
TABR1: TAMING GRPO FOR TABULAR REASONING LLMs

Pengxiang Cai, Zihao Gao, Jintai Chen*

AI Thrust, Information Hub, HKUST(GZ)

ABSTRACT

Tabular prediction has traditionally relied on gradient-boosted decision trees and specialized deep learning models, which excel within tasks but provide limited interpretability and weak transfer across tables. Reasoning large language models (LLMs) promise cross-task adaptability with transparent reasoning traces, yet their potential has not been fully realized for tabular data. This paper presents **TabR1**, the first reasoning LLM for tabular prediction with multi-step reasoning. At its core is **Permutation Relative Policy Optimization (PRPO)**, a simple yet efficient reinforcement learning method that encodes column-permutation invariance as a structural prior. By constructing multiple label-preserving permutations per sample and estimating advantages both within and across permutations, PRPO transforms sparse rewards into dense learning signals and improves generalization. With limited supervision, PRPO activates the reasoning ability of LLMs for tabular prediction, enhancing few-shot and zero-shot performance as well as interpretability. Comprehensive experiments demonstrate that TabR1 achieves performance comparable to strong baselines under full-supervision fine-tuning. In the zero-shot setting, TabR1 approaches the performance of strong baselines under the 32-shot setting. Moreover, TabR1 (8B) substantially outperforms much larger LLMs across various tasks, achieving up to **53.17%** improvement over **DeepSeek-R1 (685B)**.

1 introduction

Tabular prediction is a central task in machine learning with wide-ranging applications in healthcare, finance, and recommendation systems. Recent advances, including gradient-boosted decision trees such as XGBoost [Chen and Guestrin, 2016] and deep learning models such as TabPFN [Hollmann et al., 2022, Toman et al., 2024], have delivered strong performance on benchmark datasets. However, most of these approaches remain constrained by limited cross-task generalization, perform poorly in zero-shot and few-shot settings, and offer limited interpretability [Ke et al., 2017, Prokhorenkova et al., 2018, Gorishniy et al., 2021], thereby impeding their deployment in real-world scenarios.

Large language models (LLMs) present a new paradigm for tabular prediction. Beyond generating predictions, LLMs can produce natural language reasoning chains, thereby enhancing transparency and trust. Their pretraining on large-scale, multi-domain corpora also endows them with strong cross-task generalization [Brown et al., 2020, Wei et al., 2022, Wang et al., 2023], enabling rapid adaptation to new tasks under few-shot and zero-shot conditions. However, this potential has not yet been fully realized for tabular data. A key obstacle lies in the modality gap in reasoning: the reasoning patterns learned from natural language and mathematical corpora do not directly transfer to table-specific reasoning, which requires both semantic and numerical understanding. Bridging this gap is therefore crucial for unlocking the latent reasoning capacity of LLMs in tabular prediction.

Recently, reinforcement learning (RL) [Schulman et al., 2017, Ouyang et al., 2022, Bai et al., 2022, Rafailov et al., 2024, DeepSeek-AI, 2024, 2025] has become a key approach for enhancing the reasoning ability of LLMs. Models such as DeepSeek-R1 [DeepSeek-AI, 2024, 2025] demonstrate that Group Relative Policy Optimization (GRPO) can substantially improve long-chain reasoning by leveraging group-relative advantage estimation. Yet, as widely observed when LLMs are extended beyond text (*e.g.*, visual LLMs) [Yao et al., 2025], tabular LLMs also experience performance degradation due to the sparse-reward problem [Yin et al., 2020, Herzig et al., 2020, Deng et al., 2020]. Feedback is typically provided only at the outcome level (*i.e.*, whether the prediction is correct), leaving intermediate reasoning

*Corresponding author. Email: jintai.chen@hkust-gz.edu.cn

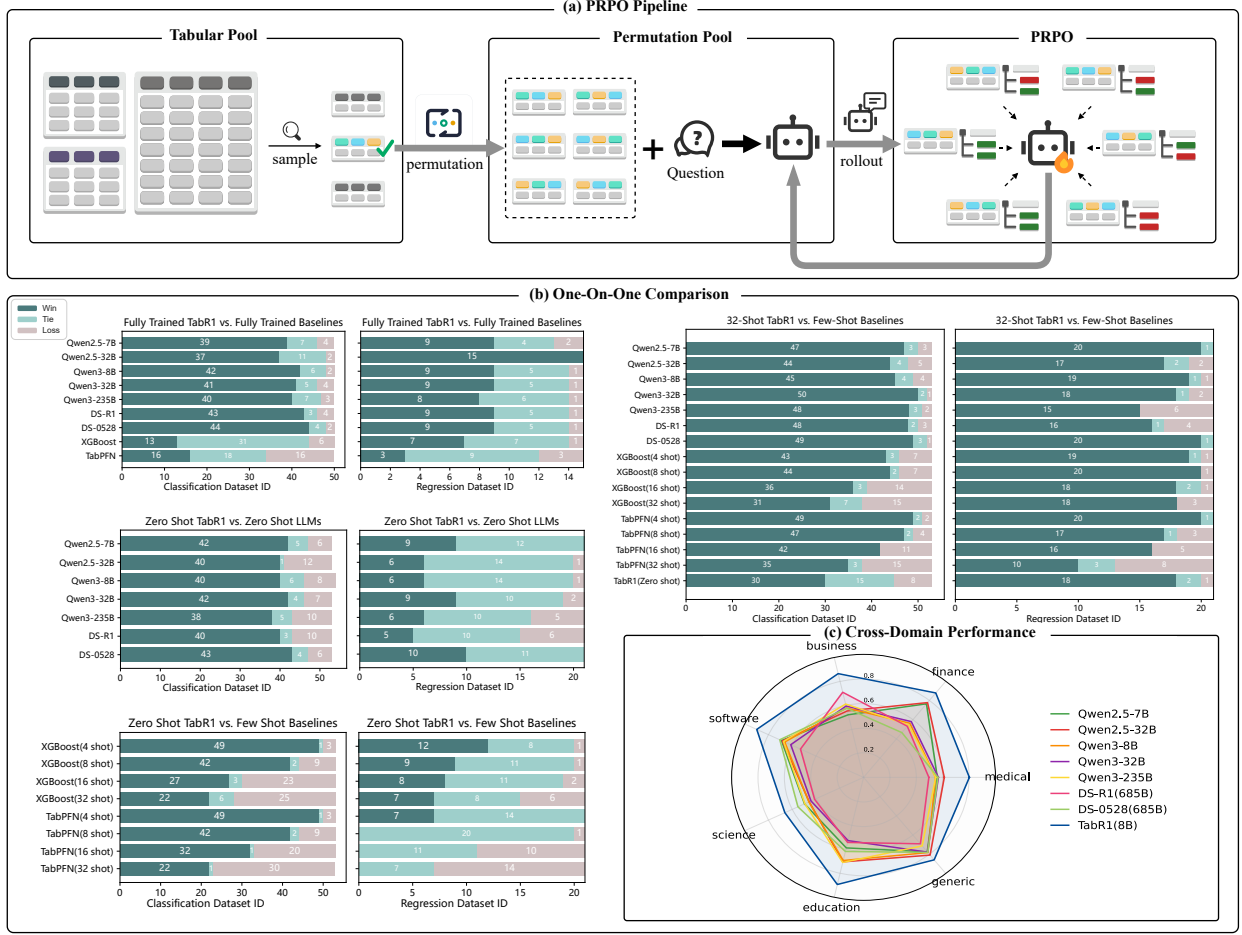


Figure 1: **(a)** We collect 139 datasets to construct a PRPO-compatible reinforcement learning dataset. Each training sample is permuted into multiple variants, paired with a prediction question, and fed into TabR1. The rollouts generate both intra-permutation and inter-permutation rewards, providing effective optimization signals for TabR1. **(b)** Win–Tie–Loss comparison between TabR1 and other models under fully trained, few-shot, and zero-shot settings. **(c)** Cross-domain performance of TabR1 and seven LLMs across seven distinct domains.

steps unsupervised. This constrains the exploitation of few-shot learning potential and hinders the incorporation of structural priors, resulting in inefficient exploration during reinforcement learning.

To address this issue, we propose **Permutation Relative Policy Optimization (PRPO)**, a reinforcement learning method specifically designed for tabular prediction. PRPO leverages the structural prior that tabular semantics remain invariant under column permutations. For each sample, it generates multiple column-permuted variants and serializes their feature names and values into reasoning instructions for the LLM. Advantages are then estimated both within and across permutations, providing richer supervisory signals from the same training sample. This mechanism converts sparse outcome-level feedback into denser learning signals while preserving reward fidelity. By operationalizing permutation invariance in this manner, PRPO mitigates inefficient exploration, stabilizes optimization, and markedly improves the generalization ability of LLMs in tabular prediction. On top of PRPO, we develop TabR1, the first reasoning LLM for tabular prediction, which not only activates the latent reasoning capacity of LLMs but also achieves strong zero-shot and few-shot performance with enhanced interpretability. Our contributions are summarized as follows:

- (i) We introduce TabR1, the first reasoning LLM tailored for tabular prediction, which integrates tabular semantics with multi-step reasoning to produce precise and interpretable predictions. At the same time, we construct a dataset for reinforcement learning with verifiable rewards to support the training of TabR1, which also provides an essential data foundation for future tabular reasoning LLMs.
- (ii) We propose Permutation Relative Policy Optimization (PRPO), a novel reinforcement learning strategy that exploits column-permutation invariance to convert sparse outcome-level rewards into dense learning signals,

thereby stabilizing training, improving generalization, and activating the tabular reasoning ability of LLMs with limited supervision.

- (iii) We validate that TabR1 achieves strong performance under full supervision and competitive results in few-shot learning, while in the zero-shot setting it approaches the 32-shot performance of strong baselines such as XGBoost and TabPFN-v2. Moreover, TabR1 surpasses models two orders of magnitude larger while maintaining transparent reasoning traces.

2 TabR1: A Reasoning LLM for Tabular Prediction

TabR1 is the first reasoning LLM tailored for tabular prediction. Previous LLM-based tabular prediction methods have mainly relied on supervised fine-tuning [Hegselmann et al., 2023, Gardner et al., 2024] or prompt engineering [Wei et al., 2022, Wang et al., 2023, Brown et al., 2020]. Although these methods bring certain performance improvements, they still fail to fully unlock the reasoning potential of LLMs in tabular prediction. In contrast, we propose Permutation Relative Policy Optimization (PRPO) and build TabR1, which effectively unleashes the reasoning capability of LLMs for tabular prediction. TabR1 comprises two key stages: (1) **Tabular serialization**. LLMs cannot directly process structured tabular data, as they are primarily trained on unstructured text. To address this, we serialize tabular data into a natural language format suitable for LLM input, enabling reinforcement learning with verifiable rewards and effectively enhancing their tabular reasoning ability. (2) **PRPO fine-tuning**. PRPO introduces a reinforcement learning method specifically designed for tabular prediction, which encodes the column-permutation invariance of tabular semantics as a structural prior, thereby transforming sparse rewards into dense learning signals. In this way, PRPO effectively mitigates inefficient exploration, stabilizes the optimization process, and significantly enhances the generalization ability of LLMs in tabular prediction.

2.1 Tabular Serialization

Since LLMs are primarily pretrained on unstructured text and cannot directly process structured tabular data, we serialize tabular data into concise and consistent textual representations so that they can serve as effective inputs for the model. We adopt a text template-based [Hegselmann et al., 2023] tabular serialization approach. Specifically, we define a function $serialize(F, x)$, where F denotes the set of column names and x represents the corresponding feature values. This function converts each feature-value pair in the table into a fixed-format natural language sentence such as “The *[feature]* is *[value]*.” and concatenates all feature descriptions sequentially according to the column order, thereby producing a coherent textual representation of the entire row. In addition to feature serialization, the LLM also receives a task-specific prediction query Q . When the serialized features $serialize(F, x)$ are combined with the task instruction Q , they form the complete LLM input ($serialize(F, x), Q$), which guides the model’s reasoning and prediction process. The detailed design and examples of the serialization template are provided in the Appendix.

2.2 PRPO Fine-tuning

Preliminary: GRPO. Group Relative Policy Optimization (GRPO) [DeepSeek-AI, 2024, 2025] stabilizes reinforcement learning for LLMs by normalizing rewards within groups. Given an input x , the policy π_θ generates G candidate outputs $\{o_1, \dots, o_G\}$, each assigned a scalar reward $\{R_1, \dots, R_G\}$. Relative advantages are computed as

$$\hat{A}_i = \frac{R_i - \mu_R}{\sigma_R}, \quad \mu_R = \frac{1}{G} \sum_{j=1}^G R_j, \quad \sigma_R = \sqrt{\frac{1}{G} \sum_{j=1}^G (R_j - \mu_R)^2}. \quad (1)$$

The policy is updated using a PPO-style clipped objective with KL regularization:

$$L^{\text{GRPO}}(\theta) = \mathbb{E}_{x, o_i \sim \pi_\theta} \left[\min(r_i(\theta) \hat{A}_i, \text{clip}(r_i(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i) \right] - \beta \text{KL}[\pi_\theta(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x)], \quad (2)$$

where

$$r_i(\theta) = \frac{\pi_\theta(o_i|x)}{\pi_{\text{ref}}(o_i|x)}. \quad (3)$$

Although GRPO has achieved strong results in reasoning tasks, it still suffers from the *sparse reward problem* in tabular prediction: only outcome-level rewards (e.g., correct/incorrect classification) are available, providing limited feedback for policy learning.

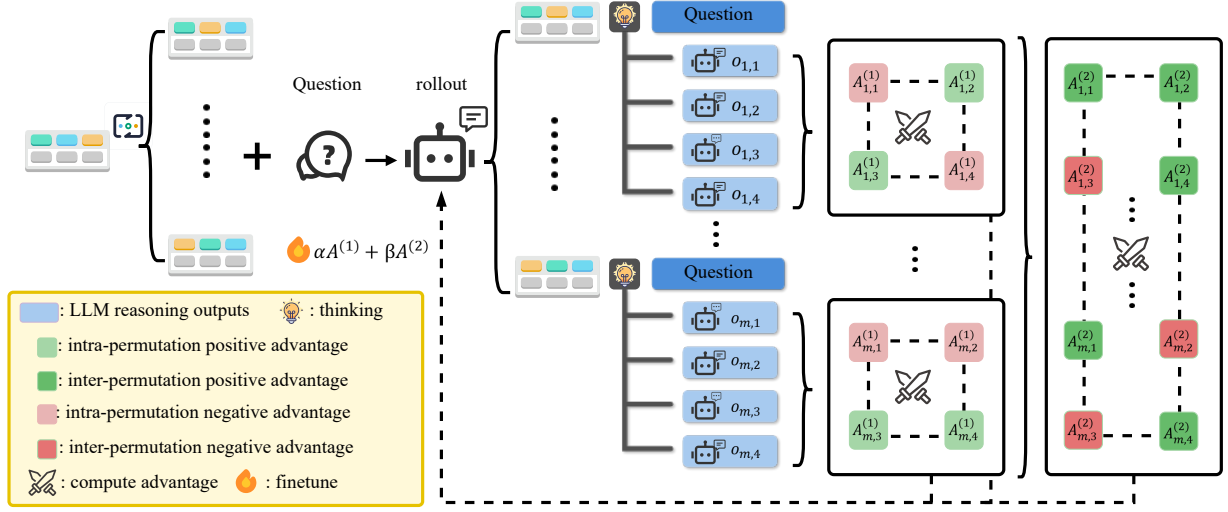


Figure 2: **Overview of PRPO.** Given a tabular sample, PRPO first generates multiple column-permuted variants that preserve label. (1) Each permuted sample is serialized and paired with the task question Q , then fed into TabR1 for rollout to produce candidate reasoning outputs $o_{i,j}$. (2) Rewards are computed through rule-based evaluation using verifiable ground-truth labels. (3) Intra-permutation advantages $A_{i,j}^{(1)}$ are estimated within each permutation group, while (4) inter-permutation advantages $A_{i,j}^{(2)}$ are aggregated across different permutations. (5) Finally, the two levels of advantages are integrated through the weighted objective $\alpha A_{i,j}^{(1)} + \beta A_{i,j}^{(2)}$ to update the policy, enabling permutation-aware reinforcement learning fine-tuning.

Permutation Relative Policy Optimization. To address this issue, we propose **Permutation Relative Policy Optimization (PRPO)**, which exploits the column-order invariance of tabular data to construct a two-level advantage estimation and densify reward signals.

Specifically, each tabular example can be represented as

$$t = \{x_1, x_2, \dots, x_n, y\}, \quad (4)$$

where x_i denotes a feature-value pair and y is the label. We define a *permutation* $\pi \in S_n$, which is a reordering of the feature index set $\{1, 2, \dots, n\}$. When applied to input t , the permuted sample is expressed as

$$\pi(t) = \{x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}, y\}. \quad (5)$$

The set of all possible permutations forms the symmetric group S_n , with cardinality $|S_n| = n!$. In practice, we sample m permutations from S_n to construct a set of column-permuted variants:

$$\{t_1, t_2, \dots, t_m\}, \quad t_j = \pi_j(t), \quad \pi_j \in S_n. \quad (6)$$

On this basis, we further define a two-level advantage estimation: intra-permutation advantages and inter-permutation advantages.

Intra-permutation advantages. For each permutation t_k , the model generates G candidates $\{o_{k,1}, \dots, o_{k,G}\}$ with rewards $R(o_{k,i})$. Intra-permutation advantages are normalized within the permutation:

$$\hat{A}_{k,i}^{(1)} = \frac{R(o_{k,i}) - \mu_k}{\sigma_k}, \quad (7)$$

where $\mu_k = \frac{1}{G} \sum_{i=1}^G R(o_{k,i})$ and σ_k is the standard deviation.

Inter-permutation advantages. All candidates across permutations are pooled into a single global group. Inter-permutation advantages are then computed as:

$$\hat{A}_{k,i}^{(2)} = \frac{R(o_{k,i}) - \mu_{\text{global}}}{\sigma_{\text{global}}}, \quad (8)$$

Algorithm 1 PRPO Fine-Tuning with Two-Level Advantage Estimation

Require: Serialized dataset $D = \{t \mid t = \{x_1, x_2, \dots, x_n, y^*\}\}$; policy π_θ ; reference policy π_{ref} ; number of permutations m ; group size G ; weight $\alpha \in [0, 1]$; KL weight β ; PPO clip ϵ

Ensure: Updated parameters θ

- 1: Initialize θ
- 2: **while** not converged **do**
- 3: Sample minibatch $\mathcal{B} \subset D$
- 4: **for all** $t = (x_1, x_2, \dots, x_n, y^*) \in \mathcal{B}$ **do**
- 5: Generate n column-permuted variants $\{t_1, \dots, t_m\}$, initialize reward set $\mathcal{R} \leftarrow \emptyset$
- 6: **for** $k = 1 \rightarrow m$ **do**
- 7: Sample G candidates $\{o_{k,1}, \dots, o_{k,G}\} \sim \pi_\theta(\cdot|t_k)$
- 8: Compute rewards $R(o_{k,i}, y^*)$ for $i = 1 \dots G$
- 9: Compute mean μ_k and std σ_k , intra-permutation advantages $\hat{A}_{k,i}^{(1)}$
- 10: Store $\{R(o_{k,i})\}$ in \mathcal{R}
- 11: **end for**
- 12: Compute mean μ_{global} and std σ_{global} over \mathcal{R} , inter-permutation advantages $\hat{A}_{k,i}^{(2)}$
- 13: Two-level aggregation: $\hat{A}_{k,i}^{\text{PRPO}} = \alpha \cdot \hat{A}_{k,i}^{(1)} + \beta \cdot \hat{A}_{k,i}^{(2)}$
- 14: Compute PPO ratios $r_{k,i}(\theta)$
- 15: Loss $\mathcal{L}^{\text{PRPO}}(\theta)$
- 16: **end for**
- 17: Update parameters: $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \mathcal{L}^{\text{PRPO}}(\theta)$
- 18: **end while**
- 19: **return** θ

where μ_{global} and σ_{global} are computed over all $\{x_1, \dots, x_n\}$.

The final PRPO advantage integrates both levels:

$$\hat{A}_{k,i}^{\text{PRPO}} = \alpha \cdot \hat{A}_{k,i}^{(1)} + (1 - \alpha) \cdot \hat{A}_{k,i}^{(2)}, \quad (9)$$

where $\alpha \in [0, 1]$ balances local and global signals.

The PRPO objective extends GRPO by incorporating the two-level advantages:

$$\mathcal{L}^{\text{PRPO}}(\theta) = \mathbb{E}_x \left[\sum_{k=1}^n \sum_{i=1}^G \min(r_{k,i}(\theta) \hat{A}_{k,i}^{\text{PRPO}}, \text{clip}(r_{k,i}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_{k,i}^{\text{PRPO}}) \right] - \beta \cdot \text{KL}, \quad (10)$$

where

$$r_{k,i}(\theta) = \frac{\pi_\theta(o_{k,i}|x_k)}{\pi_{\text{ref}}(o_{k,i}|x_k)}. \quad (11)$$

In summary, PRPO extends GRPO by incorporating the structural prior of column-permutation invariance, enabling it to better adapt to tabular data within the reinforcement learning framework. The overall training process is illustrated in Algorithm 1. This design alleviates inefficient exploration, stabilizes optimization, and maintains prediction consistency under column-order variations. Through this permutation-aware learning mechanism, PRPO significantly enhances the reasoning capability and generalization of LLMs in tabular prediction.

3 EXPERIMENTS

TabR1 demonstrates strong cross-task zero-shot capability, requiring full supervision on only a subset of datasets to activate this ability. Once activated, TabR1 learns the reasoning patterns underlying tabular data rather than merely memorizing data distributions, thereby exhibiting genuine reasoning-based generalization across datasets. To comprehensively and systematically evaluate TabR1’s effectiveness, we focus on two key aspects: (1) its performance under full supervision, and (2) its zero-shot generalization ability after reasoning activation. To this end, we collect 139 OpenML datasets for experimentation and organize them following the methodology described in Section 2.1, constructing a dataset designed for the training and inference of tabular reasoning large language models (LLMs). The entire experimental pipeline follows a two-stage process: first, TabR1 is trained on a subset of datasets under full

supervision to activate its reasoning capability; then, the model is directly evaluated on the remaining datasets in a zero-shot manner without additional training. This setup allows us to systematically examine both the fully supervised performance and the zero-shot transferability and generalization ability of TabR1.

3.1 Datasets and Experimental Setup

We collect 139 datasets from OpenML, including 103 classification datasets and 36 regression datasets. These datasets span a broad range of real-world domains, such as healthcare, finance, software engineering, business, education, and science. Following the tabular serialization procedure, we process each dataset and define a task-specific prediction question, constructing a reinforcement learning dataset with verifiable rewards for the training and evaluation of TabR1.

We comprehensively evaluate the overall performance of TabR1 through a two-stage experimental setup. In the first stage, 50 classification datasets and 15 regression datasets are used for fully supervised training and testing. TabR1 is fine-tuned using PRPO on the training sets and evaluated on the test sets to assess its performance under the fully supervised tabular prediction setting. After this stage, TabR1’s reasoning capability for tabular data is effectively activated. In the second stage, we conduct zero-shot evaluation on the remaining 53 classification and 21 regression datasets that TabR1 has never encountered before. This setup enables us to rigorously assess TabR1’s cross-task zero-shot generalization and reasoning-based transferability after reasoning activation.

TabR1 is post-trained on Qwen3-8B using PRPO. Our implementation is based on the VERL framework. All experiments are conducted on a node equipped with 8 NVIDIA H100-80G GPUs. A comprehensive list of hyperparameters and training configurations is provided in Appendix A.

3.2 Baselines

We compare TabR1 against three categories of baselines: **(1) General LLMs.** We evaluate several LLMs under the zero-shot setting, including models with comparable parameter scales to TabR1, such as Qwen2.5-7B, Qwen2.5-32B, Qwen3-8B, and Qwen3-32B, as well as large-scale reasoning-oriented LLMs including DeepSeek-R1-0528-685B, DeepSeek-R1-685B, and Qwen3-235B. For these LLMs, we design text-template prompts suitable for zero-shot generation. Detailed configurations of the text-template prompts are provided in the Appendix. **(2) TabLLM.** We evaluate TabLLM, which is a representative LLM specifically tuned for tabular prediction, under both few-shot learning and fully supervised fine-tuning settings. **(3) Strong Tabular Baselines.** Two strong tabular prediction baselines, TabPFN v2 and XGBoost, both are evaluated under full-supervision and few-shot learning settings. These models serve as strong, task-specific baselines known for their robustness and efficiency. Through this comprehensive comparison, we aim to thoroughly assess TabR1 across fully supervised, few-shot, and zero-shot learning settings, demonstrating how reinforcement learning with column-order structural priors effectively unlocks the reasoning potential of LLMs for tabular prediction.

3.3 Main experiment

Performance under Full Supervision. We conduct fully supervised training of TabR1 on 50 classification datasets and 15 regression datasets, evaluating accuracy for classification tasks and NMAE for regression tasks on their corresponding test sets. In this section, the experimental settings for the three baseline categories are as follows. **(1) Traditional Tabular Models** include TabPFN v2 and XGBoost, both trained under full supervision. **(2) Tabular LLMs** include TabLLM, which is evaluated under the fully supervised setting. Since TabLLM does not support regression tasks, it is only evaluated on classification datasets. **(3) General LLMs** include the Qwen and DeepSeek-R1 series, all evaluated under the zero-shot setting without any dataset-specific training.

In this experiment, TabR1 is trained on $8 \times$ NVIDIA H800-80G GPUs for 3000 steps. As shown in Table 1, TabR1 significantly outperforms general-purpose LLMs on classification tasks under full supervision. Compared with strong fully supervised baselines, it achieves higher average accuracy and leads on most datasets, demonstrating strong competitiveness in fully supervised tabular prediction. As shown in Table 3, TabR1 also achieves excellent performance on regression tasks, substantially surpassing general LLMs, achieving higher average performance than XGBoost, and remaining highly competitive with TabPFN v2.

Zero-Shot and Few-Shot Performance. We first evaluate the zero-shot performance of TabR1 on the remaining 53 classification and 21 regression datasets, where the model is directly applied to unseen datasets without any task-specific training. We report accuracy for classification tasks and normalized mean absolute error (NMAE) for regression tasks. Next, we perform few-shot fine-tuning of TabR1 on the same 53 classification and 21 regression datasets, and evaluate its performance on the corresponding test sets in terms of accuracy (for classification) and NMAE (for regression). In this section, the experimental settings for the three categories of baselines are as follows: **Traditional Tabular**

Table 1: Accuracy (\uparrow) comparison across models on classification datasets under the fully trained setting. Darker green cells indicate higher accuracy. All LLM-based methods except TabR1 are evaluated under the zero-shot setting.

Dataset	Qwen2.5-7B	Qwen2.5-32B	Qwen3-8B	Qwen3-32B	Qwen3-235B	DS-R1	DS-0528	XGBoost	TabPFN	TabR1
ada	0.7330	0.7943	0.7856	0.8118	0.7615	0.8249	0.8009	0.8313	0.8553	0.7834
Amazo	0.0600	0.0600	0.0850	0.3650	0.2400	0.9200	0.0700	0.9425	0.9425	0.9400
arsen	0.8750	0.8571	0.8036	0.6071	0.4107	0.0714	0.7500	0.8571	0.8723	0.8571
art	0.9678	0.9678	0.9678	0.9678	0.9356	0.8663	0.9604	0.9678	0.9678	0.9678
AVIDa	0.8021	0.8021	0.8021	0.8021	0.7995	0.6791	0.8021	0.8019	0.8019	0.8021
blood	0.4000	0.6133	0.4400	0.6800	0.7600	0.6667	0.4400	0.7600	0.7533	0.7600
breas	0.5172	0.5517	0.5862	0.6552	0.5172	0.6207	0.5517	0.7069	0.7500	0.7586
breas	0.9143	0.9143	0.8714	0.9857	0.9714	0.9571	0.8143	0.9429	0.9614	0.9571
bwin	0.5283	0.5849	0.3585	0.3962	0.3962	0.3774	0.4717	0.6698	0.6708	0.6604
Click	0.3750	0.8300	0.8050	0.4800	0.6050	0.5900	0.8000	0.8325	0.8305	0.8300
coil2	0.9400	0.8210	0.9379	0.9247	0.9217	0.8891	0.8566	0.9405	0.9405	0.9400
confe	0.4000	0.3600	0.6400	0.3600	0.5600	0.8000	0.4800	0.8800	0.8800	0.8800
datas	0.4200	0.4400	0.3200	0.3900	0.4900	0.5200	0.3200	0.7000	0.7045	0.7100
datat	0.8462	0.9231	0.8462	0.0769	0.5385	0.0769	0.7692	0.9231	0.9115	0.9231
depre	0.8322	0.7762	0.3007	0.2378	0.2797	0.2028	0.4336	0.8322	0.8322	0.8322
dis	0.9788	0.9815	0.9815	0.9762	0.9815	0.9762	0.9709	0.9841	0.9917	0.9841
doa	0.5849	0.4151	0.4906	0.5283	0.4528	0.3585	0.5660	0.6604	0.6321	0.6604
Fraud	0.8966	0.8942	0.7260	0.6635	0.4014	0.1971	0.3341	0.8990	0.9012	0.8990
haber	0.1935	0.6452	0.5484	0.5806	0.6452	0.5806	0.5484	0.7419	0.7484	0.7419
imdb	0.7875	0.8375	0.7875	0.8500	0.8625	0.8375	0.8000	0.7500	0.7556	0.8250
ipums	0.4766	0.6061	0.7210	0.6262	0.7784	0.7063	0.4806	0.8945	0.8951	0.8825
iris	1.0000	0.9333	1.0000	1.0000	1.0000	1.0000	1.0000	0.6667	1.0000	0.9333
irish	0.5600	0.9400	0.9000	1.0000	1.0000	1.0000	0.8400	1.0000	0.9910	1.0000
kc1	0.6445	0.5213	0.3886	0.7393	0.7346	0.6398	0.5592	0.8460	0.8507	0.8436
kc3	0.8261	0.9130	0.8913	0.7391	0.6739	0.5870	0.8696	0.9022	0.8804	0.9130
kick	0.7850	0.8650	0.6850	0.5550	0.1750	0.1250	0.4950	0.8775	0.8775	0.8750
Loan	0.7742	0.8226	0.7903	0.8065	0.7903	0.7903	0.7097	0.6911	0.7724	0.8226
Marke	0.6473	0.7366	0.7679	0.7946	0.7723	0.6429	0.7902	0.8504	0.8891	0.8750
mc1	0.6610	0.7360	0.7276	0.6589	0.6051	0.8353	0.8680	0.9926	0.9947	0.9916
mc2	0.8235	0.6471	0.5294	0.5294	0.5294	0.5882	0.5882	0.6667	0.6061	0.7059
meta	0.5472	0.5094	0.7925	0.6415	0.5472	0.0566	0.6415	0.8962	0.9208	0.9057
mw1	0.7073	0.6341	0.7073	0.6098	0.7561	0.5610	0.9268	0.9259	0.9284	0.9268
pc1	0.7297	0.4955	0.3694	0.5045	0.7387	0.3874	0.4144	0.9324	0.9423	0.9279
pc2	0.7943	0.9893	0.8372	0.7048	0.8354	0.6261	0.8927	0.9955	0.9954	0.9964
pc3	0.5669	0.7261	0.7452	0.4904	0.6561	0.5032	0.6624	0.8978	0.8994	0.8981
pc4	0.6096	0.6644	0.6438	0.5068	0.6644	0.5205	0.6233	0.8801	0.9199	0.8767
plasm	0.4375	0.2812	0.4688	0.4375	0.2500	0.5000	0.4375	0.5714	0.4762	0.5625
polis	0.7377	0.7614	0.3672	0.4450	0.4129	0.5482	0.1997	0.9306	0.9657	0.9306
polle	0.5091	0.4286	0.4935	0.4753	0.4831	0.3766	0.4857	0.4870	0.4688	0.5013
profb	0.0588	0.1029	0.0000	0.0294	0.2794	0.0735	0.0000	0.6741	0.6630	0.6618
quake	0.5000	0.5183	0.5596	0.5229	0.5229	0.5183	0.5321	0.5550	0.5532	0.5275
regim	1.0000	0.9524	1.0000	0.9048	0.9524	0.8571	1.0000	0.8049	0.9512	1.0000
seism	0.5869	0.6371	0.5637	0.4942	0.6062	0.3707	0.5830	0.9342	0.9323	0.9344
sf	0.8800	0.8150	0.8750	0.8250	0.7600	0.4600	0.8450	0.8775	0.8775	0.8800
solar	0.4019	0.2336	0.1776	0.2056	0.3458	0.2617	0.6822	0.8271	0.8037	0.8318
Speed	0.6301	0.4212	0.2780	0.6885	0.6420	0.6516	0.4654	0.8353	0.8648	0.8353
taiwa	0.9384	0.9677	0.0323	0.0367	0.0367	0.1349	0.0499	0.9677	0.9705	0.9677
tic	0.6146	0.9271	0.9896	0.9896	1.0000	1.0000	0.9792	0.7083	0.9880	1.0000
wilt	0.0579	0.0661	0.0909	0.1178	0.2831	0.1756	0.2955	0.9463	0.9889	0.9463
WMO	0.4851	0.4692	0.5050	0.5129	0.5109	0.5308	0.4771	0.5114	0.4955	0.5149
Mean	0.6409	0.6678	0.6196	0.5986	0.6175	0.5608	0.6187	0.8234	0.8413	0.8436
Rank	5.28	5.04	5.28	5.34	5.10	5.84	5.50	2.48	2.18	2.08

Models, including TabPFN v2 and XGBoost, both trained under few-shot settings. **Tabular LLMs**, including TabLLM, evaluated under few-shot settings. Since TabLLM does not support regression tasks, it is only evaluated on classification datasets. **General LLMs**, including the Qwen and DeepSeek-R1 series, all evaluated under the zero-shot setting without any dataset-specific training. In the zero-shot setting, TabR1 is directly applied to unseen datasets without any additional training. In the few-shot setting, TabR1 is fine-tuned on $8 \times$ NVIDIA H800-80G GPUs for 500 steps. As shown in Table 2, TabR1 significantly outperforms general-purpose LLMs on classification tasks in the zero-shot setting. Compared with strong baselines such as TabPFN v2 and XGBoost, TabR1 achieves higher average accuracy than their 4-, 8-, and 16-shot results and leads on most datasets. Notably, its zero-shot performance approaches the 32-shot results of these supervised models. This demonstrates that after training on other datasets, TabR1 learns transferable tabular reasoning patterns rather than memorizing dataset-specific distributions, effectively activating its reasoning capability and exhibiting strong zero-shot generalization in classification tasks. As shown in Table 4, TabR1 also shows

Table 2: Accuracy (\uparrow) comparison across different models on classification datasets under zero-shot and few-shot settings. Darker green cells indicate higher accuracy. **TabPFN** and **XGBoost** are trained and evaluated under **few-shot** settings, while **TabR1** and other LLM-based methods are evaluated under the **zero-shot** setting on unseen datasets.

Dataset	Qwen-2.5-7B	Qwen-2.5-32B	Qwen3-8B	Qwen3-32B	Qwen3-235B	DS-R1	DS-0528	TabLLM				XGBoost				TabPFN				TabR1	
								4 shot	8 shot	16 shot	32 shot	4 shot	8 shot	16 shot	32 shot	4 shot	8 shot	16 shot	32 shot	zero shot	32 shot
adult	0.6700	0.7350	0.7250	0.7600	0.7550	0.7900	0.7600	0.7600	0.7600	0.4500	0.4737	0.6100	0.6900	0.6950	0.7100	0.4780	0.5497	0.6355	0.7047	0.7800	0.7950
airf1	0.4400	0.3400	0.5600	0.5250	0.5300	0.5450	0.5500	0.4450	0.4450	0.4516	0.4993	0.4500	0.5300	0.5400	0.4950	0.4795	0.5022	0.5035	0.5082	0.5500	0.5550
anlic	0.1458	0.1667	0.5833	0.5417	0.7917	0.8333	0.3542	0.3750	0.8750	0.4811	0.5000	0.8333	0.7500	0.4792	0.8750	0.5400	0.6274	0.6747	0.7874	0.6875	0.8750
anlea	0.2444	0.3778	0.4111	0.2778	0.5556	0.4333	0.2889	0.7667	0.7667	0.5000	0.5000	0.6889	0.4111	0.7778	0.8667	0.0000	0.0000	0.0000	0.0000	0.7556	0.7667
arthy	0.6739	0.5870	0.5435	0.5435	0.5652	0.6087	0.6304	0.4348	0.5870	0.5000	0.5161	0.5652	0.4348	0.5435	0.5000	0.5220	0.5264	0.6231	0.6220	0.5652	0.6304
autoM	0.5500	0.7000	0.7250	0.7000	0.8500	0.8000	0.7000	0.5250	0.5250	0.5000	0.5284	0.5500	0.7500	0.7500	0.8250	0.7280	0.8337	0.8638	0.8650	0.7750	0.8250
Bank	0.7290	0.7480	0.4530	0.4050	0.4430	0.4110	0.3870	0.7960	0.7960	0.5000	0.5450	0.2450	0.4410	0.5500	0.6930	0.4263	0.5611	0.5682	0.6456	0.7960	0.7960
blast	0.7149	0.7674	0.6000	0.6213	0.6099	0.6227	0.5489	0.4000	0.7348	0.5000	0.5500	0.4780	0.4227	0.6610	0.7007	0.5386	0.5505	0.6337	0.6971	0.7348	0.7348
brzi	0.6429	0.6190	0.3810	0.3571	0.3333	0.5238	0.2857	0.7619	0.7619	0.5007	0.5500	0.7281	0.7143	0.5000	0.5000	0.5361	0.5880	0.5747	0.6723	0.7619	0.7619
calif	0.6200	0.3650	0.6700	0.7200	0.7200	0.7200	0.5050	0.5000	0.5000	0.5100	0.5550	0.5100	0.5950	0.7350	0.7480	0.5493	0.6597	0.7027	0.7718	0.7000	0.7450
chole	0.4839	0.4516	0.4516	0.4516	0.4516	0.5161	0.4516	0.4516	0.4516	0.5263	0.5870	0.5484	0.3548	0.4194	0.4516	0.5262	0.5115	0.5344	0.5410	0.5161	0.6161
churn	0.7840	0.7920	0.7480	0.5680	0.6900	0.5040	0.7340	0.1420	0.5800	0.5435	0.6000	0.7560	0.6460	0.1800	0.6820	0.4147	0.4433	0.5414	0.5901	0.8580	0.8580
cieve	0.6774	0.7097	0.4839	0.5484	0.5806	0.4839	0.5806	0.4516	0.7742	0.5455	0.6400	0.6452	0.5806	0.8065	0.7097	0.5852	0.6601	0.7197	0.7672	0.6774	0.8419
colic	0.3784	0.5405	0.6216	0.6406	0.5946	0.6216	0.6216	0.5484	0.6429	0.4054	0.6486	0.7297	0.7297	0.7108	0.7297	0.5284	0.7892	0.8216	0.7400	0.8216	0.8216
commu	0.5950	0.6300	0.6800	0.7000	0.6550	0.6100	0.6000	0.5750	0.4250	0.5850	0.8629	0.5950	0.7300	0.7600	0.7600	0.5935	0.7020	0.7644	0.7855	0.6400	0.8300
compa	0.3920	0.4792	0.5814	0.6212	0.5947	0.5852	0.6023	0.4697	0.4830	0.6000	0.6456	0.5417	0.5303	0.4811	0.5701	0.5203	0.5413	0.7312	0.5937	0.6420	0.6307
dgrf	0.6082	0.4561	0.8216	0.7778	0.8129	0.8450	0.7398	0.8363	0.8363	0.6216	0.6493	0.1813	0.7368	0.8947	0.8743	0.4220	0.7726	0.8158	0.8873	0.8363	0.8363
diabi	0.6234	0.6623	0.6753	0.5974	0.6234	0.6234	0.6104	0.3506	0.4675	0.6229	0.6494	0.7662	0.6494	0.6753	0.6883	0.5630	0.6045	0.6612	0.6474	0.7662	0.7494
dye	0.7436	0.7949	0.9744	0.8462	0.9487	0.9487	0.7436	0.7692	0.7179	0.6250	0.6500	0.8974	0.8974	0.8974	0.9231	0.8195	0.8688	0.8818	0.9195	0.8974	0.9231
Englo	0.5408	0.6609	0.5429	0.5880	0.5300	0.3562	0.4571	0.3433	0.3433	0.6429	0.6580	0.6416	0.5837	0.5794	0.6545	0.5814	0.6404	0.6161	0.6291	0.6502	0.6507
eye	0.5007	0.7507	0.4993	0.4665	0.4599	0.4796	0.5033	0.5007	0.5007	0.6456	0.6567	0.5138	0.5085	0.4915	0.5020	0.5007	0.5003	0.5168	0.5138	0.5059	0.4993
flags	0.7000	0.5500	0.6000	0.5000	0.4500	0.4000	0.4500	0.3500	0.6500	0.6490	0.6875	0.3000	0.3000	0.4000	0.7000	0.4282	0.5128	0.5846	0.6205	0.6000	0.6500
huyes	0.6429	0.5000	0.6429	0.6429	0.4286	0.7857	0.6429	0.6429	0.6429	0.6493	0.7143	0.3571	0.7143	0.7857	0.8371	0.5222	0.5556	0.5963	0.6667	0.6429	0.6729
hepat	0.2500	0.5000	0.4375	0.3125	0.5000	0.5000	0.3750	0.8125	0.8125	0.6500	0.7404	0.5625	0.6250	0.3625	0.6250	0.5781	0.7387	0.7323	0.7484	0.8750	0.8750
HMEQ	0.6376	0.7886	0.4262	0.5101	0.5268	0.3792	0.7097	0.8003	0.8003	0.6550	0.7412	0.7987	0.5923	0.5000	0.7450	0.4859	0.5800	0.6525	0.7255	0.8020	0.7987
hypot	0.4683	0.7063	0.4127	0.5344	0.4524	0.3704	0.3651	0.9233	0.9233	0.6567	0.7433	0.7593	0.5344	0.8466	0.8148	0.0000	0.0000	0.0000	0.0000	0.6561	0.9153
ibm	0.7755	0.7823	0.3401	0.3741	0.3878	0.2585	0.3469	0.8367	0.8367	0.6857	0.7500	0.6122	0.2585	0.6054	0.7551	0.5143	0.4810	0.5469	0.5344	0.8367	0.8367
jungl	0.4407	0.7415	0.6992	0.7161	0.7288	0.7076	0.6780	0.4407	0.4407	0.7348	0.7604	0.5932	0.6483	0.7585	0.8263	0.6257	0.6711	0.7985	0.9361	0.9368	0.7373
kc2	0.4340	0.3774	0.2453	0.2842	0.5094	0.4528	0.3208	0.7925	0.7925	0.7348	0.7619	0.2830	0.5283	0.6226	0.7170	0.6210	0.6714	0.6381	0.7562	0.7925	0.7925
kdd	0.6456	0.5443	0.6076	0.6203	0.5190	0.6076	0.5190	0.4557	0.4557	0.7412	0.7667	0.4430	0.3671	0.8481	0.8098	0.5287	0.7102	0.7879	0.8293	0.5606	0.8456
lungc	0.6522	0.7826	0.3043	0.2609	0.3043	0.1739	0.4783	0.8261	0.8261	0.7500	0.7800	0.4783	0.7826	0.5652	0.4783	0.4783	0.6696	0.6239	0.7196	0.8696	0.8261
NAT1C	0.6427	0.6293	0.6653	0.6467	0.6307	0.1893	0.6547	0.6493	0.6493	0.7570	0.7925	0.6787	0.8640	0.8120	0.8387	0.5857	0.6758	0.7800	0.8517	0.6480	0.8493
newto	0.4286	0.5000	0.5714	0.5714	0.5000	0.4286	0.5714	0.5000	0.3571	0.7600	0.7947	0.4286	0.5000	0.7143	0.6429	0.4821	0.5571	0.6077	0.6679	0.6429	0.8133
no2	0.5000	0.4600	0.5000	0.4800	0.4800	0.5000	0.5000	0.5000	0.5000	0.7604	0.7960	0.4800	0.4400	0.5200	0.6600	0.5100	0.5110	0.5280	0.5210	0.5200	0.8205
page	0.5912	0.6642	0.7310	0.8978	0.8631	0.4380	0.8978	0.8978	0.8978	0.7619	0.8500	0.2464	0.6387	0.6387	0.7836	0.8520	0.8978	0.8978	0.8978	0.8978	0.8978
phary	0.5500	0.6000	0.6000	0.4500	0.5000	0.4500	0.5000	0.6000	0.6000	0.6000	0.7500	0.5000	0.6000	0.6000	0.6000	0.6154	0.5667	0.7385	0.7513	0.6000	0.6000
pm10	0.4800	0.5000	0.5000	0.5000	0.4800	0.5000	0.5000	0.5000	0.5000	0.7925	0.8108	0.5000	0.6200	0.5600	0.5000	0.4980	0.5020	0.5200	0.5140	0.5000	0.5000
PostP	0.5298	0.6026	0.5894	0.5894	0.5629	0.6225	0.5894	0.6490	0.6490	0.7960	0.8156	0.7417	0.4702	0.7682	0.7748	0.5934	0.6595	0.6890	0.7684	0.6689	0.6490
prmn	0.1000	0.2000	0.5500	0.8000	0.4000	0.5000	0.3500	0.5000	0.5000	0.8003	0.8261	0.5500	0.5500	0.6500	0.8000	0.5175	0.8025	0.9800	1.0000	0.5500	0.5000
road	0.5350	0.3300	0.3500	0.6400	0.5400	0.5550	0.4000	0.5100	0.5000	0.8125	0.8363	0.5000	0.6400	0.5300	0.6300	0.5390	0.5545	0.5577	0.5565	0.6500	0.8850
segme	0.8225	0.7186	0.4848	0.8528	0.8442	0.8355	0.8442	0.8571	0.8571	0.8261	0.8367	0.3247	0.6364	0.8097	0.8918	0.5929	0.6781	0.8833	0.8918	0.7222	0.8571
Skin	0.7043	0.8565	0.8652	0.8174	0.8652	0.8870	0.8565	0.9261	0.9261	0.8363	0.8387	0.9174	0.9261	0.9435	0.9261	0.6576	0.7172	0.7593	0.8787	0.9174	0.9043
spamb	0.6052	0.7657	0.6161	0.5748	0.6920	0.5141	0.6703	0.6052	0.6052	0.8367	0.8563	0.6334	0.5597	0.8178	0.8330	0.5544	0.7403	0.8347	0.8010	0.7137	0.7202
SPECT	0.7143	0.5143	0.7429	0.6571	0.7143	0.7143	0.6857	0.7143	0.7143	0.8563	0.8571	0.6286	0.6571	0.6571	0.6286	0.5800	0.6814	0.6986	0.7271	0.7143	0.7143
spect	0.8889	0.7778	0.8889	0.1852	0.6481	0.1667	0.6111	0.8889	0.8889	0.8571	0.8580	0.1481	0.3519	0.8889	0.8704	0.5869	0.6963	0.8280	0.8738	0.7222	0.8889
telco	0.6411	0.7538	0.5702	0.5957	0.6241	0.5901	0.7470	0.7448	0.8300	0.8644	0.5021	0.4936	0.6950	0.6624	0.5313	0.5566	0.6241	0.6977	0.7362	0.7348	0.7348
Tour	0.6562	0.5833	0.3958	0.3438	0.3854	0.3125	0.3438	0.7604	0.7604	0.8750	0.8750	0.7292	0.4792	0.7396	0.7188	0.6236	0.6298	0.6812	0.6942	0.7604	0.7604
triaz	0.5263	0.5263	0.4737	0.4737	0.4211	0.4211	0.5263	0.7579	0.3684	0.8889	0.8889	0.5263	0.6842	0.7368	0.7895	0.4895	0.5905	0.6002	0.6500	0.6316	

Table 4: NMAE (\downarrow) comparison across different models on regression datasets under zero-shot and few-shot settings. Darker green cells indicate lower NMAE. **TabPFN** and **XGBoost** are trained and evaluated under **few-shot** settings, while **TabR1** and other LLM-based methods are evaluated under the **zero-shot** setting on unseen datasets.

Dataset	Qwen-2.5-7B	Qwen-2.5-32B	Qwen3-8B	Qwen3-32B	Qwen3-235B	DS-R1	DS-0528	XGBoost				TabPFN				TabR1	
								4 shot	8 shot	16 shot	32 shot	4 shot	8 shot	16 shot	32 shot	zero shot	32 shot
ames	0.0839	0.0997	0.0919	0.0578	0.0635	0.0605	0.1009	0.2751	0.0926	0.0777	0.0858	0.1003	0.0706	0.0541	0.0432	0.0868	0.0335
aucti	0.1831	0.1838	0.1804	0.1816	0.1819	0.1826	0.1846	0.3189	0.2286	0.1876	0.3318	0.1931	0.1879	0.1512	0.0732	0.1829	0.0828
Bosto	0.1346	0.1121	0.1506	0.1360	0.0619	0.0675	43.1375	0.2525	0.1170	0.1077	0.0794	0.1856	0.1127	0.0782	0.0756	0.1251	0.0687
cars	0.3299	0.2764	0.3575	0.2941	0.3314	0.2907	0.4177	0.0974	0.0905	0.0867	0.0469	0.1790	0.1578	0.1051	0.0681	0.1567	0.0584
coltr	0.2892	0.2378	0.3233	1.0247	1.1183	0.4831	0.8425	0.2406	0.1890	0.1710	0.2072	0.3113	0.2052	0.1733	0.1420	0.2426	0.0906
coner	0.2237	0.1704	0.1961	0.2316	0.1106	0.1139	0.1685	0.2326	0.1630	0.1661	0.1230	0.1829	0.1567	0.1194	0.0851	0.1685	0.1499
CPMP	0.3852	0.3860	0.3854	0.3835	0.3845	0.3793	0.3856	0.4157	0.6947	0.5518	0.4200	0.3977	0.3880	0.3705	0.3780	0.3855	0.2864
emplo	0.1665	0.1804	0.0599	0.1034	0.0697	0.0418	0.1888	0.1216	0.1249	0.0746	0.0642	0.1048	0.0720	0.0506	0.0285	0.0860	0.0209
healt	0.1836	0.1512	0.1537	0.1247	0.1517	0.1478	0.1656	0.4697	0.2409	0.1843	0.1023	0.2098	0.1501	0.1005	0.0828	0.1503	0.0947
house	0.0937	0.0675	0.0571	0.0578	0.0405	0.0774	0.0697	0.2050	0.0715	0.0660	0.0574	0.0890	0.0691	0.0510	0.0381	0.0632	0.0651
Lisbo	0.1661	0.3716	0.3182	0.1666	0.1269	0.1269	0.6827	0.1881	0.2410	0.2204	0.1948	0.1744	0.1298	0.1125	0.0918	0.1267	0.0570
lowbw	0.1755	0.1520	0.2667	0.1732	0.1393	0.1268	0.2929	0.1541	0.1797	0.1601	0.1683	0.2024	0.1490	0.1275	0.1171	0.1759	0.1592
mauna	0.2041	0.0333	0.0573	0.0352	0.0244	0.0246	0.0832	0.1577	0.1278	0.0779	0.0696	0.1126	0.0445	0.0298	0.0303	0.0508	0.0531
newto	0.4029	1.6552	0.8940	0.3716	0.3384	0.3670	0.5957	0.2934	0.2184	0.2649	0.3309	0.2807	0.2521	0.2307	0.2001	0.2429	0.2857
NHANE	0.2972	0.2798	0.2676	0.2925	0.2644	0.2652	0.2933	0.4441	0.3740	0.3218	0.3794	0.2833	0.2602	0.2502	0.2299	0.2636	0.1861
place	0.1765	0.5358	0.5702	0.4590	0.4478	0.4549	0.3526	0.1953	0.2105	0.4233	0.1866	0.2181	0.1899	0.1740	0.1653	0.1925	0.1618
pm10	0.1888	0.1992	0.1909	0.1893	0.1819	0.1883	0.2176	0.2090	0.1736	0.1809	0.2337	0.2220	0.1681	0.1591	0.1457	0.1891	0.1202
seoul	0.1747	0.1223	0.1412	0.1181	0.1088	0.1110	0.1333	0.1644	0.2444	0.5734	0.1514	0.1531	0.1690	0.1306	0.1095	0.1326	0.1219
std	0.2026	1.2481	0.2262	0.3650	0.9962	1.2384	0.7261	0.2380	0.2125	0.5603	0.2984	0.2429	0.2114	0.2146	0.2045	0.2238	0.2005
weath	0.0325	0.0325	0.0325	0.0642	0.0324	0.0325	0.0325	0.1638	0.1558	0.0610	0.0612	0.1360	0.0371	0.0264	0.0207	0.0325	0.0125
yacht	0.1502	0.1112	0.1396	0.1475	0.1673	0.1106	0.1425	0.3972	0.2245	0.2645	0.0695	0.1998	0.0434	0.0400	0.0207	0.1113	0.0216
Mean	0.2021	0.3146	0.2409	0.2370	0.2544	0.2329	2.3435	0.2492	0.2083	0.2277	0.1744	0.1990	0.1536	0.1309	0.1119	0.1614	0.1110
Rank i	11.05	10.76	10.81	9.86	7.24	7.24	13.10	13.14	11.33	10.33	9.19	12.52	7.43	3.71	2.00	8.19	3.10

and Guestrin, 2016, Ke et al., 2017] and specialized deep architectures (TabNet, TabTransformer) [Arik and Pfister, 2021, Huang et al., 2020] achieve strong supervised performance but lack interpretability and cross-task generalization [Hegselmann et al., 2023, Gupta et al., 2023, Zhang et al., 2023, Gardner et al., 2024, Touvron et al., 2023, Bai et al., 2024]. Approaches like TabPFN [Hollmann et al., 2022, Toman et al., 2024] introduce pretrained priors for few-shot settings, yet remain task-specific.

Reasoning LLMs (e.g., GPT-4 [OpenAI, 2023], DeepSeek-R1 [DeepSeek-AI, 2025], Qwen3 [Bai et al., 2023, 2024]) demonstrate strong multi-step reasoning with reinforcement learning methods such as GRPO, but their ability does not naturally transfer to tabular data due to a modality gap and sparse outcome-level rewards.

Tabular LLMs such as TabLLM explore prompt engineering and supervised fine-tuning for structured prediction, but improvements remain limited in zero-shot and few-shot scenarios. Prior studies suggest that incorporating structural priors is essential for robust adaptation.

Reinforcement learning for tabular reasoning extends this line by leveraging structural properties to densify feedback. Our PRPO builds on GRPO, encoding column-permutation invariance to transform sparse rewards into dense learning signals, thereby stabilizing training and enhancing cross-task generalization.

5 CONCLUSIONS

We presented TabR1, the first reasoning LLM for tabular prediction with Permutation Relative Policy Optimization (PRPO). By exploiting column-permutation invariance, TabR1 densifies sparse rewards and activates tabular reasoning ability with limited supervision. Experiments show that TabR1 achieves competitive supervised results and strong zero-shot transfer, surpassing larger LLMs while offering interpretable predictions.

References

- Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *KDD*, 2016.
- Noah Hollmann, Samuel Müller, et al. TabPFN: A transformer that solves small tabular classification problems in a second. In *NeurIPS*, 2022.
- Jan Toman, Noah Hollmann, et al. TabPFN v2: Expanded benchmarks and stronger baselines for tabular learning. *arXiv preprint arXiv:2406.XXXX*, 2024.
- Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, et al. Lightgbm: A highly efficient gradient boosting decision tree. In *NeurIPS*, 2017.
- Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: Unbiased boosting with categorical features. In *NeurIPS*, 2018.
- Yury Gorishniy, Ivan Rubachev, Valentin Khurlov, and Artem Babenko. Revisiting deep learning models for tabular data. In *NeurIPS*, 2021.

- Tom B Brown, Benjamin Mann, et al. Language models are few-shot learners. In *NeurIPS*, 2020.
- Jason Wei et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.
- Xuezhi Wang, Jason Wei, et al. Self-consistency improves chain of thought reasoning in language models. In *ICLR*, 2023.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Long Ouyang, Jeff Wu, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.
- Yuntao Bai et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Raphael Rafailov et al. Direct preference optimization: Your language model is secretly a reward model. In *ICLR*, 2024.
- DeepSeek-AI. Group relative policy optimization for llm reasoning. *arXiv preprint arXiv:2412.XXXX*, 2024.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning via reinforcement learning. *arXiv preprint arXiv:2501.XXXX*, 2025.
- Huanjin Yao, Qixiang Yin, Jingyi Zhang, Min Yang, Yibo Wang, Wenhao Wu, Fei Su, Li Shen, Minghui Qiu, Dacheng Tao, et al. R1-sharevl: Incentivizing reasoning capability of multimodal large language models via share-grpo. *arXiv preprint arXiv:2505.16673*, 2025.
- Pengcheng Yin, Graham Neubig, et al. Tabert: Pretraining for joint understanding of textual and tabular data. In *ACL*, 2020.
- Jonathan Herzig et al. Tapas: Weakly supervised table parsing via pre-training. In *ACL*, 2020.
- Xiang Deng, Huan Sun, et al. Turl: Table understanding through representation learning. In *VLDB*, 2020.
- Stefan Hegselmann et al. Tabllm: Few-shot classification of tabular data with large language models. *arXiv preprint arXiv:2305.XXXX*, 2023.
- Joshua P. Gardner, Juan C. Perdomo, and Ludwig Schmidt. Large-scale transfer learning for tabular data via language modeling. *arXiv preprint arXiv:2406.12031*, 2024. TabuLa-8B.
- Sercan O. Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *AAAI*, 2021.
- Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. In *NeurIPS*, 2020.
- Sergei Popov, Stanislav Morozov, and Artem Babenko. Neural oblivious decision ensembles for deep learning on tabular data. In *ICLR*, 2019.
- Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, Arjun Bansal, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. In *NeurIPS*, 2021.
- Vitaly Borisov, Tobias Leemann, Julian Seifert, et al. Deep neural networks and tabular data: A survey. *IEEE TPAMI*, 2022.
- Vishal Gupta et al. Tablegpt: Towards llms for table-based reasoning. *arXiv preprint arXiv:2305.13455*, 2023.
- Han Zhang et al. Gpt4tab: Can llms empowered by gpt-4 perform well on tabular data? *arXiv preprint arXiv:2306.01607*, 2023.
- Hugo Touvron et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- Li Bai et al. Qwen technical report. *arXiv preprint arXiv:2407.XXXX*, 2024.
- OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, and Tianhang Zhu. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

A PRPO Fine-tuning Setting for TabR1

The hyperparameters and training configurations of PRPO for TabR1 is provided in Table 5.

Table 5: Key hyperparameters and environment settings for PRPO fine-tuning on **TabR1**.

Parameter	Value
Base model	Qwen3-8B-Base
Training batch size	128
PPO mini-batch size	32
Micro-batch per GPU	4
Max prompt length	5120
Max response length	1024
Learning_rate	1×10^{-6}
KL_loss_coefficient	0.001
KL loss type	low_var_kl
Use KL in reward	False
Entropy coefficient	0
Rollout parallel size	2
Number of rollouts per sample	5
GPU memory utilization	0.6
Gradient checkpointing	Enabled
FSDP parameter offload	False
FSDP optimizer offload	False
Number of GPUs per node	8
Number of nodes	1
Total training epochs	30
CUDA version	12.6