TabSieve: Explicit In-Table Evidence Selection for Tabular Prediction

Yongyao Wang[1*], **Ziqi Miao**[1*], **Lu Yang**[2], **Haonan Jia**[3]
Wenting Yan[4], **Chen Qian**[5], **Lijun Li**[1†]
[1] Shanghai Artificial Intelligence Laboratory
[2] The Hong Kong Polytechnic University
[3] Beihang University
[4] Zhejiang University
[5] Renmin University of China
{lilijun}@pjlab.org.cn

Abstract

Tabular prediction can benefit from in-table rows as few-shot evidence, yet existing tabular models typically perform instance-wise inference and LLM-based prompting is often brittle. Models do not consistently leverage relevant rows, and noisy context can degrade performance. To address this challenge, we propose TabSieve, a select-then-predict framework that makes evidence usage explicit and auditable. Given a table and a query row, TabSieve first selects a small set of informative rows as evidence and then predicts the missing target conditioned on the selected evidence. To enable this capability, we construct TabSieve-SFT-40K by synthesizing high-quality reasoning trajectories from 331 real tables using a strong teacher model with strict filtering. Furthermore, we introduce TAB-GRPO, a reinforcement learning recipe that jointly optimizes evidence selection and prediction correctness with separate rewards, and stabilizes mixed regression and classification training via dynamic task-advantage balancing. Experiments on a held-out benchmark of 75 classification and 52 regression tables show that TabSieve consistently improves performance across shot budgets, with average gains of 2.92% on classification and 4.56% on regression over the second-best baseline. Further analysis indicates that TabSieve concentrates more attention on the selected evidence, which improves robustness to noisy context.

1 Introduction

Tabular analysis serves as the essential diagnostic foundation for understanding structured data in applications such as climate science, healthcare, finance and energy (Cachay et al., 2021; Liu et al., 2025b; Addo et al., 2018; Colverd et al., 2025), while tabular prediction transforms those historical insights into actionable foresight to drive high-stakes decision-making. Unlike isolated instance prediction, real-world tables typically contain other rows that are statistically correlated with the target row. Therefore, in-context learning (ICL) in tabular prediction is vital to achieve robust generalization and stable performance.

However, most existing methods still perform instance-wise inference (Arik and Pfister, 2020; Huang et al., 2020; Gorishniy et al., 2023). This design becomes brittle when transferring across tables whose schemas, column semantics, or data distributions differ. Although recent foundation models incorporate table context, they remain constrained by limited reasoning capabilities for textual metadata and high sensitivity to context composition. Conversely, Large Language Models (LLMs) exhibit strong reasoning and in-context learning capabilities (Brown et al., 2020; Guo et al., 2025) to cast tabular prediction as language modeling by serializing headers and rows into natural language (Nam et al., 2024; Hegselmann et al., 2023; Wen et al., 2024). Naively concatenating a few-shot set into the prompt does not reliably yield controllable gains in practice. Two risks are particularly salient. (i) The model may not reliably condition its predictions on the provided context during inference, particularly for non-reasoning models (Cheng et al., 2026; Sun et al., 2025). (ii) Recent studies suggest that many failures of strong LLMs arise from ignoring contextual details or incorrectly applying the context (Dou et al., 2026). In realistic deployments, available demonstrations are often noisy or weakly relevant. The lack of a mechanism to select informative rows can cause the model to rely on unhelpful evidence, resulting in degraded accuracy and robustness (Gao et al., 2024).

To address these issues, we propose TabSieve, a *select-then-predict* reasoning framework for tabular prediction. As shown in Figure 1, given a table and a query row, TabSieve first analyzes the column semantics to form an initial hypothesis about the relationship between input features and the target.

[*]Equal contribution.
[†]Corresponding author.

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 28555–28574
October 24-29, 2026 ©2026 Association for Computational Linguistics

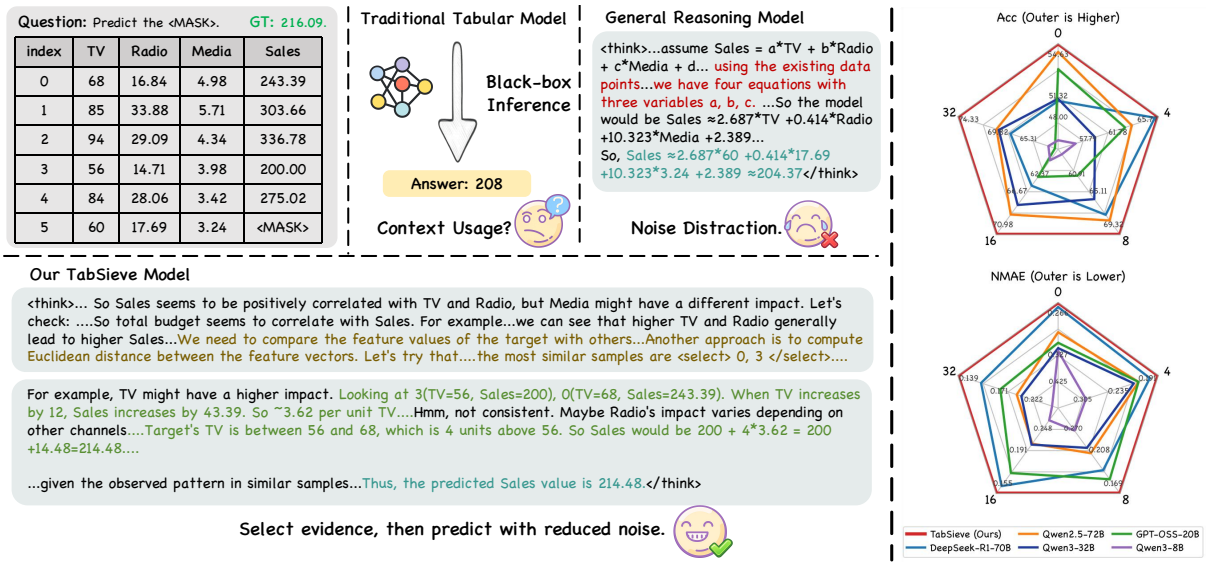

Figure 1: We contrast three prediction paradigms. Traditional models cannot explicitly interpret how context is used. Strong LLMs with in-context prompting can be distracted by noisy context. TabSieve first performs evidence selection and then conducts noise-filtered reasoning to produce the final prediction.

It then explicitly selects a small set of evidential rows that are most informative for the query and finally predicts the missing value. Compared to passively appending demonstrations, TabSieve turns row selection into a constrained intermediate decision, enabling the model to prioritize high-value evidence under a limited context budget and reducing sensitivity to noisy rows.

To equip the model with these capabilities, we build a rigorous data synthesis pipeline that converts 331 tables from existing collections (Wen et al., 2024; Yan et al., 2024) into a large-scale set of training instances with intermediate reasoning traces. Specifically, we construct a teacher-driven workflow that performs structure understanding and feature analysis, explicitly selects effective reference rows, and then completes the final prediction. By applying strict filtering and rejection sampling to retain only high-quality trajectories, we construct a cold-start supervised dataset TabSieve-SFT-40K that provides direct supervision for both evidence selection and prediction.

Building on this dataset, we further introduce TAB-GRPO for reinforcement learning (RL) by designing separate rewards for evidence selection and prediction to jointly improve selection quality and answer accuracy. In addition, a practical complication in tabular prediction is the coexistence of classification and regression objectives. We observe that in early training, regression tasks often exhibit larger advantage scales, which can dominate the optimization direction and induce early-stage bias. To mitigate this effect, we propose a task-advantage balancing mechanism to adaptively rescale the regression advantages. This rescaling dampens overly strong updates driven by regression at the beginning of training, leading to more stable joint optimization. Our contributions are summarized as follows:

- We propose TabSieve, a reasoning-based tabular predictor that explicitly selects in-table evidence and uses it to support prediction. We also construct TabSieve-SFT-40K to provide high-quality supervision.
- We introduce TAB-GRPO, a dynamic task-advantage balancing strategy, to mitigate early-stage optimization imbalance between classification and regression.
- We evaluate TabSieve on 75 classification tables and 52 regression tables. TabSieve consistently outperforms baselines, yielding average gains of 2.92% on classification and 4.56% on regression over the second-best method.

2 Diagnosing the Need to Optimize Tabular In-Context Learning

Previous studies have investigated the role of attention in hallucination and grounding failures, suggesting that insufficient attention allocation to evidential context is a potential factor behind ungrounded outputs (Huang et al., 2024; Huo et al., 2025; Liu et al., 2025a). At the same time, several

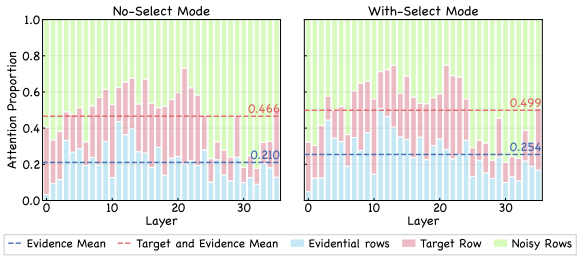

Figure 2: Explicit evidence selection concentrates attention on informative rows.

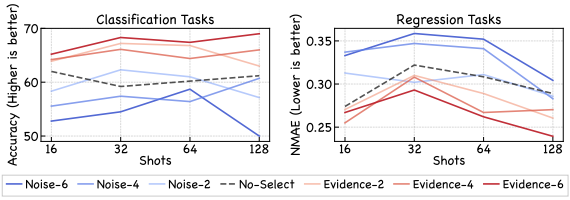

Figure 3: Effect of explicit row selection under evidential or noisy in-table context.

studies indicate that tokens with higher attention weights tend to have a stronger influence on the final decision (DeYoung et al., 2020; Eilertsen et al., 2025). Motivated by these observations, in this section, we use attention-based analysis to examine how general LLMs utilize tabular context during prediction. Specifically, we center our study on two research questions:

RQ1: Can introducing an explicit evidence selection declaration in the reasoning trace steer the model's attention toward evidential rows?

RQ2: If the model is guided to emphasize noisy context, can the context become actively misleading and drive the model to incorrect answers?

2.1 Evidence Focus Improves with an Explicit Selection Trace

Prior work (Thomas et al., 2024) shows that training tabular models with the k-nearest neighbours of a target row can improve accuracy. Inspired by this observation, we categorize the table rows into target row, evidential rows, and noisy rows. The target row is the query instance. Evidential rows are defined as the rows with high embedding similarity to the target row, while the remaining context rows are treated as noise. We then conduct a comparative analysis of the attention allocation of Qwen3-8B (Yang et al., 2025) to these token groups under two inference modes. In the With-Select mode, we insert a short `<select>` block into the reasoning trace, listing the indices of the evidential rows that the model will use for prediction. In the No-Select mode, the model performs standard inference and outputs the answer directly.

An explicit `<select>` declaration shifts attention toward evidential rows and away from noisy context. Figure 2 shows the layer-wise attention proportions over table tokens under the two inference modes. After inserting `<select>`, the model allocates more attention to evidential rows and less attention to noisy rows. This suggests that explicit selection can serve as an effective intermediate scaffold for guiding evidence-grounded prediction.

2.2 Noisy Context Can Be Actively Misleading Under In-Context Learning

We further examine whether incorrect evidence selection can actively mislead tabular prediction, rather than being merely unhelpful. For each shot setting, we sample 80 prediction tasks. Within each task, we label in-context rows as evidence or noise using the same embedding-similarity criterion. We then compare three inference settings that differ only in the inserted selection trace: (i) No-Select, which performs standard inference without an explicit selection trace; (ii) Evidence-S, which inserts a `<select>` trace listing S evidential rows; and (iii) Noise-S, which inserts the same trace but lists S rows sampled from the noisy set.

Selecting noisy rows degrades performance, showing that incorrect evidence focus can actively mislead prediction. As shown in Figure 3, Evidence-S generally outperforms No-Select, whereas Noise-S hurts performance, with the gap typically widening as S increases. These results highlight the importance of explicitly optimizing the model's ability to identify valid evidence, as attending to noisy rows can bias inference and reduce prediction quality. At the same time, the gains from Evidence-S further suggest that the embedding-based criterion provides a useful proxy for rows that improve prediction.

3 Method

The above analysis shows that tabular ICL exhibits weak evidence focus by default, while inserting an explicit `<select>` trace can shift attention from noisy rows toward evidential rows. However, if the model mistakenly treats noisy rows as evidence, prediction performance degrades. To address these issues, we propose a training strategy to train TAB-SIEVE, as illustrated in Figure 4. We first conduct cold-start initialization on TabSieve-SFT-40K, a synthesized dataset of select-then-predict trajectories. We then perform RL with TAB-GRPO to

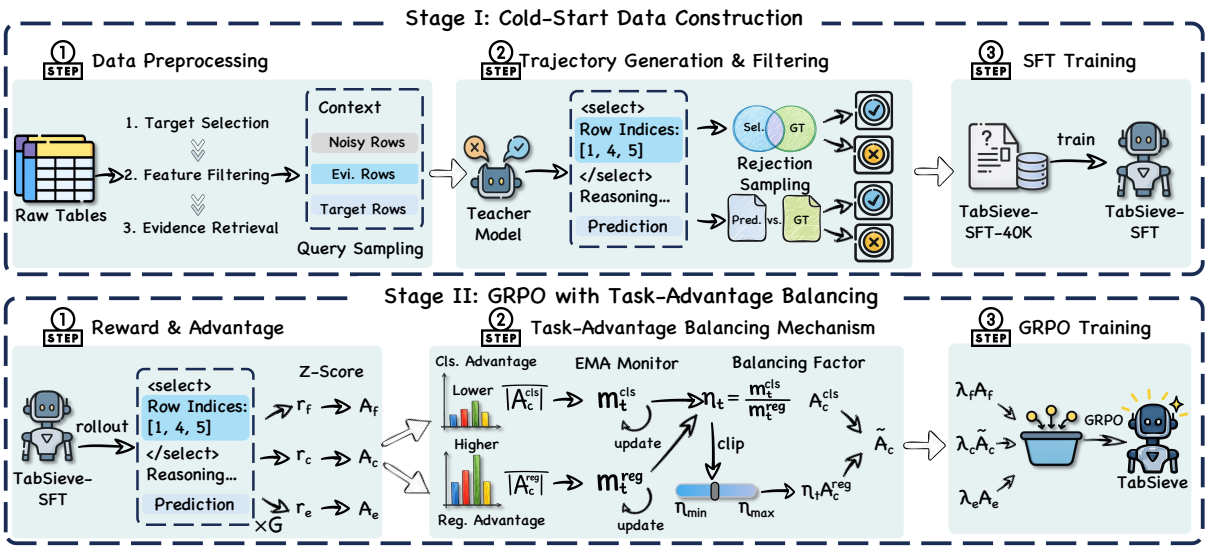

Figure 4: Training pipeline of TabSieve. We synthesize select-then-predict trajectories from a teacher model to build SFT dataset. Starting from the SFT-initialized model, TAB-GRPO extends GRPO with task-advantage balancing to mitigate optimization imbalance and strengthen context selection for robust in-context learning.

jointly refine evidence selection and improve prediction accuracy. To stabilize early-stage optimization, TAB-GRPO balances task advantages across regression and classification objectives.

3.1 Preliminary

Tabular Prediction Tasks. We consider a labeled tabular dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with N instances and d feature columns. Each row $x_i = (x_{i1}, \ldots, x_{id})$ lies in the instance space $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ and $y_i \in \mathcal{Y}$ is the target. For C-class classification, $\mathcal{Y} = \{0, 1, \ldots, C-1\}$; for regression, $\mathcal{Y} \subseteq \mathbb{R}$. The tabular prediction problem is to learn a predictor $f_\theta : \mathcal{X} \to \mathcal{Y}$ such that, given a query row $x \in \mathcal{X}$, the model outputs $\hat{y} = f_\theta(x)$.

In-Context Learning for Tabular Data. In the ICL setting, we test on a new tabular task with dataset $\mathcal{D}'$. Besides a query instance $x^{(q)} \in \mathcal{X}'$, the model is provided with a context set of K labeled examples $\mathcal{C} = \{(x^{(k)}, y^{(k)})\}_{k=1}^{K} \subseteq \mathcal{X}' \times \mathcal{Y}'$, typically drawn from the same table as the query. The model predicts by conditioning on the context $\hat{y}^{(q)} = f_{\theta'}(x^{(q)}, \mathcal{C})$. Equivalently, an ICL predictor implements a mapping $f_{\theta'} : (\mathcal{X}' \times \mathcal{Y}')^K \times \mathcal{X}' \to \mathcal{Y}'$ and adapts to $\mathcal{D}'$ without updating parameters.

3.2 Cold-Start Data Construction

To bootstrap TabSieve with both evidence selection and prediction capability, we construct a cold-start supervised dataset, TabSieve-SFT-40K, which provides explicit signals for the select-then-predict process. Since existing tabular prediction datasets lack chain-of-thought reasoning traces, we build a reasoning corpus from 331 raw tables collected by previous methods (Wen et al., 2024; Yan et al., 2024). Specifically, we first preprocess them into task instances, and then apply a teacher-driven workflow to synthesize select-then-predict trajectories.

Data Preprocessing. We convert raw tables into task instances through three steps. (i) *Target selection.* We prompt a teacher to evaluate each column as a candidate target based on its name semantics and inferability from the remaining columns with self-consistency. This yields reverse reasoning tasks that map original targets back to the features, improving data diversity (Chen et al., 2025). (ii) *Weak correlation feature filtering.* We first remove identifier columns, then rank the remaining features by mutual information with the target and retain the minimal feature subset whose cumulative mutual information exceeds 90%, thereby shortening prompt length and improving sample efficiency. (iii) *Evidence retrieval.* Motivated by the observations in Section 2.2 and findings in prior work (Thomas et al., 2024), we remove the target column and retrieve the top-32 nearest neighbors of the query row as the evidence pool using the cosine similarity of row embeddings. We then randomly draw $K \in \{0, 4, 8, 16, 32\}$ rows from the evidence pool and the full table as the in-context support set, while keeping the evidence ratio within $[0, 0.5]$ to prevent the task from becoming trivially easy.

Trajectory Generation and Filtering. Given a preprocessed task, a teacher model synthesizes a select-then-predict trajectory through two steps. (i) *Evidence selection.* We ask the teacher to analyze the feature relationships in the table and select E evidential rows among the K candidates. We then prompt the teacher to rewrite the selection trace, removing any statements that indicate prior knowledge of E, so that the final trajectory does not leak this information. (ii) *Target prediction.* Starting from the accepted selection trace, the teacher performs up to five attempts at predicting the target value. For classification tasks, we keep only samples whose predictions exactly match the ground truth. For regression tasks, we accept predictions with $\mathrm{MAPE} < 25\%$ and further prompt the teacher to refine its reasoning so that the final value matches the ground truth exactly. Finally, we remove trajectories with abnormal length and apply an LLM-as-judge to filter out traces with logical errors, implausible reasoning, and potential answer leakage. Distribution of the resulting TabSieve-SFT-40K is detailed in Appendix A.2.

3.3 GRPO with Task-Advantage Balancing

We further strengthen TabSieve using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to jointly improve evidence selection and target prediction. In addition, we introduce a task-advantage balancing mechanism to alleviate the early-training imbalance where regression tends to dominate gradient updates, thereby stabilizing joint learning over classification and regression.

Reward Design. Each training instance is a quadruple $(x, y, E^\star, \tau)$, where x is the prompt, y is the ground-truth target, $E^\star$ is the pseudo-evidence set, and $\tau \in \{\mathrm{cls}, \mathrm{reg}\}$ indicates the task type. Given a sampled output o, we define three rewards. The *format reward* $r_f \in \{0, 1\}$ verifies whether o contains a valid `<select>` block and a `\boxed{·}` answer. The *evidence reward* r_e is the F_1 score between the selected row indices $\hat{E}$ parsed from `<select>` and the pseudo set $E^\star$. The *correctness reward* r_c evaluates prediction quality in a task-dependent manner. For classification, we use exact match between the prediction and the ground truth, i.e., $r_c = \mathbf{1}[\hat{y} = y]$. For regression, we apply an exponential shaping on the mean absolute percentage error, i.e., $r_c = \exp(-\gamma \cdot \mathrm{MAPE}(\hat{y}, y))$ with $\mathrm{MAPE}(\hat{y}, y) = |\hat{y} - y|/|y|$.

Advantage Computation. For each prompt x, we sample a group of G outputs $\{o_i\}_{i=1}^G$ from the behavior policy π_{θ^-} and, for each reward component $k \in \{f, e, c\}$, compute a group-relative advantage by whitening rewards within the group:

$$A_i^{(k)} = \frac{r_i^{(k)} - \mathrm{mean}\big(\{r_j^{(k)}\}_{j=1}^G\big)}{\mathrm{std}\big(\{r_j^{(k)}\}_{j=1}^G\big)}. \tag{1}$$

Task-Advantage Balancing Mechanism. *Motivation.* In mixed-task training, regression samples exhibit a larger correctness-advantage magnitude $|A^{(c)}|$ than classification samples in the early stage. Since GRPO weights policy updates by advantages, this imbalance lets regression dominate the update direction and may prematurely bias the model toward a regression local optimum before robust cross-task behaviors emerge.

Balancing factor. To mitigate this, we track the task-wise advantage scale via exponential moving averages (EMA) of $|A^{(c)}|$ with decay $\beta \in (0, 1)$:

$$m_t^\tau = \beta\, m_{t-1}^\tau + (1 - \beta)\, \mathbb{E}\Big[|A^{(c)}| \,\Big|\, \tau\Big]. \tag{2}$$

We then construct a bounded balancing factor that only rescales regression advantages:

$$\eta(\tau) = \begin{cases} 1, & \tau = \mathrm{cls}, \\ \mathrm{clip}\big(\frac{m_t^{\mathrm{cls}}}{m_t^{\mathrm{reg}}},\, \eta_{\min},\, \eta_{\max}\big), & \tau = \mathrm{reg}, \end{cases} \tag{3}$$

and apply it to obtain the balanced correctness advantage $\widetilde{A}_i^{(c)} = \eta(\tau)\, A_i^{(c)}$, where $0 < \eta_{\min} \leq \eta_{\max} \leq 1$. This dampens overly strong regression updates and narrows the optimization gap between the two task types.

GRPO Objective. We aggregate the three component advantages into a single advantage $A_i = \lambda_f A_i^{(f)} + \lambda_e A_i^{(e)} + \lambda_c \widetilde{A}_i^{(c)}$, where $\lambda_f, \lambda_e, \lambda_c$ control their relative importance. Following GRPO, we optimize:

$$\max_\theta \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G} \min\Big(\frac{\pi_\theta(o_i \mid x)}{\pi_{\theta^-}(o_i \mid x)} A_i, \\ \mathrm{clip}\big(\frac{\pi_\theta(o_i \mid x)}{\pi_{\theta^-}(o_i \mid x)}, 1 - \epsilon_{low}, 1 + \epsilon_{high}\big) A_i\Big)\Bigg], \tag{4}$$

where ϵ_{low} and ϵ_{high} are the clipping thresholds.

Model	Classification (Accuracy ↑)							Regression (NMAE ↓)						
	0	4	8	16	32	Rank_z	Rank_i	0	4	8	16	32	Rank_z	Rank_i
Traditional Tabular Models														
XGBoost (Chen and Guestrin, 2016)	–	49.93	51.79	58.38	62.98	–	10.63	–	0.237	0.200	0.184	0.160	–	8.23
CatBoost (Prokhorenkova et al., 2019)	–	51.25	56.42	65.88	72.64	–	8.95	–	0.235	0.192	0.179	0.159	–	7.56
FT-Transformer (Gorishniy et al., 2023)	–	54.31	59.74	68.60	71.45	–	8.37	–	0.363	0.363	0.357	0.354	–	11.56
TabPFN (Hollmann et al., 2023)	–	52.68	60.70	69.15	73.67	–	8.09	–	0.228	0.186	0.165	0.144	–	6.70
General Large Language Models														
Qwen2.5-72B-Instruct (Team, 2024)	54.07	63.67	67.95	68.98	70.87	6.00	6.12	0.297	0.206	0.205	0.198	0.178	5.19	7.15
DeepSeek-R1-70B (Guo et al., 2025)	51.04	65.52	67.39	66.06	69.62	6.37	6.76	0.270	0.197	0.188	0.160	0.152	4.81	6.40
QwQ-32B (Team, 2025)	52.07	64.11	62.46	65.62	67.07	6.59	8.28	0.293	0.220	0.196	0.164	0.158	4.90	6.61
Qwen3-32B (Yang et al., 2025)	51.18	60.74	65.84	68.01	70.67	6.16	6.17	0.317	0.210	0.211	0.197	0.182	6.52	8.33
GPT-OSS-20B (OpenAI et al., 2025)	53.05	63.14	63.53	65.17	65.59	6.39	8.27	0.310	0.206	0.180	0.170	0.165	5.33	7.21
GLM-4-32B-0414 (GLM et al., 2024)	54.06	63.73	66.45	68.21	68.99	6.27	6.93	0.303	0.214	0.182	0.182	0.182	6.71	7.76
Qwen3-8B (Yang et al., 2025)	48.18	58.77	60.91	63.01	65.67	7.44	8.56	0.325	0.268	0.239	0.235	0.222	7.14	9.10
Llama-3.1-8B (Grattafiori et al., 2024)	38.32	46.77	48.88	48.02	47.50	10.08	13.51	0.567	0.351	0.338	0.331	0.319	12.14	14.07
Mistral-7B-Instruct-v0.3 (Choi et al., 2024)	45.94	57.18	55.78	54.37	53.82	7.37	10.51	0.668	0.416	0.405	0.370	0.365	12.67	14.57
Tabular Large Language Models														
Tabula-8B (Gardner et al., 2024)	39.92	41.77	41.62	44.13	42.70	9.84	14.45	0.648	0.420	0.395	0.382	0.445	13.00	15.63
TableLLM-13B (Zhang et al., 2025a)	30.09	40.20	41.60	39.74	33.31	10.44	13.02	0.709	0.476	0.581	0.571	0.541	12.43	15.94
TableGP2-7B (Su et al., 2024)	36.29	41.84	43.62	45.99	47.93	10.76	14.03	0.511	0.504	0.473	0.477	0.508	10.95	16.68
P2T (Qwen3-8B) (Nam et al., 2024)	48.72	59.26	61.87	63.46	66.34	7.80	8.36	0.315	0.259	0.231	0.225	0.215	6.43	9.25
GTL-13B (Wen et al., 2024)	48.62	60.35	62.53	64.27	67.15	7.33	7.92	0.304	0.250	0.226	0.215	0.203	6.29	8.89
TabSieve (Ours)	**54.63**	**65.77**	**69.32**	**70.98**	**74.33**	**5.76**	**5.46**	**0.266**	**0.192**	**0.169**	**0.158**	**0.139**	**3.95**	**5.95**
Δ *over base*	+6.45	+7.00	+8.42	+7.97	+8.65	-1.68	-3.10	-0.059	-0.076	-0.071	-0.078	-0.082	-3.19	-3.15

Table 1: Classification (Accuracy ↑) and Regression (NMAE ↓) results averaged over all datasets under zero-shot and few-shot settings. Rank_z denotes the average rank in the zero-shot setting; Rank_i is averaged across all datasets and all few-shot settings. Δ denotes the performance gain relative to Qwen3-8B.

4 Experiments

4.1 Experimental Setup

Implementation Details. To construct the SFT dataset, we process 331 tables collected from previous works (Wen et al., 2024; Yan et al., 2024). We use Qwen3-Embedding-8B (Zhang et al., 2025b) to encode each table row and Qwen3-Next-80B-Thinking (Yang et al., 2025) as the teacher model to synthesize trajectories, and adopt Qwen3-235B-A22B as the judge model for rejection sampling. In the SFT stage, we initialize from Qwen3-8B (Yang et al., 2025) and fine-tune it with LLaMA-Factory (Zheng et al., 2024). For RL stage, our implementation of TAB-GRPO is based on the VeRL (Sheng et al., 2025) framework[1]. Hyperparameter settings are reported in Appendix A.4.

Evaluation. We evaluate on the benchmark released by GTL and TP-BERTa (Wen et al., 2024; Yan et al., 2024), which contains 75 classification tables and 52 regression tables. To prevent data leakage, we manually filtered out any training tables that appear in the evaluation benchmark. Following common tabular in-context learning protocols, we report results under multiple shot budgets (Cai et al., 2026; Hegselmann et al., 2023; Nam et al., 2024). For each shot setting, we uniformly sample 5K query rows to form the evaluation set and construct the in-context support set by sampling rows from the same table with three different random seeds. We use accuracy (Acc) for classification and normalized mean absolute error (NMAE) for regression. To reduce evaluation variance in few-shot regression and limit the impact of extreme outliers, we clip the NMAE for each sample at 1.0 before aggregation (Wen et al., 2024).

Baselines. Our baselines fall into three categories. (i) *Traditional Tabular Models.* Except for TabPFN, all predictors are fine-tuned on the labeled data for each dataset, so we report their performance starting from the few-shot regime rather than the zero-shot setting. (ii) *General Large Language Models.* We evaluate a diverse set of open-source models spanning different scales, and test them under a unified prompting protocol. (iii) *Tabular Large Language Models.* We follow the prompting templates and input formatting used in the official implementations.

4.2 Main Results

Table 1 presents a comprehensive comparison of TabSieve against all baselines on both classification and regression tasks. Compared with the second-best baseline, TabSieve yields average gains of 2.92% on classification and 4.56% on regression.

Comparison to traditional tabular models. Traditional tabular models require task-specific fitting on each table and therefore are not applicable in the zero-shot setting. In contrast, TabSieve performs cross-table reasoning without per-table fine-tuning

[1] https://github.com/Kazeya27/TabSieve

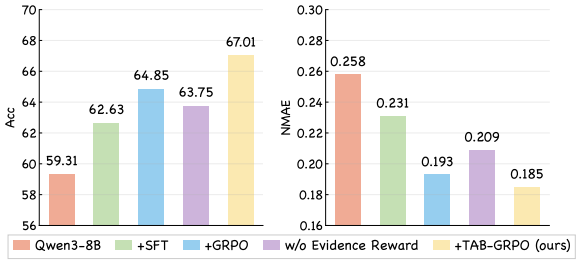

Figure 5: Ablation on training stages and RL components.

by analyzing table schemas and leveraging semantic reasoning to infer relationships between features and targets. This enables TabSieve to generalize effectively across heterogeneous tables while achieving the best overall performance.

Comparison to general LLMs. TabSieve consistently outperforms all evaluated general LLMs, including models with up to 70B parameters. Compared with the base Qwen3-8B, TabSieve delivers substantial gains on both tasks, and these gains generally become larger as the shot budget increases. This trend suggests that TabSieve can exploit in-context examples more effectively. Its select-then-predict procedure prioritizes evidential rows and suppresses noisy context, leading to more effective context utilization.

Comparison to tabular LLMs. Prior tabular LLMs typically lack strong reasoning ability and do not provide an explicit chain-of-thought, which limits their generalization under heterogeneous table schemas and task distributions. In contrast, TabSieve explicitly learns to reason over in-table evidence and to filter contextual rows through a dedicated selection step. This design is effective in in-context learning scenarios where demonstrations may be noisy, resulting in more robust and accurate predictions.

4.3 Ablation Study

In this section, we conduct ablations on the training stages, key RL components, and the evidence selection. We report the mean results averaged over zero-shot and all few-shot configurations.

Training stages. Figure 5 summarizes the incremental gains from SFT and RL stage. Starting from the backbone QWEN3-8B, SFT on TabSieve-SFT-40K (+SFT) yields a clear improvement, showing that the synthesized reasoning trajectories provide an effective initialization. Standard GRPO (+GRPO)

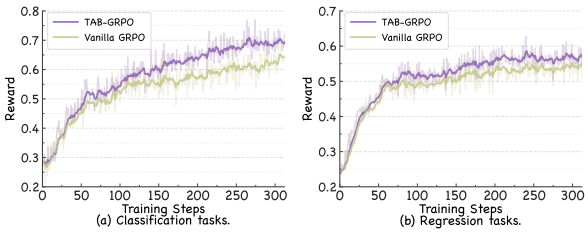

Figure 6: Training reward curves of TAB-GRPO and Vanilla GRPO.

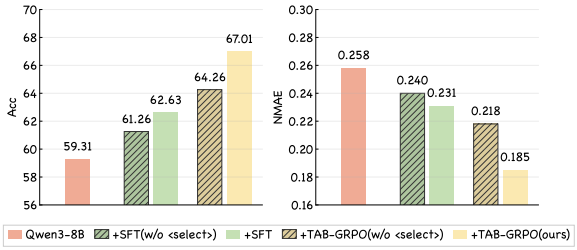

Figure 7: Ablation on evidence selection.

further improves both tasks, indicating that our reward design provides a reliable optimization signal that effectively refines the selection policy and reinforces prediction quality beyond pure imitation.

RL components. Figure 5 also reports an ablation over key RL components. Removing the evidence reward (`w/o Evidence Reward`) leads to a clear degradation, indicating that optimizing evidence selection improves the model's in-context learning ability. In addition, replacing vanilla GRPO with our TAB-GRPO yields a large improvement in classification. We further report the training curves in Figure 6, which show that TAB-GRPO achieves higher rewards during training. This supports our task-advantage balancing design, which reduces early-stage imbalance and stabilizes joint training.

Evidence selection. We construct `SFT(w/o <select>)` by removing the `<select>` step from the synthesized trajectories and fine-tuning the backbone on the modified dataset. We then continue RL training from `SFT(w/o <select>)` while disabling the evidence reward, yielding `TAB-GRPO(w/o <select>)`. Figure 7 shows that both variants perform worse than TabSieve at the corresponding training stages. These results indicate that explicit evidence selection is a key capability for unlocking the full benefits of the select-then-predict framework.

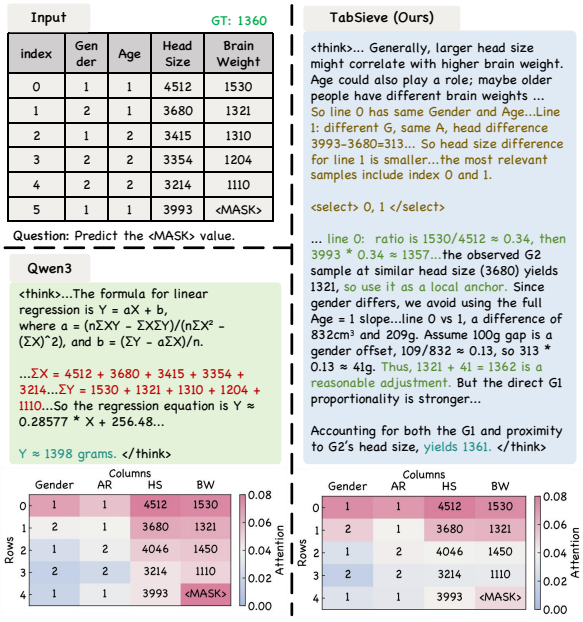

Figure 8: Case study of evidence selection.

4.4 Benefits of Evidence Selection Analysis

We present a case study in Figure 8 to illustrate the benefits of explicit evidence selection. Qwen3-8B tends to fit over all contextual rows, which in few-shot settings with sparse context is more easily affected by noisy or weakly relevant rows. In contrast, TabSieve follows a select-then-predict procedure that first identifies informative rows, reduces the influence of noise, and then predicts based on the selected evidence, yielding a better prediction.

We further follow prior work (Tu et al., 2025) and aggregate attention weights from the final 12 transformer layers into a cell-level heatmap. As shown in Figure 8, Qwen3-8B distributes attention broadly across the context, assigning substantial attention to nearly every row and even to the masked target cell, although it contains no task-relevant information. In contrast, TabSieve concentrates attention much more on the selected rows. This suggests that explicit evidence selection not only improves prediction accuracy, but also makes context usage more interpretable and auditable. Additional case studies are provided in Appendix B.5.

5 Related Works

Deep Learning Models for Tabular Data. Tabular prediction has long been a central problem in machine learning. Gradient-boosted decision trees have historically been strong performers, and prior studies show that tree ensembles often remain highly competitive and outperform many deep models (Shwartz-Ziv and Armon, 2021; Grinsztajn et al., 2022; Nam et al., 2023). To narrow this gap, researchers have proposed attention-based neural architectures to better capture feature interactions in tabular data (Arik and Pfister, 2020; Gorishniy et al., 2023; Somepalli et al., 2021). LoCalPFN (Thomas et al., 2024) shows that training with the k-nearest neighbours of the query row can further improve performance. Beyond single table learning, cross-table generalization methods address schema heterogeneity and knowledge transfer (Wang and Sun, 2022; Zhu et al., 2023; Yan et al., 2024). Despite these advances, most deep tabular models follow an instance-wise inference paradigm and do not exploit within-table evidence during inference. In-context learning models further highlight the potential of context-aware inference, where predictions are produced by conditioning on a set of labeled rows within the table at test time (Hollmann et al., 2023; Qu et al., 2025).

Large Language Models for Tabular Prediction. Large language models have recently gained traction for tabular prediction due to their in-context learning and reasoning abilities. Early works serialize tables into text and rely on prompting to elicit predictions (Hegselmann et al., 2023; Nam et al., 2024). Later studies emphasize improving data quality and training signals to better adapt LLMs to heterogeneous tables (Wang et al., 2024; Wen et al., 2024; Gardner et al., 2024). TabR1 further explores optimization beyond supervised learning by introducing reinforcement learning (Cai et al., 2026). Despite these developments, context usage in existing LLM-based approaches is still largely implicit and heavily dependent on prompt designs, which can be brittle when the context budget is limited and demonstrations are noisy. In contrast, TabSieve makes evidence selection an explicit intermediate action and jointly optimizes it with prediction, enabling more reliable suppression of noisy contextual rows and more consistent use of few-shot evidence in in-context learning.

6 Conclusion

In this paper, we propose TabSieve, a *select-then-predict* framework for tabular prediction that makes in-table evidence usage explicit. Rather than passively concatenating demonstrations, TabSieve first selects a small set of informative rows as evidence and then performs prediction conditioned on the selected rows, thereby focusing the context budget on salient evidence and reducing sensitivity

to noisy rows. To train this capability, we construct TabSieve-SFT-40K through data synthesis and strict filtering, and further propose TAB-GRPO to stabilize training on mixed regression and classification tasks. Extensive experiments across 127 tables show that TabSieve consistently improves over strong baselines under different shot budgets, while ablation studies and case studies further validate the effectiveness of its key design choices.

Limitations

First, we focus exclusively on text-only tabular prediction and do not investigate whether TabSieve can generalize to multimodal tables that involve images or other non-textual fields. Since many real-world tabular scenarios are inherently multimodal, extending the framework to such settings would be an important direction for future work. Second, due to computational constraints, our current study is limited to the 8B scale. It remains to be explored whether the benefits of TabSieve can be further amplified with larger backbones and broader training budgets. Third, on classification tasks TabSieve only outputs a single predicted class label rather than a probability distribution over the candidate labels, which limits its use in settings that require calibrated class posteriors. We leave extending the output format to per-label probabilities for future work. Finally, our current design mainly optimizes prediction performance. A more general and capable tabular model should ideally also develop stronger table understanding abilities, such as better schema comprehension and structural reasoning. Improving these capabilities could not only broaden the applicability of TabSieve, but also potentially bring further gains in prediction performance.

Ethical Considerations

This work aims to improve the prediction performance of tabular LLMs in few-shot settings. All experiments are conducted on publicly available tabular datasets and benchmarks. As in prior work, these data sources and tabular prediction methods may contain social biases, sampling biases, or other imperfections that can affect model behavior and evaluation outcomes. Beyond the risks already associated with tabular prediction methods on existing public datasets, we do not identify additional ethical risks introduced specifically by our method. We encourage continued attention to data quality, fairness, transparent evaluation, and responsible reporting of model capabilities and limitations.

References

Peter Martey Addo, Dominique Guegan, and Bertrand Hassani. 2018. Credit risk analysis using machine and deep learning models. *Risks*, 6(2):38.

Sercan O. Arik and Tomas Pfister. 2020. Tabnet: Attentive interpretable tabular learning. *Preprint*, arXiv:1908.07442.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Salva Rühling Cachay, Venkatesh Ramesh, Jason N. S. Cole, Howard Barker, and David Rolnick. 2021. Climart: A benchmark dataset for emulating atmospheric radiative transfer in weather and climate models. *Preprint*, arXiv:2111.14671.

Pengxiang Cai, Zihao Gao, Wanchen Lian, Guocong Li, and Jintai Chen. 2026. Strengthening llms for tabular prediction with structural priors. *Preprint*, arXiv:2510.17385.

Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, and Tomas Pfister. 2025. Reverse thinking makes llms stronger reasoners. *Preprint*, arXiv:2411.19865.

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*, KDD '16, page 785–794. ACM.

Xiang Cheng, Chengyan Pan, Minjun Zhao, Deyang Li, Fangchao Liu, Xinyu Zhang, Xiao Zhang, and Yong Liu. 2026. Revisiting chain-of-thought prompting: Zero-shot can be stronger than few-shot. *Preprint*, arXiv:2506.14641.

Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy yong Sohn. 2024. Linq-embed-mistral technical report. *Preprint*, arXiv:2412.03223.

Grace Colverd, Ronita Bardhan, and Jonathan Cullen. 2025. Machine learning methods for domestic energy prediction and retrofit potential for small-neighbourhoods at national scales in england and wales. *Energy and Buildings*, page 116388.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. *Preprint*, arXiv:1911.03429.

Shihan Dou, Ming Zhang, Zhangyue Yin, Chenhao Huang, Yujiong Shen, Junzhe Wang, Jiayi Chen, Yuchen Ni, Junjie Ye, Cheng Zhang, Huaibing Xie, Jianglu Hu, Shaolei Wang, Weichao Wang, Yanling Xiao, Yiting Liu, Zenan Xu, Zhen Guo, Pluto Zhou, and 8 others. 2026. Cl-bench: A benchmark for context learning. *Preprint*, arXiv:2602.03587.

Brage Eilertsen, Røskva Bjørgfinsdóttir, Francielle Vargas, and Ali Ramezani-Kebrya. 2025. Aligning attention with human rationales for self-explaining hate speech detection. *Preprint*, arXiv:2511.07065.

Hongfu Gao, Feipeng Zhang, Wenyu Jiang, Jun Shu, Feng Zheng, and Hongxin Wei. 2024. On the noise robustness of in-context learning for text generation. *Preprint*, arXiv:2405.17264.

Josh Gardner, Juan C. Perdomo, and Ludwig Schmidt. 2024. Large scale transfer learning for tabular data via language modeling. *Preprint*, arXiv:2406.12031.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, and 40 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2023. Revisiting deep learning models for tabular data. *Preprint*, arXiv:2106.11959.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on tabular data? *Preprint*, arXiv:2207.08815.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

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. *Preprint*, arXiv:2210.10723.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. 2023. Tabpfn: A transformer that solves small tabular classification problems in a second. *Preprint*, arXiv:2207.01848.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. *Preprint*, arXiv:2311.17911.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. 2020. Tabtransformer: Tabular data modeling using contextual embeddings. *Preprint*, arXiv:2012.06678.

Fushuo Huo, Wenchao Xu, Zhong Zhang, Haozhao Wang, Zhicheng Chen, and Peilin Zhao. 2025. Self-introspective decoding: Alleviating hallucinations for large vision-language models. *Preprint*, arXiv:2408.02032.

Chengzhi Liu, Zhongxing Xu, Qingyue Wei, Juncheng Wu, James Zou, Xin Eric Wang, Yuyin Zhou, and Sheng Liu. 2025a. More thinking, less seeing? assessing amplified hallucination in multimodal reasoning models. *Preprint*, arXiv:2505.21523.

Zheng Liu, Wenqi Shu, Teng Li, Xuan Zhang, and Wei Chong. 2025b. Interpretable machine learning for predicting sepsis risk in emergency triage patients. *Scientific Reports*, 15(1):887.

Jaehyun Nam, Woomin Song, Seong Hyeon Park, Jihoon Tack, Sukmin Yun, Jaehyung Kim, Kyu Hwan Oh, and Jinwoo Shin. 2024. Tabular transfer learning via prompting llms. *Preprint*, arXiv:2408.11063.

Jaehyun Nam, Jihoon Tack, Kyungmin Lee, Hankook Lee, and Jinwoo Shin. 2023. Stunt: Few-shot tabular learning with self-generated tasks from unlabeled tables. *Preprint*, arXiv:2303.00918.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2019. Catboost: unbiased boosting with categorical features. *Preprint*, arXiv:1706.09516.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2025. Tabicl: A tabular foundation model for in-context learning on large data. *Preprint*, arXiv:2502.05564.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM.

Ravid Shwartz-Ziv and Amitai Armon. 2021. Tabular data: Deep learning is not all you need. *Preprint*, arXiv:2106.03253.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C. Bayan Bruss, and Tom Goldstein. 2021. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *Preprint*, arXiv:2106.01342.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, and 14 others. 2024. Tablegpt2: A large multimodal model with tabular data integration. *Preprint*, arXiv:2411.02059.

Hao Sun, Chenming Tang, Gengyang Li, and Yunfang Wu. 2025. Lost in the passage: Passage-level in-context learning does not necessarily need a "passage". *Preprint*, arXiv:2502.10634.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. 2024. Retrieval & fine-tuning for in-context tabular models. *Preprint*, arXiv:2406.05207.

Songjun Tu, Qichao Zhang, Jingbo Sun, Yuqian Fu, Linjing Li, Xiangyuan Lan, Dongmei Jiang, Yaowei Wang, and Dongbin Zhao. 2025. Perception-consistency multimodal large language models reasoning via caption-regularized policy optimization. *Preprint*, arXiv:2509.21854.

Zifeng Wang, Chufan Gao, Cao Xiao, and Jimeng Sun. 2024. Meditab: Scaling medical tabular data predictors via data consolidation, enrichment, and refinement. *Preprint*, arXiv:2305.12081.

Zifeng Wang and Jimeng Sun. 2022. Transtab: Learning transferable tabular transformers across tables. *Preprint*, arXiv:2205.09328.

Xumeng Wen, Han Zhang, Shun Zheng, Wei Xu, and Jiang Bian. 2024. From supervised to generative: A novel paradigm for tabular deep learning with large language models. *Preprint*, arXiv:2310.07338.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Z. Chen, Jimeng Sun, Jian Wu, and Jintai Chen. 2024. Making pre-trained language models great on tabular prediction. *Preprint*, arXiv:2403.01841.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Yang Li, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, Jifan Yu, Shu Zhao, Juanzi Li, and Jie Tang. 2025a. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. *Preprint*, arXiv:2403.19318.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *Preprint*, arXiv:2506.05176.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. *Preprint*, arXiv:2403.13372.

Bingzhao Zhu, Xingjian Shi, Nick Erickson, Mu Li, George Karypis, and Mahsa Shoaran. 2023. Xtab: Cross-table pretraining for tabular transformers. *Preprint*, arXiv:2305.06090.

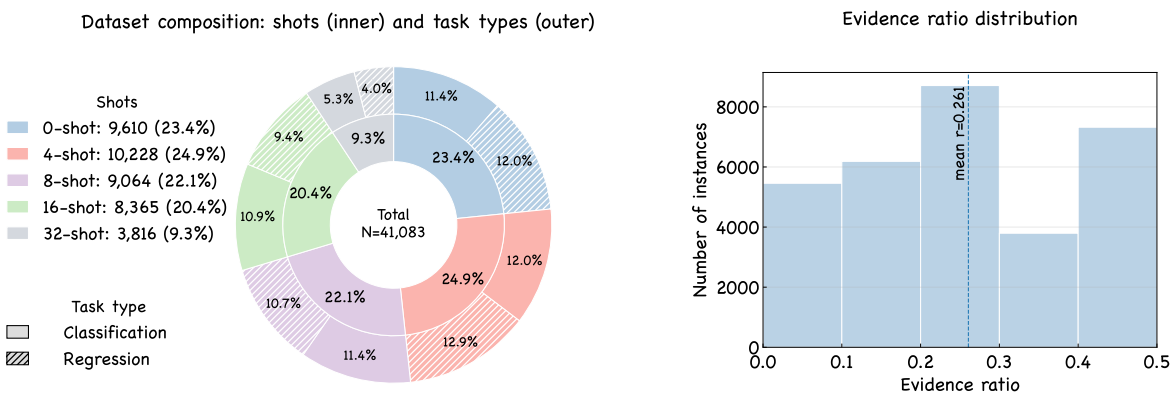

Figure 9: Data distribution of the SFT dataset. **Left:** dataset composition across the five shot budgets, with each slice further decomposed into classification and regression trajectories. **Right:** distribution of the evidence ratio for $K > 0$.

A Implementation Details.

A.1 Details of Feature Filtering

We consider a labeled tabular dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ with N instances and d feature columns. We first discard identifier-like columns, such as user IDs or email addresses, since their near-unique values can spuriously yield high mutual information with the target and dominate MI-based ranking, despite being uninformative for prediction. After this removal, we retain d' candidate feature columns. We denote the j-th remaining feature column as a random variable X_j and the target column as Y. For each candidate feature X_j, we compute its MI score with the target as:

$$s_j = I(X_j; Y) = \sum_u \sum_v p(u, v) \log \frac{p(u, v)}{p(u)\, p(v)}, \tag{5}$$

where u and v enumerate possible values of X_j and Y, respectively.

We then rank features by MI scores in descending order, obtaining $s_{(1)} \geq \cdots \geq s_{(d')}$. We select the smallest prefix whose cumulative MI reaches a fixed fraction ρ of the total MI mass:

$$m^\star = \min \left\{ m \,\middle|\, \sum_{t=1}^{m} s_{(t)} \geq \rho \sum_{t=1}^{d'} s_{(t)} \right\}, \tag{6}$$

where $\rho = 0.9$. Finally, we keep the top-$m^\star$ features together with the target column to form the filtered table. This procedure removes weakly related fields and improves the signal-to-noise ratio, while maintaining diverse dependency patterns and moderate residual noise that better reflects real-world tabular prediction scenarios.

A.2 Distribution of TabSieve-SFT-40K

We summarize the data distribution of our SFT dataset in Figure 9. After strict trajectory filtering, TabSieve-SFT-40K contains 41,083 select-then-predict trajectories. As shown in the left panel, each instance is constructed under a shot budget $K \in \{0, 4, 8, 16, 32\}$ and belongs to either a classification or a regression task. Across the five shot budgets, the dataset contains 9,610, 10,228, 9,064, 8,365, and 3,816 instances, corresponding to 23.4%, 24.9%, 22.1%, 20.4%, and 9.3%. The task types are approximately balanced overall, and both types are present under every shot budget. For $K > 0$, we further report the evidence ratio r, defined as the fraction of evidential rows among the sampled in-context candidates. As shown in the right panel of Figure 9, r ranges from 0 to 0.5 with an average value of $\bar{r} = 0.261$. The histogram exhibits a broad coverage of evidence densities, including both sparse-evidence cases and relatively dense-evidence cases, which improves distributional diversity and better reflects realistic tabular prediction scenarios where useful context can be sparse and unevenly distributed.

A.3 Reinforcement Learning Dataset Construction

We build the reinforcement learning training set TabSieve-RL-40K by filtering instances that can destabilize optimization or provide limited learning signal. We first discard examples for which the teacher model fails to produce an acceptable prediction within five attempts during SFT trajectory synthesis, as these cases often indicate insufficient supporting evidence, excessive difficulty, or an ill-posed target. We further remove examples

Hyperparameters	Value
Epochs	4.0
Micro batch size	4
Gradient accumulation	4
Learning rate	1×10^{-4}
Warmup ratio	0.1
LR scheduler	Cosine
GPUs	$8 \times$ H200
Training time (h)	14.11

Table 2: Core SFT hyperparameters for cold-start.

Hyperparameters	Value
Epochs	2.0
Train batch size	256
Rollout sampling num	8
Rollout temperature	1.0
KL coefficient	0.001
Learning rate	1×10^{-6}
Max prompt length	4096
Max response length	5120
Clip ratio ϵ_{low}	0.2
Clip ratio ϵ_{high}	0.28
λ_f	0.1
λ_e	0.2
λ_c	0.7
γ	1.0
β	0.99
$\eta_{\min}$	0.8
$\eta_{\max}$	1.0
GPUs	$8 \times$ H200
Training time (h)	25.08

Table 3: Core RL hyperparameters for TAB-GRPO.

that Qwen3-8B (Yang et al., 2025) solves correctly with a single forward evaluation. Since Qwen3-8B shares the same pretraining knowledge as our backbone, such instances are typically near-trivial and yield limited benefit from reinforcement learning. After filtering, TabSieve-RL-40K concentrates on medium difficulty instances, which improves training stability and makes the learning signal more informative.

A.4 Parameters Setup

We report the key hyperparameters used in our two-stage training pipeline, and present the essential settings for SFT and RL training with TAB-GRPO in Tables 2 and 3, respectively. In particular, we set ϵ_{high} to 0.28 following (Yu et al., 2025) to encourage exploration, and set the EMA decay factor β to 0.99 following common practice. Omitted options follow the default values of the training framework.

Evidence	4	8	16	32
$\lvert\mathcal{E}^{\text{inf}}\rvert$	4592	8594	17791	34313
$\lvert\mathcal{E}^{\text{emb}}\rvert$	5681	10327	23717	46235
$\lvert\mathcal{E}^{\text{emb}} \cap \mathcal{E}_i^{\text{inf}}\rvert$	3551	6816	14968	28567
$\frac{\lvert\mathcal{E}^{\text{emb}} \cap \mathcal{E}^{\text{inf}}\rvert}{\lvert\mathcal{E}^{\text{inf}}\rvert}$	77.33%	79.31%	84.13%	83.25%

Table 4: Overlap between embedding-based evidence and influence-based evidence under different shot budgets.

Evidence	0	4	8	16	32
Qwen3-8B	48.18	58.77	60.91	63.01	65.67
Influence-based	54.41	65.23	69.06	70.72	74.42
Embedding-based	54.63	65.77	69.32	70.98	74.33

Table 5: Classification results of different evidence construction strategies. We report accuracy (↑) averaged over all classification datasets under all shot settings.

Evidence	0	4	8	16	32
Qwen3-8B	0.325	0.268	0.239	0.235	0.222
Influence-based	0.262	0.184	0.171	0.149	0.132
Embedding-based	0.266	0.192	0.169	0.158	0.139

Table 6: Regression results of different evidence construction strategies. We report NMAE (↓) averaged over all regression datasets under all shot settings.

A.5 Prompt Template

In this section, we describe the prompt templates used in our experiments. Figure 10 presents the prompt for TabSieve. It first asks the model to analyze the table structure and the relationships among feature columns, and then to select evidential rows based on this understanding before making the final prediction. The model is finally required to output the answer in the `\boxed{}` format. Figure 11 shows the prompt used for the general LLM baselines. Compared with the prompt of TabSieve, we only remove the instructions related to evidence selection, so as to minimize prompt-induced bias. For the tabular LLM baselines, we follow the prompt templates provided in their official implementations for inference.

B Additional Analysis

B.1 Comparison with Influence-Based Evidence

In the main experiments, we use embedding-based evidence as the supervision signal for explicit row selection. Although Section 2.2 has shown that embedding-based evidence can improve prediction, one may ask whether a more direct, performance-based criterion can provide better evidence labels. To further validate the reasonableness of embedding-based evidence, we compare it with an influence-based evidence construction strategy.

For each training instance i, let $\mathcal{E}_i^{\text{emb}}$ denote the evidence set selected by the embedding-based construction, and let $k_i = |\mathcal{E}_i^{\text{emb}}|$ be the corresponding evidence budget. We construct an influence-based evidence set $\mathcal{E}_i^{\text{inf}}$ using a leave-one-out criterion. Specifically, we first run Qwen3-8B on the original context and compute the Avg@5 performance. We then remove each candidate row $x_{i,j}$ from the context and rerun the same Avg@5 evaluation. The influence of $x_{i,j}$ is defined by the performance degradation caused by removing it. We construct $\mathcal{E}_i^{\text{inf}}$ by selecting up to k_i rows with the largest positive influence scores. Note that not every row removal leads to a performance drop. Therefore, for some instances, the number of positively influential rows is smaller than k_i.

Table 4 shows that $\mathcal{E}^{\text{emb}}$ covers a substantial portion of the influential rows. This suggests that $\mathcal{E}^{\text{emb}}$ already captures many rows that are important for prediction, while much cheaper to construct. Moreover, as shown in Tables 5 and 6, $\mathcal{E}^{\text{inf}}$ achieves comparable performance to $\mathcal{E}^{\text{emb}}$, but does not yield a consistent advantage. Since constructing $\mathcal{E}^{\text{inf}}$ requires repeated leave-one-out inference, we use embedding-based evidence in the main experiments as an effective and efficient supervision signal for evidence selection.

B.2 More Ablation Studies

Prompt Design. Table 7 compares the standard prompt with the TabSieve prompt on general LLMs. Using the TabSieve prompt improves the performance, indicating that explicit instructions for evidence selection are beneficial for tabular prediction. However, prompt engineering alone still falls behind TabSieve, suggesting that jointly optimizing evidence selection and prediction is still necessary for achieving more robust cross-table generalization.

Model	Acc. ↑	NMAE ↓
Qwen2.5-72B w. Std. Prompt	65.11	0.217
Qwen2.5-72B w. TabSieve Prompt	65.92	0.209
Qwen3-8B w. Std. Prompt	59.31	0.258
Qwen3-8B w. TabSieve Prompt	59.79	0.251
TabSieve (Ours)	**67.01**	**0.185**

Table 7: Prompt ablation. We report average classification accuracy and regression NMAE over all shot settings.

Hyperparameter	Value	Acc. ↑	NMAE ↓
G	4	65.97	0.209
	8	**67.01**	0.185
	12	66.82	**0.179**
$\eta_{\min}$	0.7	66.57	0.197
	0.8	**67.01**	**0.185**
	0.9	66.14	0.192

Table 8: Sensitivity analysis of key RL hyperparameters.

Variant	Acc. ↑	NMAE ↓
w/o EMA Smoothing	65.12	0.207
w/o $\eta_{\min}$	66.04	0.202
TabSieve	**67.01**	**0.185**

Table 9: Component-level ablation of the task-advantage balancing mechanism in TAB-GRPO. We report average classification accuracy and regression NMAE over all shot settings.

Effects of group size G and lower bound η_{min} in TAB-GRPO. Table 8 presents the sensitivity analysis of two key hyperparameters in RL training. Increasing the group size G generally leads to better performance, since a larger group provides more diverse rollouts and more reliable group-relative advantage estimation. However, it also increases the computational cost of each update. The lower bound $\eta_{\min}$ controls the minimum scale of regression advantages in the task-advantage balancing mechanism. When $\eta_{\min}$ is too small, regression samples may be overly suppressed in the early stage, reducing learning efficiency. When $\eta_{\min}$ is too large, the balancing effect becomes weaker, which limits its ability to mitigate the optimization imbalance between classification and regression.

Component-level ablation of TAB-GRPO. The task-advantage balancing mechanism of TAB-GRPO combines two ingredients: the EMA of Eq. (2), which tracks the task-wise advantage scale with decay β, and the lower bound $\eta_{\min}$ on the rescaling factor of Eq. (3). To isolate their individ-

Model	0	4	8	16	32
DeepSeek-R1-70B	41.06	52.21	54.75	53.12	59.93
GPT-OSS-20B	40.75	48.99	50.15	52.48	53.87
Qwen3-8B	26.76	34.62	43.22	47.74	50.92
TabSieve	**43.19**	**54.10**	**56.06**	**55.92**	**60.31**

Table 10: Macro-F1 (↑) averaged over all classification datasets under zero-shot and few-shot settings. The best result in each column is in bold and the second best is underlined.

ual contributions, we conduct component-level ablations in Table 9. The *w/o EMA Smoothing* variant sets $\beta = 0$, which reduces m_t^τ to the current-step statistics and is therefore more sensitive to local fluctuations. The *w/o* $\eta_{\min}$ variant removes the lower-bound protection, allowing the regression advantage to become too small at some training steps and weakening its effective learning signal. Both variants underperform the full method, supporting the contributions of the EMA estimate and the bounded rescaling factor. The sensitivity analysis in Table 8 also provides additional evidence for the bound, since both an overly small and an overly large $\eta_{\min}$ degrade performance relative to the chosen value.

B.3 Macro-F1 on Classification Tasks

Accuracy can be dominated by majority classes on imbalanced tables, so it does not fully characterize classification quality. We therefore additionally report Macro-F1, which weights every class equally regardless of its frequency. Table 10 compares TabSieve with the strongest general LLM baselines and with the Qwen3-8B backbone under all shot budgets. TabSieve achieves the best Macro-F1 under every shot budget. The margin over the backbone is especially large in the low-shot regime, where Qwen3-8B tends to collapse onto the majority class, while TabSieve still recovers minority classes by grounding its prediction on explicitly selected evidence rows. These results confirm that TabSieve has stronger classification performance under a class-balanced evaluation, and that its accuracy gains in Table 1 are not an artifact of majority-class bias.

B.4 Utility of the Selected Rows

We further evaluate the utility of the rows selected by TabSieve. Specifically, for each query we take the evidence rows that TabSieve selects, remove these rows from the in-context support set, and evaluate Qwen3-8B on the remaining context. As shown in Table 11, removing the selected rows results in 26.5% lower accuracy and 30.2% higher NMAE, directly showing that the selected rows contain useful predictive information.

Variant	Acc. ↑	NMAE ↓
Qwen3-8B	**59.31**	**0.258**
Qwen3-8B w/o Selected Rows	43.57	0.336

Table 11: Utility of the rows selected by TabSieve. We remove the selected rows from the context and re-evaluate Qwen3-8B on the remaining rows. Results are averaged over all shot settings.

B.5 More Case Studies

In this section, we present additional examples to illustrate the reasoning patterns of TabSieve under different inference settings. Figure 12 and Figure 13 show TabSieve in the zero-shot regime. Since no examples are available for in-context learning, the model relies entirely on general knowledge to model relationships among the features and predicts the missing value based on the inferred mapping. Figure 14 presents a few-shot example, where the model first infers and constructs feature relations from the context, then selects a small set of evidential rows, and finally uses the selected evidence to produce the final prediction.

Prompt Template

Role

You are an expert in Data Mining and Logical Reasoning. Your core expertise lies in deeply comprehending the intrinsic logic of tabular data, analyzing the structural and feature-target dependencies, identifying samples similar to the target, and utilizing these insights to predict the ‘[MISSING]‘ value.

Task Description

Given a sampled sub-table and its metadata:

- The last cell of the final row is marked as ‘[MISSING]‘, serving as the prediction target.
- You must conduct a step-by-step analysis within ‘<think>‘ tags, adhering to the ‘Reasoning Requirements‘, to derive a **specific** predicted value.

Reasoning Requirements

- Comprehend the table structure and the semantics of feature columns.
- Analyze potential relationships between features and the target (e.g., strong correlations, causality).
- Identify samples similar to the target. During the reasoning process, enclose the indices of these selected samples within ‘<select>‘ tags (e.g., ‘<select> 2, 4, 7 </select>‘).
- Synthesize the analysis to model the logic connecting features to the target and predict the ‘[MISSING]‘ value.
- Construct a complex and comprehensive logical derivation from features to the target, simulating a predictive model.
- Reject lazy reasoning: Deriving the result solely by calculating the mean, median, or mode of similar samples is strictly prohibited.

Output Format

Output ONLY the final answer, excluding any additional text, and put your final answer within ‘\boxed{}‘.

Metadata of the table named

`{table_name}`

`{meta_data}`

Sampled Sub-table

`{sub_table}`

Candidate Values

`{range_of_target}`

Please reason step by step, output ONLY the final answer, put your final answer within \boxed{}.

Figure 10: Training and inference prompt templates of the TabSieve.

Prompt Template

Role

You are an expert in Data Mining and Logical Reasoning. Your core expertise lies in deeply comprehending the intrinsic logic of tabular data, analyzing the structural and feature-target dependencies, and predicting the ‘[MISSING]‘ value.

Task Description

Given a sampled sub-table and its metadata:

- The last cell of the final row is marked as ‘[MISSING]‘, serving as the prediction target.
- You must conduct a step-by-step analysis within ‘<think>‘ tags, adhering to the ‘Reasoning Requirements‘, to derive a **specific** predicted value.

Reasoning Requirements

- Comprehend the table structure and the semantics of feature columns.
- Analyze potential relationships between features and the target (e.g., strong correlations, causality).
- Synthesize the analysis to model the logic connecting features to the target and predict the ‘[MISSING]‘ value.
- Construct a complex and comprehensive logical derivation from features to the target, simulating a predictive model.
- Reject lazy reasoning: Deriving the result solely by calculating the mean, median, or mode of similar samples is strictly prohibited.

Output Format

Output ONLY the final answer, excluding any additional text, and put your final answer within ‘\boxed{}‘.

Metadata of the table named

`{table_name}`

`{meta_data}`

Sampled Sub-table

`{sub_table}`

Candidate Values

`{range_of_target}`

Please reason step by step, output ONLY the final answer, put your final answer within \boxed{}.

Figure 11: Inference prompt templates of the general LLMs.

Case Study 1:

index	215
gender	Female
age	79.0
hypertension	0
heart_disease	1
ever_married	No
work_type	Private
Residence_type	Urban
avg_glucose_level	205.33
bmi	31.0
smoking_status	smokes
stroke	[MISSING]

Question

Metadata of the table named stroke-prediction-dataset.
The column 'age' represents 'Age of the individual in years' and is a 'numerical' variable.

...

Candidate Values
The possible values are: '1', '0'.

Please reason step by step, output ONLY the final answer, put your final answer within '\boxed{}'.

Ground Truth: 1

TabSieve Output.
<think>
Okay, let's tackle this problem step by step. The task is to predict the missing stroke value for the target row...

First, looking at the table structure. The dataset is called stroke-prediction-dataset. The target column is 'stroke', which is categorical with possible values 0 or 1...

Now, considering the relationships between features and the target. Based on general knowledge, stroke risk factors typically include hypertension (high blood pressure), heart disease, smoking... For example, hypertension is a major risk factor for stroke. Similarly, having a heart disease could increase the risk...

<select></select>

Stroke risk factors include age (older age increases risk), heart disease (yes), smoking, high glucose (diabetes), BMI (obesity), etc...However, heart disease is present. Also, smoking is a risk factor. High glucose could indicate diabetes, which is a risk factor. Age 79 is very high risk...

Given that heart disease is a strong predictor of stroke, and she has it, and she smokes, the risk is elevated. High glucose further increases this risk. Therefore, despite no hypertension, the overall risk is high, and the most probable outcome is stroke=1.
</think>

$\boxed{1}$

Figure 12: Zero-shot case in classification tasks. TabSieve first analyzes the semantic meaning of the patient attributes and then predicts the label based on medical knowledge about stroke risk.

Case Study 2:

index	9725	Q1	-1.0
Run	146644.0	E2	3.70897
Event	1.49×10^9	px2	-0.0277326
E1	86.0024	py2	-2.4756
px1	-1.30757	pz2	-2.76172
py1	16.8533	pt2	2.47576
pz1	84.3248	eta2	-0.960737
pt1	16.904	phi2	-1.582
eta1	2.31017	Q2	1.0
phi1	1.64823	M	[MISSING]

Question

Metadata of the table named cern-electron-collision-data. The column 'E1' represents 'Energy of electron 1' and is a 'numerical' variable. The column 'px1 ' represents 'Momentum of electron 1 in the x direction'...

...

Candidate Values

The value of M ranges from 2.00008 to 109.999 (inclusive).

Please reason step by step, output ONLY the final answer, put your final answer within '\boxed{}'.

Ground Truth: 34.4543

TabSieve Output.

<think>

Okay, let's tackle this problem step by step.First, let's look at the metadata. The table is about CERN electron collision data. The target column is 'M', which is the invariant mass of the two-electron system...

Now, the target is M, the invariant mass. In physics, the invariant mass of a system is calculated from the total energy and momentum. For two particles, the formula is $M = sqrt((E1 + E2)^2 - (p_x1 + p_x2)^2 - (p_y1 + p_y2)^2 - (p_z1 + p_z2)^2)$. So M should be directly determined by the sum of the energies and the sum of the momenta components...

Since there are no other samples available for comparison, no samples can be selected. <select></select>

...So the answer should be the computed value of M using the formula. Let's compute it for the given row... First, sum of energies: E1 + E2 = 86.0024 + 3.70897 = 89.71137... Now, compute the sum of squared momenta: $(px1 + px2)^2 = (-1.3353026)^2 \approx 1.7830$... Total sum of squared momenta: $1.7830 + 206.72 + 6652.54 \approx 6861.043$.... Then $M^2 = 8048.140 - 6861.046 = 1187.097$ and $M = sqrt(1187.097) \approx 34.454$

This value falls within the expected range of 2.00008 to 109.999, confirming the calculation is consistent. Therefore, the missing value is approximately 34.454.

</think>

$\boxed{34.454}$

Figure 13: Zero-shot case in regression tasks. TabSieve recognizes that the target is the invariant mass of a two-electron collision system, and then derives the final prediction by applying the physics formula.

Case Study 3:

index	type_school	accreditation	gender	interest	residence	parent_salary	parent_was_in_college	will_go_to_college
779	Vocational	B	Male	Less Interested	Rural	6710000	True	True
771	Vocational	B	Female	Interested	Urban	3580000	False	False
438	Academic	A	Male	Less Interested	Urban	4180000	False	False
569	Academic	B	Male	Very Interested	Urban	6640000	False	True
91	Vocational	B	Male	Interested	Urban	3240000	False	[MISSING]

Question

Metadata of the table named go-to-college-dataset.

The column 'type_school' represents 'The type of school the student attends, either Academic or Vocational' and is a 'categorical' variable...

...

Candidate Values

The possible values are: 'True', 'False'.

Please reason step by step, output ONLY the final answer, put your final answer within '\boxed{}'.

Ground Truth: *False*

TabSieve Output.

<think>

Okay, let's tackle this problem step by step. The task is to predict the missing information in the 'will_go_to_college' column for the target sample...Let's think about general knowledge first. Typically, the type of school...urban vs rural could have different access to college resources. Parent's salary might indicate financial ability to afford college. If the parent was in college, maybe that's a positive indicator for the student to go too...

Line 771: Vocational, B... So same school type (Vocational), same school accreditation (B)... But maybe gender isn't a strong factor here. So line 771 is very similar... Line 569: Academic, B,... So same gender (Male), same school accreditation (B), similar interest (Very Interested vs Interested), same residence (Urban), parent not in college. Parent salary higher. So maybe this is the second similar...

<select> 771, 569 </select>

Line 771 has the same school type... The only difference is gender, but gender might not be a strong predictor. For example, line 438 is male...So gender might not be a major factor here...So maybe for Vocational, if parent not in college, they don't go to college. So target would be False. But line 569 is Academic, B...but went True...So school type and salary matter.

So for the target, which is Vocational, B, parent not in college. So maybe target is False.

</think>

$\boxed{False}$

Figure 14: Few-shot case. TabSieve follows a select-then-predict procedure. It first analyzes the table semantics and the influence of features on the target, then selects the evidence rows, and finally performs prediction mainly based on these selected rows.