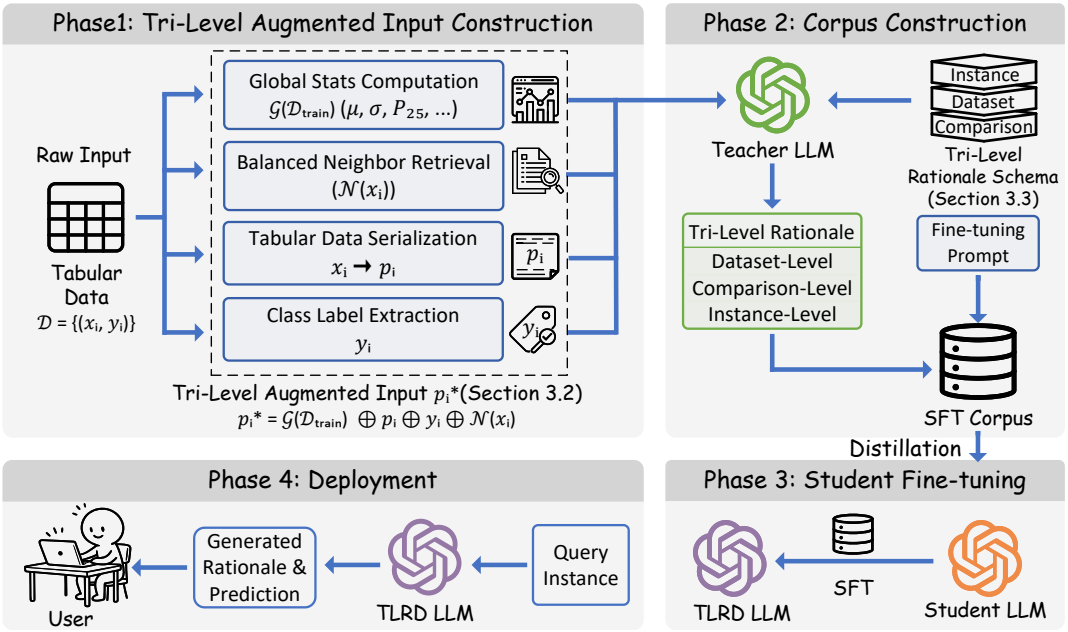

Figure 2: Overview of the TLRD framework. Starting from a label-only tabular dataset, we construct Tri-Level Augmented Inputs for the teacher model. Guided by the Tri-Level Rationale Schema, the teacher generates structured rationales, which are then distilled into a student LLM through supervised fine-tuning. During deployment, the student takes only the raw serialized row as input and outputs both the prediction and the tri-level rationale.

tectures introduce specialized mechanisms, such as TabGemma's (Schindler et al., 2025) numeric canonicalization with retrieval-augmented generation, and highly efficient ICL strategies (Wu and Hou, 2025). Despite these advances, LLMs still struggle to capture dataset-specific patterns, motivating the development of compact, task-specific models that provide grounded explanations.

2.2 LLM Distillation and Fine-Tuning

To deploy LLMs efficiently, researchers frequently use knowledge distillation. While traditional distillation transfers only predictive distributions, recent NLP breakthroughs emphasize transferring reasoning abilities from large to small language models and rationale distillation (Hsieh et al., 2023; Magister et al., 2023). However, rationale distillation remains largely underexplored for tabular data, where reasoning often depends on dataset-specific knowledge that cannot be directly acquired from pre-training. Fine-tuning tabular models strictly on labels risks "explanation collapse (Yang et al., 2025b)," where models optimize for accuracy at the expense of generative reasoning faculties. Recent works focus on the performance of few-shot settings (Gardner et al., 2024; Wang et al., 2026), which are less practical in industrial scenarios where large labeled datasets are usually available. Moreover, their designs do not focus on explanation generation. TLRD bridges these gaps by distilling teacher-generated tri-level rationales based on tri-level evidence and designed schema, enabling the student model to retain grounded, readable reasoning while maintaining strong performance for real-world tabular decision-making settings.

3 Proposed Method

3.1 Problem Formulation

Let $\mathcal{X}$ denote a tabular feature space with both numerical and categorical variables, such that $\mathcal{X} = \mathbb{R}^{d_{\text{num}}} \times \mathcal{V}^{d_{\text{cat}}}$, where d_{num} and d_{cat} are the numbers of numerical and categorical features, respectively, and $\mathcal{V}$ denotes the categorical value space. For a labeled tabular dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathcal{X}$ is the i-th instance and $y_i \in \mathcal{Y}$ (e.g., $\mathcal{Y} = \{0, 1\}$) is its ground-truth label. Let $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ denote the training split. Each table row is then converted into a textual prompt $p_i = \text{Serialize}(\mathbf{x}_i)$.

The overall workflow is illustrated in Figure 2. Our goal is to fine-tune an LLM to output both a prediction and a structured, evidence-grounded explanation rationale, i.e., (y_i, r_i) with $r_i \in \mathcal{R}$ following a fixed schema. We seek a mapping that converts label-only dataset into a supervised corpus $\mathcal{C} = \{(p_i, o_i) \mid (\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}\}$, where $o_i = (r_i, y_i)$, by synthesizing schema-compliant rationales from a stronger teacher or via self-distillation.

3.2 Tri-Level Augmented Input

To address the limitation that LLMs struggle to understand dataset-specific patterns, we construct a *Tri-Level Augmented Input* that extends the serialized row with global statistics and retrieved neighbors, exposing the teacher to the distributional and contrastive evidence needed for grounded tri-level rationale generation. We first serialize each row $\mathbf{x} \in \mathcal{X}$ in the form of `<feature> is <value>`, using a fixed order, explicit missing-value placeholders, and append the ground-truth label. We refer to this block as *Current Sample Features* and then augment it with two additional sources of evidence: *Dataset Statistics* and *Retrieved Neighbors*.

Dataset statistics. To provide global distributional context, we compute and append dataset-level statistics to the teacher prompt. For classification tasks, we compute, for each target class on $\mathcal{D}_{\text{train}}$, numerical-feature statistics including the mean, standard deviation, quartiles (P25, P50, P75), minimum, and maximum, as well as class-conditional category frequencies for categorical features. For regression tasks, we discretize the continuous target into four percentile bins using the 25th, 50th, and 75th percentiles of the target distribution on $\mathcal{D}_{\text{train}}$ as thresholds, and compute the corresponding numerical and categorical feature statistics within each bin. We denote the resulting dataset-level statistical context by $\mathcal{G}(\mathcal{D}_{\text{train}})$.

Retrieved neighbors. To provide a contrastive case-based context, we retrieve a label-balanced set of historical instances; the query instance itself is excluded from retrieval. Let $\mathbf{x}_q$ be the query instance, and let $\mathcal{F}_{\text{cat}}$ denote the index set of categorical features. To maintain semantic consistency, we first define a candidate pool $\Omega(\mathbf{x}_q)$ by enforcing categorical feature matches:

$$\Omega(\mathbf{x}_q) = \{\mathbf{x}_i \in \mathcal{D}_{\text{train}} \mid \mathbf{x}_i[f] = \mathbf{x}_q[f], \forall f \in \mathcal{F}_{\text{cat}}\}.$$

If this exact-match pool is too small, we progressively relax the categorical matching constraints until a sufficiently large candidate pool is obtained.

Within this structurally aligned pool, we compute a similarity score using standardized numerical features. Let $\mathbf{v}_i$ and $\mathbf{v}_q$ denote the standardized numerical feature vectors of $\mathbf{x}_i$ and $\mathbf{x}_q$, respectively. Their numerical similarity is defined as:

$$\text{sim}(\mathbf{x}_i, \mathbf{x}_q) = \frac{\mathbf{v}_i^\top \mathbf{v}_q}{\|\mathbf{v}_i\|_2 \|\mathbf{v}_q\|_2}.$$

To support contrastive analysis, we use label-balanced retrieval under a fixed total budget $K = 16$. For classification tasks, we allocate the budget across classes and retrieve the top-k_c most similar instances from each class: $\forall \tilde{y} \in \mathcal{Y}$,

$$\mathcal{N}_{\tilde{y}}(\mathbf{x}_q) = \underset{\mathbf{x}_i \in \Omega(\mathbf{x}_q),\, y_i = \tilde{y}}{\arg\text{top-}k_{\tilde{y}}} \ \text{sim}(\mathbf{x}_i, \mathbf{x}_q).$$

The final retrieved context is the union $\mathcal{N}(\mathbf{x}_q) = \bigcup_{\tilde{y}=0}^{|\mathcal{Y}|-1} \mathcal{N}_{\tilde{y}}(\mathbf{x}_q)$, where $\sum_{\tilde{y}=0}^{|\mathcal{Y}|-1} k_{\tilde{y}} = K$. For regression, we split targets at the global median and apply the same balanced retrieval strategy.

The resulting *Tri-Level Augmented Input* combines dataset statistics, the serialized row, the ground-truth label, and retrieved neighbors. Formally, for each training instance $(\mathbf{x}_i, y_i)$, we define

$$p_i^* = \mathcal{G}(\mathcal{D}_{\text{train}}) \oplus p_i \oplus y_i \oplus \mathcal{N}(\mathbf{x}_i).$$

3.3 Tri-Level Rationale Schema

To convert the provided evidence into rationales for student fine-tuning, the teacher is prompted to perform in-depth analysis of the Tri-Level Augmented Input p_i^* under a fixed three-part schema: *instance-level reasoning*, *dataset-level reasoning*, and *comparison-level reasoning*. This structure grounds the prediction in the sample itself, contextualizes it with global distributional statistics, and refines it using contrastive evidence from retrieved neighbors. For brevity, we present only the schematic definitions here, while the full prompt template is deferred to Appendix C.

Instance-level reasoning (Self feature-based): At this level, the model analyzes each feature strictly using general domain logic and task-relevant common sense. The prompt requires the model to explain its baseline intuition and state the directional impact of each feature on the target variable.

> **(1). Self feature-based reasoning:**
> - Analyze each feature from "Current Sample Features" strictly using <general domain logic and task-relevant common sense>.

Dataset-level reasoning (Global statistics-based): At this level, the rationale contextualizes the sample by comparing its feature values with the provided dataset statistics. The prompt requires explicit citation of numerical statistics so that the explanation is grounded in global distributional evidence.

> **(2). Global statistics-based reasoning:**
> - Contextualize the sample by comparing its key feature values against the provided dataset statistics for EACH class.
> - Prioritize those with the clearest class separation or most extreme placement for this sample.

Comparison-level reasoning (Similar-cases): At this level, the model uses the retrieved neighbors as contrastive anchors to extract feature-combination patterns and critical deviations that further support the prediction. To encourage abstraction rather than record-level reproduction, this part is constrained to two structured blocks: <*Patterns*> and <*Deviations*>. The former summarizes recurring feature-combination alignments, while the latter highlights the most decision-relevant mismatches between the current sample and the retrieved neighbors.

> **(3). Similar-cases reasoning:**
> - Use the provided similar historical cases to extract feature-combination alignment patterns and critical deviations to further strengthen and validate your prediction logic.

3.4 Teacher Generation and Distillation

Teacher generation. $\mathcal{M}_T$ denotes a teacher LLM. For each instance $(\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}$, we prompt $\mathcal{M}_T$ with the Tri-Level Augmented Input p_i^* defined in Section 3.2 and schema outlined in Section 3.3. Given the ground-truth label, the teacher acts as an expert that explains *why* the label is plausible, rather than predicting the outcome from scratch. The generation tri-level rationale r_i justifies y_i, yielding the fine-tuning corpus $\mathcal{C} = \{(p_i, o_i) \mid (\mathbf{x}_i, y_i) \in \mathcal{D}_{\text{train}}\}$, where $o_i = (r_i, y_i)$.

Student fine-tuning. The student receives *only* the serialized feature p_i and is trained to generate a unified output sequence formed by concatenating the reasoning trace and label: $[\mathbf{r}_i \oplus y_i]$. We train student parameters Θ using the standard causal language modeling objective:

$$\mathcal{L}_{SFT}(\Theta) = -\frac{1}{|\mathbf{r}_i \oplus y_i|} \sum_{t=1}^{|\mathbf{r}_i \oplus y_i|} \log P_\Theta(w_t \mid p_i, w_{<t})$$

where w_t represents the t-th token in the target sequence. Fine-tuning indirectly injects global distributional knowledge and case-based alignment patterns into the model parameters Θ. During online deployment, the student model operates entirely autonomously without external evidence, making it efficient while preserving reasoning capabilities.

4 Experiments

We answer the following research questions (RQ):

- **RQ1**: How do base LLMs perform on tabular reasoning, and do they benefit from our proposed evidence sources when provided at inference time?
- **RQ2**: Under raw-feature-only inference, how does TLRD improve base LLMs performance on tabular reasoning, and can it be competitive with strong tabular baselines?
- **RQ3**: Compared with standard LLM distillation, how does TLRD perform in terms of predictive performance and rationale grounding?
- **RQ4**: How do different supervision sources (i.e., self-distillation and teacher-model distillation) affect the performance of TLRD?
- **RQ5**: How small can an LLM be while maintaining competitive performance under TLRD?

4.1 Experiment Setup

Datasets. We evaluate TLRD on six datasets covering both classification and regression. The classification benchmarks are ADULT (Becker and Kohavi, 1996), HOME CREDIT (HC) (Montoya et al., 2018), OKCUPID (OK) (Kim and Escobedo-Land, 2015), and DIABETES130US (D130) (Strack et al., 2014), spanning social, financial, and clinical domains with varying class imbalance and feature heterogeneity. For regression, we use CALIFORNIA (Kelley Pace and Barry, 1997) and DIAMONDS (Dia.) (Villanueva and Chen, 2019). To reduce computational cost, we downsample DIABETES130US and HOME CREDIT to 50k instances each. Dataset statistics and preprocessing details are given in Appendix A. All datasets are split into training, validation, and test sets with an 8:1:1 ratio.

Models & Baselines. We consider instruction-tuned LLMs including *Llama 3.1-8B* (Grattafiori et al., 2024), *Gemma 3-12B* (Team et al., 2025), *Qwen3-8B*, *Qwen3-Next-80B-A3B-Instruct* (Yang et al., 2025a), and *GPT-OSS-20B/120B* (Agarwal et al., 2025). As non-LLM baselines, we use a diverse set of strong tabular methods: *XGBoost* and *CatBoost* (gradient-boosted decision trees (GBDTs)), *TabM* (Gorishniy et al., 2025) (a recent deep tabular model), and *TabPFN* (Hollmann et al., 2023) (a tabular foundation model).

Evaluation Metrics. For binary classification tasks, given the prevalence of class imbalance in financial and risk datasets (e.g., HOME CREDIT), we report *F1-score*. For multi-class classification, we report *Macro-F1*, i.e., the unweighted average of per-class F1 scores. For regression tasks, we consider *Root Mean Squared Error (RMSE)*.

Implementation Details. Experiments are run on 4×A100 GPUs. For *XGBoost*, *CatBoost*, and *TabM* baselines, we perform hyperparameter optimization on the validation set with 50 Optuna trials; the search spaces are in Appendix Table 6. Due to limited computation resources, we conduct LoRA fine-

Model	Adult ↑	HC ↑	D130 ↑	OK ↑	California ↓	Dia. ↓
TabPFN(v2.5)	68.9	29.7	34.3	55.9	40,231.0	<u>496.0</u>
TabPFN(v2.0)	67.8	24.0	33.3	52.6	<u>39,434.4</u>	505.7
XGBoost	<u>72.0</u>	<u>29.8</u>	42.7	58.8	43,253.8	527.7
CatBoost	71.3	28.1	42.8	<u>58.9</u>	42,654.1	502.6
TabM	70.1	28.0	<u>43.7</u>	58.2	48,855.8	519.7
GPT-OSS-20B (Zero-shot)	57.9	16.6	14.2	49.4	201,829.8	2,477.1
Tri-level Reasoning	**66.9**	24.3	36.0	54.5	75,992.7	705.1
Qwen3-Next-80B (Zero-shot)	61.9	15.5	8.9	44.5	333,534.9	1,289.5
Tri-level Reasoning	63.5	23.5	30.4	52.6	73,383.1	698.4
GPT-OSS-120B (Zero-shot)	58.2	16.2	21.1	50.4	114,749.6	2,632.1
Tri-level Reasoning	66.6	25.2	37.3	55.0	**62,906.6**	**649.7**
Llama 3.1-8B (Zero-shot)	51.3	15.5	19.7	46.0	$3.02_{\times 10^7}$	$3.64_{\times 10^8}$
Tri-level Reasoning	63.9	17.8	27.1	51.5	$1.12_{\times 10^7}$	2,193.4
TLRD (GPT-OSS-120B)	64.7	25.1	34.9	54.2	86,572.7	1,531.9
TLRD (Qwen3-Next-80B)	63.1	22.7	35.7	53.2	112,008.0	1,453.7
Qwen3-8B (Zero-shot)	57.9	20.4	6.1	53.6	175,459.5	2,280.9
Tri-level Reasoning	65.5	26.0	36.6	54.0	65,660.6	691.0
TLRD (GPT-OSS-120B)	65.7	**26.3**	33.4	55.6	86,341.3	1,316.6
TLRD (Qwen3-Next-80B)	63.0	22.7	36.9	55.0	91,877.5	1,326.1
Gemma 3-12B (Zero-shot)	55.1	15.0	9.2	50.3	568,304.3	2,589.5
Tri-level Reasoning	63.1	18.8	33.2	51.1	70,633.5	1,044.9
TLRD (GPT-OSS-120B)	63.4	25.7	34.5	**56.1**	85,646.6	945.9
TLRD (Qwen3-Next-80B)	64.8	22.3	**38.1**	55.6	96,473.5	1,060.4

Table 1: Results across all datasets. Bold values denote the best result among LLM-based methods, and underlined values denote the strongest non-LLM tabular baseline for each dataset. Among fine-tuned student backbones, TLRD variants achieve the strongest classification performance within every backbone group.

tuning (Hu et al., 2022) on three relatively compact models: *Llama 3.1-8B*, *Qwen3-8B*, and *Gemma 3-12B*. Full details are provided in Appendix B.

4.2 Results

RQ1: Baseline LLM performance and impact of extra evidence sources. In base LLMs, we follow *TabLLM* (Xie et al., 2024; Hegselmann et al., 2023; Wolff and Hulsebos, 2025) and use the **zero-shot** setting as baseline. We then study a **Tri-level Reasoning** variant, in which the query is augmented with dataset statistics and retrieved similar samples to provide tri-level context at inference time.

Baseline performance. As shown in Table 1, base LLMs exhibit weak and unstable zero-shot performance. They perform better on ADULT and OK, where patterns are usually more aligned with common social semantics that LLMs have captured during pre-training. In contrast, performance drops substantially on HC and D130, which rely more on dataset-specific statistical regularities, class imbalance, and specialized feature semantics. For example, *Qwen3-Next-80B* achieves only 8.9 on D130. The limitation is even more pronounced on regression, where several models produce extremely large RMSE values, such as *Llama 3.1-8B*, which reaches 3.02×10^7 on CALIFORNIA. Tabular regression requires accurate scale awareness and fine-grained numerical mappings from inputs to targets. Without explicit grounding, LLMs can easily produce severe scale mismatches.

Tri-level Reasoning performance. Tri-level Reasoning improves the performance of base LLMs across all models and datasets. For example, *GPT-OSS-20B* improves from 14.2 to 36.0 on D130. For the challenges faced by zero-shot, dataset-level reasoning provides dataset statistical context, allowing the model to **judge which class a feature value is more aligned with and how informative it is.** Comparison-level reasoning further adds local contrastive evidence through balanced retrieved neighbors, helping the model **capture discriminative feature combinations**. These benefits also extend to regression. Dataset-level reasoning provides coarse calibration signals, while comparison-level reasoning supplies exact target-value references. Together, they **help the model better capture the output scale and infer a more plausible prediction range.** This motivates our next hypothesis: can we let a strong teacher analyze these evidence sources more deeply under the ground-truth label, and then distill the resulting evidence-grounded knowledge into a compact student model?

RQ2: TLRD performance. Motivated by this hypothesis, TLRD uses the tri-level schema during

Task: Based on the following applicant and loan features, predict whether the applicant will default on the loan.

EXT_SOURCE_2=0.225; AMT_GOODS_PRICE=270000; AMT_CREDIT=298512; FLAG_OWN_CAR=N; ORGANIZATION_TYPE=Government; FLAG_DOCUMENT_3=1

Gemma3-12B with Standard Distillation

Prediction: NO ❌ − EXT_SOURCE_2 is 0.225 – A relatively low value. A low values are associated with higher default risk, pulling the assessment toward default. AMT_GOODS_PRICE is 270 000, while the requested AMT_CREDIT is 298 512. The loan amount is only slightly higher than the price of the goods, indicating the applicant is not over-leveraged relative to the purchase. ORGANIZATION_TYPE is "Government", which typically signals stable, regular income. FLAG_OWN_CAR = N does not add any additional risk. The FLAG_DOCUMENT_3 = 1. shows that required documentation is complete, reducing information-asymmetry risk. Overall, while the low EXT_SOURCE_2 score raises concern, the applicant's stable government employment, reasonable loan-to-price ratio, complete documentation, and lack of additional liabilities (no car ownership) collectively point toward a lower likelihood of default.

Gemma3-12B with TLRD

Prediction: YES ✅ (1). Self feature-based reasoning: Not owning a car removes a potential asset that could be liquidated; neutral to slightly adverse(2). Global statistics-based reasoning: AMT_GOODS_PRICE | 270 k *non-default stats* | *default stats.* **Slightly above the 25p of both classes; neutral to mildly favorable.** AMT_CREDIT | 298 k | *non-default stats* | *default stats.* Just above the 25p of both groups; neutral. **FLAG_DOCUMENT_3 (1) | 77.8 % of defaults vs 70.4 % of non-defaults | More prevalent in defaults → ↑ risk.**(3). Similar-cases reasoning: <Patterns> FLAG_OWN_CAR = N, ORGANIZATION_TYPE = Government, FLAG_DOCUMENT_3 = 1 – all these categorical attributes are identical across the similar default records. <Deviations> **EXT_SOURCE_2 is lower (0.225) than many similar non-default records,** this pushes the sample toward the higher-risk side.

Figure 3: A qualitative case study comparing the same model under standard distillation and TLRD. For a sample with ground-truth label YES, the standard distillation model predicts incorrectly, whereas TLRD predicts correctly and provides a more grounded tri-level rationale.

corpus construction to let a teacher model generate evidence-grounded analyses conditioned on the given label, and distills the resulting knowledge into the student through fine-tuning. Although the fine-tuned student receives only raw features, **it consistently achieves the strongest classification performance within its backbone family.** For example, under *GPT-OSS-120B* supervision, *Qwen3-8B* reaches 26.3 on HC. Under *Qwen3-Next-80B* supervision, *Gemma 3-12B* reaches 38.1 on D130. Compared with zero-shot, these gains are substantial, showing that TLRD can effectively convert teacher-side evidence analysis into useful supervision for compact student models. On regression, TLRD still improves substantially over zero-shot, but it often remains below Tri-level Reasoning. Regression requires much finer scale control and numerical calibration than discrete class prediction. Compressing such precise quantitative mappings into the student parameters remains challenging.

Table 1 shows that **TLRD substantially narrows the gap between LLM-based methods and strong tabular baselines on classification tasks.** On OK, TLRD achieves a score of 56.1, slightly surpassing *TabPFN* v2.5 (55.9) and approaching *TabM* (58.2), *CatBoost* (58.9), and *XGBoost* (58.8). On D130, all three fine-tuned backbones outperform both *TabPFN* variants, further showing that TLRD is competitive with modern tabular models.

RQ3: Comparison between TLRD and standard rationale distillation. For this comparison, we fix *GPT-OSS-120B* as the teacher model for corpus construction. The standard distillation is trained on teacher-generated free-form explanations.

Predictive performance. Table 2 shows that TLRD outperforms standard distillation across all three backbones on the classification datasets. On DIA., the advantage is less uniform, but TLRD still achieves the strongest overall result. These findings suggest that TLRD provides more effective supervision by organizing the explanation around tri-level evidence under the designed schema.

Method	HC ↑	OK ↑	Dia. ↓
Llama 3.1-8B Standard Distill.	19.9	46.8	1201.7
Llama 3.1-8B TLRD	25.1	54.2	1531.9
Qwen3-8B Standard Distill.	23.7	51.9	1317.5
Qwen3-8B TLRD	26.3	55.6	1316.6
Gemma 3-12B Standard Distill.	20.0	44.8	1152.1
Gemma 3-12B TLRD	25.7	56.1	945.9

Table 2: Performance of standard distillation and TLRD.

Rationale grounding. Figure 3 compares two *Gemma 3-12B* variants on the same HC instance. Although standard distillation generates a fluent explanation, its reasoning is largely based on generic commonsense assumptions that appear plausible but are not supported by concrete evidence. By contrast, TLRD structures its explanation into instance-level, dataset-level, and comparison-level reasoning, yielding a more grounded explanation. At the dataset level, it captures the counterintuitive but dataset-supported pattern that `FLAG_DOCUMENT_3 = 1` is more prevalent among default applicants. At the comparison level, identifies `EXT_SOURCE_2 = 0.225` as a key deviation from similar non-default profiles. Together, these elements indicate that TLRD combines reasonable local intuition with dataset-specific evidence, rather than relying only on potentially misleading narrative associations.

RQ4: Effect of different supervision sources. We consider self-distillation and distillation from multiple teacher models, including *GPT-OSS-20B*,

Student	Teacher	HC ↑	OK ↑	Dia. ↓
Llama 3.1-8B	Self	9.7	52.6	1606.1
	GPT-OSS-20B	24.3	54.8	1475.3
	Qwen3-Next-80B	22.7	53.2	1453.7
	GPT-OSS-120B	25.1	54.2	1531.9
	GPT-5.2	25.0	55.6	1391.7
Qwen3-8B	Self	24.9	53.2	1417.2
	GPT-OSS-20B	25.3	56.3	1135.7
	Qwen3-Next-80B	22.7	55.0	1326.1
	GPT-OSS-120B	26.3	55.6	1316.6
	GPT-5.2	24.7	54.4	1295.5
Gemma 3-12B	Self	20.8	56.0	990.1
	GPT-OSS-20B	24.8	53.5	859.6
	Qwen3-Next-80B	22.3	55.6	1060.4
	GPT-OSS-120B	25.7	56.1	945.9
	GPT-5.2	24.4	57.0	906.2

Table 3: Performance under different supervision sources for each student backbone.

Qwen3-Next-80B, *GPT-OSS-120B*, and *GPT-5.2*.

Table 3 shows that under the TLRD framework, mid-sized teacher *GPT-OSS-20B* already provides strong supervision, and self-distillation can also be competitive in some cases. Meanwhile, performance does not always scale with the supervisors, meaning that TLRD typically does not require very large teacher models. In several settings, relatively smaller supervisors even achieve performance similar to larger ones. Overall, for tabular tasks, **TLRD does not necessarily require massive teacher LLMs. Rather, it can successfully rely on a mid-sized teacher model to enable efficient and resource-friendly deployment.**

	Gemma 3					Qwen3				
Dataset	27B	12B	4B	1B	270M	14B	8B	4B	1.7B	0.6B
HC ↑	26.6	25.7	23.9	13.7	9.5	26.1	26.3	24.2	23.4	23.1
OK ↑	59.7	57.0	56.5	40.9	34.1	56.8	56.3	55.7	55.4	43.4
Dia. ↓	712.8	859.6	1376.9	4457.2	14932.6	879.2	1135.7	1507.0	2008.7	3421.9

Table 4: Performance of *Qwen3* and *Gemma 3* families. The selected supervisors are *GPT-OSS-120B* for HC, *GPT-5.2* for *Gemma 3* on OK, and *GPT-OSS-20B* for *Qwen3* on OK and for both families on DIA.

RQ5: Performance of models from the same family at different scales. We conduct experiments on the *Qwen3* and *Gemma 3* families. Since our focus is on how small the student can be while maintaining performance, we report the results obtained with the best teacher supervisor.

Table 4 summarizes the results across three representative datasets. Overall, increasing the student model scale generally improves TLRD performance. In addition, both 4B-scale students still show strong performance on OK and only mild degradation on HC. A particularly encouraging result appears on OK, where *Gemma 3-27B* reaches 59.7 Macro-F1, **even surpassing the strongest baseline, *CatBoost* (58.9)**.

Combined with the analysis in RQ4, these results show that TLRD offers a potential practical direction for resource-constrained scenarios: **using a 4B-scale student model supervised by a mid-sized teacher to enable effective and resource-friendly deployment under classification tasks**, while larger students or stronger teachers can be used when further performance gains are needed.

4.3 Design and Analyses on Hallucination, Consistency, and Human Annotation; Ablation Study; Further Discussion

Design and Analyses. We present design safeguards and conduct additional quantitative analyses in Appendix D, including hallucination, consistency, and human annotation, to further assess the reliability and usefulness of TLRD.

Ablation. We provide ablations in Appendix E to examine key designs of TLRD, including extra evidence sources and the designed schema.

Further Discussion. From a performance perspective, both tree-based methods and TabM have mature optimization pipelines and class-imbalance handling, whereas we did not conduct systematic hyperparameter optimization for LLMs due to the high cost and time-consuming. However, TLRD still surpasses them with *Gemma 3-27B* on OK, which demonstrates the potential of TLRD. We also explored several task-specific optimization directions in Appendix F. We hope our work encourages more mature and practical deployment of LLM-based methods for tabular tasks.

5 Conclusion

We proposed TLRD, a framework that converts label-only tabular datasets into structured rationale supervision for LLMs. TLRD provides the teacher with a Tri-Level Augmented Input, which extends the labeled sample with dataset statistics and retrieved neighbors, and uses a Tri-Level Schema to guide grounded tri-level rationale generation. By distilling these rationales into student models, TLRD enables efficient inference from raw features only. Experiments show that TLRD achieves competitive performance against strong tabular baselines, while producing grounded and human-readable explanations. Overall, TLRD provides a practical assistive tool for supporting transparent decision support, highlighting structured rationale distillation as an effective bridge between tabular prediction and explanation.

Limitations

Our study has several limitations that suggest caution and opportunities for future work.

- **Distillation of Spurious Correlations.** TLRD depends on the quality of teacher-generated rationales. Although we condition the teacher on the ground-truth label to anchor the rationale to the correct outcome, the teacher may still produce spurious correlations, subtle logical leaps, or plausible but unsupported justifications. Since the student is trained to mimic this corpus, undetected artifacts in teacher outputs may be distilled into the student model.
- **Scalability to High-Dimensional Features and Classes.** For datasets with many columns, we rely on feature selection to construct concise natural-language prompts, which may omit long-tail variables or weak feature interactions. In addition, our dataset-level reasoning schema appends class-specific statistics; when the number of classes is large, this can substantially increase input length and dilute the useful context. Nevertheless, many practical tabular decision tasks are binary or low-class classification problems, where TLRD remains directly applicable.
- **Use of LoRA Instead of Full Supervised Fine-Tuning.** Due to limited computational resources, we adopt LoRA-based parameter-efficient fine-tuning rather than full supervised fine-tuning (full SFT). While this choice enables training on multiple datasets and backbones within a practical budget, it also limits our ability to examine whether full SFT could yield larger improvements in predictive performance or rationale quality. Since full-parameter adaptation may internalize distilled knowledge differently, this remains an important direction for future work.
- **Inherent Limitations of LLM-Generated Rationales and Human Evaluation.** TLRD inherits the general limitations of LLM-based generation, including possible hallucination, overconfident wording, and sensitivity to prompt or decoding settings. We conduct quantitative analyses and a small-scale human study to assess whether the generated rationales are grounded, consistent, and useful. However, human evaluation of tabular rationales is inherently task- and domain-dependent, and a larger study with domain experts would be needed to fully assess their impact in real-world decision-making. Therefore, TLRD should be deployed strictly as a decision-support tool, not as an autonomous decision-maker.

Ethical Considerations

- **Bias Inheritance.** Tabular datasets frequently include sensitive attributes like race, gender, and socioeconomic status. There is a risk that the teacher generates biased outcomes and leads the student model to learn biased rationales. Practitioners may conduct rigorous algorithmic fairness audits on the data before applying the TLRD pipeline.
- **Data Privacy.** Our methodology makes a concerted effort to protect data privacy. By explicitly prohibiting the teacher model from referencing neighbor IDs or precise historical records, and by completely discarding the retrieved neighbors during the student fine-tuning phase, TLRD structurally mitigates the risk of verbatim training data leakage at deployment. Nevertheless, there remains a theoretical risk that highly unique feature combinations (outliers) could be implicitly memorized and reconstructed in the generalized <*Patterns*> or <*Deviations*> blocks.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20.

Tianqi Chen and Carlos 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.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. 2026. TabArena: A living benchmark for machine learning on tabular data. In *The Thirty-ninth Annual Conference on Neural*

Information Processing Systems Datasets and Benchmarks Track.

Chufan Gao, Jintai Chen, and Jimeng Sun. 2025. Utilizing training data to improve LLM reasoning for tabular understanding. *arXiv preprint arXiv:2508.18676*.

Josh Gardner, Juan C Perdomo, and Ludwig Schmidt. 2024. Large scale transfer learning for tabular data via language modeling. *Advances in Neural Information Processing Systems*, 37:45155–45205.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. 2025. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *The Thirteenth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.

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 *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 5549–5581. PMLR.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. TabPFN: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

R. Kelley Pace and Ronald Barry. 1997. Sparse spatial autoregressions. *Statistics Probability Letters*, 33(3):291–297.

Albert Y Kim and Adriana Escobedo-Land. 2015. Okcupid data for introductory statistics and data science courses. *Journal of Statistics Education*, 23(2).

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Anna Montoya, inversion, KirillOdintsov, and Martin Kotek. 2018. Home credit default risk. `https://kaggle.com/competitions/home-credit-default-risk`. Kaggle.

Bo Pan, Zhen Xiong, Guanchen Wu, Zheng Zhang, Yifei Zhang, Yuntong Hu, and Liang Zhao. 2025. GraphNarrator: Generating textual explanations for graph neural networks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–42, Vienna, Austria. Association for Computational Linguistics.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. CatBoost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31.

Günther Schindler, Maximilian Schambach, Michael Medek, and Sam Thelin. 2025. TabGemma: Text-based tabular icl via LLM using continued pretraining and retrieval. *arXiv preprint arXiv:2511.03570*.

Ravid Shwartz-Ziv and Amitai Armon. 2022. Tabular data: Deep learning is not all you need. *Information fusion*, 81:84–90.

Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. 2014. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014(1):781670.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Randle Aaron M Villanueva and Zhuo Job Chen. 2019. ggplot2: elegant graphics for data analysis.

Ruiyu Wang, Zifeng Wang, and Jimeng Sun. 2023. Unipredict: Large language models are universal tabular classifiers. *arXiv preprint arXiv:2310.03266*.

Yi-Siang Wang, Kuan-Yu Chen, Yu-Chen Den, and Darby Tien-Hao Chang. 2026. BoostLLM: Boosting-inspired LLM fine-tuning for few-shot tabular classification. *Preprint*, arXiv:2605.06117.

Cornelius Wolff and Madelon Hulsebos. 2025. How well do LLMs reason over tabular data, really? In *Proceedings of the 4th Table Representation Learning Workshop*, pages 241–250.

Jie Wu and Mengshu Hou. 2025. An efficient retrieval-based method for tabular prediction with LLM. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9917–9925.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, and 1 others. 2024. Finben: A holistic financial benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:95716–95743.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yining Yang, Ruihong Huang, and Yang Shen. 2025b. Beyond labels: Explanatory collapse due to instruction tuning in protein LLMs. In *NeurIPS 2025 2nd Workshop on Multi-modal Foundation Models and Large Language Models for Life Sciences*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pages 400–410.

Dataset	Task	N (raw)	#Num	#Cat	#Classes	Imbalance (%)	License
Adult	Binary cls.	48,842	6	8	2	23.9:76.1	CC BY 4.0
Home Credit	Binary cls.	307,511	106	16	2	8.1:91.9	Competition data
OkCupid	Multi cls.	50,789	2	17	3	71.6:18.8:9.6	Public-permissioned
Diabetes130US	Multi cls.	69,990	8	37	3	59.3:31.7:9.0	CC BY 4.0
California	Regression	20,640	8	1	–	–	CC0: Public Domain
Diamonds	Regression	53,940	6	3	–	–	CC BY 4.0

Table 5: Dataset statistics. For binary datasets, **Imbalance (%)** reports the positive-to-negative class ratio (P:N) in percentages. Preprocessing and subsampling details are provided in Appendix A.

A Datasets

We use six real-world tabular datasets: two binary classification, two multi-class classification, and two regression datasets. Below we provide detailed dataset descriptions and preprocessing details. Unless stated otherwise, we use the full dataset and partition it into training, validation, and test splits with an 8:1:1 ratio. Summary statistics are reported in Table 5.

Adult (Becker and Kohavi, 1996).[1] A binary classification dataset from the 1994 U.S. Census, where the goal is to predict whether an individual's annual income exceeds $50K based on demographic and employment attributes.

Home Credit (Montoya et al., 2018).[2] A binary credit-risk dataset where the goal is to predict whether an applicant defaults on the loan. The dataset contains 307,511 labeled applications, is highly imbalanced, with only about 8.07% positive (default) cases, and includes 122 features. During LLM inference and corpus construction, we retain only the top 15 features ranked by CatBoost SHAP importance. To mitigate optimization bias from severe class imbalance during LLM fine-tuning while also satisfying the computational constraints of Transformer-based in-context inference with long contexts, we apply undersampling on the training split when building LLM inference corpora and when forming TabPFN (Hollmann et al., 2023) contexts. Specifically, we perform ratio-preserving undersampling with a 3:1 negative-to-positive target and continue sampling until the training subset reaches 40,000 examples; this reduces majority-class dominance while keeping a reasonable approximation to the original class prior, as well as token budget, GPU memory usage, and training time tractable. This treatment is particularly important for TabPFN: beyond the quadratic context-length cost of concatenating labeled training examples in a single forward pass, using the original highly imbalanced pool without undersampling makes in-context predictions strongly majority-class biased, leading to systematic under-detection of defaults. For evaluation efficiency, we further subsample the test split to 5,000 examples while preserving the original class ratio, improving throughput and reproducibility without altering the underlying data distribution.

OkCupid (Kim and Escobedo-Land, 2015).[3] A multi-class classification dataset where the goal is to predict the profile author's occupation category (STEM / non-STEM / student). Inputs are mixed-type attributes describing demographics and self-reported profile fields, with a predominance of categorical variables.

Diabetes130US (1999–2008) (Strack et al., 2014).[4] A multi-class clinical prediction dataset containing inpatient encounters for diabetes patients from 130 U.S. hospitals (1999–2008), with 101,766 instances. We consider the original readmission prediction setting, where the label indicates the readmission outcome as multiple categories (commonly `NO`, `<30`, and `>30`). Data preprocessing follows the cohort-construction logic of Strack et al. Specifically, we remove encounters with hospice/death discharge outcomes, keep only the first encounter per patient to ensure patient-level independence, drop high-missing columns (`weight` and `payer_code`), and impute missing `medical_specialty` as `missing`. We then follow Strack-style binning to map ICD-9 diagnosis codes into coarse clinical groups and to rebin age into three categories. Our downstream setup intentionally differs from Strack et al. Their main aim is explanatory statistical modeling (logistic-regression-based analysis of HbA1c and readmis-

[1] https://archive.ics.uci.edu/dataset/2/adult
[2] https://kaggle.com/competitions/home-credit-default-risk
[3] https://www.openml.org/d/42734
[4] https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008

sion), whereas our main aim is predictive classification. We use the original 3-class prediction target, `readmitted` $\in \{$`<30`, `>30`, `NO`$\}$. The dataset includes mixed numerical and categorical attributes describing demographics, diagnoses, procedures, lab tests, and medications. After preprocessing, we retain 69,990 samples with 45 features. The full preprocessed dataset is used for training and evaluating the non-LLM baselines. During LLM inference and corpus construction, we retain only the top 15 features ranked by CatBoost SHAP importance. Using an 8:1:1 split, we further subsample the training set to 40,000 examples for corpus construction and the test set to 5,000 examples for efficient LLM inference.

California (Kelley Pace and Barry, 1997).[5] A regression dataset derived from the 1990 California census, where the target is the median house value for a census block group. We use the Kaggle version of the dataset released as `camnugent/california-housing-prices`.

Diamonds (Villanueva and Chen, 2019).[6] A regression dataset of round-cut diamonds with features describing the "4 Cs" (carat, cut, color, clarity) and physical measurements (e.g., depth, table, and x/y/z), and the target being price in USD. We follow standard setup steps consistent with common tabular benchmarks.

B Experimental Setup

B.1 Baselines

We use a diverse set of strong tabular baselines: XGBoost (Chen and Guestrin, 2016) and CatBoost (Prokhorenkova et al., 2018) (gradient-boosted decision trees (GBDTs)), TabM (Gorishniy et al., 2025) (deep tabular model), and TabPFN (Hollmann et al., 2023) (foundation tabular predictor).

TabPFN (v2.5 / v2.0). TabPFN is a transformer-based foundation model for tabular prediction. It is pretrained on a large distribution of synthetic supervised learning tasks and performs inference in an in-context manner, conditioning directly on the observed training data to make predictions for test instances. We use the latest major release (TabPFN-2.5) and TabPFN-v2 (v2.0). Both are evaluated using their standard inference protocol.

XGBoost and CatBoost. XGBoost and CatBoost are gradient boosting decision tree methods that construct an ensemble of trees sequentially, with each new tree trained to correct the errors of the current ensemble; CatBoost additionally includes specialized handling of categorical features. For both XGBoost and CatBoost, we reference the hyperparameter spaces reported in TabArena (Erickson et al., 2026). We perform HPO using the TPE sampler in Optuna (Akiba et al., 2019) with 50 trials for each dataset–model pair, select the best configuration on the validation split, and then evaluate that selected configuration on the test split. For XGBoost and CatBoost, we use early stopping with a patience of 300 rounds. Hyperparameter details are in Table 6. To address class imbalance, we apply fixed loss weighting; implementation details are provided in Table 7.

TabM. TabM is a parameter-efficient ensemble-based tabular deep learning model. It efficiently emulates an ensemble of MLPs by training multiple predictors in parallel while sharing a large portion of their parameters. Hyperparameter settings are reported in Table 6, and class-imbalance handling is reported in Table 7. Our configuration follows the official TabM repository recommendations. We used early stopping on the validation set with a patience of 16 epochs, selecting the checkpoint with the best validation selection score; validation was evaluated after every epoch, and training stopped if the score did not improve for 16 consecutive epochs.

B.2 LLM Hyperparameters

Base LLM backbones. We use the same configuration for both base-model inference and corpus construction. We set the maximum context length to 32,768 tokens, use temperature 0.6, and apply nucleus sampling with $p = 0.95$.

LLM Fine-tuning. We use parameter-efficient fine-tuning (PEFT) with LoRA (Hu et al., 2022), implemented with LLaMA-Factory (Zheng et al., 2024). Fine-tuning hyperparameters are summarized in Table 8.

Fine-tuned LLM inference. After dataset-specific fine-tuning, the model is already aligned to the target prediction task. Therefore, we change the temperature to 0 to obtain stable and deterministic predictions.

[5] https://www.kaggle.com/datasets/camnugent/california-housing-prices

[6] https://doi.org/10.5281/zenodo.3522106

Model	Hyperparameter	Search Space / Setting
XGBoost	learning_rate	LogUniform([0.005, 0.1])
	max_depth	LogUniformInt([4, 10])
	min_child_weight	LogUniform([0.001, 5.0])
	subsample	Uniform([0.6, 1.0])
	colsample_bylevel	Uniform([0.6, 1.0])
	colsample_bynode	Uniform([0.6, 1.0])
	reg_alpha	Uniform([1e-4, 5.0])
	reg_lambda	Uniform([1e-4, 5.0])
	grow_policy	Choice([`depthwise`, `lossguide`])
	max_cat_to_onehot	LogUniformInt([8, 100])
	max_leaves	LogUniformInt([8, 1024])
	n_estimators	Fixed to 10,000
CatBoost	learning_rate	LogUniform([0.005, 0.1])
	bootstrap_type	Bernoulli
	subsample	Uniform([0.7, 1.0])
	grow_policy	Choice([`SymmetricTree`, `Depthwise`])
	depth	UniformInt([4, 8])
	colsample_bylevel	Uniform([0.85, 1.0])
	l2_leaf_reg	LogUniform([1e-4, 5])
	leaf_estimation_iterations	LogUniformInt([1, 20])
	one_hot_max_size	LogUniformInt([8, 100])
	model_size_reg	LogUniform([0.1, 1.5])
	max_ctr_complexity	UniformInt([2, 5])
	boosting_type	Plain
	max_bin	254
	n_estimators	Fixed to 10,000
TabM	embedding_method	Fixed to `piecewise linear embedding`
	k	Const([32]) (not tuned)
	n_blocks	UniformInt([1, 4])
	d_block	UniformInt([64, 1024], step=16)
	lr	LogUniform([1e-4, 5e-3])
	weight_decay	$\{0, \text{LogUniform}[1\text{e}{-4}, 1\text{e}{-1}]\}$
	n_bins	UniformInt([2, 128])
	d_embedding	UniformInt([8, 32], step=4)

Table 6: Hyperparameter search spaces used in HPO.

Model	Task	Balancing Rule
XGBoost	Binary	`scale_pos_weight = N_neg / N_pos`
	Multiclass	`w_k = N_max / N_k`, `sample_weight_i = w_y_i`
CatBoost	Binary	`class_weights = [1.0, N_neg / N_pos]`
	Multiclass	`class_weights[k] = N_max / N_k`
TabM	Binary	`pos_weight = N_neg / N_pos`
	Multiclass	`class_weights[k] = N_max / N_k`

Table 7: Class imbalance handling rules used for baselines (fixed, not tuned).

C Prompt

We use different prompt templates for (i) zero-shot inference, (ii) w/o Statistics, (iii) w/o Comparison, (iv) Tri-level Reasoning, (v) standard fine-tuning (FT) corpus construction, (vi) TLRD corpus construction, and (vii) TLRD model inference. The zero-shot template is also used for inference with the standard fine-tuned model. After each template, we list feature explanations and feature importance (high to low), ordered by CatBoost SHAP importance. Bold angle-bracketed tokens (e.g., **`<DATASET_DOMAIN>`**) are instantiated with dataset-specific metadata/domain knowledge. Specifically, templates for (i)–(iv) are shown in Table 9, template (v) is shown in Table 10, template (vi) is shown in Table 11, and template (vii) is shown in Table 12.

Hyperparameter	Value
learning_rate	0.0001
train_batch_size	4
eval_batch_size	8
seed	42
distributed_type	multi-GPU
num_devices	4
gradient_accumulation_steps	8
lr_scheduler_type	cosine
num_epochs	1.0 (only California: 2.0)
optimizer	`adamw_torch`
lora_rank	8
lora_alpha	16
lora_dropout	0
lora_target	`all`

Table 8: LLM fine-tuning hyperparameters.

Zero-shot / w/o Statistics / w/o Comparison / Tri-level Reasoning Prompt Template

```
You are a <DATASET_DOMAIN> classifier/estimator.
Based on the following <ENTITY> features, predict <TARGET_DESCRIPTION>.
Please explain your reasoning.
At the very end of your reasoning, on a separate line, output exactly:
Prediction is <value>.
where <value> is the <PREDICTED_TARGET_VALUE_DESCRIPTION>.
Do not add any other text after this line.
[If applicable: w/o Statistics]
You are also provided with similar historical cases to assist your reasoning.
[If applicable: w/o Comparison]
You are also provided with dataset statistics to assist your reasoning.
[If applicable: Tri-level Reasoning]
You are also provided with dataset statistics and similar historical cases to assist your reasoning.
```

Table 9: Prompt template used for Zero-shot / w/o Statistics / w/o Comparison / Tri-level Reasoning.

Standard FT Corpus Construction Prompt Template

```
You are a <DATASET_DOMAIN> classifier/estimator.
Based on the following <ENTITY> features, explain why THE GIVEN PREDICTION LABEL for
<TARGET_DESCRIPTION> is plausible. <LABEL_SEMANTICS_DESCRIPTION>
At the very end of your reasoning, on a separate line, output exactly:
Prediction is <value>.
Do not add any other text after this line.
```

Table 10: Prompt template for standard and Evidence-Augmented distillation corpus construction.

TLRD Corpus Construction Prompt Template

You are a **<DATASET_DOMAIN>** classifier/estimator.
Based on the following **<ENTITY>** features, conduct in-depth qualitative analysis to explain why **THE GIVEN PREDICTION LABEL** for **<TARGET_DESCRIPTION>** is plausible.
You are also provided with **DATASET STATISTICS** and **SIMILAR HISTORICAL CASES** to assist your reasoning.
Please explain your reasoning in the following way.

Reasoning format:
(1). Self feature-based reasoning:
- Analyze each feature from "Current Sample Features" strictly using domain logic and common sense.
Execution Rules:
- Briefly explain your intuition, and state its directional impact on the target.
- Focus on the self feature logical connection.
Prohibitions:
- DO NOT use any dataset-level quantities (e.g., mean, percentile) from "Dataset Statistics" or similar cases in this step.

(2). Global statistics-based reasoning:
- Contextualize the sample by comparing its key feature values against the provided dataset statistics for EACH class.
- Prioritize those with the clearest class separation or most extreme placement for this sample.
- For each feature, explicitly cite the relevant statistics (quote the numbers) when stating where the sample falls relative to the stats (e.g., closer to which class mean/median, or which percentile), highlight deviations/alignments between the sample and the statistical benchmarks, then explain the directional implication.

(3). Similar-cases reasoning:
Use the provided similar historical cases to extract feature-combination alignment patterns and critical deviations, to further strengthen and validate your prediction logic.
Execution Rules:
- Extract consistent feature-combination alignment patterns (e.g., [A + B + C] matches).
- Identify critical feature deviations (e.g., [D] differs).
- For every pattern or deviation mentioned, explicitly cite the current sample's corresponding features.
- Clearly explain how the current sample's features align with patterns or diverge from deviations.
Prohibitions:
- Do NOT reference similar historical cases using numeric identifiers (IDs, order/rank, similarity scores).
- Do NOT use the words "Example", "record", or "case".
- Structure your response in two sections:
1. <Patterns>: List all consistent feature-combination alignment patterns, each explicitly linked to the current sample's matching features.
2. <Deviations>: List the most critical feature deviations, each explicitly linked to the current sample's differing features.

(4). Final Resolution:
Weigh the evidence from the above steps. If there are conflicts, explicitly explain how you resolve them.
Use a layered, step-by-step reasoning chain: start from feature logic, then refine with statistics, then validate with similar historical cases.

At the very end of your reasoning, on a separate line, output exactly:

Prediction is <value>.

where <value> is the **<PREDICTED_TARGET_VALUE_DESCRIPTION>**.

Do not add any other text after this line.

Table 11: Prompt template for TLRD corpus construction.

TLRD Model Inference Prompt Template

You are a **<DATASET_DOMAIN>** classifier/estimator.
Based on the following **<ENTITY>** features, conduct in-depth qualitative analysis to predict **<TARGET_DESCRIPTION>**.
You should use **DATASET STATISTICS** and **SIMILAR HISTORICAL CASES** to assist your reasoning.
Please explain your reasoning in the following way.

Reasoning format:
(1). Self feature-based reasoning:
(2). Global statistics-based reasoning:
(3). Similar-cases reasoning:
(4). Final Resolution:

At the very end of your reasoning, on a separate line, output exactly:

Prediction is <value>.

where <value> is the **<PREDICTED_TARGET_VALUE_DESCRIPTION>**.

Do not add any other text after this line.

Table 12: Prompt template for TLRD model inference.

D Design and Analyses on Hallucination, Consistency, and Human Annotation

Given the inherent hallucination risk of LLMs, we explicitly design code-level safeguards, and we further conduct quantitative experiments to assess whether student-generated rationales are evidence-grounded and prediction-consistent. However, hallucination cannot be fully eliminated in current LLMs, and it remains an inherent limitation of LLM-based generation rather than a risk specific to TLRD. Therefore, our design and analyses should be understood as efforts to improve and verify the practical reliability of TLRD, rather than as a claim that hallucination is completely solved or absent. We further conduct a small-scale human study to assess whether the generated rationales are understandable, useful, and well structured from a decision-support perspective.

D.1 Corpus Construction Safeguards

Corpus consistency control. In the corpus-construction prompt, we state at the beginning that the teacher should explain why *the given prediction label* is plausible. However, in the subsequent rationale-generation steps, we do not repeatedly expose or emphasize the label. This design reduces the chance that the teacher simply rationalizes the label in an unconstrained way. In addition, to prevent data leakage and the hallucination of specific training records during student fine-tuning, the prompt imposes explicit generation constraints: the teacher is forbidden from echoing similarity scores, numeric identifiers, or rank orders in its final output. We also apply a rerun mechanism: if the teacher's final generated prediction does not match the given label, we discard the output and rerun the sample.

Corpus quality control. In our quality-control process, we conduct an abnormal-length check on generated rationales. Specifically, we use the IQR rule to identify outputs above the upper fence and manually inspect these long-rationale outliers. We find that about 1% of the generated rationales contain relatively weak and verbose reasoning. In our practical pipeline, we automatically drop these abnormal generations before fine-tuning.

Evidence support. During preliminary validation and optimization, we conducted a small-scale review to assess the usefulness of the selected evidence and to examine whether it provided reasonable support for the generated rationale. However, in complex real-world tabular datasets, especially HOME CREDIT, the decision boundary can be intrinsically noisy. The dataset contains more than 300K loan-application samples, and even strong tabular models achieve relatively low F1 scores on this task. According to our observation, in such datasets, highly similar samples, even with similarity scores as high as 0.99, can still have different labels. Therefore, we do not claim that every generated rationale identifies a uniquely decisive feature pattern. Instead, TLRD aims to provide a general framework for helping LLMs better understand and reason over tabular data by grounding rationales in instance-level, dataset-level, and comparison-level evidence.

D.2 Quantitative Analysis of Student Outputs

We conduct quantitative experiments to assess whether student-generated rationales are evidence-grounded and prediction-consistent. Experiments are conducted on four classification datasets by randomly sampling 100 data points from the validation set of each dataset. We exclude regression datasets because their numerical outputs can fluctuate substantially, making this evaluation protocol inapplicable to regression tasks. Specifically, we define the following four metrics: Prediction Stability, Prediction–Rationale Consistency, Cited-Number Error, and Similar-Case Support.

Prediction Stability. Adapted from SelfCheck-GPT for hallucination detection (Manakul et al., 2023), we build a hallucination-detection metric tailored to our tabular prediction setting. This can be explained by the probability-distribution mechanism of LLM generation. When a model is confident about a factual statement, the probability mass of the correct answer tends to dominate the output distribution. For example, for the question "What is the capital of the United States?", the model is expected to consistently generate "Washington, D.C." even when moderate randomness is introduced during decoding, such as using a higher temperature or increasing the top-p threshold to enlarge the candidate sampling pool. In our setting, this idea corresponds to measuring the consistency rate of predictions for the same sample under different temperature and top-p configurations.

For each test instance, we generate $R = 6$ outputs under different decoding configurations, using two temperature values $\{0.1, 0.2\}$ and three top-p values $\{0.9, 0.95, 1.0\}$. These settings introduce

Dataset	Model	Stab. ↑	Cons. ↑	Cite Err. ↓	Case ↑
ADULT	*Gemma 3-12B*	0.973	100%	1/4326	97%
	Qwen3-8B	0.965	99%	0/4598	96%
	Llama 3.1-8B	0.935	97%	1/4441	98%
HOME CREDIT	*Gemma 3-12B*	0.953	100%	0/5925	98%
	Qwen3-8B	0.943	100%	1/5979	99%
	Llama 3.1-8B	0.938	100%	1/6401	98%
OKCUPID	*Gemma 3-12B*	0.917	100%	2/3307	98%
	Qwen3-8B	0.910	100%	2/3165	98%
	Llama 3.1-8B	0.903	100%	1/3549	96%
DIABETES130US	*Gemma 3-12B*	0.905	100%	2/9132	96%
	Qwen3-8B	0.913	97%	1/8943	93%
	Llama 3.1-8B	0.888	98%	3/9307	94%

Table 13: Quantitative analyses of hallucination and consistency on classification datasets. Stab. denotes prediction stability averaged over all sampled instances; for each instance, 1.000 means that all repeated generations produce the same prediction, while 0.833 means that five out of six generations agree, i.e., one prediction shift. Cons. denotes the percentage of samples that pass the prediction–rationale consistency check; 100% means that all evaluated samples pass this test. Cite Err. denotes cited-number error, reported as the number of incorrect cited numbers over the total number of cited numbers. Case denotes the percentage of samples whose comparison-level rationales are supported by history cases. The selected supervisor is *GPT-OSS-120B*.

mild decoding perturbations by varying both sampling temperature and nucleus sampling thresholds, while avoiding overly random generation. We compute prediction stability as the fraction of repeated outputs that agree with the majority prediction:

$$\mathrm{PredStab}(x) = \frac{1}{R} \max_{y} \sum_{r=1}^{R} \mathbf{1}[\hat{y}^{(r)} = y],$$

where $\hat{y}^{(r)}$ denotes the final prediction in the r-th generated output. A higher score indicates that the model makes more stable decisions under decoding variations. Since $R = 6$ yields a discrete stability score, we regard an instance as prediction-stable if at least five out of six generations produce the same final prediction, i.e., $\mathrm{PredStab}(x) \geq 5/6 \approx 0.833$. This threshold allows one minor sampling-induced deviation while filtering out cases where the final prediction changes under multiple decoding configurations.

Prediction–Rationale Consistency. Prior studies have used simulatability as a faithfulness-oriented criterion for evaluating natural-language explanations, where a useful and faithful explanation should help an observer infer the model's output (Hase et al., 2020; Pan et al., 2025). Inspired by this idea, we evaluate whether the generated rationale itself provides sufficient support for the model's final prediction. For each output, we remove the final resolution statement and the explicit prediction line to avoid direct answer leakage, leaving only the intermediate reasoning process as input to the verifier. We then provide this reasoning text to an independent verifier model, GPT-5.5, and ask it to infer the most supported label from the rationale alone. If the verifier-inferred label matches the model's original final prediction, we count the output as prediction–rationale consistent. This metric measures whether the explanation is aligned with and supportive of the final decision, rather than merely appearing alongside it.

Cited-Number Error. We check whether the numerical values cited in the rationales are correct, including both dataset-level statistics and current-sample feature values. For dataset-level evidence, we perform tool-assisted verification by giving the verifier access to the precomputed statistics and checking whether each cited statistic matches the corresponding value in the statistics table. Although less likely, the model may also misquote values from the current input sample; therefore, we additionally verify whether cited feature values in the rationale are consistent with the actual input features. We report cited-number errors as the number of incorrect cited numbers over the total number of cited numbers. This metric directly measures numerical hallucination in cited evidence.

Similar-Case Support Rate. Finally, we evaluate whether comparison-level evidence is grounded in the retrieved similar cases. For rationales that mention similar historical samples or comparison-level patterns, we provide the verifier with access to the corresponding retrieved training-set references. For a sample, if the cited similar-case evidence can be found in the retrieved context, we count it as

supported; otherwise, we count it as unsupported. We report the support rate over all 100 samples.

Experimental Results. Table 13 shows that TLRD-generated rationales are generally stable and evidence-grounded. Across all classification datasets and student models, prediction stability remains high, suggesting that TLRD enables student models to internalize evidence-grounded reasoning patterns rather than relying on unstable or arbitrary generation. Prediction–rationale consistency is close to perfect in most cases, indicating that the rationales usually support the model's own predictions. Numerical hallucination is rare, with only a few incorrect cited numbers among thousands of cited values. Importantly, these errors are occasional rather than systematic: a statistic that is incorrectly cited in one rationale is not consistently cited incorrectly whenever it appears in other rationales. This suggests that the errors are more likely caused by the inherent limitations of LLM generation rather than by a systematic flaw in the proposed method. In addition, the similar-case support rate remains high across datasets, showing that comparison-level claims are usually grounded in the retrieved examples. Overall, these results suggest that TLRD produces rationales that are not only readable, but also stable, internally consistent.

D.3 Human Evaluation.

We conduct a small-scale human study to evaluate the usability of generated rationales. The study involves five annotators, all of whom have a master's-level or higher academic background in computer science. Evaluations are conducted on four classification datasets by randomly sampling 100 data points from the validation set of each dataset. We focus on classification tasks, where discrete decisions provide a clear basis for judging whether a rationale supports the predicted outcome. Regression rationale evaluation is less directly comparable because continuous-valued predictions require task-specific tolerance ranges, numerical calibration criteria, and stronger domain expertise.

Evaluation settings. Annotators rate each rationale on a 5-point scale for each dimension, allowing half-point increments: Understandability, Decision-Support Utility, and Structuredness.

- **Understandability** measures whether the rationale is easy to read and understand.
- **Decision-Support Utility** measures whether the rationale provides useful insights for understanding or verifying the model decision in a practical decision-making setting.
- **Structuredness** measures whether the rationale is well organized and presents the reasoning in a clear and coherent structure.

Rating scale.

- **1**: Very poor quality, where the rationale clearly performs poorly in the evaluated dimension.
- **2**: Poor quality, where the rationale has major weaknesses but may still contain some useful elements.
- **3**: Acceptable quality, where the rationale partially satisfies the evaluated dimension but has clear limitations.
- **4**: Good quality, where the rationale mostly satisfies the evaluated dimension with only minor weaknesses.
- **5**: Excellent quality, where the rationale fully satisfies the evaluated dimension.

Data	Method	Und. ↑	Util. ↑	Struc. ↑
ADULT	*Base*	3.7	3.0	3.2
	Std. Distill.	3.9	3.6	3.8
	TLRD	**4.3**	**4.4**	**4.8**
HC	*Base*	3.2	2.4	2.6
	Std. Distill.	3.8	3.6	3.5
	TLRD	**4.2**	**4.3**	**4.8**
OK	*Base*	3.1	2.6	2.4
	Std. Distill.	3.7	3.6	3.6
	TLRD	**4.1**	**4.2**	**4.6**
D130	*Base*	3.2	2.9	3.0
	Std. Distill.	4.0	3.7	4.0
	TLRD	**4.4**	**4.6**	**4.8**

Table 14: Human evaluation results on classification datasets using *Gemma 3-12B* as the backbone and *GPT-OSS-120B* as the teacher model. *Base*, *Std. Distill.*, and *TLRD* denote the non-finetuned model, standard distillation, and our method, respectively. Und., Util., and Struc. denote understandability, decision-support utility, and structuredness.

Evaluation Results. Table 14 shows that standard distillation improves over the non-finetuned base model in most cases, confirming the benefit of rationale supervision. TLRD further achieves the best scores across all datasets and criteria. This indicates that tri-level rationale distillation produces more readable, useful, and structured explanations for tabular decision support.

E Ablation Study

Before presenting the experiments, we first briefly outline the design motivation of TLRD to help readers better understand our ablation studies. In the domain of LLMs for tabular data, base models

Model	Adult (F1 ↑)	Home Credit (F1 ↑)	Diabetes130US (Macro-F1 ↑)	OkCupid (Macro-F1 ↑)	California (RMSE ↓)	Diamonds (RMSE ↓)
Llama 3.1-8B (Zero-shot)	51.3	15.5	19.7	46.0	3.02×10^7	3.64×10^8
w/o Statistics	58.9	17.9	29.1	50.2	4.38×10^6	4522.6
w/o Comparison	58.6	15.6	21.3	40.2	1.25×10^7	7315.3
Tri-level Reasoning	63.9	17.8	27.1	51.5	1.12×10^7	2193.4
Standard Distillation	62.5	19.9	25.3	46.8	91399.0	1201.7
Evidence-Augmented Distillation	63.1	23.7	24.2	47.4	92592.7	1094.9
TLRD	64.7	25.1	34.9	54.2	86572.7	1531.9
Qwen3-8B (Zero-shot)	57.9	20.4	6.1	53.6	175459.5	2280.9
w/o Statistics	63.6	18.5	36.3	53.3	65895.3	736.6
w/o Comparison	61.4	22.3	29.5	49.9	88828.4	1783.5
Tri-level Reasoning	65.5	26.0	36.6	54.0	65660.6	691.0
Standard Distillation	61.4	23.7	28.0	51.9	79511.6	1317.5
Evidence-Augmented Distillation	53.1	24.0	33.7	47.1	82779.9	1294.7
TLRD	65.7	26.3	33.4	55.6	86341.3	1316.6
Gemma 3-12B (Zero-shot)	55.1	15.0	9.2	50.3	568304.3	2589.5
w/o Statistics	65.3	16.1	32.6	52.1	68114.6	800.0
w/o Comparison	61.1	16.1	24.1	44.2	118604.5	2297.6
Tri-level Reasoning	63.1	18.8	33.2	51.1	70633.5	1044.9
Standard Distillation	63.2	20.0	27.5	44.8	80762.5	1152.1
Evidence-Augmented Distillation	57.7	21.4	34.2	50.3	81151.4	872.1
TLRD	63.4	25.7	34.5	56.1	85646.6	945.9
GPT-OSS-20B (Zero-shot)	57.9	16.6	14.2	49.4	201829.8	2477.1
w/o Statistics	63.4	15.9	36.7	51.6	67616.8	706.0
w/o Comparison	64.9	21.8	36.4	52.9	105755.3	1691.2
Tri-level Reasoning	66.9	24.3	36.0	54.5	75992.7	705.1
Qwen3-Next-80B (Zero-shot)	61.9	15.5	8.9	44.5	333534.9	1289.5
w/o Statistics	62.2	20.6	34.4	51.9	68405.7	730.6
w/o Comparison	62.5	18.7	28.6	50.8	98311.9	1565.6
Tri-level Reasoning	63.5	23.5	30.4	52.6	73383.1	698.4
GPT-OSS-120B (Zero-shot)	58.2	16.2	21.1	50.4	114749.6	2632.1
w/o Statistics	64.3	19.7	37.4	52.0	61736.4	653.1
w/o Comparison	65.0	23.6	38.0	52.6	93657.9	1431.0
Tri-level Reasoning	66.6	25.2	37.3	55.0	62906.6	649.7

Table 15: Ablation study of tri-level reasoning and distillation variants across classification and regression datasets.

typically underperform in zero-shot or few-shot settings. While fine-tuning on labels is a natural solution, it introduces the critical problem of explanation collapse. Therefore, we aim to address these challenges via knowledge distillation from a teacher model.

TLRD provides the teacher with extra evidence and uses a designed schema to guide structured rationale generation. **For the ablation of extra evidence, we conduct the ablation experiments of Tri-level Reasoning**, which removes dataset-level statistics or comparison-level retrieval, showing that the full tri-level evidence is generally the most robust. Then it is natural to ask whether we can inject such evidence into a compact student model through fine-tuning, and that's why we construct the Tri-Level Rationale Schema. The corpus construction process of **TLRD can be viewed as extending standard distillation with two additional factors: additional evidence and the designed schema**. Table 2 of our paper compares standard rationale distillation with TLRD and shows that TLRD achieves better performance. **To further disentangle these two factors, we add a new distillation ablation** where the teacher receives the same additional evidence from dataset-level statistics and comparison-level retrieval as TLRD, but is not constrained by the designed schema. We denote this variant as **Evidence-Augmented Distillation**.

E.1 Ablation of Extra Evidence

Table 15 shows that both partial variants, *w/o Statistics* and *w/o Comparison*, can already improve LLM prediction performance, suggesting that comparison-level context and dataset-level statistics each serve as useful standalone evidence sources. Across classification tasks, the combination of evidence generally delivers the best results, making it the most robust configuration overall. These results indicate that the two components are

generally complementary, with their integration usually leading to the strongest performance. We also provide a more detailed analysis in Section E.3 by examining the outputs of different model variants, hoping to offer useful insights for interested readers and future researchers.

E.2 Ablation of Distillation

Table 15 disentangles the role of extra evidence and the designed schema. Evidence-Augmented Distillation shows that providing the teacher with dataset statistics and retrieved neighbors can already help the student acquire dataset-specific tabular knowledge. However, this improvement is unstable and can even fall below standard distillation, indicating that extra evidence alone does not guarantee useful supervision. Under this setting, the teacher generates explanations such as: "The most similar historical cases (similarity ≥ 0.98) are overwhelmingly labeled non-STEM, reinforcing that this sample falls into the non-STEM class." Such explanations are low-information and potentially noisy supervision signals. In contrast, under the Tri-Level Rationale Schema, we explicitly prohibit referencing similarity scores and require the model to extract feature-combination patterns and critical deviations from similar cases. This encourages the teacher to generate more structured, evidence-grounded, and informative rationales, which can provide higher-quality supervision.

For regression, the results are more mixed. Regression requires fine-grained numerical calibration rather than discrete decision-boundary learning, so retrieved examples may sometimes provide direct numerical anchors that are easier to exploit in a less constrained format. Thus, while TLRD remains effective compared with zero-shot reasoning, compressing precise quantitative mappings into structured rationales and student parameters is still more challenging. This suggests that regression may benefit from additional calibration-oriented schema designs.

E.3 Further Analysis

Beyond these direct quantitative results, we further conduct a more in-depth analysis by examining dataset characteristics and qualitative patterns in model outputs, hoping to bring useful insight for future researchers.

A notable overall trend is that, on GPT-OSS-20B and GPT-OSS-120B, *w/o Comparison* performs better than *w/o Statistics* in nearly all cases, indicating that these models make more effective use of dataset-level summaries alone. This advantage is closely related to their strong thinking ability. Compared with other models, they analyze feature-level statistics more precisely during the thinking process, which allows them to better understand statistical signals and reason from them more effectively. By contrast, for most other backbones, comparison-level context is often more beneficial than statistics alone, suggesting that local case-based evidence is easier to exploit than abstract global summaries. The effectiveness of comparison-level context, however, is highly constrained by retrieval quality. On HOME CREDIT, severe class imbalance makes it difficult to construct stable and sufficiently balanced local neighborhoods, which substantially weakens the *w/o Statistics* variant. In contrast, on DIABETES130US, comparison-level reasoning remains highly informative because the dataset contains many code-like categorical features, such as diagnoses and medications, for which similar patient records provide meaningful local evidence. Moreover, dataset-level statistics capture global marginal trends, while retrieved comparisons emphasize local patterns around the query instance. When these signals focus on different aspects of the data, combining them may not provide additional gains, especially if one source already offers the dominant predictive cue.

For regression, the results are more mixed. Retrieved neighbors may provide useful numerical anchors through concrete target values, helping with scale calibration. However, dataset-level statistics can still help when global ranges or categorical structures are informative. This suggests that regression may require more calibration-oriented designs, rather than directly relying on the same schema used for classification.

F Task-Specific Optimization Directions

TLRD is designed as a general framework for converting label-only tabular datasets into structured rationale supervision. Beyond the default setting used in our main experiments, there are several practical directions that may further improve predictive performance and decision-support utility. These include standard optimization choices such as systematic hyperparameter search, checkpoint selection, and full-parameter supervised fine-tuning. TLRD currently uses direct dataset-level statistics, such as class-conditional means and cat-

Backbone	Method	Home Credit Overhead (ms)	OkCupid Overhead (ms)	Diamonds Overhead (ms)
Llama3-8B	Zero-shot	0.00	0.00	0.00
	w/o Statistics	246.56	28.53	21.96
	w/o Comparison	29.38	4.16	1.61
	Tri-level Reasoning	263.79	35.16	22.99
	TLRD(*GPT-OSS-120B*)	0.00	0.00	0.00
Gemma-3-12B	Zero-shot	0.00	0.00	0.00
	w/o Statistics	252.12	28.83	21.66
	w/o Comparison	32.47	4.72	1.27
	Tri-level Reasoning	284.71	34.22	23.48
	TLRD(*GPT-OSS-120B*)	0.00	0.00	0.00
Qwen3-8B	Zero-shot	0.00	0.00	0.00
	w/o Statistics	246.34	28.70	21.94
	w/o Comparison	38.10	6.22	2.11
	Tri-level Reasoning	273.38	33.28	22.62
	TLRD(*GPT-OSS-120B*)	0.00	0.00	0.00

Table 16: Average prompt-construction overhead before the LLM generation call. Overhead is measured in milliseconds and defined as *Comparison Build (ms)* + *Stats Compute (ms/sample)*. Zero-shot and TLRD incur no test-time overhead because they do not require prompt-time statistics computation or retrieval.

egorical frequencies. A promising extension is to expose higher-order statistical evidence to the teacher model such as the likelihood ratio. For example, conditional likelihood ratios can describe how much additional information a pattern provides after another pattern is already observed, such as whether pattern B remains informative conditioned on pattern A. Carefully selecting and presenting such higher-order evidence may help the teacher identify more discriminative feature interactions. In addition, we explored several task-specific directions that are not included in our experiments but may be useful for future work and for deploying TLRD in specific application scenarios.

Domain-specific reasoning workflow. First, even before fine-tuning, the reasoning behavior of LLMs can be improved a little by adapting the prompt to the task-specific decision process while keeping the input features unchanged. For example, in HOME CREDIT, the goal is to assess whether an applicant may default on a loan. Instead of asking the model to reason in a generic way, we can prompt it to separately analyze *Financial Resilience* and *Behavioral Risk Signals*, which may follow a practical credit-risk assessment workflow. Incorporating this structured approach into rationale construction may yield both better performance and enhanced readability.

Task-specific decision preference. Second, the model behavior can be meaningfully affected by explicitly specifying the downstream decision preference. For instance, in loan-default detection, missing a true default applicant may be more costly than incorrectly flagging a non-default applicant. We find that simply adding an instruction such as "identifying default is more important than avoiding false alarms" can noticeably shift the model behavior toward higher recall and lower precision, which may improve F1 under imbalanced settings. This observation suggests that LLM-based tabular decision systems can be adjusted according to task-specific cost preferences, rather than treating all prediction errors as equally important.

G Inference Efficiency

We further report inference-efficiency statistics for different settings. Specifically, we include three quantities: (1) **overhead**, (2) **input tokens**, and (3) **output tokens**. All measurements are conducted with `vLLM` using a batch size of 1 to simulate an online serving scenario. For each dataset–model configuration, we measure the **overhead** before the actual LLM generation call, i.e., the time required to construct the final prompt for the query instance, including any auxiliary processing such as assembling retrieved context and dataset statistics when applicable. Overhead is reported in milliseconds and averaged over all evaluated queries. In addition, we report the average **input length** (number of prompt tokens fed into the model) and the average **output length** (number of generated tokens in the model response).

Table 16 reports the average prompt-

Backbone	Method	HC		OK		Diamonds	
		InputTok	OutTok	InputTok	OutTok	InputTok	OutTok
Llama3-8B	Zero-shot	484.15	553.67	385.68	610.40	376.36	296.90
	w/o Statistics	3309.39	582.22	2646.93	369.68	1776.80	237.79
	w/o Comparison	2164.45	749.40	3599.93	725.39	2401.88	370.74
	Tri-level Reasoning	4976.69	759.78	5848.18	361.33	3789.32	250.30
	TLRD(*GPT-OSS-120B*)	542.15	1930.00	442.68	1879.42	438.36	1219.98
Gemma-3-12B	Zero-shot	541.17	708.72	392.10	787.09	376.85	645.62
	w/o Statistics	4225.43	446.93	2728.87	389.15	1936.50	346.99
	w/o Comparison	2804.11	834.65	4115.62	849.74	2893.23	487.26
	Tri-level Reasoning	6474.37	505.82	6438.39	349.91	4438.88	506.99
	TLRD(*GPT-OSS-120B*)	607.17	2318.35	458.10	2154.08	446.85	1414.09
Qwen3-8B	Zero-shot	503.08	646.39	366.33	518.79	368.85	399.23
	w/o Statistics	4009.86	658.01	2706.64	701.66	1928.50	571.38
	w/o Comparison	2695.82	902.07	4014.88	1165.16	2828.31	600.08
	Tri-level Reasoning	6189.60	934.62	6342.19	1228.84	4374.96	701.49
	TLRD(*GPT-OSS-120B*)	561.08	2005.50	423.33	1971.29	430.85	1255.31

Table 17: Average input and output token counts for each dataset–model configuration. HC = HOME CREDIT, OK = OKCUPID. Token counts are computed using the tokenizer corresponding to each backbone model.

construction overhead in milliseconds for each dataset–model configuration. In general, the overhead is dominated by comparison-context construction rather than statistics computation, as the *w/o Statistics* setting consistently incurs much higher cost than *w/o Comparison* across all datasets. This gap is especially pronounced on HOME CREDIT, where retrieval-based prompt construction is substantially more expensive than on OKCUPID and DIAMONDS.

Table 17 reports the corresponding average input and output lengths, computed using the tokenizer of each backbone model. We do not separately emphasize end-to-end inference latency here, because in our setting it is largely dominated by generated output length, which is in turn controllable through the length of teacher-generated rationales during corpus construction. We therefore report token statistics mainly to provide a practical view of the prompting cost under different settings. Excluding TLRD, the main pattern is that prompt-augmented inference trades substantially longer inputs for only moderate changes in output length: *w/o Statistics*, *w/o Comparison*, and especially *Tri-level Reasoning* increase the input length by attaching external context at test time, while their generated outputs remain in a similar range or grow only moderately depending on the backbone and dataset. By contrast, TLRD keeps the input length close to zero-shot prompting because it requires no test-time statistics or retrieval, but produces much longer outputs. This is consistent with our design: the student internalizes a richer reasoning style distilled from *GPT-OSS-120B*, shifting more of the reasoning cost from prompt construction to response generation. In practice, this output cost could be reduced when needed by constraining teacher-generated rationales to be more concise.

Method	HC ↑	OK ↑	Dia. ↓
Llama 3.1-8B Label-only FT	28.3	51.3	1100.9
Qwen3-8B Label-only FT	19.0	53.0	1682.1
Gemma 3-12B Label-only FT	28.1	58.7	688.7

Table 18: Results of label-only fine-tuning. HC = HOME CREDIT, OK = OKCUPID, and Dia. = DIAMONDS.

H Label-only Fine-tuning Results

To observe the explanation-collapse phenomenon discussed in prior work (Yang et al., 2025b), we also conduct label-only fine-tuning experiments. Table 18 reports the prediction outcomes of label-only fine-tuned models when they are evaluated in the same format as fine-tuning, i.e., mapping the input directly to the final prediction label without generating any rationale. Although some results seem competitive, they are accompanied by an almost complete loss of explanation ability. When we additionally prompt the label-only fine-tuned model to provide explanations, the model often fails to follow the instruction, producing either malformed outputs or still label-only responses. Even when it

does follow the explanation instruction, the generated rationales are shallow and come with a large performance drop. For instance, after being explicitly asked to explain its decision, *Gemma 3-12B* drops to 49.0 Macro-F1 on OKCUPID, even below its zero-shot performance. In one representative error, the model misclassifies a non-STEM job profile as `student` and explains: "The profile indicates the education level is a 'masters program', which strongly suggests a student. The other features, such as age, sex, location, etc. are not as important for determining the job category in this case." This explanation is low-information and reveals that the model no longer understands the feature semantics: it unreasonably conflates a master's-level educational background with the student job category.

The experimental observations are consistent with prior work, when optimized solely with label supervision, the model is encouraged to minimize the discriminative objective with the shortest valid output, rather than to preserve free-form explanation ability. As a result, LLMs can behave like a discriminative Transformer-style predictor over serialized rows and may achieve performance comparable to some neural tabular models on certain tasks. At the same time, it almost completely loses its ability to generate meaningful explanations, and forcing explanation generation can further degrade prediction performance.

I Explainability of Tree-Based Methods and TLRD

Tree-based methods such as XGBoost can only provide relatively coarse explanation signals, such as feature importance or attribution scores. However, for complex real-world tabular data, these signals cannot provide case-by-case analysis value. Moreover, modern tree ensembles typically consist of thousands of estimators, making case-by-case path tracing practically valueless.

For example, on HOME CREDIT, XGBoost can produce feature importance such as:

ORGANIZATION_TYPE	0.242968
OCCUPATION_TYPE	0.077036
NAME_TYPE_SUITE	0.028893
NAME_INCOME_TYPE	0.026812
WALLSMATERIAL_MODE	0.026786
NAME_EDUCATION_TYPE	0.025851
NAME_HOUSING_TYPE	0.023091
WEEKDAY_APPR_PROCESS_START	0.022930
NAME_FAMILY_STATUS	0.017258
FONDKAPREMONT_MODE	0.016404

Such feature-importance results tell us which columns are generally useful for prediction, but they do not directly explain why a specific applicant is predicted as default or non-default. In contrast, TLRD generates case-specific rationales that connect the current feature values with dataset-level statistical evidence and similar-case contrastive evidence. This allows the explanation to answer questions such as: whether a feature value is statistically unusual, whether it is discriminative between classes, and how the current sample aligns with or deviates from similar historical samples. Below, we show the final resolution from an example TLRD-generated rationale:

> The self-feature analysis flags several high-risk signals: low external scores, missing external scores, short employment tenure, young age, high annuity, and a male laborer with secondary education. Global statistics confirm that the applicant's `EXT_SOURCE_2`, age, employment length, and annuity are closer to the default distributions, while the modest goods price and recent ID update provide only minor offsets. Although most highly similar historical samples with the same categorical profile have a non-default outcome, the key quantitative deviations, including lower `EXT_SOURCE_2`, younger age, and higher annuity, tilt the balance toward higher default risk. Consequently, the overall evidence supports the given prediction of default.

J Efficiency Comparison with Tabular Baselines

Prior work has shown that, in practical tabular prediction settings, tree-based methods such as XGBoost and CatBoost remain highly competitive and efficient (Shwartz-Ziv and Armon, 2022; Grinsztajn et al., 2022). In Table 1, XGBoost and CatBoost remain among the strongest baselines for pure tabular prediction across most datasets. We further conduct efficiency experiments and find that this observation still holds in our setting. All measurements are conducted on a 24GB NVIDIA RTX A5000 GPU. For Qwen3-8B TLRD, we use *GPT-OSS-120B* as the teacher model and vLLM for inference. The elapsed-time values for Qwen3-8B TLRD report inference time only and do not include fine-tuning time. In our experiments, fine-tuning the LLM backbones typically takes around 3–6 hours on 4×80GB A100 GPUs, and inference requires at least a 24GB GPU. For reference, on HOME CREDIT, a fine-tuned Qwen3-8B model supervised by *GPT-OSS-120B* takes a median of 9.71 seconds to generate one complete rationale.

Tables 19 and 20 confirm the strong efficiency advantage of tree-based methods. XGBoost re-

Model	Adult	Home Credit	Diabetes	OkCupid	California	Diamonds
TabPFN(v2.5)	584.10	10198.61	972.57	833.54	58.45	280.38
TabPFN(v2.0)	418.17	12852.28	1598.24	847.72	48.45	359.95
XGBoost	1921.03	7108.51	3216.24	4594.69	3106.04	800.34
CatBoost	1040.01	768.20	1152.22	737.84	1578.13	714.41
TabM	2317.10	18271.61	3314.52	3120.38	3333.91	6620.48
Qwen3-8B TLRD	2390.20	2740.44	2691.00	2388.54	1455.57	2361.72

Table 19: Total elapsed wall-clock time in seconds. Lower is better.

Model	Adult	Home Credit	Diabetes	OkCupid	California	Diamonds
TabPFN(v2.5)	1.34	23.46	5.40	2.63	1.84	6.30
TabPFN(v2.0)	3.34	23.42	12.74	4.37	2.08	7.34
XGBoost	0.33	0.48	0.52	0.79	0.36	0.33
CatBoost	∼22.5	∼22.5	∼22.5	∼22.5	∼22.5	∼22.5
TabM	0.60	1.40	1.34	1.81	1.21	1.21
Qwen3-8B TLRD	∼21.3	∼21.3	∼21.3	∼21.3	∼21.3	∼21.3

Table 20: Peak GPU memory usage in GB. Values marked with ∼ are approximate peak allocations.

mains the most resource-efficient baseline in terms of GPU memory usage across all datasets. On larger datasets HOME CREDIT, XGBoost and CatBoost also achieve faster execution than TabPFN. For CatBoost, the reported GPU memory mainly reflects its default GPU memory reservation behavior. By default, CatBoost uses `gpu_ram_part = 0.95`, so it may reserve a large fraction of available GPU memory after initialization. For Qwen3-8B TLRD, the reported GPU memory also reflects the vLLM configuration. We launch vLLM with `-gpu-memory-utilization 0.9`. Therefore, the reported value should be interpreted as observed peak allocation under the default configuration, rather than the strictly required model memory.

We emphasize that TLRD is not designed to compete with tree-based methods in high-throughput deployment. Instead, TLRD aims to help bridge the gap between deep learning methods and traditional tabular models in tabular decision-making: it improves the ability of LLM-based models to produce competitive predictions while also generating case-specific rationales that offer valuable reference for decision-making. Therefore, tree-based models and TLRD can be used in a complementary workflow. Tree-based models can serve as efficient high-throughput predictors, while TLRD can be applied in latency-tolerant stages to generate case-specific explanations for selected samples. These explanations can support downstream workflows such as assisting data scientists in understanding dataset-specific patterns, auditing individual model decisions, and helping machine learning engineers diagnose or tune tree-based models.

K LLM Usage

In compliance with the ACL Rolling Review (ARR) guidelines regarding AI assistance, we hereby clarify the role of Large Language Models (LLMs) in this work. Since LLMs serve as the primary focus of our empirical investigation, all reported metrics and outputs are derived strictly from our direct execution and evaluation of these models. Furthermore, we employed LLMs solely as text-editing utilities to refine our prose and enhance readability. We confirm that no AI tools were involved in formulating the research hypotheses, designing the framework, or interpreting the scientific findings.