How Does Text Help in Tables? A Systematic Evaluation of Text-Rich Tabular Learning

Anonymous authors
Paper under double-blind review

Abstract

Real-world tabular datasets often contain free-form text alongside numerical and categorical features, yet existing methods incorporate such textual information through substantially different interfaces and are typically evaluated under separate protocols. We present a systematic evaluation of text-rich tabular learning that compares four strategies under a unified framework: tabular-only prediction, embedding-augmented learning with frozen text representations, native text-aware modeling, and large language model (LLM) direct prediction across 44 datasets spanning binary classification, multiclass classification, and regression. Through controlled experiments and analyses, we establish three key findings: first, incorporating even basic text representations consistently enhances supervised tabular learners; second, the strongest embedding-augmented model outperforms nearly all native text-aware models; and third, LLM direct prediction underperforms both supervised paradigms due to numerical insensitivity and output compression. Together, these results show that the effectiveness of text-rich tabular prediction depends critically on how textual evidence is represented and integrated with structured supervision. Code and evaluation pipelines are available at `https://anonymous.4open.science/r/TextTableBench`.

1 Introduction

Real-world tabular datasets often contain more than numerical and categorical variables. Many practical prediction problems also involve high-cardinality strings and free-form text, including product descriptions, user-generated content, and professional records (Sebastiani, 2002). Such textual attributes can carry predictive signals that cannot be adequately captured by treating them merely as categorical identifiers (Turney & Pantel, 2010). A general-purpose tabular learning system should therefore leverage both interactions among structured features and the semantic information embedded in text. Understanding how to effectively incorporate these complementary sources of information is thus essential for advancing tabular prediction in real-world settings.

Mainstream tabular benchmarks predominantly emphasize numerical and low-cardinality categorical features, while string-valued columns are often excluded or converted into fixed representations during dataset-specific preprocessing (Bischl et al., 2017; Blayer et al., 2026). Reported rankings can therefore conflate model capacity with decisions about how textual features are represented, making it difficult to determine whether gains arise from stronger tabular learning, richer text representations, or their integration (McElfresh et al., 2023; Gijsbers et al., 2024). A meaningful comparison therefore demands a unified framework that evaluates text-integration strategies under consistent conditions.

Existing approaches to text-rich tabular prediction differ primarily in how textual information is exposed to the predictor. They may discard free text and rely solely on structured features, transform text into fixed representations using a frozen text encoder before tabular learning (Mráz et al., 2025), or jointly model textual and structured information within a unified architecture (Wang & Sun, 2022; Kim et al., 2024). General-purpose large language models (LLMs) provide a fourth route by serializing table rows as prompts and producing predictions directly (Hegselmann et al., 2023). These four strategies differ substantially in their representations, supervision mechanisms, inductive biases, and computational requirements, motivating a systematic comparison across tabular prediction tasks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing evaluations of text-rich tabular prediction span different modeling strategies and task settings. Native text integration has been evaluated through models such as TabSTAR (Arazi et al., 2026) and TabPFN-3-Plus (Grinsztajn et al., 2026), while Mráz et al. (2025) examine embedding-based text representations combined with conventional tabular learners. Complementing these supervised strategies, Grushina et al. (2026) evaluate direct prediction with general-purpose LLMs, primarily for binary classification. However, existing evaluations fail to cover all text-integration strategies, making it difficult to systematically compare the effectiveness of different strategies in utilizing textual information. A unified, large-scale evaluation across text-integration strategies and task types is therefore still needed.

We study how different strategies for integrating textual information with structured features shape predictive performance on text-rich tables. To this end, we compare four strategies—tabular-only baselines, embedding-augmented representations, native text-aware architectures, and LLM direct prediction—across 44 datasets spanning binary classification, multiclass classification, and regression within a unified evaluation framework. LLM direct prediction is further evaluated under multiple in-context learning settings. We assess performance through both aggregate comparisons and rigorous dataset-level paired analyses.

Our evaluation yields three main findings. First, incorporating textual information substantially improves supervised tabular learning on average, with even simple text representations yielding performance gains. Second, embedding-augmented and native text-aware approaches show mixed relative performance. Surprisingly, the strongest embedding-augmented learner outperforming nearly all native text-aware models. Third, LLM direct prediction remains less effective than both embedding-augmented and native text-aware approaches, with diagnostic analyses revealing limited numerical sensitivity and substantial output compression of LLM. Adding demonstrations further compresses the predictions and degrades predictive signal rather than improving performance.

Together, our study systematically evaluates how different strategies for integrating textual information with structured supervision shape predictive performance on text-rich tables. Our main contributions are:

- We establish a unified evaluation of four text-integration strategies across classification and regression, enabling controlled comparisons across diverse text-rich tabular prediction tasks.
- We characterize text gains, native-versus-frozen differences, and LLM prediction behavior through dataset-level paired analyses, encoder ablations, inference-time perturbations, and prediction diagnostics, providing complementary evidence for our key empirical findings.
- Our findings demonstrate that even basic text representations consistently enhance tabular prediction; the strongest embedding-augmented model outperforms nearly all native text-aware models; LLM direct prediction underperforms both supervised paradigms due to numerical insensitivity and output compression.

2 Related Work

Tabular prediction. Tabular prediction has evolved from task-specific tree ensembles and deep neural architectures to pretrained foundation models that perform inference across datasets. Gradient-boosted decision trees, including XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), and CatBoost (Prokhorenkova et al., 2018), capture nonlinear feature interactions and irregular decision boundaries while remaining relatively robust to irrelevant features. Deep tabular models instead learn task-specific representations: TabNet (Arik & Pfister, 2021) performs attention-based feature selection, whereas FT-Transformer (Gorishniy et al., 2021) models interactions among feature tokens. However, neither family is uniformly superior because they impose different inductive biases on the data (Grinsztajn et al., 2022). More recently, tabular foundation models such as TabPFN (Hollmann et al., 2025) and TabICL (Qu et al., 2025) have learned transferable priors from synthetic prediction tasks and condition on labeled rows to make predictions without conventional per-task optimization. Despite their methodological diversity, these models have been developed and evaluated primarily on conventional tables dominated by numerical and categorical features, leaving their ability to exploit semantically rich textual fields less systematically characterized.

Text-rich tabular learning. Text-rich tabular learning approaches differ primarily in how textual semantics are exposed to the predictor. Modular approaches provide frozen text representations to downstream tabular learners, separating semantic encoding from supervised prediction (Mráz et al., 2025). Native text-aware models, including CARTE (Kim et al., 2024), TabSTAR (Arazi et al., 2026), and TabPFN-3-Plus (Grinsztajn et al., 2026), more closely integrate textual and structured information, allowing text representations to interact with table structure or adapt to the prediction target. The principal distinction is therefore the degree to which textual representation and supervised tabular learning are coupled.

LLMs for tabular prediction. General-purpose LLMs perform tabular prediction by serializing rows as text and generating outputs from task instructions and, optionally, in-context examples (Dinh et al., 2022; Hegselmann et al., 2023). This interface can exploit pretrained knowledge but is sensitive to row serialization, demonstration selection, and output constraints, particularly for multiclass and regression tasks. LLMTabBench (Grushina et al., 2026) systematically evaluates direct LLM prediction under zero- and few-shot settings but focuses on binary classification. How this paradigm compares to tabular-only, embedding-augmented, and native text-aware approaches under a unified protocol—and across classification and regression—remains unexplored.

3 Benchmark and Evaluation Protocol

3.1 Problem Formulation

We consider supervised learning on text-rich tabular data, where each instance comprises both structured attributes and free-text fields. Formally, let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{t}_i, y_i)\}_{i=1}^{n}$ denote a dataset of n labeled samples. For the i-th sample, $\mathbf{x}_i$ represents its numerical and categorical features, while $\mathbf{t}_i = (t_{i1}, \ldots, t_{im_i})$ contains its $m_i \geq 1$ text fields. The target y_i is a discrete label $y_i \in \{1, \ldots, K\}$ for classification, or a continuous value $y_i \in \mathbb{R}$ for regression. The objective is to learn a predictive mapping

$$f : (\mathbf{x}_i, \mathbf{t}_i) \mapsto y_i$$

that jointly leverages structured feature interactions and unstructured textual semantics to generalize to unseen data.

3.2 Text-column Identification and Preprocessing

We identify free-text columns based on string characteristics and cardinality statistics. Specifically, a string-type column is classified as text if its entries have an average character length of at least 50 and an average token count of at least five. To distinguish semantic text from categorical variables or identifiers, we apply a low-cardinality exception: columns with fewer than 100 unique values and a uniqueness ratio $R < 20\%$ are retained as categorical features. This ratio is defined as

$$R = \frac{\text{Unique Count}}{N},$$

where N is the total number of instances. Consequently, high-cardinality, sentence-like columns are treated as free text, while repetitive string columns remain categorical.

Tabular-only baselines exclude these identified text fields entirely, whereas text-aware models integrate them natively. For embedding-augmented methods, frozen pre-trained encoders process these fields into dense representations, treating any missing values as empty strings. To prevent data leakage, PCA is fitted exclusively on the training split of each fold and then applied to transform the corresponding validation and test splits.

3.3 Dataset Collection and Characterization

We initially source 50 datasets from the public TabSTAR benchmark suite. To ensure the presence of genuine natural-language descriptions rather than short, low-cardinality categorical strings, we filter this corpus using the formal criteria outlined in Section 3.2. The resulting benchmark comprises 44 text-rich tabular datasets spanning diverse application domains, encompassing 4 binary classification, 11 multiclass classification, and 29 regression tasks. This curated collection provides

a highly heterogeneous evaluation landscape across varying sample sizes, feature dimensionalities, and textual characteristics. To ensure robust conclusions, all datasets are evaluated under a unified protocol with fixed data splits and task-specific metrics, focusing on aggregate performance rather than dataset-specific optimization.

3.4 Four Model Groups

We group the evaluated methods according to how text is exposed to the predictor. **Tabular-only models**, including XGBoost (Chen & Guestrin, 2016), LightGBM (Ke et al., 2017), TabPFN-2 (Hollmann et al., 2025), TabICLv2 (Qu et al., 2025), and TabPFN-3 (Grinsztajn et al., 2026), use only structured features, providing a reference for estimating the value of text. **Embedding-augmented models** combine fixed text representations with structured features before applying a tabular learner, thereby separating semantic encoding from supervised prediction. In our setup, we encode text fields using E5-small-v2 (Wang et al., 2022), reduce the embeddings to 50 dimensions via PCA, and concatenate them with structured features for the aforementioned tabular models. **Native text-aware models**, encompassing CatBoost (Prokhorenkova et al., 2018), CARTE (Kim et al., 2024), TabSTAR (Arazi et al., 2026), and TabPFN-3-Plus (Grinsztajn et al., 2026), jointly process textual and structured inputs within a unified architecture. **LLM direct prediction** serializes each row as a prompt and generates predictions under zero-shot or in-context settings without task-specific training, which we evaluate using the Qwen3 suite (Yang et al., 2025), GPT-4o (Achiam et al., 2023), GLM-5.2 (Zeng et al., 2026), and DeepSeek-V4 (Xu et al., 2026). Together, these groups span four text interfaces—exclusion, external representation, joint modeling, and LLM direct inference—and form the basis of our unified comparison.

3.5 LLM Prompting Protocol

We evaluate LLMs as direct predictors for tabular tasks using a unified prompting framework, where each prompt encapsulates the task description, dataset context, output schema, and a serialized query instance formatted as key-value pairs. Models are instructed to output predictions in a constrained JSON format, enabling consistent parsing across both classification and regression tasks.

To study demonstration scaling dynamics in detail, we first evaluate the Qwen3 suite (4B, 8B, and 14B) and DeepSeek-V4-Flash by systematically varying the demonstration budget across $k \in \{0, 4, 8, 16, 32, 64\}$. Furthermore, to isolate the performance gains brought by in-context learning across diverse architectures, we compare the zero-shot and 32-shot regimes across all six evaluated models—incorporating the Qwen3 suite alongside GPT-4o, GLM-5.2, and DeepSeek-V4-Pro. Across all few-shot settings, labeled demonstrations are sampled from the training split using class-balanced sampling for classification tasks to minimize selection bias, with evaluation conducted on up to 256 test instances per dataset under budget constraints.

During inference, we employ greedy decoding ($T = 0$, top-$p = 1.0$) and parse responses based on the predefined JSON schema. Any transient inference failures or formatting violations are automatically resolved through a retry mechanism, ensuring that reported metrics accurately reflect true predictive capacity rather than incidental generation artifacts.

3.6 Metrics and Statistical Analysis

To ensure consistent comparisons across heterogeneous datasets and task types, we adopt task-specific predictive metrics alongside dataset-level statistical analyses. We utilize ROC AUC as the primary classification metric, employing standard ROC AUC for binary tasks and macro-averaged one-vs-rest ROC AUC for multiclass scenarios. Regression performance is evaluated using R^2. For both metrics, higher values indicate superior performance.

Across datasets, we report the mean rank, win rate, and paired performance differences. Following standard practices for evaluating algorithms over multiple datasets (Demšar, 2006), statistical significance is assessed using two-sided Wilcoxon signed-rank tests with Holm's correction for multiple comparisons (Wilcoxon, 1945; Holm, 1979), while bootstrap confidence intervals quantify the uncertainty of paired differences. To account for initialization variance, each experiment is repeated over five independent runs with distinct random seeds (e.g., $42, 43, \ldots$). The resulting scores are averaged within each dataset before conducting dataset-level paired statistical tests.

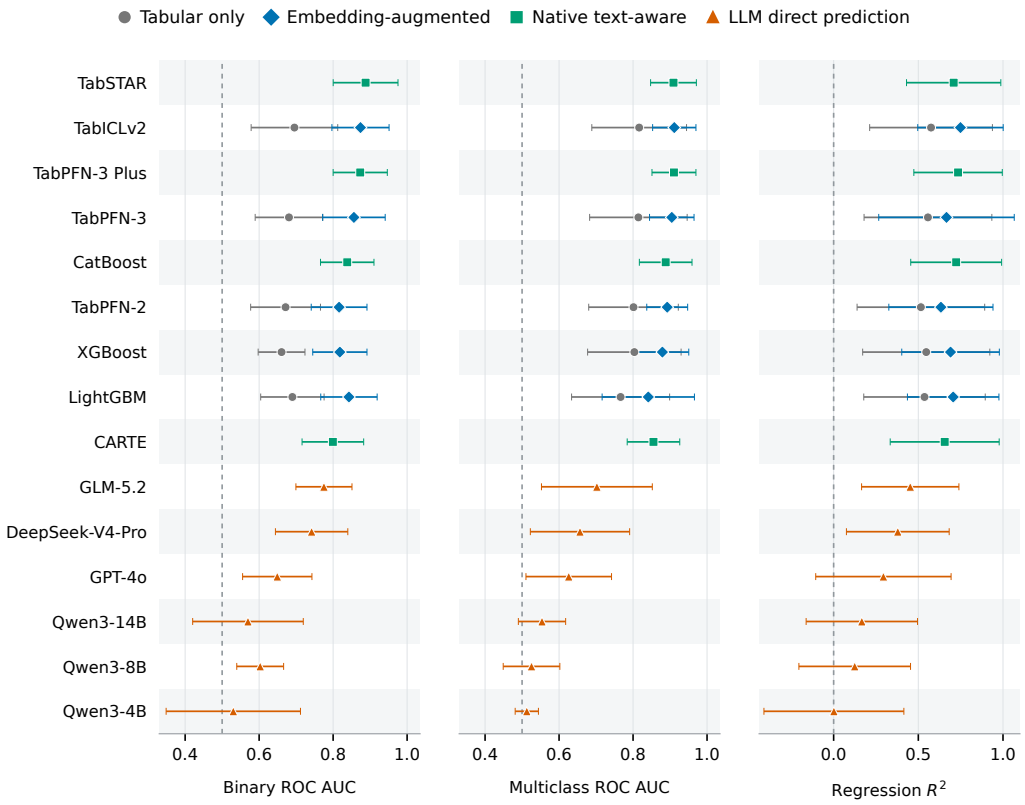

Figure 1: Overall performance across the four text-use strategies. Each row represents one model, with tabular-only and frozen-embedding variants aligned where applicable; all LLM results use zero-shot prompting. Binary and multiclass panels report ROC AUC, while regression reports R^2, with higher values indicating better performance.

4 Main Results

4.1 Overall Comparison across Four Strategies

As shown in Figure 1, the overall comparison reveals clear differences among the four text-integration strategies. Supervised methods that incorporate text consistently outperform their tabular-only counterparts, while embedding-augmented and native text-aware approaches achieve the strongest overall performance. At the model level, embedding-augmented TabICLv2 performs comparably to the leading native models, TabPFN-3-Plus and TabSTAR. This proximity suggests that native integration does not confer a uniform predictive advantage over strong externally encoded representations.

The relative performance of these strategies nevertheless varies across task types. Native models are particularly competitive in classification, whereas embedding-augmented approaches show a slight edge in regression. Tabular-only learners consistently trail methods that incorporate text, while direct LLM prediction shows its largest deficits in regression and multiclass classification. Overall, text provides broad predictive benefits, but the relative effectiveness of different integration strategies remains task-dependent. The following subsection tests the consistency of these text gains through dataset-level paired analyses, while Appendix A.5 further examines how model rankings and encoder uplift vary across application domains.

4.2 Does Text Help the Same Learner?

We isolate the contribution of text through paired comparisons between tabular-only and embedding-augmented variants of the same five learners. This controlled design holds the underlying learner

Table 1: Dataset-level effect of adding frozen text representations. Δ is embedding-augmented minus tabular-only, so positive values and W/T/L favour +text.

Comparison	n	Median Δ [95% CI]	W/T/L	p_{Holm}	r_{rb}
XGBoost + text	44	$+0.060\,[+0.023, +0.115]$	36/0/8	4.81×10^{-8}	$+0.873$
LightGBM + text	44	$+0.073\,[+0.035, +0.163]$	39/0/5	1.43×10^{-8}	$+0.899$
TabPFN-2 + text	44	$+0.052\,[+0.016, +0.100]$	36/0/8	3.09×10^{-6}	$+0.754$
TabICLv2 + text	44	$+0.092\,[+0.025, +0.132]$	37/0/7	5.66×10^{-9}	$+0.917$
TabPFN-3 + text	44	$+0.041\,[+0.019, +0.120]$	36/0/8	2.46×10^{-6}	$+0.778$

fixed, allowing the performance difference to be attributed to the addition of text representations. Across binary classification, multiclass classification, and regression, the resulting dataset-level deltas are predominantly positive, indicating that text gains are broadly distributed rather than driven by a small subset of datasets.

As shown in Figure 1 and Table 1, all five learners benefit significantly from text, with median paired gains ranging from $+0.041$ to $+0.092$. TabICLv2 shows the largest effect ($r_{\text{rb}} = 0.917$), while even the smallest median gain, observed for TabPFN-3, has a strictly positive 95% confidence interval ($[+0.019, +0.120]$). Embedding-augmented variants outperform their tabular-only counterparts on 36–39 of the 44 datasets, and all paired comparisons remain significant after Holm correction ($p_{\text{Holm}} \leq 3.09 \times 10^{-6}$).

The magnitude of these gains nevertheless varies across datasets. Exploratory analyses in Appendix B.4 show smaller gains on wider tables and those with more text columns across all five learners (average $\rho = -0.61$ and -0.48, respectively). Although these associations are correlational, they suggest that the marginal contribution of fixed-dimensional text representations may diminish as the feature space grows.

4.3 Frozen Embeddings versus Native Text-Aware Models

We evaluate the relative effectiveness of native text integration by comparing four native text-aware models against a fixed embedding-augmented baseline (TabICLv2 with E5/PCA50) across all 44 datasets. Differences are defined as native minus frozen performance, so negative values favour the frozen approach.

As shown in Table 2, native integration does not systematically improve predictive performance. Three of the four native models—TabSTAR, CatBoost, and CARTE—perform significantly below the frozen baseline, although the magnitude of the difference varies substantially across models. TabSTAR shows a moderate deficit, whereas CARTE underperforms on nearly all datasets. TabPFN-3-Plus is the exception, with performance statistically indistinguishable from the frozen baseline. These results establish embedding-augmented learning as a strong reference that native architectures do not uniformly surpass.

The relative performance of native and frozen approaches also varies with dataset size. Exploratory analysis in Appendix B.4 shows a positive association between the native-minus-frozen difference and dataset size across all four native models ($\rho = +0.34$ with $\log n$), indicating that native models become relatively more competitive as training data grows. This pattern is consistent with native integration benefiting from greater supervision, but the association is correlational and should not be interpreted as evidence of a causal scaling advantage.

5 Analysis and Ablations

5.1 Encoder and PCA Ablation

We examine whether the observed text benefit is robust to the choice of encoder and embedding dimension. The primary configuration uses E5-small-v2 with 50 PCA components, while the ablation additionally considers E5-small-v2/PCA30 and all-MiniLM-L6-v2 with either 30 or 50 components. As shown in Table 3, all four configurations are evaluated with the same five downstream learners.

Table 2: Dataset-level comparison of native text-aware models against the frozen-embedding baseline (TabICLv2 with E5/PCA50). Δ is native minus frozen, so negative values favour frozen.

Comparison	n	Median Δ [95% CI]	W/T/L	p_{Holm}	r_{rb}
CARTE − frozen	44	$-0.059\,[-0.083, -0.037]$	3/0/41	6.74×10^{-10}	-0.943
CatBoost − frozen	44	$-0.025\,[-0.038, -0.011]$	8/0/36	3.70×10^{-6}	-0.778
TabSTAR − frozen	44	$-0.006\,[-0.015, +0.000]$	16/0/28	0.0104	-0.477
TabPFN-3-Plus − frozen	44	$-0.002\,[-0.007, +0.002]$	20/0/24	0.1611	-0.244

Table 3: Encoder and PCA ablation across tasks. E30 and E50 denote E5-small-v2 with 30 and 50 PCA components, while M30 and M50 denote the corresponding all-MiniLM-L6-v2 configurations. E50 is the primary setting.

	Binary AUC				Multiclass AUC				Regression R^2			
Model	E30	**E50**	M30	M50	E30	**E50**	M30	M50	E30	**E50**	M30	M50
TabPFN-2	0.814	0.816	0.796	0.794	0.893	0.892	0.881	0.881	0.646	0.633	0.640	0.628
XGBoost	0.809	0.818	0.788	0.797	0.881	0.879	0.867	0.865	0.685	0.690	0.672	0.688
LightGBM	0.839	0.843	0.829	0.827	0.839	0.841	0.828	0.823	0.711	0.706	0.706	0.709
TabPFN-3	0.853	0.856	0.836	0.834	0.905	0.905	0.892	0.891	0.701	0.666	0.619	0.597
TabICLv2	0.865	0.874	0.857	0.859	0.912	0.911	0.900	0.899	0.752	0.749	0.748	0.744

The main pattern is stable across encoders and PCA dimensions. E5 generally yields higher classification performance than MiniLM, but the differences are modest relative to the gains over tabular-only inputs. Increasing the representation from 30 to 50 components does not produce a uniform improvement: its effect depends on both the learner and task, particularly in regression. Nevertheless, all configurations preserve the broader advantage of incorporating text. The ablation therefore supports the robustness of the main conclusion rather than identifying a universally optimal encoder–dimension combination. Moreover, a matched 50-dimensional TF-IDF (Salton & Buckley, 1988) baseline, reported in Appendix B.5, recovers much of the uplift on multiclass and regression tasks. Thus, substantial text gains do not appear to require pretrained semantic representations.

5.2 Text Sensitivity Ablation

Section 4.3 showed that native text-aware models do not consistently outperform a frozen-embedding baseline, leaving open whether this reflects weaker use of textual information. To test this possibility, we conduct an inference-time sensitivity ablation that measures how much predictive performance depends on aligned text. Each fitted model is evaluated without refitting under three conditions: **intact**, using the original text; **shuffled**, jointly permuting text rows to break text–label alignment while preserving the marginal text distribution; and **blanked**, replacing all text values with empty strings. Because blanking induces a severe distribution shift and can produce extreme negative R^2, we use $\Delta_{\text{shuffle}} = \text{score}_{\text{intact}} - \text{score}_{\text{shuffled}}$ as the primary measure of text reliance, where positive values indicate that aligned text contributes to prediction. The analysis covers all 44 datasets (15 classification and 29 regression), with five seeds per condition.

Table 4 shows substantial text reliance across all pipelines: median Δ_{shuffle} ranges from $+0.178$ to $+0.413$ and is positive on 40–44 of the 44 datasets. The two strongest native models, TabSTAR ($+0.413$) and TabPFN-3-Plus ($+0.376$), are also the most sensitive overall, exceeding every frozen pipeline. However, dataset-level paired tests against the shared frozen reference (TabICLv2 with E5/PCA50) reveal model-specific rather than family-wide differences. As shown in Table 5, after Holm correction, only TabPFN-3-Plus is significantly more sensitive, CatBoost is significantly less sensitive, while CARTE and TabSTAR do not differ significantly from the reference. Except for CatBoost, which relies on categorical statistics, these comparisons therefore provide no evidence that native models rely systematically less on text than the frozen baseline.

Table 4: Median scores and paired deltas under the three inference conditions across all 44 datasets, where Δ_{shuffle} = intact − shuffled and Δ_{blank} = intact − blanked.

Model	**Intact**	**Shuffled**	**Blanked**	$\boldsymbol{\Delta}_{\text{shuffle}}$	$\boldsymbol{\Delta}_{\text{blank}}$
Embedding-augmented (E5/PCA50)					
XGBoost	0.829	0.519	0.598	+0.280	+0.200
LightGBM	0.834	0.554	0.608	+0.242	+0.195
TabPFN-2	0.821	0.509	0.582	+0.178	+0.151
TabICLv2	0.870	0.521	0.638	+0.295	+0.215
TabPFN-3	0.867	0.508	0.562	+0.339	+0.221
Native text-aware					
CARTE	0.814	0.511	0.520	+0.307	+0.309
CatBoost	0.836	0.636	0.703	+0.215	+0.113
TabSTAR	0.851	0.481	0.505	+0.413	+0.350
TabPFN-3-Plus	0.877	0.492	0.605	+0.376	+0.276

Table 5: Dataset-level paired tests of text sensitivity against the shared frozen reference (TabICLv2 with E5/PCA50). $\Delta\Delta = \Delta^{\text{ref}}_{\text{shuffle}} - \Delta^{\text{native}}_{\text{shuffle}}$, so *negative* values indicate that the native model relies on text more than the reference. Two-sided Wilcoxon signed-rank tests with Holm correction within each stratum; p values are from the overall stratum.

	Overall (n=44)				Median $\Delta\Delta$	
Native model	Median $\Delta\Delta$ [95% CI]	W/T/L	p_{Holm}	r_{rb}	Cls	Reg
CARTE	$+0.049\,[+0.021, +0.103]$	31/0/13	0.0570	+0.378	+0.052	+0.025
CatBoost	$+0.088\,[+0.065, +0.124]$	42/0/2	1.1×10^{-11}	+0.984	+0.065	+0.127
TabSTAR	$-0.015\,[-0.140, +0.016]$	18/0/26	0.1442	−0.255	−0.015	+0.001
TabPFN-3-Plus	$-0.022\,[-0.086, +0.002]$	16/0/28	0.0264	−0.448	−0.006	−0.042

These results narrow the explanation for the performance differences observed in Section 4.3: native models generally depend on aligned text at least as strongly as the frozen baseline, yet this reliance does not consistently translate into higher predictive accuracy, suggesting that the difference lies less in text reliance itself than in how textual and structured evidence are jointly exploited. Importantly, the sensitivity ablation does not identify the underlying mechanism: greater performance degradation under shuffling indicates stronger dependence on aligned text, but not necessarily more effective text representations or better cross-modal integration. Rather, it shows that the native models' relative performance cannot generally be explained by a failure to use textual information. Appendix B.6 provides results for the blanked condition, task-specific scores, and the full paired comparisons.

5.3 General-Purpose LLM Direct Prediction

We first examine shot scaling within the Qwen3 family and DeepSeek-V4-Flash. As Figure 2 shows, performance varies non-monotonically from 0 to 64 shots: occasional gains at 4–8 shots do not persist, and all four models perform worse overall at 32 shots, driven primarily by regression, while the two smaller models show no degradation on multiclass classification. Dataset-level paired comparisons confirm that this pattern is not an artefact of averaging. As detailed in Appendix B.3, all six LLMs favour zero-shot on a majority of datasets, with strictly positive bootstrap intervals and significance retained after Holm correction.

Despite near-zero R^2, zero-shot regression retains meaningful predictive signal. Table 6 shows Spearman correlations of 0.37–0.70, while predicted standard deviations reach only 41–64% of the observed ones. Oracle isotonic recalibration using test labels raises R^2 to 0.28–0.62, suggesting that much of the zero-shot error reflects scale mis-calibration rather than a lack of predictive signal. This signal weakens markedly with demonstrations: at 32 shots, Spearman correlations fall to 0.15–0.24, oracle-recalibrated R^2 to 0.04–0.14, and the predicted-to-observed standard deviation ratio to 0.20–0.32. The compression extends to multiclass classification, where all six models produce predictions with lower normalized entropy than the label distribution, as detailed in Appendix B.7.

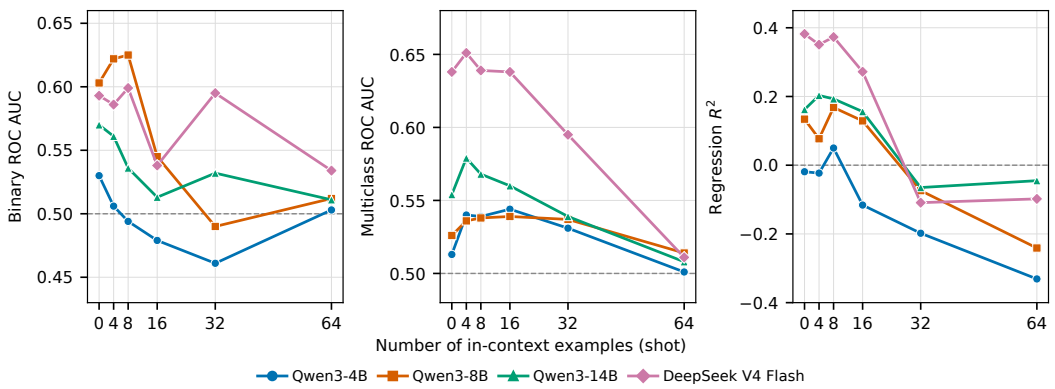

Figure 2: Shot-scaling behavior of the Qwen3 models and DeepSeek-V4-Flash. Performance is reported for 0, 4, 8, 16, 32, and 64 in-context examples. Binary and multiclass panels report mean ROC AUC, while regression reports mean R^2.

Table 6: Diagnostics for LLM regression predictions on the 29 regression datasets. ρ is the Spearman correlation with the true target. R^2_{iso} applies an isotonic recalibration fitted on the test labels and is therefore an oracle upper bound. Scale is the ratio of predicted to observed standard deviation. Entries are means over dataset–seed pairs.

	Zero-shot				32-shot			
Model	R^2	ρ	R^2_{iso}	Scale	R^2	ρ	R^2_{iso}	Scale
Qwen3-4B	0.002	0.37	0.28	0.41	−1.310	0.18	0.06	0.27
Qwen3-8B	0.124	0.44	0.36	0.43	−0.101	0.15	0.06	0.20
Qwen3-14B	0.166	0.49	0.38	0.48	−0.103	0.24	0.09	0.21
GPT-4o	0.294	0.64	0.56	0.64	−0.110	0.18	0.12	0.32
DeepSeek-V4-Pro	0.379	0.67	0.61	0.62	−0.021	0.18	0.13	0.28
GLM-5.2	0.452	0.70	0.62	0.61	−0.054	0.22	0.14	0.24

These diagnostics explain part of the weak LLM performance but not the broader gap to supervised learning. Even under favorable oracle recalibration for regression and tie resolution for classification, the strongest LLM results remain below embedding-augmented TabICLv2. Thus, miscalibration and output compression contribute to the observed gap but do not fully explain it.

These analyses are diagnostic rather than causal, but they reveal a consistent effect of demonstrations: both output dispersion and predictive ordering deteriorate as examples are added. One possible explanation is that longer contexts dilute attention to the query row and bias predictions toward the example-label distribution, consistent with prior observations in long-context settings (Liu et al., 2024). The precise mechanism behind this compression remains an open question.

6 Conclusion

We present a unified evaluation of four text-integration strategies across 44 datasets spanning binary classification, multiclass classification, and regression. Our results show that incorporating even basic text representations substantially improves supervised tabular learning, while embedding-augmented and native text-aware approaches exhibit mixed relative performance, with the strongest embedding-augmented learner outperforming nearly all native models. In contrast, LLM direct prediction remains less effective than both supervised approaches, with diagnostic analyses revealing limited numerical sensitivity and substantial output compression. Overall, our findings provide a systematic empirical characterization of how different text-integration strategies shape predictive performance on text-rich tables.

AI USE STATEMENT

In this work, we used generative AI tools for drafting parts of the research paper, editing the manuscript to improve readability, formatting references, and suggesting structure for the paper. We used these tools for brainstorming, sourcing and searching for information, and identifying relevant literature. Additionally, we used generative AI tools for implementing methods and assisting with experimental code. We did not use generative AI tools for generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proving mathematical claims, proposing or refining hypotheses, designing research methodology or experiments, cleaning or reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results.

We have reviewed all AI-assisted work. AI-generated code was verified and tested for correctness by the authors. AI-assisted text was checked against the source data and experimental results for accuracy. All AI-generated reference suggestions were manually verified against the original sources.

We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study does not involve human subjects, human annotations, or user studies. All experiments use publicly available benchmark datasets previously released for tabular prediction research; we do not collect, process, or release any personally identifiable information.

We foresee no direct harmful applications arising from this work. All methods evaluated are standard supervised learning and prompting techniques applied to public data; no new capabilities that lower the barrier to harmful use are introduced.

Our benchmark spans diverse domains and is not intended as a fairness audit. We do not claim that any evaluated method is fair or unbiased in any absolute sense; our goal is methodological comparison, not model certification for deployment.

We have no conflicts of interest or external sponsorship to declare. All LLM outputs were generated with documented model versions and configurations to ensure reproducibility. This research raises no legal compliance, IRB, or research integrity concerns.

REPRODUCIBILITY STATEMENT

To ensure full reproducibility of our benchmark and experimental findings, we have made the complete source code, evaluation scripts, and dataset preprocessing pipelines publicly available in an anonymous repository at `https://anonymous.4open.science/r/TextTableBench`. Detailed descriptions of the dataset curation process, including the strict text-field filtering criteria, are provided in the main text and supplementary materials. Furthermore, all experimental setups—encompassing the unified evaluation protocol, specific data splits, and precise model hyperparameters for the tabular-only, embedding-augmented, native text-aware, and LLM direct prediction pipelines—are thoroughly documented in the experimental sections and the Appendix to facilitate exact replication of our results.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alan Arazi, Eilam Shapira, and Roi Reichart. Tabstar: A tabular foundation model for tabular data with text fields. *Advances in Neural Information Processing Systems*, 38:172108–172161, 2026.

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 6679–6687, 2021.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael G Mantovani, Jan N van Rijn, and Joaquin Vanschoren. Openml benchmarking suites. *arXiv preprint arXiv:1708.03731*, 2017.

Gioia Blayer, Myung Jun Kim, Félix Lefebvre, Lennart Purucker, Alan Arazi, Eilam Shapira, Roi Reichart, Frank Hutter, Marine Le Morvan, David Holzmüller, et al. Strable: Benchmarking tabular machine learning with strings. *arXiv preprint arXiv:2605.12292*, 2026.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. *Advances in Neural Information Processing Systems*, 35:11763–11784, 2022.

Pieter Gijsbers, Marcos LP Bueno, Stefan Coors, Erin LeDell, Sébastien Poirier, Janek Thomas, Bernd Bischl, and Joaquin Vanschoren. Amlb: an automl benchmark. *Journal of Machine Learning Research*, 25(101):1–65, 2024.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

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

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Daria Grushina, Kseniia Kuvshinova, Alina Kostromina, Aziz Temirkhanov, Mile Mitrovic, and Dmitry Simakov. Llmtabbench: Evaluating llms on binary tabular classification from zero to few shots. *arXiv preprint arXiv:2605.24417*, 2026.

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

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pp. 65–70, 1979.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Myung Jun Kim, LĂŠ Grinsztajn, and GaĂŤl Varoquaux. Carte: pretraining and transfer for tabular learning. *arXiv preprint arXiv:2402.16785*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *Advances in neural information processing systems*, 36:76336–76369, 2023.

Martin Mráz, Breenda Das, Anshul Gupta, Lennart Purucker, and Frank Hutter. Towards benchmarking foundation models for tabular data with text. *arXiv preprint arXiv:2507.07829*, 2025.

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

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

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Zifeng Wang and Jimeng Sun. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915, 2022.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83, 1945.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

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

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

A Appendix

A.1 Dataset Construction and Statistics

The final benchmark contains 44 text-rich tabular datasets retained from the TabSTAR benchmark, including 4 binary classification, 11 multiclass classification, and 29 regression datasets, as summarized in Table 7.

The retained 44 datasets exhibit substantial variation in both data scale and feature structure. Across the benchmark, the number of samples ranges from 800 to more than 7.7 million, while the number of curated features ranges from 2 to 112. In terms of textual features, the benchmark contains more than 200 text columns across all datasets. The average text length varies considerably, from under 20 characters per cell in some short-attribute datasets to over 2,000 characters in description-rich datasets, reflecting the diversity of textual inputs — from short categorical-like text attributes to long-form natural-language descriptions. Accordingly, the benchmark provides heterogeneous evaluation conditions for text-aware tabular prediction.

Table 7: Dataset statistics for datasets with textual features

Dataset	Task	n	m	#Text	Avg. Text Len	#Classes
used-cars-dataset-cardekho	Regression	37,814	112	14	146.0	—
animeplanet-recommendation	Regression	14,391	14	4	92.0	—
Babies-R-Us	Regression	5,085	12	3	20.2	—
employee_salaries	Regression	9,228	11	4	21.2	—
california_house_price	Regression	37,951	39	8	112.8	—
coffee-scrap-coffeereview	Regression	2,440	17	7	153.7	—
BikeWale	Regression	9,003	6	1	26.7	—
bookprice_prediction	Regression	4,989	8	5	221.5	—
ae_price_prediction	Regression	22,662	12	5	143.9	—
filmtv-movies-dataset	Regression	41,399	17	5	164.4	—
free-7-million-company-dataset	Regression	7,173,426	7	4	27.5	—
museums	Regression	22,290	21	9	19.4	—
vivino-wine-data	Regression	8,650	6	1	22.2	—
wikilq-dataset	Regression	12,569	12	4	126.6	—
beer-profile-and-ratings	Regression	3,197	24	4	70.0	—
korean-drama	Regression	1,647	9	6	190.5	—
videogamesales	Regression	16,598	5	1	24.0	—
zomato-bangalore-restaurants	Regression	41,665	15	6	2,093.6	—
the-movies-dataset	Regression	45,460	20	12	69.4	—
nba-draft-basketball	Regression	1,669	22	1	12.9	—
Goodreads	Regression	3,967	14	8	122.8	—
Rotten-Tomatoes	Regression	7,158	15	10	52.6	—
top-ramen-ratings-2022	Regression	4,105	4	1	29.5	—
Journal-Score-SJR	Regression	31,136	21	6	30.9	—
chocolate-bar-ratings	Regression	1,795	8	2	13.0	—
merci_price_suggestion100K	Regression	100,000	9	3	67.3	—
wine-price-on-polish-market	Regression	2,247	18	5	21.6	—
clear-corpus	Regression	4,724	30	6	187.5	—
jc_penney_products	Regression	10,860	5	2	195.2	—
spotify-tracks-dataset	Classification	114,000	18	3	18.1	114
women_clothing_review	Classification	18,788	10	2	164.2	5
us-accidents	Classification	7,728,394	42	3	35.6	4
data_scientist_salary	Classification	15,841	6	3	69.2	6
imdb_genre_prediction	Classification	800	11	4	62.5	2
product_sentiment_machine_hack	Classification	5,091	2	1	104.5	4
google_qa_question_type_reason	Classification	4,863	39	8	240.0	5
michelin-guide-restaurants-2021	Classification	17,735	11	5	138.6	5
fake_job_postings2	Classification	12,725	5	3	430.2	2
jigsaw_unintended_bias100K	Classification	100,000	40	2	161.7	2
yelp-reviews-dataset	Classification	10,000	5	1	710.7	5
news_channel	Classification	20,284	17	1	54.5	6
wine_reviews	Classification	84,123	5	1	243.2	30
kick_starter_funding	Classification	86,502	9	5	56.3	2
melbourne_airbnb	Classification	18,316	89	22	216.6	10

A.2 Dataset Clusters

The 44 retained datasets form several heterogeneous clusters along two dimensions: dataset scale and domain–task composition. As shown in Figure 3(a), the number of instances spans several orders of magnitude, from fewer than 10^3 to more than 10^7, while the number of features ranges from only a few to more than 100. Most datasets are located in the low- to medium-scale region, whereas a small number of large datasets, such as the CarDekho and US Accidents datasets, form distinct large-scale cases. The plot also shows that dataset dimensionality is not strongly determined

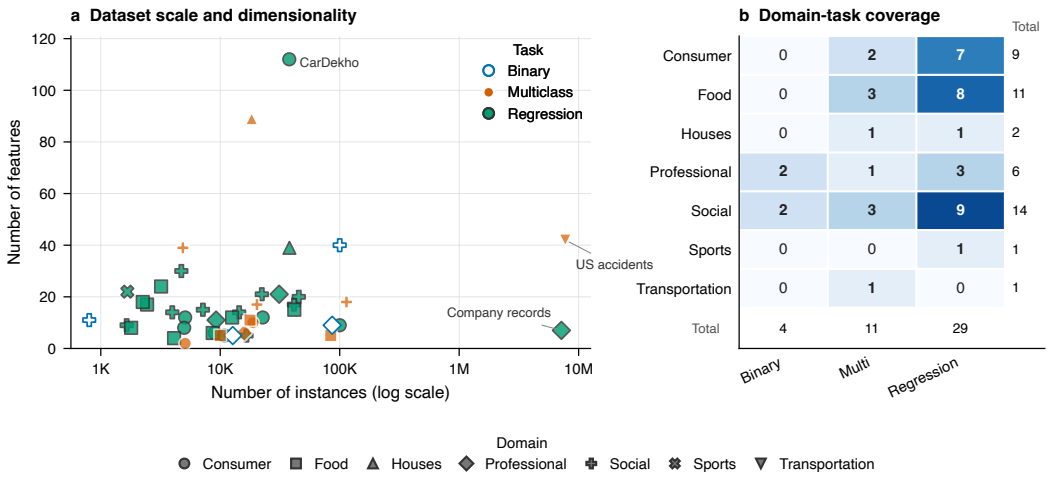

Figure 3: Scale, dimensionality, and domain–task composition of the 44 retained text-rich tabular datasets. Panel (a) shows the number of instances on a logarithmic scale against the number of curated features, with marker shapes indicating application domains and colors indicating task types. Panel (b) reports the number of datasets for each domain–task combination; the margins show the corresponding task and domain totals.

by sample size: datasets with similar numbers of instances may differ substantially in their numbers of features.

Panel (b) summarizes the domain–task composition of the benchmark. Regression datasets constitute the largest group, with 29 datasets, followed by 11 multiclass and 4 binary classification datasets. Social, Food, and Consumer domains contribute the largest numbers of datasets, whereas Houses, Sports, and Transportation are represented more sparsely. The benchmark therefore covers both differences in prediction objectives and differences in application domains, enabling comparisons across diverse text-rich tabular settings rather than classification-only tasks.

A.3 Data splits and sample limits

Supervised Models. We apply the same data-curation and splitting pipeline across model families. Each run uses a random seed from 42–46. The nominal test size is 10% of the dataset and is capped at 20,000 instances. For datasets exceeding the downstream training limit, the data are first subsampled to retain at most 100,000 training instances together with the corresponding test portion. Classification splits are stratified whenever feasible, with explicit handling of binary and singleton-class cases. Models requiring validation data draw 10% of the available training partition, capped at 1,000 instances; this validation subset is used only for model fitting and selection.

LLM direct prediction. The LLM evaluator reuses the same curated data and outer train–test splitting pipeline, with a default training cap of 100,000 instances. The evaluated LLM runs use seeds 42–46, and each seed is evaluated on at most 256 test instances. Binary and regression test sets are truncated to the first 256 rows of the already randomized test split, whereas multiclass subsets are sampled reproducibly while ensuring that every test-set class is represented. Evaluation fails if the row budget cannot cover all classes.

The LLM evaluator uses a few-shot prompting setting. Classification demonstrations are selected to cover the label set before filling the remaining positions, while regression demonstrations are selected across target quantiles. Demonstrations are drawn only from the training partition. LLM decoding is deterministic, with temperature $T = 0$ and top-$p = 1$.

A.4 LLM PROMPTING PROTOCOL

We evaluate direct large language models as row-level predictors. For each dataset, the model receives the task type, dataset identifier, optional dataset description, feature information, and one query row. For few-shot evaluation, the prompt additionally contains labeled training examples sampled exclusively from the training partition. For zero-shot evaluation, the example list is empty. The model must return a single JSON object that conforms to a task-specific schema; explanatory text, Markdown, and additional fields are disallowed.

For binary classification, the output contains the predicted label and a continuous score for the positive class. The latter is used to compute ROC AUC and is not required to be a calibrated probability, as shown in Figure 4. For multiclass classification, the model returns the predicted class index and one score for each class, with the score order matching the class list, as shown in Figure 5. For regression, the model returns one numeric prediction, as illustrated in Figure 6. The prompt also provides lightweight training-set summaries, such as class frequencies for classification and target quantiles for regression, which serve as contextual information rather than additional test data.

LLM decoding uses temperature $T = 0$ and top-$p = 1$. Invalid or non-conforming outputs are parsed and recorded as invalid rather than being treated as valid predictions.

Figure 4: Binary-classification prompt.

```
System:
You are a careful assistant that follows instructions and outputs only
 valid JSON.
Return only a single valid JSON object. Do not include analysis text.

User:
You are evaluating exactly one tabular row.
Do not copy any input section.
Return only OUTPUT_JSON and nothing else.
TASK: binary_classification
DATASET: BIN_PROFESSIONAL_FAKE_JOB_POSTING
LABELS_JSON: {"negative": "0", "positive": "1"}

DATASET_INTRO:
Predict whether online job postings are real or fake (binary
classification).
The target variable 'fraudulent' has two classes: '0' (real) and '1' (
fake).
Features include the job title, salary range, job description,
required experience,
and required education. ...

OUTPUT_JSON_SCHEMA:
{"prediction": "1", "score_for_positive": 0.0}

RULES:
- Return ONLY the JSON object.
- The prediction must be either '0' or '1'.
- score_for_positive must be a number between 0 and 1.
- Do not add extra keys.
- Use the query row values whenever possible.

EXAMPLES_JSON_ARRAY:
[]

QUERY_JSON:
{"x": {"title": "All round web design superstar for cutting edge non-
profit websites",
       "salary_range": null,
       "description": "UsRaising IT is growing quickly! ...",
```

```
       "required_experience": null,
       "required_education": null}}
```

Figure 5: Multiclass-classification prompt.

```
System:
You are a careful assistant that follows instructions and outputs only
 valid JSON.
Return only a single valid JSON object. Do not include analysis text.

User:
You are evaluating exactly one tabular row.
Do not copy any input section.
Return only OUTPUT_JSON and nothing else.
TASK: multiclass_classification
DATASET: MUL_CONSUMER_PRODUCT_SENTIMENT
CLASSES_JSON:
["Cannot Say", "Negative", "No Sentiment", "Positive"]

CLASS_DISTRIBUTION_JSON:
{"Cannot Say": {"count": 87, "prior": 0.01899},
 "Negative": {"count": 292, "prior": 0.06373},
 "No Sentiment": {"count": 1488, "prior": 0.32475},
 "Positive": {"count": 2715, "prior": 0.59254}}

DATASET_INTRO:
Classify the sentiment of user reviews for products into four
categories:
'Cannot Say', 'Negative', 'No Sentiment', and 'Positive'. The target
variable
'Sentiment' is numeric with values 0 to 3 corresponding to these
classes.
The dataset includes product review text and product type. ...

OUTPUT_JSON_SCHEMA:
{"prediction_index": 0, "scores": [0.0, 0.0, 0.0, 0.0]}

RULES:
- Return ONLY the JSON object.
- prediction_index is the zero-based index in CLASSES_JSON.
- The scores array must match the class order and length.
- Each score must be between 0 and 1.
- Do not add extra keys.

EXAMPLES_JSON_ARRAY:
[]

QUERY_JSON:
{"x": {"Product_Description":
       "Tomlinson of TX Observer says Apple subscription data holding
...",
       "Product_Type": 2}}
```

Figure 6: Regression prompt.

```
System:
You are a careful assistant that follows instructions and outputs only
 valid JSON.
Return only a single valid JSON object. Do not include analysis text.
```

```
User:
You are evaluating exactly one tabular row.
Do not copy any input section.
Return only OUTPUT_JSON and nothing else.
TASK: regression
DATASET: REG_CONSUMER_AMERICAN_EAGLE_PRICES
TARGET_NAME: price

DATASET_INTRO:
Predict the price of inner-wear items sold by American Eagle using
text features
such as product name and description, categorical features such as
brand and
product category, and numerical features such as rating and review
count. ...

TARGET_SUMMARY_JSON:
{"min": 3.99, "p05": 9.5, "median": 13.47,
 "p95": 31.46, "max": 51.09}

FEATURE_SUMMARY_JSON:
{"product_name": {"type": "text_or_categorical", ...},
 "brand_name": {"type": "text_or_categorical", ...},
 "rating": {"type": "numeric", ...},
 "review_count": {"type": "numeric", ...}}

OUTPUT_JSON_SCHEMA:
{"prediction": 0.0}

RULES:
- Return ONLY the JSON object.
- Return a numeric prediction.
- Use the examples and target summary to calibrate the target scale.
- Use the query feature values rather than only the median.
- Keep the prediction within a plausible target range.
- Do not add extra keys.

EXAMPLES_JSON_ARRAY:
[]

QUERY_JSON:
{"x": {"product_name": "Happy Unlined Plunge Bra",
       "mrp": "29.95 USD",
       "brand_name": "AERIE",
       "product_category": "Plunge",
       "rating": 4.4,
       "review_count": 63.0,
       "description": "Hello, Happy Bras. Happiness is feeling like
the real you. ..."}}
```

A.5 Domain-Level Model Profiles and Text Uplift

To examine how model behavior varies across application settings, we report two complementary domain-level views. Figure 7 shows mean within-dataset percentile ranks within each text-integration paradigm, with 0 denoting the lowest-ranked model and 100 the highest-ranked model; classification and regression receive equal weight within each domain. Because each panel is normalized separately, the profiles describe within-paradigm model ordering rather than absolute performance differences between paradigms. The leading model varies across domains, while Sports and Transportation are omitted because the task-balanced profile requires both classification and regression coverage.

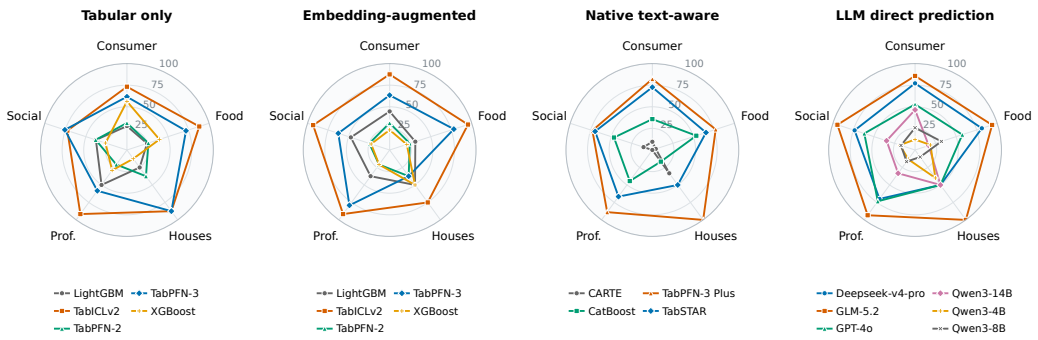

Figure 7: Domain-level model profiles for the four text-integration paradigms. Each vertex is the mean within-dataset percentile rank within the corresponding paradigm, where 0 denotes the lowest-ranked model and 100 the highest-ranked model. Classification and regression receive equal weight; only domains represented in both task groups are shown. Scores are normalized separately within each panel and therefore do not measure absolute differences between paradigms.

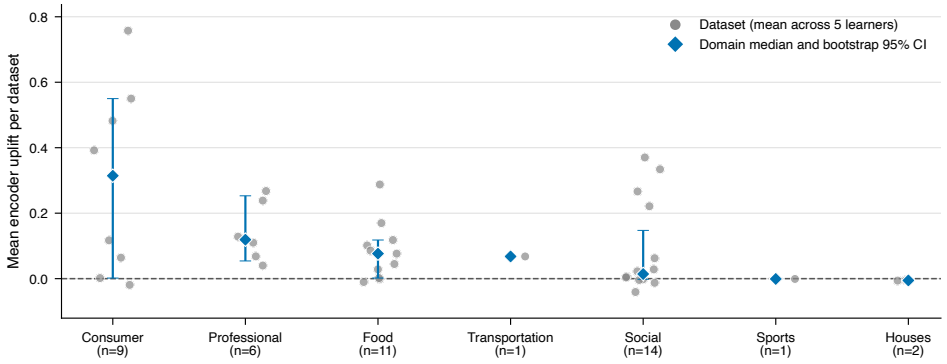

Figure 8: Dataset-level encoder uplift by application domain. Each grey point is one dataset after averaging the E5-small-v2/PCA50 minus tabular-only delta across five supervised learners. Blue diamonds show domain medians and error bars show 10,000-resample bootstrap 95% confidence intervals for those medians. Domains are ordered by median uplift.

Figure 8 addresses the narrower question of where frozen text embeddings help. To avoid pseudo-replication, we first average the E5-small-v2/PCA50 minus tabular-only delta across the five supervised learners within each dataset, leaving datasets as the independent units. Consumer has the largest median uplift, while Professional and Food also have positive domain medians. Social has a near-zero median despite substantial between-dataset variation. Estimates for Transportation ($n = 1$), Sports ($n = 1$), and Houses ($n = 2$) remain descriptive because of limited coverage. Together, the two views distinguish domain-dependent model ordering from the incremental benefit of text.

A.6 Hardware Specifications and Model Runtime

Hardware Specifications We conduct our experiments across two server configurations. Supervised models run on NVIDIA Tesla V100-SXM2 GPUs with 32 GB memory, powered by dual Intel Xeon Gold 6133 CPUs @ 2.50 GHz with 80 cores. Qwen3 runs on an NVIDIA GeForce RTX 4090 GPU with 48 GB memory, powered by dual Intel Xeon Platinum 8368Q CPUs @ 2.60 GHz with 152 cores.

Training, Inference, and Runtime Configurations We document the training requirements and inference deployment for all evaluated models in Table 8, providing transparency for reproducibility and cost-awareness.

Models fall into three categories based on their training paradigm. The gradient-boosted trees (XGBoost, LightGBM, and CatBoost) and CARTE are trained from scratch on each dataset, with CatBoost additionally supporting native text input. TabSTAR requires fine-tuning its pre-trained components on the target dataset. In contrast, TabPFN variants and TabICLv2 are used in a prior-fitting fashion without per-dataset training; they directly produce predictions given the training set. LLMs are used without any task-specific training, relying solely on prompting.

For the Qwen3 series models, we use vLLM (Kwon et al., 2023) as the inference backend, which provides efficient serving through PagedAttention and continuous batching. Commercial LLMs—GPT-4o, GLM-5.2, and DeepSeek-V4-Pro—are accessed through their respective APIs.

For embedding-augmented pipelines, the tabular learner is applied after encoding text fields; the additional inference cost is incurred by the encoder forward pass and PCA projection. This overhead is included in the reported inference times for these models.

A.7 Computational Efficiency

As illustrated in Figure 9, the recorded wall-clock time varies drastically across different integration strategies. Unsurprisingly, tabular-only baselines are the most time-efficient, uniformly completing execution within 25 minutes. LLM direct prediction paradigms also maintain moderate and consistent execution times, largely bounded between 18 and 75 minutes regardless of whether they utilize local deployment or remote APIs. Meanwhile, extracting and integrating text embeddings introduces substantial computational overhead across the remaining learners.

Among native text-aware architectures, execution times exhibit a vast spectrum. Most notably, TabSTAR requires a staggering 3,070 minutes on local GPUs, reflecting the extensive fine-tuning needed to effectively align tabular and textual signals. In stark contrast, TabPFN-3-Plus circumvents these local training bottlenecks via a remote API, achieving text-aware predictions in just 53 minutes. Ultimately, these results demonstrate that the computational footprint of incorporating text varies widely across paradigms, governed primarily by model training intensity and execution infrastructure.

Table 8: Training requirements, inference deployment, and time ratios for all evaluated models. The "Ratio" column indicates the multiplicative increase in inference time when text embedding augmentation is applied.

Model	Training	Inference route	Base/native/direct	Embedding-augmented	Ratio
XGBoost	✓ Fit	Local GPU	13 min	129 min	9.9×
TabPFN-3	×	Local GPU	20 min	170 min	8.4×
TabPFN-2	×	Local GPU	15 min	181 min	11.9×
Qwen3-14B	×	Local GPU	59 min	—	—
Qwen3-4B	×	Local GPU	18 min	—	—
Qwen3-8B	×	Local GPU	36 min	—	—
DeepSeek-V4-Pro	×	Remote API	33 min	—	—
GLM-5.2	×	Remote API	75 min	—	—
GPT-4o	×	Remote API	62 min	—	—
LightGBM	✓ Fit	Local GPU	18 min	56 min	3.1×
TabICLv2	×	Local GPU	24 min	118 min	4.9×
CatBoost	✓ Fit	Local GPU	440 min	—	—
TabSTAR	✓ Fine-tune	Local GPU	3070 min	—	—
CARTE	✓ Fit	Local GPU	127 min	—	—
TabPFN-3 Plus	×	Remote API	53 min	—	—

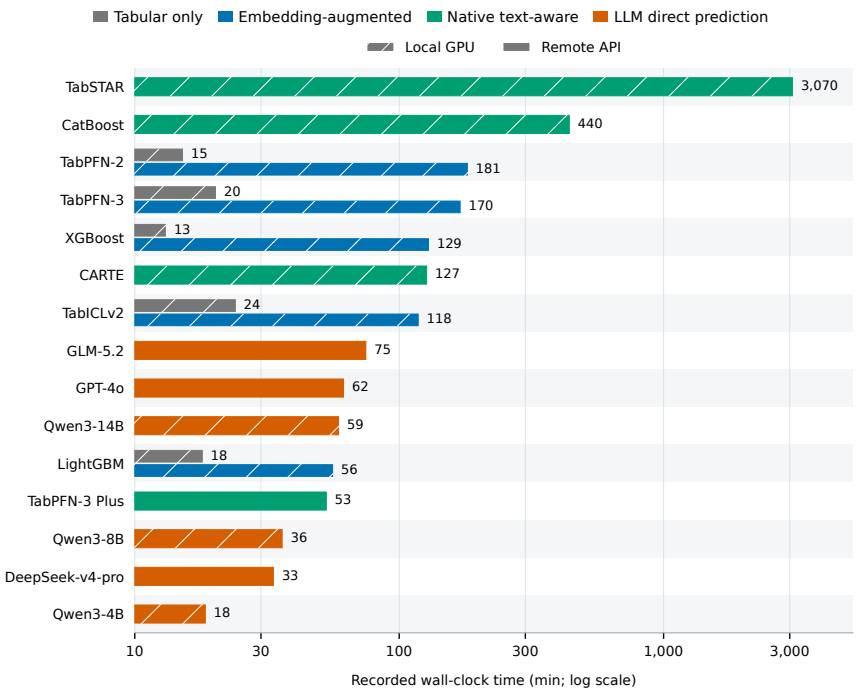

Figure 9: Recorded wall-clock time (in minutes, log scale) across the evaluated integration strategies. The chart distinguishes between execution environments, where hatched bars indicate computation on local GPUs and solid bars denote delegation via remote APIs.

B Supplementary Results

B.1 Complete Model Results

Table 9 reports the complete benchmark results for all evaluated models under the unified evaluation protocol, covering binary classification, multiclass classification, and regression. The results are organized according to the four model groups considered in our study: tabular-only, embedding-augmented, native text-aware, and LLM direct prediction. For embedding-augmented models, we report the main E5-small-v2 with 50 PCA components configuration, where frozen text representations are reduced to 50 dimensions before being combined with the structured features. For LLM direct prediction, the table reports the zero-shot results. Together, these complete results provide the numerical basis for the cross-paradigm comparisons presented in the main text.

B.2 Complete LLM Direct Prediction Results

We present the complete zero-shot and 32-shot results for all six evaluated LLMs across the three task types in Table 10. The table reports aggregate mean performance for each model under both prompting regimes, alongside the best supervised text-aware result from our main evaluation for reference.

Overall, these detailed results confirm that direct LLM prediction, while capable of producing reasonable performance on some text-rich tabular tasks, does not approach the level of supervised text-aware models. The systematic zero-shot advantage over 32-shot is not an artifact of aggregation but is driven predominantly by regression, where all six models are significant with large effect sizes; on classification the effect is directionally consistent but not resolvable at $n = 15$, as detailed in Appendix B.3. This refines rather than contradicts the conclusions of the main text.

Table 9: Aggregate performance across text-use strategies on 44 datasets. Values are reported as mean ± standard deviation across datasets.

Model / setting	**Strategy**	**Binary AUC**	**Multiclass AUC**	**Regression R^2**
XGBoost	Tabular only	0.661 ± 0.063	0.804 ± 0.126	0.547 ± 0.375
LightGBM	Tabular only	0.690 ± 0.086	0.766 ± 0.133	0.537 ± 0.359
TabPFN-2	Tabular only	0.671 ± 0.094	0.801 ± 0.121	0.515 ± 0.377
TabPFN-3	Tabular only	0.681 ± 0.091	0.814 ± 0.132	0.557 ± 0.377
TabICLv2	Tabular only	0.695 ± 0.117	0.817 ± 0.128	0.575 ± 0.363
XGBoost	Embedding-augmented	0.818 ± 0.073	0.879 ± 0.071	0.690 ± 0.288
LightGBM	Embedding-augmented	0.843 ± 0.076	0.841 ± 0.125	0.706 ± 0.270
TabPFN-2	Embedding-augmented	0.816 ± 0.075	0.892 ± 0.055	0.633 ± 0.308
TabPFN-3	Embedding-augmented	0.856 ± 0.085	0.905 ± 0.060	0.666 ± 0.400
TabICLv2	Embedding-augmented	0.874 ± 0.077	0.911 ± 0.059	0.749 ± 0.252
CatBoost	Native text-aware	0.838 ± 0.072	0.888 ± 0.071	0.723 ± 0.268
CARTE	Native text-aware	0.799 ± 0.083	0.855 ± 0.071	0.656 ± 0.322
TabSTAR	Native text-aware	0.888 ± 0.088	0.909 ± 0.062	0.709 ± 0.278
TabPFN-3-Plus	Native text-aware	0.873 ± 0.073	0.911 ± 0.059	0.735 ± 0.261
Qwen3-4B	LLM direct prediction	0.530 ± 0.182	0.513 ± 0.031	0.002 ± 0.413
Qwen3-8B	LLM direct prediction	0.603 ± 0.063	0.526 ± 0.077	0.124 ± 0.330
Qwen3-14B	LLM direct prediction	0.570 ± 0.150	0.554 ± 0.064	0.166 ± 0.329
GLM-5.2	LLM direct prediction	0.775 ± 0.076	0.702 ± 0.150	0.452 ± 0.287
DeepSeek-V4-Pro	LLM direct prediction	0.742 ± 0.098	0.657 ± 0.134	0.379 ± 0.303
GPT-4o	LLM direct prediction	0.649 ± 0.094	0.626 ± 0.116	0.294 ± 0.400

Table 10: Direct LLM performance under zero-shot and 32-shot prompting

	Binary AUC		Multiclass AUC		Regression R^2	
Model	0-shot	32-shot	0-shot	32-shot	0-shot	32-shot
Qwen3-4B	0.530	0.461	0.513	0.531	0.002	−1.311
Qwen3-8B	0.603	0.490	0.526	0.537	0.124	−0.101
Qwen3-14B	0.570	0.532	0.554	0.539	0.166	−0.103
GPT-4o	0.649	0.526	0.626	0.590	0.294	−0.110
DeepSeek-V4-Pro	0.742	0.645	0.657	0.609	0.379	−0.021
GLM-5.2	0.775	0.703	0.702	0.624	0.452	−0.054
Best supervised	0.888		0.911		0.749	

B.3 Task-Stratified Significance Tests

The paired tests reported in the main text pool ROC AUC deltas from classification datasets with R^2 deltas from regression datasets into a single signed-rank distribution. Because the two metrics differ in scale and support, the pooled medians are convenient summaries rather than directly interpretable effect magnitudes. To verify that our conclusions are not artefacts of this aggregation, we repeat every paired comparison separately on the 15 classification datasets (4 binary and 11 multiclass) and the 29 regression datasets. Within each stratum, seed replicates are averaged per dataset before a two-sided Wilcoxon signed-rank test, Holm correction is applied within the stratum and experiment family, and confidence intervals are 10,000-resample bootstrap intervals for the paired median delta.

Supervised comparisons. Tables 11 and 12 confirm that the text-uplift result is robust in both strata: all five learners improve significantly on classification and on regression, with strictly positive confidence intervals throughout. The native-versus-frozen comparison, by contrast, is stratum-dependent. CARTE and CatBoost lose to the frozen baseline on every classification dataset ($0/0/15$, $r_{\mathrm{rb}} = -1.000$) and remain significantly behind on regression. TabSTAR and TabPFN-3 Plus are

statistically indistinguishable from the frozen baseline on classification ($p_{\text{Holm}} = 1.0000$, $8/0/7$ in both cases), but TabSTAR falls significantly behind on regression (-0.010, $p_{\text{Holm}} = 0.0054$) while TabPFN-3 Plus does not (-0.005, $p_{\text{Holm}} = 0.0920$). Consequently, the pooled conclusion that native integration does not uniformly outperform strong frozen representations holds in both strata, and the residual deficit of TabSTAR observed in the main text is attributable to regression tasks rather than to classification.

Table 11: Classification-only supervised comparisons on 15 datasets. Positive text-uplift deltas favour E5/PCA50; negative native-minus-frozen deltas favour the frozen TabICLv2 baseline.

Comparison	n	Median Δ [95% CI]	W/T/L	p_{Holm}	r_{rb}
XGBoost + text	15	$+0.066\,[+0.007, +0.215]$	12/0/3	0.0040	$+0.850$
LightGBM + text	15	$+0.057\,[+0.014, +0.233]$	13/0/2	0.0054	$+0.783$
TabPFN-2 + text	15	$+0.084\,[+0.017, +0.217]$	14/0/1	0.0012	$+0.950$
TabICLv2 + text	15	$+0.095\,[+0.022, +0.132]$	14/0/1	9.16×10^{-4}	$+0.967$
TabPFN-3 + text	15	$+0.076\,[+0.019, +0.263]$	14/0/1	0.0012	$+0.950$
CARTE − frozen	15	$-0.065\,[-0.071, -0.037]$	0/0/15	2.44×10^{-4}	-1.000
CatBoost − frozen	15	$-0.025\,[-0.035, -0.010]$	0/0/15	2.44×10^{-4}	-1.000
TabSTAR − frozen	15	$+0.001\,[-0.007, +0.004]$	8/0/7	1.0000	-0.050
TabPFN-3 Plus − frozen	15	$+0.000\,[-0.003, +0.007]$	8/0/7	1.0000	$+0.033$

Table 12: Regression-only supervised comparisons on 29 datasets. Positive text-uplift deltas favour E5/PCA50; negative native-minus-frozen deltas favour the frozen TabICLv2 baseline.

Comparison	n	Median Δ [95% CI]	W/T/L	p_{Holm}	r_{rb}
XGBoost + text	29	$+0.044\,[+0.013, +0.146]$	24/0/5	1.66×10^{-5}	$+0.871$
LightGBM + text	29	$+0.074\,[+0.034, +0.190]$	26/0/3	1.02×10^{-6}	$+0.949$
TabPFN-2 + text	29	$+0.041\,[+0.001, +0.097]$	22/0/7	0.0018	$+0.641$
TabICLv2 + text	29	$+0.088\,[+0.014, +0.185]$	23/0/6	1.59×10^{-5}	$+0.880$
TabPFN-3 + text	29	$+0.032\,[+0.003, +0.129]$	22/0/7	0.0018	$+0.678$
CARTE − frozen	29	$-0.053\,[-0.115, -0.017]$	3/0/26	7.99×10^{-6}	-0.899
CatBoost − frozen	29	$-0.028\,[-0.041, -0.007]$	8/0/21	0.0024	-0.683
TabSTAR − frozen	29	$-0.010\,[-0.027, -0.005]$	8/0/21	0.0054	-0.641
TabPFN-3 Plus − frozen	29	$-0.005\,[-0.025, +0.003]$	12/0/17	0.0920	-0.361

LLM prompting. Tables 13 and 14 show that the zero-shot advantage is driven predominantly by regression. On regression, all six models favour zero-shot with large effect sizes (r_{rb} from $+0.554$ to $+0.894$) and every comparison survives Holm correction. On classification, the direction is unchanged for all six models, but the magnitudes shrink by roughly an order of magnitude (median Δ between $+0.004$ and $+0.069$) and only GLM-5.2 remains significant after correction. The three Qwen3 models yield confidence intervals that straddle zero, indicating that at $n = 15$ the classification-side effect is not resolvable. The stratified view therefore refines rather than contradicts the main claim: additional demonstrations are clearly harmful for numeric targets, whereas for classification the evidence is directionally consistent but statistically weak.

B.4 Conditional Strategy Analysis

The paired tests establish average effects but not the conditions under which each integration strategy is preferable. To probe strategy-selection boundaries, we relate per-dataset deltas to eight schema-level attributes: the number of text columns, their share of all columns (*text ratio*), the median word and character counts of the concatenated text fields, the total column count m, $\log n$, the number of classes, and the mean unique-value rate of the categorical columns. Following the convention used throughout, positive +text deltas favour E5/PCA50 over tabular-only input, and positive native-minus-frozen deltas favour the native model over the TabICLv2 E5/PCA50 baseline. Tables 15

Table 13: Classification-only comparison of zero-shot and 32-shot prompting on 15 datasets. Positive values favour zero-shot.

Model	n	Median Δ [95% CI]	W/T/L	p_{Holm}	r_{rb}
Qwen3-4B	15	$+0.004\,[-0.010, +0.041]$	10/0/5	1.0000	$+0.117$
Qwen3-8B	15	$+0.011\,[-0.001, +0.060]$	10/0/5	0.5629	$+0.400$
Qwen3-14B	15	$+0.038\,[-0.059, +0.066]$	10/0/5	1.0000	$+0.200$
GPT-4o	15	$+0.039\,[-0.003, +0.084]$	11/0/4	0.0623	$+0.717$
DeepSeek-V4-Pro	15	$+0.048\,[+0.011, +0.094]$	11/0/4	0.1650	$+0.600$
GLM-5.2	15	$+0.069\,[+0.003, +0.136]$	13/0/2	0.0403	$+0.767$

Table 14: Regression-only comparison of zero-shot and 32-shot prompting on 29 datasets. Positive values favour zero-shot.

Model	n	Median Δ [95% CI]	W/T/L	p_{Holm}	r_{rb}
Qwen3-4B	29	$+0.187\,[+0.074, +0.396]$	24/0/5	0.0070	$+0.628$
Qwen3-8B	29	$+0.093\,[+0.023, +0.556]$	21/0/8	0.0080	$+0.554$
Qwen3-14B	29	$+0.286\,[+0.031, +0.408]$	21/0/8	0.0076	$+0.600$
GPT-4o	29	$+0.412\,[+0.126, +0.652]$	24/0/5	3.93×10^{-4}	$+0.775$
DeepSeek-V4-Pro	29	$+0.368\,[+0.174, +0.616]$	24/0/5	9.19×10^{-5}	$+0.834$
GLM-5.2	29	$+0.496\,[+0.250, +0.714]$	26/0/3	1.43×10^{-5}	$+0.894$

and 16 report Spearman correlations per learner; the *Dataset average* row first averages deltas across learners within each dataset, so that the dataset remains the independent unit. At $n = 44$, $|\rho| > 0.30$ corresponds to a nominal two-sided $p < 0.05$.

Table 15: Spearman correlations between the per-dataset +text delta and dataset attributes. Positive deltas favour E5/PCA50 over tabular-only input.

	Text attributes				Table attributes			
Model	ratio	cols	words	chars	$\log n$	m	classes	uniq. rate
XGBoost	$+0.216$	-0.277	$+0.036$	-0.006	$+0.102$	-0.457	-0.036	$+0.100$
LightGBM	$+0.286$	-0.261	$+0.012$	-0.015	$+0.055$	-0.526	-0.075	$+0.036$
TabPFN-2	$+0.076$	-0.405	$+0.113$	$+0.066$	$+0.040$	-0.490	$+0.121$	$+0.162$
TabICLv2	$+0.213$	-0.390	-0.024	-0.073	$+0.002$	-0.555	-0.015	$+0.189$
TabPFN-3	$+0.203$	-0.451	$+0.097$	$+0.059$	$+0.033$	-0.615	$+0.083$	$+0.193$
Average	$+0.156$	-0.476	$+0.063$	$+0.022$	$+0.016$	-0.607	$+0.034$	$+0.105$

When do frozen embeddings help most? Table 15 shows that the dominant moderator is not the amount of text but its share of the feature space. The total column count m is the single strongest correlate ($\rho = -0.607$ on dataset averages, negative for all five learners), and the number of text columns is also consistently negative (-0.476). Text ratio moves in the opposite, positive direction for every learner ($+0.076$ to $+0.286$), although the dataset-average magnitude ($+0.156$) does not reach nominal significance at $n = 44$. Raw text length is essentially uninformative ($\rho = +0.063$ for words and $+0.022$ for characters), as are $\log n$ ($+0.016$), the number of classes ($+0.034$), and the categorical unique-value rate ($+0.105$). Taken together, these signs are consistent with a dilution account: a fixed 50-dimensional text representation contributes a diminishing fraction of the available signal as the tabular schema widens, so the uplift is largest on narrow, text-dominated tables and smallest on wide tables in which text columns are a minor component. The negative correlation with the number of text columns is compatible with the same mechanism, since datasets with many text columns in our catalogue also tend to be wide overall, and concatenating several fields into one embedding may blur field-specific signal.

Table 16: Spearman correlations between the per-dataset native-minus-frozen delta and dataset attributes. Positive deltas favour the native model over the TabICLv2 E5/PCA50 baseline.

	Text attributes				Table attributes			
Model	ratio	cols	words	chars	$\log n$	m	classes	uniq. rate
CARTE	-0.311	$+0.075$	-0.058	-0.017	$+0.316$	$+0.349$	$+0.024$	-0.158
CatBoost	-0.071	$+0.260$	-0.089	-0.068	$+0.240$	$+0.346$	-0.046	-0.100
TabSTAR	-0.152	$+0.100$	$+0.332$	$+0.347$	$+0.331$	$+0.204$	$+0.277$	$+0.067$
TabPFN-3 Plus	-0.117	-0.046	$+0.189$	$+0.169$	$+0.105$	$+0.005$	$+0.124$	-0.026
Average	-0.420	-0.037	$+0.061$	$+0.090$	$+0.336$	$+0.302$	$+0.135$	-0.098

When does native integration become competitive? Table 16 identifies two moderators with opposite implications. First, the native-minus-frozen delta increases with dataset size for all four models ($\rho = +0.336$ on dataset averages, from $+0.105$ to $+0.331$), which is consistent with native architectures requiring sufficient supervision to learn a text–target mapping that a frozen encoder supplies for free. Second, the delta decreases with text ratio for all four models (-0.420 on dataset averages), so the native approach is at its greatest relative disadvantage precisely on the text-dominated datasets identified above as most favourable to frozen representations. The positive correlation with m ($+0.302$) is the mirror image of the $+$text result rather than an independent effect: on wide tables the frozen baseline gains little from its text channel, which mechanically narrows the gap. Among individual models, TabSTAR is the only one whose delta correlates appreciably with raw text length ($+0.332$ for words and $+0.347$ for characters), indicating that its end-to-end encoder does extract additional value from longer text, though not enough to overturn the aggregate deficit reported in Section 4.3.

B.5 Lexical versus Semantic Text Representations

The embedding-augmented results in Section 4.2 establish that adding frozen text features helps, but they do not identify whether the benefit requires pretrained semantic representations or is already available from surface lexical statistics. To separate these two accounts, we replace E5-small-v2 with a purely lexical pipeline: the concatenated text fields of each row are vectorised with TF-IDF and compressed to 50 dimensions by truncated SVD, matching the 50-dimensional budget of the E5/PCA50 setting. Because TF-IDF is sparse and high-dimensional before compression, we use truncated SVD rather than PCA, which would require densifying the matrix. Both the vectoriser and the projection are fitted on the training split of each fold only. We evaluate the three learners that appear in both configurations, keeping datasets, splits, and five-seed aggregation identical; seed results are averaged within each dataset before the mean and standard deviation across datasets are computed.

Most of the uplift is available from lexical features. Table 17 shows that the semantic encoder retains an advantage, but a modest one relative to the overall value of text. On multiclass classification the gap is $+0.012$ for XGBoost and $+0.014$ for TabICLv2 and reverses to -0.018 for LightGBM; on regression it is $+0.010$, $+0.019$, and -0.008 respectively. These differences are an order of magnitude smaller than the median paired gains of $+0.041$ to $+0.092$ that text integration delivers over tabular-only inputs (Table 1). A lexical representation compressed to the same 50 dimensions therefore recovers the large majority of the benefit attributed to text, which indicates that much of what the frozen encoder contributes on this benchmark is accessible from term-frequency statistics rather than from pretrained semantics. Binary classification is the one setting in which E5 is clearly ahead, by $+0.053$ to $+0.080$ across all three learners, but this stratum contains only four datasets and the standard deviations exceed the gap, so we regard the pattern as suggestive rather than established.

Implications for the native-versus-frozen comparison. This result strengthens rather than weakens the interpretation of Section 4.3. If a frozen sentence encoder is only marginally better than TF-IDF on the two larger task strata, then the frozen baseline that three of four native text-aware models fail to beat is itself close to what a classical text-feature pipeline achieves. The bar that tighter archi-

Table 17: Lexical TF-IDF/SVD50 versus semantic E5/PCA50 under an identical 50-dimensional budget. Binary, multiclass, and classification columns report ROC AUC; regression reports R^2. Δ is E5/PCA50 minus TF-IDF/SVD50, so positive values favour the semantic encoder.

Learner	Text input	Binary (n=4)	Multiclass (n=11)	Classification (n=15)	Regression (n=29)
XGBoost	TF-IDF/SVD50	0.755 ± 0.101	0.867 ± 0.082	0.837 ± 0.098	0.680 ± 0.328
	E5/PCA50	0.818 ± 0.073	0.879 ± 0.071	0.863 ± 0.075	0.690 ± 0.288
	Δ	$+0.063$	$+0.012$	$+0.026$	$+0.010$
LightGBM	TF-IDF/SVD50	0.763 ± 0.101	0.859 ± 0.068	0.833 ± 0.086	0.714 ± 0.284
	E5/PCA50	0.843 ± 0.076	0.841 ± 0.125	0.842 ± 0.111	0.706 ± 0.270
	Δ	$+0.080$	-0.018	$+0.009$	-0.008
TabICLv2	TF-IDF/SVD50	0.821 ± 0.110	0.897 ± 0.065	0.877 ± 0.083	0.730 ± 0.260
	E5/PCA50	0.874 ± 0.077	0.911 ± 0.059	0.901 ± 0.063	0.749 ± 0.252
	Δ	$+0.053$	$+0.014$	$+0.024$	$+0.019$

tectural integration must clear is therefore lower than the use of a pretrained encoder might suggest, which makes the observed deficits of CARTE, CatBoost, and TabSTAR more difficult to attribute to an unusually strong reference point.

B.6 Text Reliance Under Input Perturbation

This section reports the full results of the sensitivity ablation summarised in Section 5.2, including the blanked condition, task-specific score levels, and the stratified paired tests.

Protocol. Each model is evaluated with its original fitted parameters under three input conditions. *Intact* retains the original text. *Shuffled* jointly permutes the text columns of the test rows, so every text value still occurs exactly once but is paired with a different row; this destroys text–label alignment while leaving the marginal distribution of text unchanged. *Blanked* replaces every text value with an empty string. No model is refitted and the structured columns are never modified. For the in-context models the context rows are left intact, so the perturbation isolates dependence on the text of the query row and matches the setting used for models with fixed weights.

Two properties of the design bear emphasis. First, shuffling is preferred to blanking as the primary probe because it preserves the input distribution: an all-empty text field is a configuration the models never encounter during fitting, so part of any degradation under blanking reflects distribution shift rather than loss of information. Second, this analysis measures *reliance*—how much a learned mapping depends on aligned text—rather than *contribution*, which would require refitting each model without text. The two coincide only if the learned mapping is also optimal for the perturbed input distribution, which we do not assume.

Every model relies on aligned text. Table 18 shows that no pipeline is insensitive to the perturbation. Median Δ_{shuffle} ranges from $+0.178$ for TabPFN-2 to $+0.413$ for TabSTAR and is positive on 40 to 44 of the 44 datasets for every model, so aligned text is load-bearing throughout. The two families are nearly indistinguishable in pooled median ($+0.292$ for frozen versus $+0.289$ for native). Within tasks, native models show higher median sensitivity on both classification ($+0.246$ versus $+0.174$) and regression ($+0.411$ versus $+0.364$), but these are unweighted averages over models of very different sensitivity, and the paired tests below show that the apparent family-level gap is driven by particular models rather than by the integration strategy as such.

Ordering of the two perturbations. For eight of nine models $\Delta_{\text{blank}} < \Delta_{\text{shuffle}}$: removing text hurts less than supplying misaligned text. This is expected, since a blanked field is merely uninformative whereas a shuffled field is actively misleading, and a model that attends to it is drawn towards the wrong row. CARTE is the exception, with $\Delta_{\text{blank}} = +0.309$ marginally exceeding $\Delta_{\text{shuffle}} = +0.307$, indicating that it discounts contradictory text no more effectively than absent text. Read alongside its being the weakest native model in Table 2 ($3/0/41$ against the frozen base-

Table 18: Median text sensitivity across 44 datasets. Δ_{shuffle} is the drop in task score when test-row text is permuted across rows; larger values indicate stronger reliance on aligned text. Classification (n=15) reports ROC AUC and regression (n=29) reports R^2.

Model	All (n=44)	Classification	Regression
Frozen embeddings (E5/PCA50)			
XGBoost	+0.280	+0.197	+0.384
LightGBM	+0.242	+0.116	+0.317
TabPFN-2	+0.178	+0.152	+0.180
TabICLv2	+0.295	+0.186	+0.376
TabPFN-3	+0.339	+0.194	+0.469
Native text-aware			
CARTE	+0.307	+0.160	+0.384
CatBoost	+0.215	+0.140	+0.246
TabSTAR	**+0.413**	**+0.289**	**+0.603**
TabPFN-3-Plus	+0.376	+0.245	+0.488

line), this points to robustness to misaligned text rather than sensitivity to text as its limiting factor. CatBoost sits at the opposite extreme, with the smallest gap between conditions and the highest shuffled and blanked scores of any model, consistent with its text handling operating largely through high-cardinality categorical statistics rather than through compositional text representations.

Table 19: Dataset-level paired tests of text sensitivity against the shared frozen reference, TabICLv2 with E5/PCA50. $\Delta\Delta = \Delta^{\text{ref}}_{\text{shuffle}} - \Delta^{\text{native}}_{\text{shuffle}}$, so negative values indicate that the native model relies on text more heavily than the reference. Two-sided Wilcoxon signed-rank tests are Holm-corrected across the four models within each stratum, and intervals are 10,000-resample bootstrap 95% confidence intervals for the paired median. W/T/L counts datasets favouring the frozen reference, tied, or favouring the native model.

Stratum	Native model	Median $\Delta\Delta$ [95% CI]	W/T/L	p_{Holm}	r_{rb}
Overall (n=44)	CARTE	$+0.049\,[+0.021, +0.103]$	31/0/13	0.0570	+0.378
	CatBoost	$+0.088\,[+0.065, +0.124]$	42/0/2	1.14×10^{-11}	+0.984
	TabSTAR	$-0.015\,[-0.140, +0.016]$	18/0/26	0.1442	−0.255
	TabPFN-3-Plus	$-0.022\,[-0.086, +0.002]$	16/0/28	0.0264	−0.448
Classification (n=15)	CARTE	$+0.052\,[+0.026, +0.112]$	12/0/3	0.3029	+0.383
	CatBoost	$+0.065\,[+0.046, +0.085]$	15/0/0	2.44×10^{-4}	+1.000
	TabSTAR	$-0.015\,[-0.097, -0.011]$	3/0/12	0.0542	−0.683
	TabPFN-3-Plus	$-0.006\,[-0.056, +0.012]$	7/0/8	0.3029	−0.433
Regression (n=29)	CARTE	$+0.025\,[-0.000, +0.127]$	19/0/10	0.2208	+0.343
	CatBoost	$+0.127\,[+0.073, +0.169]$	27/0/2	2.83×10^{-7}	+0.968
	TabSTAR	$+0.001\,[-0.305, +0.101]$	15/0/14	0.3145	−0.218
	TabPFN-3-Plus	$-0.042\,[-0.112, -0.000]$	9/0/20	0.0730	−0.476

Paired tests locate the effect in individual models. Table 19 compares each native model against the frozen reference on a per-dataset basis, which removes the dataset-difficulty variation that inflates the spread of the marginal medians. The picture is one of model-specific heterogeneity rather than a family-level property. TabPFN-3-Plus is significantly more text-sensitive than the reference overall ($\Delta\Delta = -0.022$, $p_{\text{Holm}} = 0.026$, $r_{\text{rb}} = -0.448$), with the effect concentrated in regression (-0.042, $20/29$ datasets favouring it). CatBoost is significantly *less* sensitive in every stratum, by a wide margin and with near-unanimous win counts ($42/0/2$ overall, $15/0/0$ on classification, r_{rb} up to $+1.000$); this is the clearest single result in the table and again reflects its categorical-statistics treatment of text. CARTE and TabSTAR do not separate from the reference after correction, although both approach the threshold in one stratum: CARTE trends towards lower sensitivity overall

($p_{\mathrm{Holm}} = 0.057$) and TabSTAR towards higher sensitivity on classification ($p_{\mathrm{Holm}} = 0.054$, $12/15$ datasets).

What the tests establish jointly is a negative result of some strength. Under Holm correction no native model is significantly less text-sensitive than the frozen reference except CatBoost, whose native text handling is the least representative of the end-to-end approaches we evaluate. The hypothesis that native architectures systematically fail to route textual signal to the output is therefore not supported, and the accuracy deficit documented in Section 4.3 must be attributed elsewhere. At the same time, the tests do not license the converse claim: the marginal medians in Table 18 suggest a native advantage in sensitivity that survives paired testing for only one of four models.

Table 20: Task-specific score levels under the three conditions, reported as mean ± SD across datasets. Panel (a) gives ROC AUC on the 15 classification datasets and panel (b) gives R^2 on the 29 regression datasets. Family rows average models within each dataset. Unlike Table 4 these are means rather than medians and are therefore sensitive to outliers, as the TabPFN-3-Plus blanked entry illustrates.

	(a) Classification ROC AUC, n=15		
Model	Intact	Shuffled	Blanked
Frozen embeddings (E5/PCA50)			
XGBoost	0.863 ± 0.075	0.645 ± 0.113	0.703 ± 0.118
LightGBM	0.842 ± 0.111	0.667 ± 0.116	0.712 ± 0.112
TabPFN-2	0.872 ± 0.068	0.673 ± 0.135	0.731 ± 0.118
TabICLv2	0.901 ± 0.063	0.669 ± 0.115	0.741 ± 0.112
TabPFN-3	0.892 ± 0.068	0.657 ± 0.126	0.717 ± 0.136
Family average	0.874 ± 0.063	0.662 ± 0.111	0.721 ± 0.106
Native text-aware			
CARTE	0.840 ± 0.076	0.644 ± 0.129	0.658 ± 0.127
CatBoost	0.875 ± 0.073	0.712 ± 0.117	0.762 ± 0.118
TabSTAR	0.904 ± 0.067	0.622 ± 0.111	0.659 ± 0.119
TabPFN-3-Plus	0.901 ± 0.063	0.628 ± 0.126	0.703 ± 0.137
Family average	0.880 ± 0.067	0.652 ± 0.109	0.696 ± 0.108
	(b) Regression R^2, n=29		
Model	Intact	Shuffled	Blanked
Frozen embeddings (E5/PCA50)			
XGBoost	0.690 ± 0.288	0.140 ± 0.639	0.199 ± 0.705
LightGBM	0.706 ± 0.270	0.239 ± 0.524	0.309 ± 0.548
TabPFN-2	0.633 ± 0.308	0.238 ± 0.471	0.301 ± 0.475
TabICLv2	0.749 ± 0.252	0.180 ± 0.575	0.356 ± 0.460
TabPFN-3	0.666 ± 0.400	0.092 ± 0.629	0.225 ± 0.714
Family average	0.689 ± 0.279	0.178 ± 0.530	0.278 ± 0.513
Native text-aware			
CARTE	0.656 ± 0.322	0.107 ± 0.601	0.131 ± 0.558
CatBoost	0.723 ± 0.268	0.332 ± 0.511	0.476 ± 0.384
TabSTAR	0.709 ± 0.278	0.060 ± 0.504	0.251 ± 0.367
TabPFN-3-Plus	0.735 ± 0.261	0.091 ± 0.609	-6.323 ± 34.764
Family average	0.706 ± 0.275	0.147 ± 0.527	-1.366 ± 8.748

Score levels and the case for median summaries. Table 20 gives the underlying levels. On classification, shuffling costs every model 0.16 to 0.28 ROC AUC, leaving scores in the 0.62–0.71 band; the two most accurate intact models, TabSTAR (0.904) and TabPFN-3-Plus (0.901), fall furthest, to 0.622 and 0.628. On regression the effect is far larger: mean R^2 drops from roughly 0.69–0.75 to 0.06–0.33, and several models approach or fall below the level of predicting the target mean once

text is misaligned. This asymmetry suggests that structured features compensate less effectively for lost text on regression tasks than on classification tasks.

The regression panel also shows why medians are used in Tables 18 and 4. The blanked mean R^2 for TabPFN-3-Plus is -6.323 with a standard deviation of 34.764, driven by a small number of datasets on which the model produces extreme predictions once text is removed; this single entry also distorts the native family average to -1.366. Since R^2 is unbounded below, one such dataset can dominate a mean while affecting the median negligibly. This is a further reason to treat shuffling rather than blanking as the primary probe: preserving the marginal text distribution keeps models inside their operating regime and avoids these pathologies.

B.7 LLM Prediction Diagnostics

This section reports the full diagnostics summarised in Section 5.3. All statistics are computed from the stored per-row predictions of the zero-shot and 32-shot runs, using five seeds and up to 256 test rows per dataset, and are averaged over dataset–seed pairs within each task group. No additional inference was performed.

Two caveats apply throughout. First, the recalibrated scores R^2_{lin} and R^2_{iso} are obtained by fitting a linear or isotonic map from predictions to targets on the same test labels used for scoring. They quantify how much of the residual error is attributable to a monotone scale mismatch and are oracle upper bounds; no practitioner could obtain them without labelled test data. Second, the prompts for several datasets did not record class priors in a recoverable form, so we omit the divergence between predicted label distributions and prompt priors rather than substituting test-set frequencies.

Table 21: Full regression diagnostics on the 29 regression datasets. Scale is the ratio of predicted to observed standard deviation and Unique is the fraction of distinct predicted values. R^2_{lin} and R^2_{iso} are oracle recalibrations fitted on the test labels.

Model	Setting	R^2	ρ	R^2_{lin}	R^2_{iso}	Scale	Unique
Qwen3-4B	0-shot	0.002	0.366	0.227	0.278	0.408	0.100
	32-shot	−1.310	0.181	0.047	0.059	0.272	0.042
Qwen3-8B	0-shot	0.124	0.436	0.284	0.358	0.426	0.100
	32-shot	−0.101	0.146	0.044	0.057	0.195	0.038
Qwen3-14B	0-shot	0.166	0.494	0.320	0.377	0.480	0.095
	32-shot	−0.103	0.236	0.072	0.087	0.209	0.042
GPT-4o	0-shot	0.294	0.641	0.500	0.562	0.641	0.104
	32-shot	−0.110	0.175	0.093	0.121	0.318	0.065
DeepSeek-V4-Pro	0-shot	0.379	0.668	0.537	0.609	0.616	0.140
	32-shot	−0.021	0.180	0.106	0.134	0.283	0.074
GLM-5.2	0-shot	0.452	0.699	0.555	0.621	0.606	0.110
	32-shot	−0.054	0.218	0.108	0.137	0.244	0.045

Reading the classification diagnostics. On multiclass tasks the pattern is uniform: the majority-label fraction rises for all six models and normalised entropy falls for all six, with the largest reductions for GLM-5.2 (-0.279) and Qwen3-14B (-0.214)—the two models that were most diverse at zero shot. On binary tasks the direction is less consistent, with GPT-4o becoming slightly more diverse under 32 shots, which is compatible with the weaker and statistically less resolvable classification effects reported in Appendix B.3. The extreme case is Qwen3-4B, whose binary predictions concentrate on a single label for 99.0% of rows at 32 shots, with normalised entropy dropping from 0.671 to 0.068. Under ROC AUC such a predictor is scored on its continuous confidence output rather than its labels, so this concentration is not automatically fatal; it does, however, indicate that the decision rule itself has become near-degenerate.

Table 22: Concentration of predicted label distributions on the 4 binary and 11 multiclass datasets. Maj. is the fraction of test rows assigned the most frequent predicted label; H_{norm} is the entropy of the predicted label distribution normalised by $\log K$. Lower entropy and higher majority fraction indicate a more concentrated, less discriminative predictor.

	Binary (n=4)			Multiclass (n=11)		
Model	Maj. 0-shot	Maj. 32-shot	ΔH_{norm}	Maj. 0-shot	Maj. 32-shot	ΔH_{norm}
Qwen3-4B	0.777	0.990	−0.603	0.712	0.740	−0.038
Qwen3-8B	0.802	0.879	−0.219	0.720	0.776	−0.043
Qwen3-14B	0.781	0.839	−0.319	0.589	0.765	−0.214
GPT-4o	0.772	0.699	+0.105	0.527	0.688	−0.183
DeepSeek-V4-Pro	0.739	0.798	−0.105	0.555	0.725	−0.155
GLM-5.2	0.747	0.716	−0.018	0.431	0.673	−0.279

B.8 Confidence Resolution and the ROC AUC ceiling

Our classification comparisons use ROC AUC, which is computed from the continuous scores an LLM emits rather than from its predicted labels. This raises a measurement concern that is specific to the LLM paradigm: a model instructed to return a probability in a JSON field tends to produce a small set of round values such as 0.1, 0.5, or 0.9, whereas a supervised classifier returns a distinct real number for almost every row. Large blocks of tied scores cap ROC AUC regardless of whether the underlying discrimination is good, so part of the observed gap between LLM and supervised classification could be an artefact of output granularity rather than a difference in predictive ability. This section quantifies that component.

For each dataset–seed pair we record the number of distinct emitted score values, the fraction of rows falling in the largest tied block, and a *tie-oracle upper bound* on ROC AUC. The bound holds the observed score buckets and their order fixed, resolves every positive–negative pair tied within a bucket in the model's favour, and leaves cross-bucket inversions unchanged. It therefore measures how much of the AUC is left undetermined by the model's own output granularity. Statistics are computed from the stored per-row predictions of the zero-shot and 32-shot runs on the 15 classification datasets using five seeds and up to 256 test rows per dataset; no additional inference was performed. Table 23 pools binary and multiclass datasets, weighting each dataset equally.

Table 23: Confidence resolution, the tie-oracle ROC AUC bound, and threshold metrics on the 15 classification datasets. Uniq. is the number of distinct emitted score values and Tie is the fraction of rows in the largest tied block. AUC_{obs} is the reported score and AUC_{tie} is the tie-oracle upper bound, with Δ_{tie} their difference. Acc, BAcc, and F1 denote accuracy, balanced accuracy, and macro-F1.

Model	Setting	Uniq.	Tie	AUC_{obs}	AUC_{tie}	Δ_{tie}	Acc	F1
Qwen3-4B	0-shot	5.4	0.729	0.518	0.813	+0.296	0.324	0.200
	32-shot	5.4	0.768	0.505	0.836	+0.332	0.229	0.140
Qwen3-8B	0-shot	11.9	0.650	0.546	0.778	+0.232	0.363	0.217
	32-shot	9.2	0.667	0.522	0.805	+0.282	0.324	0.199
Qwen3-14B	0-shot	21.6	0.477	0.558	0.711	+0.153	0.389	0.279
	32-shot	6.4	0.744	0.537	0.863	+0.326	0.370	0.230
GPT-4o	0-shot	18.3	0.545	0.632	0.816	+0.184	0.484	0.356
	32-shot	15.9	0.519	0.573	0.751	+0.178	0.356	0.272
DeepSeek-V4-Pro	0-shot	19.7	0.496	0.679	0.813	+0.134	0.540	0.400
	32-shot	19.0	0.637	0.618	0.844	+0.225	0.452	0.312
GLM-5.2	0-shot	32.5	0.326	0.722	0.801	+0.080	0.520	0.401
	32-shot	30.6	0.446	0.645	0.789	+0.144	0.456	0.305

Resolution accounts for part of the AUC shortfall. Emitted scores are coarse across the board: under zero-shot prompting the models produce between 5.4 and 32.5 distinct values for up to 256 test rows, and the largest tied block covers between 33% and 73% of rows. The tie-oracle bound exceeds the observed AUC by 0.080 to 0.296, and this headroom is closely tied to granularity: across the six models the number of distinct scores and Δ_{tie} have a Spearman correlation of -0.94. Qwen3-4B, which emits about five distinct values, leaves 0.296 of AUC undetermined, whereas GLM-5.2, with roughly six times as many levels, leaves only 0.080. Reported ROC AUC therefore understates the discriminative content of the coarser models, and comparisons among LLMs on this metric partly reflect how finely each model verbalises confidence.

The gap to supervised learning survives this correction. The correction is nonetheless insufficient to close the gap we report in the main text. GLM-5.2 is both the strongest and the finest-grained LLM, yet its tie-oracle bound of 0.801 remains 0.100 below the 0.901 attained by embedding-augmented TabICLv2 on the same 15 datasets, and that bound assumes every within-bucket tie is resolved in the model's favour. The most favourable reading available to any LLM in our evaluation thus still falls short of a frozen-embedding pipeline evaluated without such assistance. Coarse confidence output is a real limitation of the prompting interface, but it does not explain the deficit in classification.

The zero-shot advantage is not a metric artefact. Because accuracy, balanced accuracy, and macro-F1 depend on predicted labels rather than on score granularity, they provide a resolution-free check on the zero-shot versus 32-shot comparison of Section 5.3. All three metrics decline from zero-shot to 32-shot for all six models: accuracy falls from a range of 0.324–0.540 to 0.229–0.456, and macro-F1 from 0.200–0.401 to 0.140–0.312. The largest degradations are for GPT-4o (-0.128 accuracy) and Qwen3-4B (-0.095 accuracy, -0.060 macro-F1). The zero-shot advantage on classification is therefore visible under threshold metrics as well and is not an artefact of scoring a coarse confidence output with a ranking metric.

Why the bound rises under 32-shot prompting. The tie-oracle bound is higher at 32 shots than at zero shot for four of six models, most strikingly for Qwen3-14B (0.863 versus 0.711). This is not evidence that demonstrations increase attainable performance. The bound grows monotonically with the amount of tied mass, and 32-shot prompting concentrates scores sharply: Qwen3-14B moves from 21.6 distinct values with a 0.477 largest tie to 6.4 values with a 0.744 largest tie. When three quarters of the rows share one score, the oracle is effectively ordering three quarters of the test set on the model's behalf, and the bound approaches vacuity. The quantity to read from these rows is Δ_{tie}, which measures indeterminacy and rises accordingly, rather than AUC_{tie}, which should not be interpreted as achievable performance. This concentration is the same phenomenon documented for predicted label distributions in Appendix B.7.

C LIMITATIONS

Our study has several limitations. First, although the benchmark spans 44 datasets and three task types, coverage is uneven: regression is substantially better represented than binary classification, and several application domains contribute only a few datasets. Second, text-column identification relies on fixed heuristic thresholds and may miss semantically meaningful attributes that fall outside them. Third, the representation ablation covers two frozen neural encoders, two PCA dimensions, and one classical sparse baseline, but a broader sweep over encoder families, pooling strategies, and compression targets remains open. Fourth, the LLM evaluation uses a single prompting and demonstration-selection protocol and is conducted on 256-example subsets, whereas supervised models are evaluated on the full test splits; the resulting cross-paradigm gaps should therefore be read as indicative rather than exact. Finally, all learners use fixed default or lightly tuned configurations, and a larger tuning budget could shift the relative ordering of individual methods, although it is unlikely to reverse the consistent dataset-level effects we report.

D BROADER IMPACT

This work provides a controlled empirical framework for evaluating how textual information is incorporated into tabular prediction. Our finding that frozen text representations remain strong baselines may encourage more rigorous comparisons before improvements are attributed to increasingly integrated architectures. Stronger predictive performance, however, does not imply fairness, robustness, or deployment readiness, particularly when textual fields encode sensitive or biased attributes. Our results should be read as methodological evidence about predictive performance, not as certification of any evaluated model for high-stakes applications.