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

Anonymous ACL submission

Abstract

Real-world tabular decision-making faces a fundamental trade-off: strong tabular predictors such as tree-based ensembles achieve high predictive performance but do not naturally produce intuitive, case-specific rationales, while Large Language Models (LLMs) can generate human-readable reasoning yet often suffer from generative degradation when fine-tuned on label-only tabular data. To address this problem, we propose Tri-Level Rationale Distillation (TLRD), a framework that converts label-only tabular datasets into structured rationale supervision for compact LLMs. TLRD uses a high-capacity teacher to synthesize a rationale corpus grounded in three complementary levels of evidence: instance-level feature analysis, dataset-level distributional context, and comparison-level contrastive reasoning from retrieved neighbors. By distilling these structured rationales into a student model, TLRD injects dataset-specific statistical and relational knowledge into the model parameters, enabling the student to generate both predictions and grounded rationales from raw serialized rows alone. Extensive experiments on multiple domain datasets demonstrate that TLRD produces more grounded and auditable explanations and closes the predictive performance gap between LLMs and state-of-the-art tree ensembles, offering a deployable solution for transparent high-stakes decision-making.

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 gradient-boosted decision trees (GBDTs) for their efficiency and strong performance. However, they usually provide only coarse interpretability signals (e.g., feature attribution scores) that often fall short of operational decision needs. However, in practice, stakeholders need concise and auditable rationales that explain (i) which specific feature values in this case are concerning, and (ii) how unusual they are relative to the broader population.

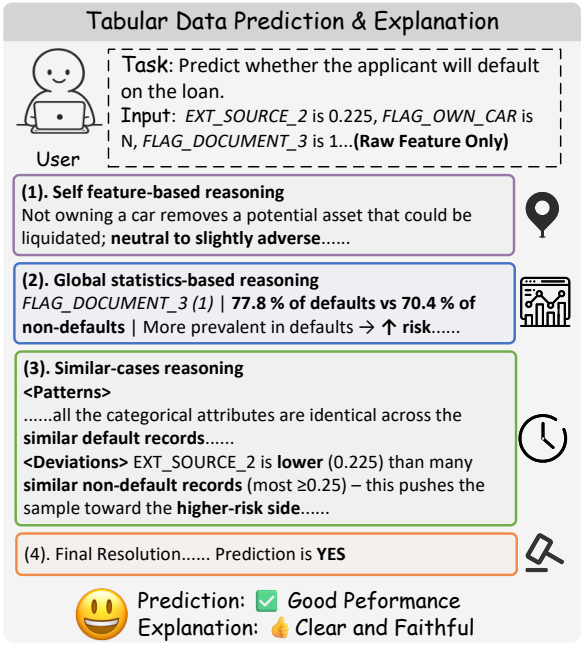

Figure 1: Illustration of TLRD on a sample from the Home Credit dataset. Given only raw input features, the TLRD student model generates tri-level rationales.

LLMs offer a natural promise to bridge this gap by combining predictive capability with human-readable reasoning. However, in zero-shot or few-shot structured prediction, LLMs often exhibit unstable performance and poor accuracy 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 to these limitations 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 after discriminative fine-tuning. 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. TLRD leverages knowledge distillation from a high-capacity teacher model. Specifically, it synthesizes a structured evidence 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 a high-capacity teacher model with a Tri-Level Augmented Input, consisting of tabular features, external statistical metadata, and retrieved neighbors. The teacher is prompted to construct the tri-level explanation, summarizing the contextual insights. A compact student model is then fine-tuned on the distilled corpus to jointly predict the label and generate the rationale from the raw feature row alone. In this way, dataset context and case-based signals are *indirectly injected* into the student through the distilled explanations, while the student remains deployable without external statistics or retrieval, avoiding added latency. Our contributions are summarized as follows:

- We formalize a structured rationale schema that decomposes justification into instance-specific evidence, global distributional statistics, and contrastive pattern summaries.
- We introduce a distillation method that uses dataset statistics and retrieved neighbors only at generation time to produce self-contained rationales for fine-tuning, enabling zero-overhead inference strictly from raw features.
- We evaluate TLRD on multiple decision-critical tabular tasks. Results show that the fine-tuned student model achieves strong predictive performance against tree-based baselines while producing more grounded and useful rationales.

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. 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 architectures 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, relying on massive LLMs for routine inference remains computationally prohibitive and prone to hallucinated reasoning, motivating the development of compact, task-specific models that preserve interpretability.

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 rationale distillation. However, rationale distillation remains largely underexplored for tabular data. Fine-tuning tabular models strictly on target labels risks "explanation collapse," where models optimize for accuracy at the expense of their generative reasoning faculties. Our Tri-Level Rationale Distillation (TLRD) builds upon text-native distillation frame-

works (Hsieh et al., 2023; Magister et al., 2023) but explicitly adapts them to the tabular domain. By enforcing a strict, multi-tiered schema, TLRD ensures the student model retains grounded, auditable reasoning without sacrificing predictive accuracy.

3 Proposed Method

3.1 Problem Formulation

Let $\mathcal{X}$ denote a tabular feature space comprising 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. We are given 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. We denote by $\mathcal{D}_{\text{train}} \subset \mathcal{D}$ the training split of the dataset. Since LLMs are not natively trained on raw tables, we consider a serialization function that converts each row into a textual prompt $p_i = \text{Serialize}(\mathbf{x}_i)$, yielding standard label-only fine-tuning examples $(p_i \rightarrow y_i)$.

Our goal is to fine-tune a language model to output both a prediction and a structured, evidence-grounded rationale, i.e., (y_i, r_i) with $r_i \in \mathcal{R}$ following a fixed schema. Because $\mathcal{D}$ contains labels but not rationales, we seek a mapping that converts label-only tabular data 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). The remainder of this section specifies $\mathcal{R}$ and describes how teacher generation and distillation populate $\mathcal{C}$.

3.2 Tri-Level Augmented Input

To enable the teacher LLM to generate grounded tri-level rationales, we construct a *Tri-Level Augmented Input* that extends the raw serialized row with global statistical context and retrieved comparison context. We first serialize a row $\mathbf{x} \in \mathcal{X}$ into a textual description for the LLM. Each feature is represented as `<feature> is <value>`, using a fixed feature order, explicit missing-value placeholders, and raw numeric values. We refer to this block as *Current Sample Features*. We then augment this block with two additional sources of information: *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 categorical-feature statistics such as class-conditional category frequencies. For regression tasks, we discretize the continuous target into four percentile bins Q_1, Q_2, Q_3, Q_4 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 structurally aligned and label-balanced set of historical instances. To avoid trivial self-matches, 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. 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 tasks, we obtain a balanced context by splitting targets at the global median and applying the same retrieval strategy.

The resulting teacher-side input forms the *Tri-Level Augmented Input*, which combines the raw serialized row with global statistical context and

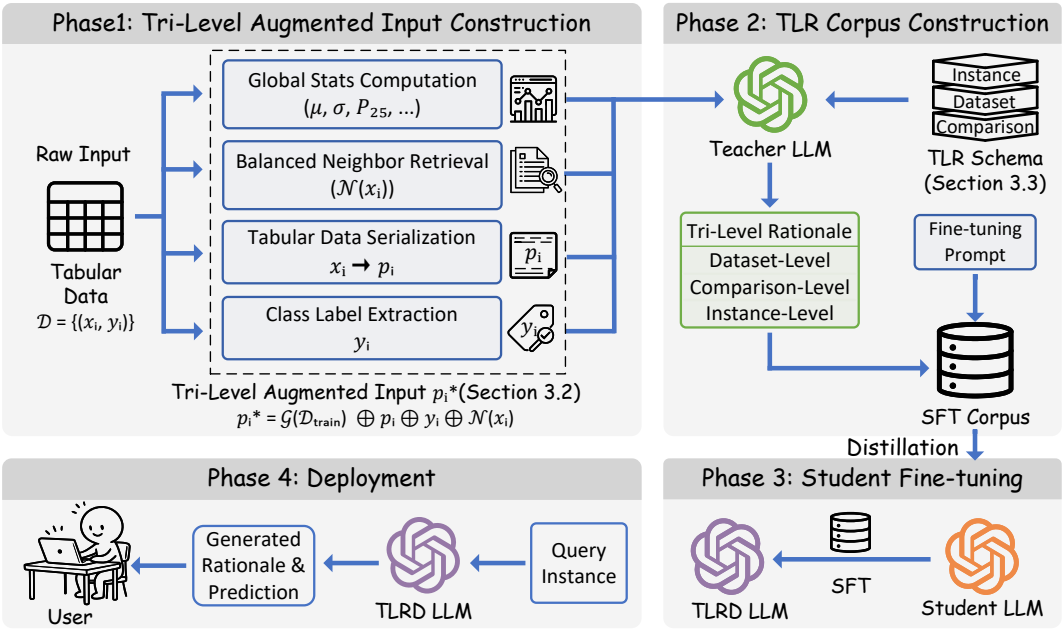

Figure 2: Overview of the TLRD framework. Starting from a label-only tabular dataset, we construct Tri-Level Augmented Inputs for the teacher model. Guided by the Tri-Level Rationale Schema, the teacher generates structured rationales, which are then distilled into a student LLM through supervised fine-tuning.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

retrieved comparison context, while also providing the ground-truth label to the teacher. 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

Given Tri-Level Augmented Input p_i^*, the teacher is prompted to generate a rationale following 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 phase. Let $\mathcal{M}_T$ denote a high-capacity teacher LLM. For each training 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. By providing the ground-truth label, we cast the teacher model as an expert annotator tasked with explaining *why* the given label is plausible, rather than requiring it to predict the outcome from scratch. Conditioned on this augmented input p_i^* and the schema outlined in Section 3.3, $\mathcal{M}_T$ generates the tri-level rationale r_i that 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 phase. The student model receives *only* the raw serialized prompt p_i. The training target is formulated as a unified output sequence formed by concatenating the reasoning trace and the ground-truth label: $[\mathbf{r}_i \oplus y_i]$. We train Θ using the standard causal language modeling objective (autoregressive next-token prediction):

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

where w_t represents the t-th token in the target sequence. The fine-tuning indirectly injects the global distributional knowledge and case-based alignment patterns into the model parameters Θ. Then, during online deployment, the student model operates entirely autonomously without requiring external retrieval databases or statistical computations, making it efficient while preserving the reasoning capabilities of the teacher model.

4 Experiments

We answer the following research questions (RQ):

- **RQ1**: How do base LLMs perform on tabular reasoning and how much can tri-level contextual information improve their performance?
- **RQ2**: How does TLRD improve LLM 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**: Do teacher models need to be larger? How about self-distillation?

4.1 Experiment Setup

Datasets. We evaluate TLRD on six tabular datasets covering both classification and regression. The classification benchmarks are ADULT (Becker and Kohavi, 1996), HOME CREDIT (Montoya et al., 2018), OKCUPID (Kim and Escobedo-Land, 2015), and DIABETES130US (Strack et al., 2014), spanning social, financial, and clinical domains with varying class imbalance and feature heterogeneity. For regression, we use CALIFORNIA HOUSING (Kelley Pace and Barry, 1997) and DIAMONDS (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 spanning a range of model scales from several families, including *Llama 3.1-8B* (Grattafiori et al., 2024), *Gemma 3-12B* (Team et al., 2025), *Qwen3-8B* and *Qwen3-Next-80B-A3B-Instruct* (Yang et al., 2025a), and *GPT-OSS-20B/GPT-OSS-120B* (Agarwal et al., 2025). As non-LLM baselines, we use a diverse set of strong tabular methods spanning tree-based and neural approaches: *XGBoost* and *CatBoost* (gradient-boosted decision trees (GBDTs)), *TabM* (Gorishniy et al., 2024) (a recent deep tabular model), and *TabPFN* (Hollmann et al., 2023) (a pretraining-based tabular predictor).

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 5. Due to limited computation resources, we conduct LoRA fine-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 tri-level contextual information. On base LLMs, we follow *TabLLM* (Xie et al., 2024; Hegselmann et al., 2023; Wolff and Hulsebos, 2025) and use the zero-shot setting as our baseline. We then study an explicit test-time prompting variant, denoted as **Tri-level Reasoning**, in which the query is augmented

Model	Adult ↑	HC ↑	D130 ↑	OK ↑	California ↓	Diamonds ↓
TabPFN(v2.5)	68.9	29.7	34.3	55.9	40231.0	496.0
TabPFN(v2.0)	67.8	24.0	33.3	52.6	39434.4	505.7
XGBoost	72.0	29.8	42.7	58.8	43253.8	527.7
CatBoost	71.3	28.1	42.8	58.9	42654.1	502.6
TabM	70.1	28.0	43.7	58.2	48855.8	519.7
GPT-OSS-20B (Zero-shot)	57.9	16.6	14.2	49.4	201829.8	2477.1
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
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
Tri-level Reasoning	66.6	25.2	37.3	55.0	**62906.6**	**649.7**
Llama 3.1-8B (Zero-shot)	51.3	15.5	19.7	46.0	3.02×10^7	3.64×10^8
Tri-level Reasoning	63.9	17.8	27.1	51.5	1.12×10^7	2193.4
TLRD (GPT-OSS-120B)	64.7	25.1	34.9	54.2	86572.7	1531.9
TLRD (Qwen3-Next-80B)	63.1	22.7	35.7	53.2	112008.0	1453.7
Qwen3-8B (Zero-shot)	57.9	20.4	6.1	53.6	175459.5	2280.9
Tri-level Reasoning	65.5	26.0	36.6	54.0	65660.6	691.0
TLRD (GPT-OSS-120B)	65.7	**26.3**	33.4	55.6	86341.3	1316.6
TLRD (Qwen3-Next-80B)	63.0	22.7	36.9	55.0	91877.5	1326.1
Gemma 3-12B (Zero-shot)	55.1	15.0	9.2	50.3	568304.3	2589.5
Tri-level Reasoning	63.1	18.8	33.2	51.1	70633.5	1044.9
TLRD (GPT-OSS-120B)	63.4	25.7	34.5	**56.1**	85646.6	945.9
TLRD (Qwen3-Next-80B)	64.8	22.3	**38.1**	55.6	96473.5	1060.4

Table 1: Results across classification and regression datasets. HC = HOME CREDIT, D130 = DIABETES130US, and OK = OKCUPID. Bold values denote the best result among LLM-based methods in each column.

with dataset statistics and retrieved neighbors to provide tri-level context at inference time.

Baseline performance. Modern LLMs remain weak and unstable performance in zero-shot. They perform better on ADULT and OK, where some patterns are more aligned with common social semantics. 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. This reflects the difficulty of zero-shot numerical prediction: 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 base LLMs in zero-shot, dataset-level reasoning provides class-conditional 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: rather than relying on such evidence only at inference time, 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 corpus construction to let a teacher model generate evidence-grounded analyses conditioned on the given label, and distills the resulting knowledge and reasoning ability 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* further reaches 38.1 on D130. Compared with zero-shot, these gains are substantial, showing that TLRD can effectively transfer stronger tabular reasoning patterns from powerful teachers into more efficient student models.

On regression, TLRD still improves substantially over zero-shot, but it often remains below Tri-level Reasoning. Regression operates over a wide continuous target range, requiring much finer scale control and numerical calibration than discrete class prediction. Although tri-level rationale distillation provides useful statistical and contrastive guidance, 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 best 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 foundation models.

RQ3: Comparison between TLRD and standard rationale distillation. We compare TLRD with a standard rationale distillation setting. For this comparison, we fix *GPT-OSS-120B* as the teacher model for corpus construction. The standard distillation baseline 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 the tri-level rationale schema provides more effective supervision by organizing the explanation around tri-level evidence.

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 under standard distillation and TLRD. HC = HOME CREDIT, OK = OKCUPID, and Dia. = DIAMONDS.

Rationale grounding. Figure 3 compares two *Gemma 3-12B* variants on the same HC instance. Although the standard distillation model generates a fluent explanation, its reasoning is largely based on generic commonsense assumptions that appear plausible but are not supported by concrete evidence, such as interpreting the absence of car ownership as reduced financial burden. By contrast, the TLRD model structures its explanation into instance-level, dataset-level, and comparison-level reasoning. This yields a more auditable rationale. At the instance level, it provides a reasonable local interpretation, e.g., treating `FLAG_OWN_CAR = N` as neutral to slightly adverse because the applicant does not possess an additional asset. At the dataset level, it explicitly places `AMT_GOODS_PRICE` and `AMT_CREDIT` in relation to class-level statistics, and correctly recognizes that although these values may appear intuitively safe because the loan amount is close to the goods price, they provide limited discriminative value here since the two classes do not differ clearly on these features. It also captures the counterintuitive but dataset-supported pattern that `FLAG_DOCUMENT_3 = 1` is more prevalent among default applicants, and, 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 can combine reasonable local intuition with dataset-specific evidence, rather than relying only on intuitive but potentially misleading narrative associations.

RQ4: Teacher model choices and self distillation. Table 3 compares multiple teacher models across three representative datasets, using both self-distillation and several external teachers.

A first notable observation is that a very large teacher is not always required: in several settings, *GPT-OSS-20B* already provides highly effective supervision. *Qwen3-8B* achieves the best results on both OK (56.3) and DIA. (1135.7). This suggests

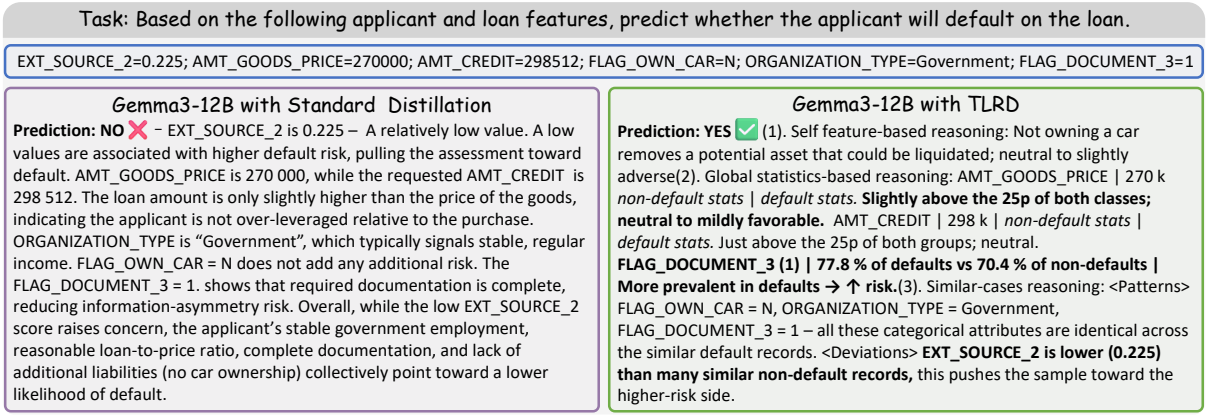

Figure 3: A qualitative case study comparing the same model under standard distillation and TLRD. For a sample with ground-truth label YES, the standard distillation model predicts incorrectly, whereas TLRD predicts correctly and provides a more grounded tri-level rationale.

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 teacher models for each student backbone. HC = HOME CREDIT, OK = OKCUPID, and Dia. = DIAMONDS.

that effective rationale distillation does not depend on scale alone; for some student–task combinations, a mid-sized teacher already produces rationales that are sufficiently strong.

At the same time, the effect of teacher scale is not uniform. On HC, stronger teachers are generally more helpful, especially for students with weaker self-distillation performance. For example, *Llama 3.1-8B* improves from 9.7 to 25.1 with *GPT-OSS-120B* and 25.0 with *GPT-5.2*. This pattern suggests that more difficult tasks requiring stronger dataset-specific statistical grounding may benefit more from stronger external supervision.

A more nuanced trend appears on OK. For *Gemma 3-12B*, performance improves as the teacher becomes stronger. This indicates that for some student–task pairs, larger teachers can provide progressively better supervision. However, self-distillation remains surprisingly competitive: *Gemma 3-12B* with self-distillation reaches 56.0 on OK, very close to the best external-teacher result, and *Qwen3-8B* with self-distillation reaches 24.9 on HC, only moderately below its best result of 26.3.

These results suggest that teacher effectiveness depends not only on model scale, but also on task characteristics, teacher–student compatibility, and the student model's own reasoning and comprehension ability. A student with stronger intrinsic instruction-following, numerical understanding, or domain-relevant reasoning may already benefit substantially from a relatively smaller teacher, whereas more challenging tasks may still require stronger external supervision. Overall, teacher choice in TLRD is better viewed as a task-dependent design decision rather than a simple larger-is-better rule.

5 Conclusion

We proposed TLRD, a framework that converts label-only tabular datasets into structured rationale supervision for small LLMs. By distilling instance-level, dataset-level, and comparison-level reasoning from teacher models, TLRD improves predictive performance, produces more grounded explanations, and preserves efficient inference from raw features alone. Experiments across multiple tabular benchmarks show that TLRD consistently strengthens LLM-based tabular prediction and narrows the gap to strong conventional baselines. Overall, our results highlight structured rationale distillation as an effective path toward transparent and practical LLMs for tabular decision-making.

Limitations

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

- **Distillation of Spurious Correlations.** Our pipeline heavily depends on the generation quality of the high-capacity teacher model. Although we condition the teacher on the ground-truth label to anchor the rationale to the correct outcome, the teacher may still generate spurious correlations, subtle logical leaps, or plausible-sounding but factually incorrect macroeconomic justifications. Because the student model is trained to mimic this corpus, any undetected reasoning artifacts in the teacher's output will be implicitly distilled into the student's parametric memory.
- **Scalability to High-Dimensional Features and Classes.** For datasets with a massive number of columns, we must rely on feature selection (e.g., retaining only the top-ranked features) to construct concise natural language prompts. While this prevents context overflow, it may cause the model to ignore the cumulative predictive power of long-tail variables or complex weak interactions. Second, our dataset-level reasoning schema requires appending global statistics for each target class. When the number of classes is very large, providing these per-class statistical benchmarks can lead to a very long input and dilute the attention mechanism. However, in real-world production environments, binary classification remains an exceptionally common scenario, making TLRD still a highly practical and immediately applicable to most tabular tasks.
- **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.

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. In the current stage, TLRD should be deployed strictly as a decision-support tool, not as an autonomous decision-maker.

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. 2025. Tabarena: A living benchmark for machine learning on tabular data. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*.

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. 2024. Tabm: Advancing tabular deep learning with parameter-efficient ensembling. *arXiv preprint arXiv:2410.24210*.

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.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2023. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, pages 5549–5581. PMLR.

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, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

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.

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

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*.

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 Housing	Regression	20,640	8	1	–	–	CC0: Public Domain
Diamonds	Regression	53,940	6	3	–	–	CC BY 4.0

Table 4: 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 4.

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., 2024) (deep tabular model), and TabPFN (Hollmann et al., 2023) (pretraining-based 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., 2025). 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 5. To address class imbalance, we apply fixed loss weighting; implementation details are provided in Table 6.

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 5, and class-imbalance handling is reported in Table 6. 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 7.

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 5: 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 6: 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 8, template (v) is shown in Table 9, template (vi) is shown in Table 10, and template (vii) is shown in Table 11.

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 7: 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 8: 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 9: Prompt template for standard fine-tuning (FT) 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 10: 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 11: Prompt template for TLRD model inference.

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
TLRD (GPT-OSS-120B)	64.7	25.1	34.9	54.2	86572.7	1531.9
TLRD (Qwen3-Next-80B)	63.1	22.7	35.7	53.2	112008.0	1453.7
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
TLRD (GPT-OSS-120B)	65.7	26.3	33.4	55.6	86341.3	1316.6
TLRD (Qwen3-Next-80B)	63.0	22.7	36.9	55.0	91877.5	1326.1
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
TLRD (GPT-OSS-120B)	63.4	25.7	34.5	56.1	85646.6	945.9
TLRD (Qwen3-Next-80B)	64.8	22.3	38.1	55.6	96473.5	1060.4
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 12: Ablation study of Tri-level Reasoning across classification and regression datasets.

D Ablation study of Tri-level Reasoning

Table 12 shows that both partial variants, *w/o Statistics* and *w/o Comparison*, can already improve LLM prediction performance in many settings, suggesting that comparison-level context and dataset-level statistics each serve as useful standalone evidence sources. Across the 24 backbone–dataset combinations on classification tasks, the full Tri-level formulation delivers the best results in 18 cases, 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. Beyond these direct quantitative results, we further conduct a more in-depth analysis by examining dataset characteristics and qualitative patterns in model outputs. A closer examination of the ablation results suggests that, when used independently, neither dataset-level statistics nor comparison-level context is uniformly superior across all settings. Instead, their relative utility depends strongly on the specific backbone and dataset combination. 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

Backbone	Method	HC Overhead (ms)	OK 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 13: Average prompt-construction overhead before the LLM generation call. Overhead is measured in milliseconds and defined as *Comparison Build (ms)* + *Stats Compute (ms/sample)*. HC = HOME CREDIT, OK = OKCUPID. Zero-shot and TLRD incur no test-time overhead because they do not require prompt-time statistics computation or retrieval.

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.

At the same time, the occasional underperformance of the full Tri-level Reasoning suggests that, in some datasets, statistical evidence and comparison-based evidence do not always align neatly. It reflects the fact that the underlying data may contain predictive patterns at different granularities. Dataset-level statistics summarize marginal class-level tendencies across the dataset, while retrieved comparisons emphasize the local patterns surrounding a particular query instance. This becomes particularly evident when features exhibit complex interactions, under which global trends do not always coincide with the local structure around a query. In such cases, combining the two sources does not necessarily produce additional gains, especially when one source already captures the dominant predictive signal.

For the regression tasks, the role of comparison-level context is more direct. Unlike classification, retrieved comparisons in regression provide not only similar feature patterns but also concrete reference target values, which can serve as highly informative local anchors for prediction. This is particularly clear on CALIFORNIA, where the features are almost entirely numerical except for one categorical attribute. On this dataset, the full Tri-level Reasoning remains very close to *w/o Statistics*, suggesting that comparison-level context already contributes most of the useful information. This is consistent with the nature of numerical regression, where nearby instances with similar feature values often provide strong guidance for estimating the target. By contrast, DIAMONDS contains both continuous measurements and structured categorical descriptors such as cut and color, making dataset-level statistics more valuable in addition to local comparisons. These results suggest that, in regression, comparison-level context is especially useful when prediction can benefit from retrieved instances with similar feature configurations and nearby target values, while dataset-level statistics can provide extra help when broader structural patterns in the data also matter.

E Inference Efficiency

We further report inference-efficiency statistics for different prompting and fine-tuning 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

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 14: 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.

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 13 reports the average prompt-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 14 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 15: Appendix results of label-only fine-tuning. HC = HOME CREDIT, OK = OKCUPID STEM, and Dia. = DIAMONDS.

F Label-only Fine-tuning Results

Label-only fine-tuning. Table 15 reports the results of label-only fine-tuning, where the model is supervised only with the target label and trained to output the final prediction without rationale supervision. Interestingly, this setting can al-

ready yield competitive predictive performance on some datasets. For example, on HOME CREDIT, *Llama 3.1-8B* and *Gemma 3-12B* reach 28.3 and 28.1 F1, respectively, which are already close to strong tabular baselines reported in the main results. On OKCUPID STEM, *Gemma 3-12B* reaches 58.7 Macro-F1, again approaching the performance range of specialized tabular predictors. This suggests that, under label-only supervision, a modern LLM can behave like a discriminative Transformer-style predictor over serialized rows, and thus may show performance comparable to some deep or neural tabular models on certain tasks.

However, these results should be interpreted together with the generative degradation discussed in Section 1. When optimized solely with label supervision (e.g., cross-entropy on the final target token sequence), 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, label-only fine-tuning may improve prediction while simultaneously biasing the model toward terse label outputs, leading to catastrophic forgetting of generative behavior or explanation collapse. This observation motivates TLRD: rather than training the model only to map serialized rows to labels, we distill structured rationales together with the target prediction, so that predictive performance can be improved without discarding grounded explanation ability.

G 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.