DO YOU EVEN NEED MORE DATA? RETHINKING LLM DEPLOYMENT FOR LABEL-SCARCE TABULAR CLASSIFICATION WITH TABLENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Every tabular classification task requires sufficient labeled data. When labels are scarce, practitioners can collect more examples, encode domain expertise into features, or accept lower predictive performance. Large language models offer a potential alternative by leveraging knowledge acquired during pretraining. This possibility raises three distinct questions: (i) can zero-shot inference reduce the need for labeled data; (ii) can a richer prompt reduce the need for collecting more examples; (iii) and can encoded domain knowledge compensate for missing data? To answer each question under explicit, testable conditions, we introduce TabLens, a benchmark of 91 real-world and synthetic datasets across eight domains, together with eight industrial datasets containing free-text columns.

First, below a complexity threshold that can be estimated from a dataset's intrinsic dimensionality before running an LLM, zero-shot inference matches the accuracy that would otherwise require up to approximately 16 labeled examples to collect. This holds whether the model generates a written answer or, more cheaply, scores candidate labels directly from their log-probabilities – the latter loses almost no accuracy except for the smallest models. Above the threshold, this advantage disappears. However, on datasets with free-text columns, zero-shot LLMs are even better than 16-shot TabPFN-3 Plus, and no longer require manual feature engineering. Second, a richer prompt reduces the number of required demonstrations only when it includes both the real column names and a plain-language task description. Providing only one of these sources of context can perform worse than providing neither, particularly at higher shot counts. Third, on the LLM-synthetic datasets, explicit if–then expert decision rules improve performance only when the prompt provides sufficient task context for the model to interpret them. Even then, they only substitute for up to about 16 real examples, after which collecting more data is more effective. Together, these results establish testable conditions under which LLMs can reduce data-collection and feature-engineering requirements for tabular classification.

1 INTRODUCTION

Tabular classification is widely used in healthcare, finance, and marketing Wang et al. (2025). Although gradient boosting methods Chen (2016); Prokhorenkova et al. (2018); Ke et al. (2017) and specialized neural architectures Chen et al. (2022); Arik & Pfister (2021); Huang et al. (2012); Gorishniy et al. perform well with sufficient labeled data, they remain limited in low-data regimes and cross-task generalization.

In-context learning methods such as TabPFN Hollmann et al. (2022) and TabICL QU et al. (2025) address data scarcity through large-scale synthetic pretraining, but still require labeled examples from the target task. Large Language Models (LLMs), by contrast, can classify from task descriptions and, optionally, a few demonstrations without parameter updates Hegselmann et al. (2023); Jaitly et al. (2023). However, their behavior on tabular data remains inconsistent Slack & Singh (2023); Manikandan et al. (2023); Cheng et al. (2025), and the respective roles of model priors, task descriptions, external knowledge, and demonstrations remain unclear.

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.

Whether large language models are good at tabular classification is too coarse a question to build a benchmark around, since any answer is meaningful only relative to the alternative a practitioner would otherwise use. In practice, a team facing scarce labeled tabular data has to pay for one of a few things: collecting more labeled examples, writing richer prompts and task descriptions, or hand-coding domain expertise as rules. The question we actually care about is which of these costs an LLM lets a practitioner skip, and under what conditions. We turn this into three concrete questions:

Q1 Can zero-shot inference substitute for collecting labeled data?

Q2 Can richer prompt context substitute for collecting additional examples?

Q3 Can encoded expert knowledge substitute for a lack of data?

We introduce TabLens to answer each question. For each one we report a concrete, checkable condition that a researcher or data scientist can test against their own data, rather than a single yes-or-no verdict. The benchmark isolates the information sources involved in each substitution – model priors, task descriptions, demonstrations, external knowledge, and inference mode – and relates their usefulness to a TabPFN-based measure of dataset complexity. We further extend the core benchmark with a dedicated suite of 8 industrial *text-tabular* datasets, letting us test whether the same substitutions hold once part of the input is free-form natural-language text rather than structured columns. As an ablation on Q1, we additionally ask whether the zero-shot-for-data substitution requires costly autoregressive generation, or whether cheap forward scoring suffices.

Our results show that each substitution has a clean boundary, developed in full in Sections 4–6: zero-shot inference substitutes for data collection only below a predictable complexity threshold (strengthening on free-form text), and does so under either inference mode – cheap forward scoring can replace autoregressive generation for mid-to-large open-weight models, though not for small models or pure zero-shot use; prompt context substitutes for demonstrations only when complete, and can hurt when partial; and encoded expert knowledge substitutes for data only alongside a matching prior, and only up to $\sim$16 shots.

Our main contributions are:

- We introduce TabLens, a benchmark for zero-shot and few-shot in-context tabular classification with LLMs, comprising 91 core datasets across eight domains plus an 8-dataset text-tabular extension.
- We answer three substitution questions – data collection, additional demonstrations, and expert knowledge – each with a controlled ablation, statistical significance testing, and an explicit boundary condition rather than an aggregate score.
- We show that the zero-shot-for-data substitution is governed by a measurable, TabPFN- and intrinsic-dimensionality-predictable complexity threshold, that it extends to feature collection itself on datasets carrying free-form text, and that it holds under cheap forward scoring as well as full autoregressive generation for mid-to-large open-weight models.
- We show that context and demonstrations are partial substitutes that can interfere destructively when only one is partially supplied, and that expert knowledge requires a contextual prior to be useful at all.

2 Related Work

Existing tabular benchmarks such as OpenML-CC18 Bischl et al. (2021), TabZilla McElfresh et al. (2023), and TabArena Erickson et al. (2025) improve evaluation consistency but primarily target supervised learning. They are not designed to isolate in-context learning factors such as prompt format, task descriptions, context budgets, or demonstration selection.

Recent studies evaluate LLMs on tabular prediction more directly. TABLET Slack & Singh (2023) reports benefits from natural-language instructions alongside limitations in instruction faithfulness, while TabuLa-8B Gardner et al. (2024) reports zero- and few-shot transfer after large-scale tabular language-model training; TabFSBench Cheng et al. (2025) independently documents that LLM tabular performance is brittle under feature shifts. However, Gorla & Puduppully (2026) argues that apparent generalization may be inflated by task aggregation, leakage, contamination, or format familiarity. Similarly, Silvestri et al. (2025) finds that LLMs may retain latent knowledge of public

Table 2: TabLens dataset suites and associated hypotheses.

Suite	#	Purpose / construction	Used by
Core real-world	24	Public OpenML/Kaggle datasets grouped into 8 domains.	H1a, H2a–d
Temporal (post-cutoff)	8	Datasets released after Oct. 2024 to reduce pretraining-contamination risk and probe leakage.	H1a, H1d
LLM-synthetic	24	LLM-generated counterparts of the real datasets, each with explicit decision rules.	H3
Complexity grid (MLP-prior)	35	Controlled difficulty: 7 levels $\times$ 5 random seeds.	H1b
Text-tabular (industrial)	8	Mixed free-text and structured columns, decomposed into full/text-only/tabular-only variants.	H1c, H2d
Total	**99**		

datasets when semantic column names and interpretable values are preserved, and Bordt et al. (2024) documents memorization and learning of tabular data in LLMs more broadly. These results motivate protocols that separate genuine in-context generalization from contamination and semantic leakage, which is why we complement our aggregate comparisons (Section 4) with explicit recognition and reconstruction probes.

Two dedicated LLM-tabular pipelines are directly relevant to our design: FeatLLM Han et al. (2024) distills LLM-generated features for a downstream few-shot classifier, and Knauer et al. Knauer et al. (2025) induce a zero-shot decision tree from a task description alone. We use both as baselines in Section 4.1 rather than only citing them, since they let us test directly whether a dedicated pipeline beats plain prompting.

Benchmarks such as Table Meets LLM Sui et al. (2023) and TableBench Wu et al. (2025) instead focus on table understanding and question answering rather than row-wise classification. A separate line of work targets *joint* text-tabular inputs, where one or more columns carry free-form natural language alongside structured features, including multimodal AutoML for tabular data with text fields Shi et al. (2021), curated multimodal benchmarks exploiting text–tabular synergy Arazi et al. (2026), and text meta-feature analyses of tabular learning Blayer et al. (2026); Mráz et al. (2025). These resources focus on dataset construction and tabular foundation models rather than on when zero-shot LLMs specifically become competitive; TabLens's text-tabular extension (Section 4.3) targets this gap directly by decomposing text-tabular datasets into text-only and tabular-only variants and tracking the zero-shot LLM vs. label-aware baseline comparison across that decomposition.

3 TabLens Overview

Terminology Table 1 defines the recurring prompt and evaluation components used without redefinition from Section 4 onward.

Table 1: Key terms used throughout the paper.

Term	Meaning	Term	Meaning
Zero-shot	Classification with $k = 0$ labeled prompt demonstrations.	Full / complete context	Schema indicators and task context are both present.
Few-shot (k-shot)	k labeled rows precede the query.	Partial / incomplete context	Only schema indicators or task context are present.
Prior knowledge / prior	Knowledge encoded during pretraining, before task-specific text.	Expert knowledge (rules)	Explicit if–then rules supplied as r (Sec. 6).
Task context (prior context)	Description of the domain, objective, and label meaning, denoted c (Sec. 6).	Forward scoring	Target-label log-probabilities read in one forward pass (Sec. 4.4).
Schema indicators	Human-readable column names rather than generic placeholders.	Generation	Autoregressive label generation followed by parsing.
Intrinsic dimensionality	Model-free feature-matrix complexity score (Sec. 4.2).	Substitution	Replacing a deployment cost—data, examples, or expertise—with an LLM capability under a testable condition.

TabLens evaluates low-data binary tabular classification while isolating serialization, example count, task knowledge, dataset complexity, and input modality.

Dataset suites TabLens contains **99** datasets across five suites (Table 2). The core suite comprises **24** public OpenML Bischl et al. (2021) and Kaggle datasets from **8** domains. All datasets use a fixed class-balanced stratified split with a 0.3 test fraction,[1] while each text-tabular dataset is additionally decomposed into *full*, *text-only*, and *tabular-only* variants to isolate free-form text from structured features.

Evaluation axes We evaluate zero-shot and 4, 8, 16, 32, and 64-shot inference using five serializations: feat_val, feat_val_mask, markdown, markdown_mask, and HTML. The complexity grid uses

[1] This reproduces the train/test partition of the k-fold cross-validation split used by TabArena Erickson et al. (2025).

the two masked formats described in the appendix. Context is provided throughout; the synthetic suite additionally includes context-free and knowledge-augmented prompts with LLM-generated decision rules. Every few-shot setting uses five independent stratified demonstration draws.

Models and baselines We evaluate GPT-4o-mini, Qwen3-1.7B, Qwen3-8B, Qwen3-14B Yang et al. (2025), and GLM-4.7-Flash; GLM has partial suite coverage, documented with its interpretation constraints in Appendix **??**. For GLM and Qwen, we compare normalized target-label *forward scoring* with parsed free-form *generation*. Baselines are TabPFN-3 Plus Grinsztajn et al. (2026), TabPFN-2.5 Plus Prior Labs (2025), TabICL QU et al. (2025), XGBoost Chen (2016), CatBoost Prokhorenkova et al. (2018)—including native text features on the text-tabular suite—Random Forest, Logistic Regression, k-nearest neighbors, and majority-class prediction, all evaluated under the same protocol. On the 32 real datasets, we also evaluate FeatLLM Han et al. (2024) and zero-shot LLM decision-tree induction Knauer et al. (2025); such methods primarily target few-shot learning, while pure zero-shot alternatives remain uncommon.

Scoring We report ROC–AUC from continuous scores because it measures positive-over-negative ranking independently of deployment-specific thresholds and error costs. Following standard ROC practice, values below 0.5 indicate a ranking that is systematically inverted rather than uninformative, since the classifier's decisions could simply be negated to obtain an equally strong classifier with AUC $1 - \mathrm{AUC}$ (Fawcett, 2006); we therefore replace any such score by $1 - \mathrm{AUC}$. Results are averaged over five demonstration draws.

Testing zero-shot performance against random guessing For H1a, the dataset—not an individual prediction or seed—is the experimental unit. For each model and each of the 24 classic datasets, we average zero-shot ROC–AUC over up to five available runs in the primary full-context generation configuration and compute $d_j = \overline{\mathrm{AUC}}_j - 0.5$. We test $H_0 : \mathrm{median}(d_j) \leq 0$ against $H_1 : \mathrm{median}(d_j) > 0$ using a one-sided Wilcoxon signed-rank test, the standard non-parametric recommendation for across-dataset comparisons Demšar (2006). Holm's step-down procedure controls the family-wise error rate across the five model-level tests Holm (1979); we report the median paired difference and matched-pairs rank-biserial correlation as effect sizes. This protocol requires only dataset-level summary ROC–AUC values, not stored per-example predictions.

4 Q1: Can Zero-Shot LLM Inference Replace Collecting Labeled Data?

We first ask whether zero-shot LLM inference can match few-shot baselines without using labeled examples from the target task. We test this claim along three dimensions: overall competitiveness against a strong few-shot baseline (H1a), the existence of a predictable complexity boundary (H1b), and robustness across input modalities, including text-tabular data (H1c). Finally, we run a fourth ablation: whether the results, and respective computational costs, depend on the inference mode (autoregressive generation and direct label scoring) (H1d).

4.1 H1a: Zero-Shot LLMs vs. k-Shot TabPFN-3 Plus

Setup We evaluate zero-shot classification with 5 LLMs on the 24 core real-world datasets plus 8 post-cutoff datasets (32 total). On the 24 classic datasets, we test whether each model exceeds random guessing (ROC-AUC $= 0.5$) with a one-sided Wilcoxon signed-rank test over dataset-level mean ROC–AUC differences, followed by Holm correction across the five models Demšar (2006); Holm (1979); this requires no per-example predictions (full protocol and visualization in Appendix F.1). We split the pool into *Classic* (the 24 established benchmarks) and *New* to probe memorization. *New* suite contains 8 datasets published after October 2024, unavailable to any model's pretraining. The sole exception is GLM 4.7 Flash: as a recently released model, it is impossible at present to assemble a sufficient number of new datasets satisfying our criteria that would also postdate its pretraining.

The corresponding per-domain comparison is reported in Figure 4 (Appendix A).

Results On the full 32-dataset pool (Table 12), every mid-to-large model approaches the performance of 16-shot TabPFN-3 Plus (0.753): GPT-4o-mini 0.712, Qwen3-8B 0.698, Qwen3-14B 0.714, GLM-4.7 Flash 0.649. Only the smallest model falls well short (Qwen3-1.7B, 0.626). The

Dataset Group	GPT-4o-mini (gen)	Qwen3-1.7B (gen)	Qwen3-8B (gen)	Qwen3-14B (gen)	GLM-4.7 Flash (gen)	TabPFN-3 Plus (16-shot)
All	0.722 ± 0.008	0.605 ± 0.007	0.684 ± 0.001	0.693	0.649 ± 0.028	0.732 ± 0.062
Classic	0.731 ± 0.009	0.608 ± 0.007	0.689	0.699	0.667 ± 0.030	0.736 ± 0.060
New	0.664 ± 0.008	0.587 ± 0.005	0.657 ± 0.001	0.657	0.607 ± 0.023	0.693 ± 0.073

Table 3: Zero-shot LLMs vs. TabPFN-3 Plus (16-shot), broken down by dataset group: *All* (32 datasets), *Classic* (24 established benchmarks), and *New* (8 post-cutoff datasets).

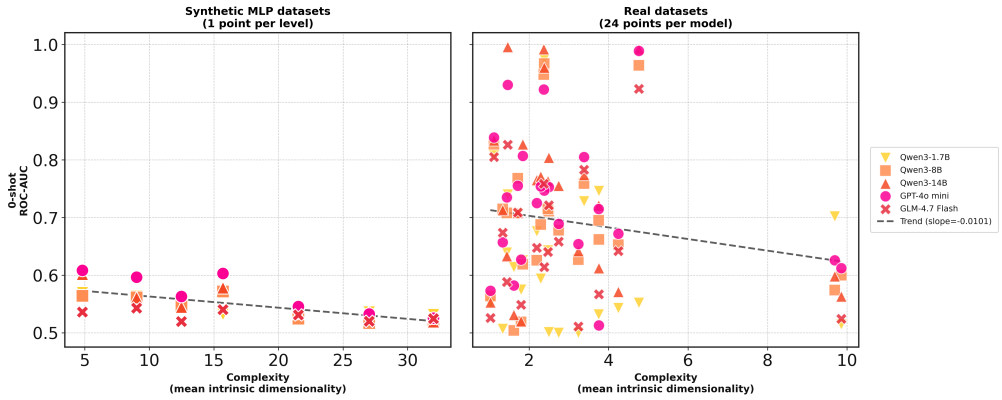

Figure 1: Intrinsic dimensionality vs. zero-shot ROC–AUC on the synthetic MLP suite and the real-world datasets.

dataset-level differences from random guessing and the Holm-adjusted Wilcoxon results are shown in Figure 6 (Appendix G.1). Splitting *Classic* vs. *New* moves every model, TabPFN-3 Plus included, down by a similar 0.05–0.08 (e.g. TabPFN-3 Plus $0.772 \rightarrow 0.695$) – a broadly shared drop consistent with the *New* datasets simply being harder, not with LLMs selectively losing a memorized advantage. Two robustness checks reinforce this reading without changing it: a two-step memorization probe (recognition + cell reconstruction; Appendix G.1) flags only 2 of 24 classic datasets as LIKELY-LEAKED, and direct prompting matches or exceeds two dedicated LLM-tabular pipelines, FeatLLM Han et al. (2024) and LLM Trees Knauer et al. (2025), at every shot count (Table 15, Appendix G.1).

Takeaway For mid-to-large models, zero-shot performance is comparable to that of TabPFN-3 Plus using 16 labeled examples; only the smallest model needs the data. The performance gap observed on post-cutoff datasets is more consistent with differences in dataset difficulty than with memorization.

4.2 H1B: A Predictable Complexity Threshold

If H1a holds only below a certain level of dataset complexity, practitioners need an LLM-independent criterion for determining whether a given dataset falls within this regime.

Setup We study controlled synthetic datasets generated from an MLP-based TabPFN-style prior Hollmann et al. (2022), with complexity varied through network depth, width, and hidden state size across seven difficulty levels and five random seeds (35 datasets total). For each level we compute an intrinsic-dimensionality score, averaged over several estimators from `scikit-dimension` Bac et al. (2021); Fukunaga & Olsen (1971); Mirkes et al. (2020); Fan et al. (2010); Levina & Bickel (2004); Facco et al. (2017); Carter et al. (2009); Gardner et al. (full protocol in Appendix B).

Results Intrinsic dimensionality rises steadily from 4.8 (MLP0) to 32.0 (MLP9; Table 22, Appendix G.7), and zero-shot ROC-AUC falls almost in lock-step: Qwen3-14B drops from 0.602 to 0.516 and GPT-4o-mini from 0.608 to 0.526, with nearly the entire decline already realized by MLP1 (dimensionality 9.0: Qwen3-14B 0.561, GPT-4o-mini 0.596) and only noise-level movement from MLP2 onward. The same ordering appears independently in 16-shot TabPFN-3 Plus's own performance on this grid (Figure 12, Appendix), and DeLong's test confirms a significant few-shot gain only at MLP0 (Appendix G.7).

Model	0-shot		Δ	64-shot		Δ
	Generation	Forward		Generation	Forward	
Qwen3-1.7B	0.6227 ± 0.1094	0.5319 ± 0.0644	-0.0908	0.6957 ± 0.1129	0.6374 ± 0.1166	-0.0583
Qwen3-8B	0.6975 ± 0.1429	0.6530 ± 0.1341	-0.0445	0.7766 ± 0.1299	0.7783 ± 0.1291	$+0.0017$
Qwen3-14B	0.7143 ± 0.1549	0.7014 ± 0.1425	-0.0129	0.7886 ± 0.1333	0.7762 ± 0.1312	-0.0124
GLM-4.7 Flash	0.6676 ± 0.1284	0.6549 ± 0.1165	-0.0127	0.7507 ± 0.1257	0.7458 ± 0.1707	-0.0050

Table 4: Forward scoring vs. generation: mean ROC-AUC $\pm$ standard deviation across the 24 classic + 8 new datasets (standard full-context prompt). Δ = forward $-$ generation.

Takeaway A dataset's intrinsic dimensionality – computable without ever calling an LLM – locates it on the same complexity scale where zero-shot performance collapses. In practice: below roughly dimensionality 9, zero-shot inference is most promising; above it, collect the data instead.

4.3 H1c: The Text-Tabular Niche

H1a and H1b are both established on structured tabular columns. Many industrial datasets, however, mix structured columns with free-form natural-language fields (product reviews, job descriptions, clinical notes). We ask whether the zero-shot-for-data substitution not only holds but *strengthens* once part of the signal is carried by text that label-aware tabular learners cannot read natively.

Setup We curate 8 industrial text-tabular datasets in which one or more columns carry free-form natural language alongside numerical and categorical features (full descriptions in Appendix H), decomposing each into a *text-only* variant (structured columns removed) and a *tabular-only* variant (text columns removed). TabPFN-3 Plus and CatBoost have no zero-shot mode, so their 0-shot cells are fixed at the random baseline (0.5). The corresponding shot trajectories are reported in Figure 5 (Appendix A).

Results At 0-shot (Table 24, Appendix H), GPT-4o-mini already scores *higher* on *text-only* (0.687) than on *full* (0.677) or *tabular-only* (0.640) – removing structured columns slightly improves performance. At 16 shots, TabPFN-3 Plus and CatBoost achieve above-chance text-only performance, but it is barely above chance (0.539 and 0.543): both stay 0.14+ ROC-AUC below GPT-4o-mini's zero-shot text-only number, and CatBoost's own *full* variant (0.578) still trails GPT-4o-mini's zero-shot *full* score (0.677). The ordering (*full* $>$ *text-only* $>$ *tabular-only* for TabPFN-3 Plus; *text-only* $\geq$ *full* $>$ *tabular-only* for the LLM) is set at zero shots and persists as shots increase (Figure 5).

Takeaway Zero-shot LLMs are most competitive when the predictive signal is concentrated in free-form text, which many conventional tabular learners cannot process directly. In this setting, LLMs may reduce not only the need for labeled examples but also the need for task-specific feature engineering.

4.4 H1d: Does Inference Mode Change the Picture?

H1a–H1c all use a fixed inference mode. We test whether the conclusions of H1a–H1c are robust to replacing autoregressive generation with forward scoring.

Generation autoregressively produces a label phrase that must be parsed, requiring multiple decoding steps and handling malformed or off-label outputs. *Forward scoring* instead obtains the two target-label log-probabilities in a single forward pass and normalizes them with a sigmoid, directly returning a normalized class probability without decoding or parsing.

Setup We compare both modes for models that expose token-level log-probabilities, at $k = 0$ and $k = 64$, across the 32 real datasets. GPT-4o-mini is omitted because its API does not expose the required target-label log-probabilities.

Results At 0-shot, forward scoring trails generation by -0.09 ROC-AUC for Qwen3-1.7B, shrinking to -0.04 for Qwen3-8B and to about -0.01 for Qwen3-14B and GLM-4.7 Flash – the gap closes steadily with model scale. At 64 shots it closes further still: -0.06 for Qwen3-1.7B, essentially zero or positive for the three larger models ($+0.002$, -0.012, -0.005). Only the smallest model keeps a non-negligible gap, and only at low shot counts. On the efficiency side, forward scoring is $\approx 4.5\times$ faster at 0-shot and $\approx 1.4\times$ faster at 64-shot, since the decoding stage it eliminates matters less once the shared few-shot prompt dominates prefill cost.

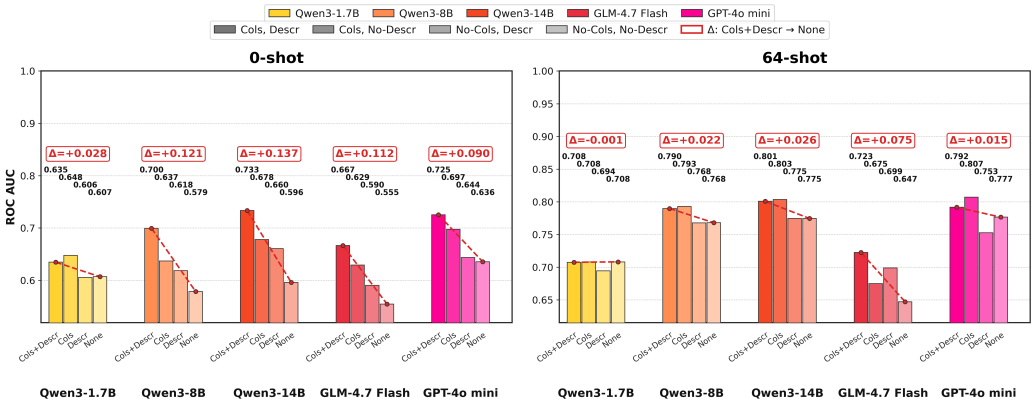

Figure 2: Context-configuration comparison on the 24 classic datasets at 0-shot (left) and 64-shot (right). Each group shows the four combinations of column names and natural-language task context for one model; the red dashed line marks the gap between the richest (Cols-Context) and poorest (NoCols-NoContext) configuration.

Takeaway Inference mode does not change H1a–H1c's conclusions. For mid-to-large open-weight models, forward scoring is a safe, substantially cheaper default; only the smallest model or the pure zero-shot regime still favors paying for generation.

5 Q2: Can Adding Context to the Prompt Replace Collecting Additional Few-Shot Examples?

If a task description or richer serialization can substitute for demonstrations, a practitioner who cannot collect more labeled examples might still improve performance simply by writing a better prompt. We evaluate this hypothesis in four steps: whether the improvement attributable to a task description alone (H2a), whether real demonstrations themselves have diminishing returns (H2b), whether *partial* context degrades performance once demonstrations are available (H2c), and whether serialization format has an effect comparable to that of adding demonstrations (H2d).

5.1 H2a: A Task Description Closes Part of the Gap

Setup We vary two channels through which in-context information reaches the model at 0-shot: *Schema indicators* (real column names vs. generic placeholders) and *Task context* (a natural-language task description vs. a neutral, uninformative instruction). Crossing the two gives four configurations, from *NoCols-NoContext* (minimal) to *Cols-Context* (maximal); full descriptions in Appendix B.

Results Moving from *NoCols-NoContext* to *Cols-NoContext* (Table 16, Appendix G.2) already recovers most of the total gain for four of five models: GPT-4o-mini $0.625 \rightarrow 0.682$, Qwen3-8B $0.579 \rightarrow 0.637$, Qwen3-14B $0.596 \rightarrow 0.678$, GLM-4.7 Flash $0.564 \rightarrow 0.637$. Stacking the task description on top (*Cols-Context*) adds a further 0.033–0.055 for those same four models, but for Qwen3-1.7B the identical step actually costs accuracy ($0.653 \rightarrow 0.640$). The joint gain (*NoCols-NoContext* $\rightarrow$ *Cols-Context*) is 0.10–0.14 ROC-AUC for the four larger models – larger than either channel alone, so the two are complementary rather than redundant.

Takeaway Schema indicators alone deliver most of the zero-shot gain from in-context information; the natural-language task description adds a further boost only at the 8 B parameter scale and above. This is a *partial* substitution: it recovers a meaningful fraction of what demonstrations would provide, but, as the next two sections show, it does not fully replace them.

5.2 H2b: Real Demonstrations Have Diminishing Returns

Setup For $k \in \{0, 4, 8, 16, 32, 64\}$ we prefix k randomly sampled labeled examples to the query and score ROC–AUC over all test instances. Significance of each k-shot gain over zero-shot is assessed with DeLong's test aggregated across the suite by Stouffer's method (Appendix F.2).

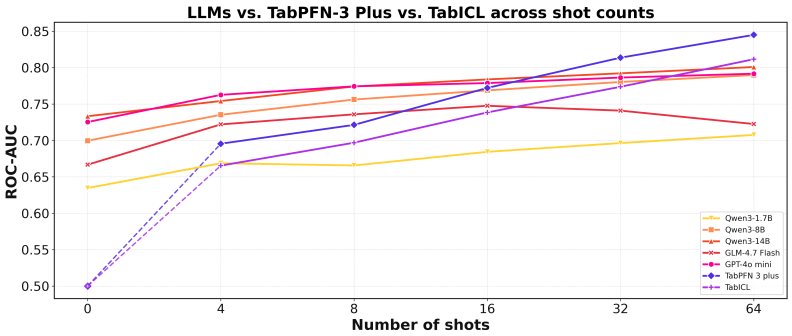

Figure 3: ROC–AUC vs. the number of in-context demonstrations on the 24 classic datasets, for all five LLMs alongside TabPFN-3 Plus and TabICL. TabPFN-3 Plus has no true zero-shot mode; its curve is extended left to chance level (0.5) as a dashed segment.

Results Every LLM's per-shot gain shrinks fast (Table 17, Appendix G.3): for Qwen3-8B the $0 \rightarrow 4$-shot step adds $+0.035$ ROC-AUC, but the entire $16 \rightarrow 64$-shot range adds only $+0.021$ combined; for GPT-4o-mini the $0 \rightarrow 4$ step alone ($+0.038$) already exceeds the $8 \rightarrow 64$ range combined ($+0.018$). GLM-4.7 Flash peaks at 16 shots (0.748) before *declining* to 0.723 by 64 shots. TabPFN-3 Plus, on the same table, shows the opposite shape: it keeps climbing without flattening across the same range ($0.696 \rightarrow 0.721 \rightarrow 0.772 \rightarrow 0.814 \rightarrow 0.845$ from 4 to 64 shots), overtaking every LLM's shot-saturated ceiling.

Takeaway LLMs do exploit demonstrations, but the marginal benefit per shot collapses quickly and is largely exhausted by 16 shots – unlike a model such as TabPFN-3 Plus, which keeps improving over the same range. Saturation here is a property of how LLMs use in-context examples, not of the underlying task. This 16-shot budget recurs in Section 6 as the point past which encoded expert knowledge stops adding value.

5.3 H2c: Incomplete Context Can Be Worse Than No Context

Setup We extend the H2a design with few-shot examples, drawing k labeled examples per (Model, Configuration, k) combination and tracking how the *Cols-Context* − *NoCols-NoContext* gap moves as k grows (Appendix B).

Results The full-vs-minimal-context gap shrinks monotonically with k for every model that uses context at all (Table 20, Appendix G.5): GPT-4o-mini falls from $+0.103$ at $k = 0$ to $+0.032$ at $k = 64$; Qwen3-8B from $+0.132$ to $+0.023$; Qwen3-14B from $+0.137$ to $+0.026$; GLM-4.7 Flash from $+0.081$ to $+0.007$. More strikingly, once enough demonstrations are present, the *complete absence* of context (*NoCols-NoContext*) can outperform the *partial*-cue configurations (*NoCols-Context*, *Cols-NoContext*) – clearest for GPT-4o-mini in the 64-shot panel of Figure 2, where the no-context, no-schema setting overtakes its intermediate counterparts.

Takeaway Context and demonstrations are partial substitutes rather than additive sources of information: context mainly improves performance when demonstrations are scarce, whereas partial context may interfere with the signal provided by demonstrations at higher shot counts.

5.4 H2d: Serialization Format vs. Shot Count

Setup We rank the five serialization formats introduced with the text-tabular suite (Section 4.3) by zero-shot ROC-AUC per (dataset, model) pair on the pooled 32-dataset set (Appendix H).

Results The best format (`markdown` or `html`, depending on the model) beats the worst by 1.0–1.5 mean rank positions, and masking column names costs a comparable amount regardless of which format carries it (Table 29, Appendix H). In absolute ROC-AUC terms this is consistently smaller than the 0.10–0.14 swing from H2a or the shot-driven gains from H2b, and the ordering is already set at zero shots.

Takeaway Serialization format is therefore a secondary optimization factor: although it can be tuned at relatively low cost, its effect is smaller than that of complete prompt context or additional demonstrations.

Prompting Regime	Qwen3-1.7B	Qwen3-8B	Qwen3-14B	GPT-4o-mini	GLM-4.7 Flash
Few-shot (16)	$\mathbf{0.620 \pm 0.058}$	0.654 ± 0.043	$\mathbf{0.695 \pm 0.046}$	0.718 ± 0.049	$\mathbf{0.722 \pm 0.046}$
Expert	0.573 ± 0.017	0.646 ± 0.010	0.645 ± 0.002	0.653 ± 0.009	0.637 ± 0.023
Prior	0.581 ± 0.007	**0.667**	0.665	$\mathbf{0.734 \pm 0.006}$	0.661 ± 0.027
Few-shot (16) + expert	0.590 ± 0.058	0.700 ± 0.036	0.719 ± 0.038	0.756 ± 0.042	0.717 ± 0.040
Few-shot (16) + prior	0.644 ± 0.042	0.772 ± 0.041	0.762 ± 0.040	0.795 ± 0.031	0.764 ± 0.041
Expert + prior	$\underline{0.732 \pm 0.006}$	0.825 ± 0.011	$\underline{0.862 \pm 0.002}$	0.779 ± 0.005	0.764 ± 0.027
Few-shot (16) + expert + prior	0.714 ± 0.060	$\underline{0.829 \pm 0.028}$	0.843 ± 0.029	$\underline{0.818 \pm 0.028}$	$\underline{0.771 \pm 0.024}$
TabPFN-3 Plus (16)	0.719 ± 0.058	0.719 ± 0.058	0.719 ± 0.058	0.723 ± 0.064	0.719 ± 0.058

Table 5: Comparison of ROC–AUC between prompting regimes with three in-context learning entities (*few-shot* demonstrations, *prior* natural-language task description, and *expert* LLM-distilled decision rules) on the LLM-synthetic suite. Best single-entity regime per model in **bold**; best overall configuration per model underlined.

6 Q3: Can Encoded Expert Knowledge Replace a Lack of Data?

A different kind of substitute for raw examples is explicit domain knowledge: if-then decision rules, feature-selection heuristics, or other codified expertise. We test whether supplying such rules substitutes for missing demonstrations, and under what conditions. We formalize what we mean by *prior context* and *expert knowledge* using the implicit Bayesian view of in-context learning (Xie et al., 2022).

6.1 H3: Expert Rules Need a Matching Prior

Setup We use the LLM-synthetic suite (Table 2): 24 datasets each generated by an LLM to mirror a real-world counterpart, together with the decision rules the same LLM produced during generation. We evaluate all seven combinations of three in-context learning entities – *few-shot* demonstrations (16), the natural-language task description (*prior*), and the LLM-distilled decision rules (*expert*; see Appendix E.3) – against a 16-shot TabPFN-3 Plus baseline (≈ 0.72; full protocol in Appendix B).

Results Alone, *Expert* never wins: it scores 0.573–0.653 across the four models (Table 21), below *Prior* alone (0.581–0.734) and below *Few-shot (16)* alone (0.620–0.718). Pairing *Expert* with *Prior* jumps to 0.732–0.862 (+0.13–0.22 over *Expert* alone), clearing the TabPFN-3 Plus baseline for three of four models; *Few-shot + expert* without *prior* still sits below *Few-shot* alone for the smallest model (0.590 vs. 0.620), while *Few-shot + expert + prior* reaches 0.714–0.843. The advantage of having rules at all shrinks as shots grow: by 16 examples, zero-shot *Expert + prior* already edges out *Few-shot + expert + prior* for two of four models – the same budget identified in H2b for saturation.

Takeaway Expert rules do not independently substitute for labeled data – they only help once paired with a contextual prior, and we have only verified this for rules consistent with the model's own knowledge. Where they do help, they compensate for missing demonstrations up to about 16 real examples; beyond that, the marginal benefit of rules is no longer observed.

7 Conclusion

In this paper, we introduce TabLens, a benchmark for evaluating whether LLM-based in-context learning can substitute for labeled data in tabular classification under data scarcity. Our results show that **Q1** zero-shot LLMs can replace limited data collection below a measurable complexity threshold, particularly for text-rich datasets; **Q2** complete semantic context can replace additional demonstrations up to approximately 16 shots; and **Q3** expert rules are beneficial only when paired with a matching contextual prior and within the same low-shot regime. The main limitation is the benchmark's focus on binary classification and a finite selection of models and datasets, which may restrict generalization to other tasks, domains, and adaptation settings.

GenAI usage disclosure. Generative AI tools were used to support the literature search, generate the synthetic datasets as described in the paper, and polish the manuscript's wording; all experimental design, analysis, and conclusions were developed and verified by the authors.

References

Alan Arazi, Eilam Shapira, Shoham Grunblat, Mor Ventura, Elad Hoffer, Gioia Blayer, David Holzmüller, Lennart Purucker, Gaël Varoquaux, and Frank Hutter. Multabench: Benchmarking

multimodal tabular learning with text and image. *arXiv preprint arXiv:2605.10616*, 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.

Jonathan Bac, Evgeny M Mirkes, Alexander N Gorban, Ivan Tyukin, and Andrei Zinovyev. Scikit-dimension: a python package for intrinsic dimension estimation. *Entropy*, 23(10):1368, 2021.

Bernd Bischl, Giuseppe Casalicchio, Matthias Feurer, Pieter Gijsbers, Frank Hutter, Michel Lang, Rafael Gomes Mantovani, Jan N van Rijn, and Joaquin Vanschoren. Openml: A benchmarking layer on top of openml to quickly create, download, and share systematic benchmarks. *NeurIPS–Track on Datasets and Benchmarks*, 2021.

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

Sebastian Bordt, Harsha Nori, Vanessa Rodrigues, Besmira Nushi, and Rich Caruana. Elephants never forget: Memorization and learning of tabular data in large language models. *arXiv preprint arXiv:2404.06209*, 2024.

Kevin M Carter, Raviv Raich, and Alfred O Hero III. On local intrinsic dimension estimation and its applications. *IEEE Transactions on Signal Processing*, 58(2):650–663, 2009.

Jintai Chen, Kuanlun Liao, Yao Wan, Danny Z Chen, and Jian Wu. Danets: Deep abstract networks for tabular data classification and regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 3930–3938, 2022.

Tianqi Chen. Xgboost: A scalable tree boosting system. *Cornell University*, 2016.

Zi-Jian Cheng, Zi-Yi Jia, Zhi Zhou, Lan-Zhe Guo, and Yu-Feng Li. TabFSBench: Tabular benchmark for feature shifts in open environments. In *International Conference on Machine Learning (ICML)*, 2025.

Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pp. 837–845, 1988.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*, 2025.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific reports*, 7(1):12140, 2017.

Mingyu Fan, Nannan Gu, Hong Qiao, and Bo Zhang. Intrinsic dimension estimation of data by principal component analysis. *arXiv preprint arXiv:1002.2050*, 2010.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

Keinosuke Fukunaga and David R Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on computers*, 100(2):176–183, 1971.

Josh Gardner, Juan C Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling, 2024. *URL https://arxiv. org/abs/2406.12031*.

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

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. Tabm: Advancing tabular deep learning with parameter-efficient ensembling, 2025. *URL https://arxiv. org/abs/2410.24210*.

Aditya Gorla and Ratish Puduppully. The illusion of generalization: Re-examining tabular language model evaluation. *arXiv preprint arXiv:2602.04031*, 2026.

Léo Grinsztajn et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026. URL https://arxiv.org/abs/2605.13986.

Sungwon Han, Jinsung Yoon, Sercan Ö. Arik, and Tomas Pfister. Large language models can automatically engineer features for few-shot tabular learning. In *International Conference on Machine Learning (ICML)*, 2024.

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, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: tabular data modeling using contextual embeddings (2020). *arXiv preprint arXiv:2012.06678*, 2012.

Sukriti Jaitly, Tanay Shah, Ashish Shugani, and Razik Singh Grewal. Towards better serialization of tabular data for few-shot classification with large language models. *arXiv preprint arXiv:2312.12464*, 2023.

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.

Ricardo Knauer, Mario Koddenbrock, Raphael Wallsberger, Nicholas M. Brisson, Georg N. Duda, Deborah Falla, David W. Evans, and Erik Rodner. "Oh LLM, I'm Asking Thee, Please Give Me a Decision Tree": Zero-Shot Decision Tree Induction and Embedding with Large Language Models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2025.

Elizaveta Levina and Peter Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in neural information processing systems*, 17, 2004.

Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. Language models are weak learners. *Advances in Neural Information Processing Systems*, 36:50907–50931, 2023.

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.

Evgeny M Mirkes, Jeza Allohibi, and Alexander Gorban. Fractional norms and quasinorms do not help to overcome the curse of dimensionality. *Entropy*, 22(10):1105, 2020.

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.

Prior Labs. TabPFN-3 (plus): Prior-data fitted networks for tabular classification. https://priorlabs.ai/tabpfn-3, 2025. Accessed 2026.

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. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=0VvD1PmNzM.

Xingjian Shi, Jonas Mueller, Nick Erickson, Mu Li, and Alexander J Smola. Benchmarking multimodal automl for tabular data with text fields. *arXiv preprint arXiv:2111.02705*, 2021.

Matteo Silvestri, Fabiano Veglianti, Flavio Giorgi, Fabrizio Silvestri, and Gabriele Tolomei. Evaluating latent knowledge of public tabular datasets in large language models. *arXiv preprint arXiv:2510.20351*, 2025.

Dylan Slack and Sameer Singh. Tablet: Learning from instructions for tabular data. *arXiv preprint arXiv:2304.13188*, 2023.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Gpt4table: Can large language models understand structured table data? a benchmark and empirical study. *arXiv preprint ArXiv:2305.13062*, 2023.

Dongjie Wang, Yanyong Huang, Wangyang Ying, Haoyue Bai, Nanxu Gong, Xinyuan Wang, Sixun Dong, Tao Zhe, Kunpeng Liu, Meng Xiao, et al. Towards data-centric ai: A comprehensive survey of traditional, reinforcement, and generative approaches for tabular data transformation. *arXiv preprint arXiv:2501.10555*, 2025.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, et al. Tablebench: A comprehensive and complex benchmark for table question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25497–25506, 2025.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations*, 2022.

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.

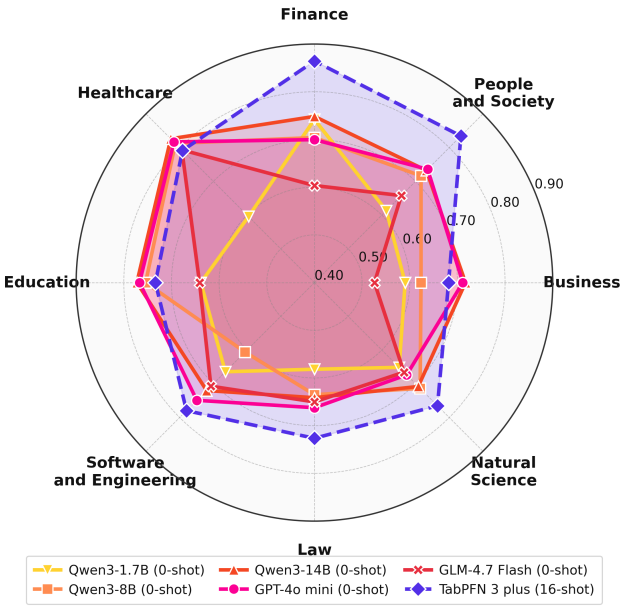

Figure 4: Per-domain mean ROC–AUC at 0-shot for the five LLMs vs. 16-shot TabPFN-3 Plus (dashed reference), on the 24 core real-world datasets grouped into 8 knowledge domains. Referenced from Section 4.1 (H1a).

A Key Result Figures

This section collects the two headline figures referenced most often from the main text – the per-domain zero-shot radar (H1a) and the combined GPT-4o-mini vs. TabPFN-3 Plus text-tabular comparison (H1c) – in one place, ahead of the per-question appendix sections that follow.

B Detailed evaluation protocol

This section states the task construction and evaluation protocol of each research question in full, expanding the condensed descriptions given in the main text.

B.1 RQ1: How competitive are LLMs in zero-shot tabular classification?

Setup This benchmark evaluates the zero-shot performance of multiple large language models (LLMs) on a binary classification task.

Experimental Design Building on this setup, we evaluate 5 LLMs, 2 inference modes, and 5 serialization regimes as mentioned in Section 3. For this RQ, 24 binary classification datasets from the core real-world suite are used (see Table 2).

Statistical Analysis To determine whether zero-shot performance is statistically superior to random guessing, we treat the 24 classic datasets as the independent experimental units. For each model, the dataset-level mean ROC–AUC is compared with 0.5 using a one-sided Wilcoxon signed-rank test; Holm-adjusted p-values control family-wise error across the five models. Full details are given in Appendix F.1.

B.2 RQ2: To what extent does zero-shot performance depend on in-context information versus LLM prior knowledge?

Setup This setting assesses whether zero-shot performance is primarily attributable to pretrained knowledge or to in-context information by measuring performance degradation under progressive removal of in-context information.

Experimental design We use the same fixed grid as above, except for additional ablations explained below.

- **Context Prompt (2 Levels):**

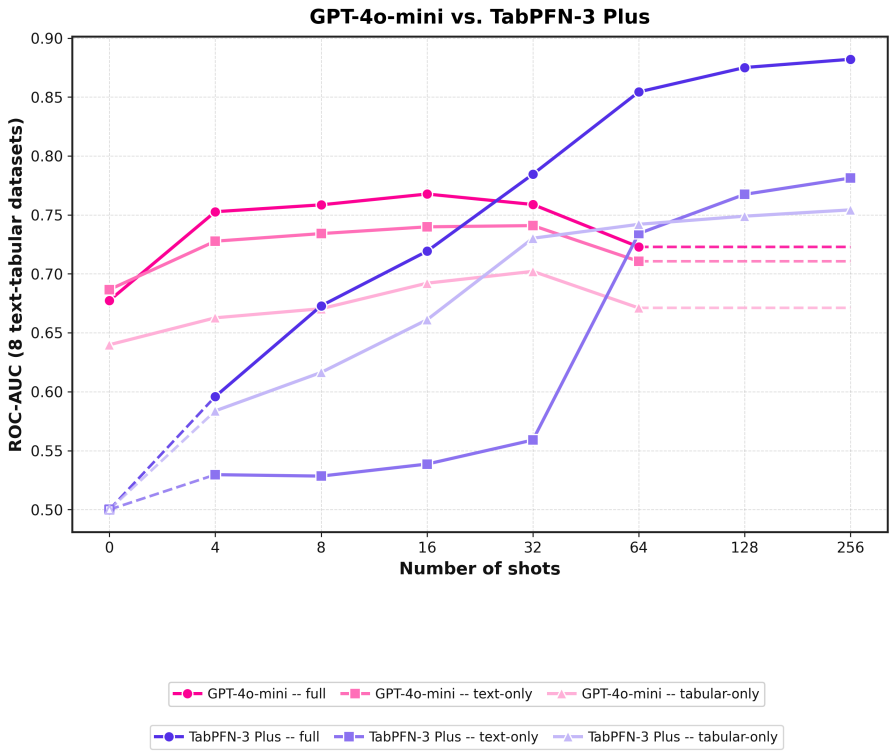

Figure 5: GPT-4o-mini vs. TabPFN-3 Plus across shot counts on the 8 text-tabular datasets, for the *full*, *text-only*, and *tabular-only* column variants. Every solid segment is a real measurement; TabPFN-3 Plus has no true zero-shot mode, so its curve is extended left to chance level (dashed) at $k = 0$, and GPT-4o-mini's last measured point (64 shots) is extended flat and dashed out to TabPFN-3 Plus's higher shot counts purely to keep both curves on a shared axis. Referenced from Section 4.3 (H1c).

Table 6: Descriptions of levels in the Graded Context Configuration setup referenced from Section 5.1.

Configuration Name	Configuration Index	Names of Columns	Prompt
Maximal Context	**Cols-Context**	Semantic	Context
Semantic Headers Only	**Cols-NoContext**	Semantic	No-context
Context Prompt Only	**NoCols-Context**	Generic	Context
Minimal Context	**NoCols-NoContext**	Generic	No-context

  - **With Context:** A prompt containing an explicit task description specifying the domain, classification objective, and target labels.
  - **Without Context:** A neutral instruction that does not disclose the domain, task, or target values (e.g., *"Perform binary classification for the following row."*).

- **Headers (2 Levels):**
  - **Semantic:** Real columns' names.
  - **Generic:** Placeholders instead of them.

- **Graded Context Configuration (4 Levels):** combining the Context Prompt (*With Context* / *Without Context*) with the Headers (*Semantic* / *Generic*) types (see Table 6 for description).

B.3 RQ4: How does removing in-context information affect few-shot performance across shot regimes?

Setup This section examines how few-shot examples interact with explicit contextual information in binary classification, specifically whether they enable models to compensate for missing task descriptions or semantic feature names.

Experimental Design This study extends the design established in RQ2 by introducing few-shot examples while maintaining the same systematic variation of contextual information.

Procedure

For each unique combination of factors (Model $\times$ Serialization $\times$ Context Prompt $\times$ Graded Configuration $\times$ k), k correctly labeled examples are sampled from the training set with a fixed seed. A prompt is then constructed by including either a context-specific prompt or a neutral instruction, appending the k few-shot examples formatted according to the chosen serialization method, and adding the test query using the same format.

To evaluate whether few-shot examples can compensate for missing contextual information, we conduct planned contrasts comparing minimal-context conditions with a high number of few-shot examples (e.g., (k = 64)) to maximal-context conditions with a small number of examples (e.g., (k = 4)). In addition, we analyze how performance scales with increasing k across different contextual configurations.

B.4 RQ5: Does combining in-context information, few-shot examples, and external instructions improve performance?

Setup This setting investigates whether combining explicit in-context information, few-shot demonstrations, and external knowledge in the form of decision rules improves predictive performance beyond using each source in isolation.

Experimental Design This study extends the design from RQ4 by introducing two novel experimental dimensions: synthetic datasets containing model-specific prior knowledge, and explicit decision rules as external instructions.

The experimental design varies two additional factors. First, we consider two dataset types: real datasets, consisting of the 24 datasets used in previous RQs, and synthetic datasets, which are model-generated counterparts created by prompting the same model to produce synthetic examples reflecting its learned priors about the domain, task, and label distribution (see Appendix E for the corresponding prompt). Second, we vary the type of external instructions provided in the prompt across three conditions: a baseline condition with no external instructions beyond context and few-shot examples; a condition where synthetic dataset examples generated by the model are included alongside or in place of real few-shot examples; and a condition where the prompt includes explicit decision rules generated by the LLM (e.g., "If $x_1 > 0.5$ and $x_3 < 2.0$, then class = 1") and expressed as if–then statements.

B.5 RQ6: How does dataset complexity influence LLM performance on tabular tasks?

Experiment Design We use the same models, inference modes, and zero-/few-shot settings as in the previous RQs. Data are serialized using two masked formats that are functionally equivalent in this context-free setting, given the absence of semantic feature headers. A single context-free prompt is employed, containing either only the formatted query instance in the zero-shot setting or k demonstration examples followed by the query instance in the few-shot setting. For each of the 35 synthetic datasets, experiments are conducted across all relevant factor combinations.

Statistical Analysis To assess whether LLMs can learn classification rules from data alone without semantic context, we compare k-shot performance to a zero-shot baseline using the same procedure as in RQ3. The null hypothesis states that k-shot performance with pure data patterns does not exceed zero-shot performance across all datasets.

Table 7: Core real-world suite (24 binary classification datasets). `ID` = identifier used in all per-dataset tables throughout the paper. #Feat. = total input features (target excluded); the sub-columns give the numerical/categorical split. % Pos. = prevalence of the positive class. *Dataset was class-stratified subsampled to 1,000 instances with seed `42`; original sizes are listed in the running text. ‡`irish`: ownership retained by Greaney & Kelleghan; usable only as an example for the development of statistical methods.

ID	Source	Size	#Feat.	Num.	Cat.	Target	% Pos.	License	Domain / Task (target = 1 means …)
Healthcare									
diabetes	OpenML-37	768	8	8	0	Outcome	34.9	CC0	Pima Indians diabetes diagnosis from physiological measurements. *Target = 1*: patient has diabetes.
transfusion	OpenML-1464	748	4	4	0	Donated 2007	23.8	CC BY 4.0	Blood-donor RFM-style features (Hsinchu, Taiwan). *Target = 1*: donor gave blood in March 2007.
cancer	OpenML-15	699	9	9	0	Class	34.5	CC BY 4.0	Wisconsin breast-cancer cell-nucleus cytology. *Target = 1*: cell is malignant; 0: benign.
Business									
marketing	OpenML-46940	2240	25	16	9	Response	14.9	CC0	Customer demographic and transactional features. *Target = 1*: customer responded to the marketing campaign.
telco	OpenML-42178	1000*	19	3	16	Churn	26.5	IBM	Telecom customer contract, billing, and usage features. *Target = 1*: customer churned (left within the last month).
hiring	Kaggle (avikumart)	1000*	17	8	9	label	81.4	Kaggle terms	HR analytics features for job candidates (interview, offer, hiring pipeline). *Target = 1*: candidate joined the company; 0: did not join.
People & Society									
adult	UCI Adult	1000*	13	6	8	income >50K	23.9	CC BY 4.0	US 1994 census demographic and employment features. *Target = 1*: annual income >$50K.
fitness	OpenML-46927	1500	6	3	3	attended	30.2	Public Domain	GoalZone fitness-club member and class registration (DataCamp, curated by TabArena). *Target = 1*: member attended the booked class.
tech_mental_health_survey	Kaggle (OSMI)	51	22	1	21	label	56.9	CC BY-SA	2014 OSMI survey of tech-industry employees (subsampled version). *Target = 1*: respondent feels employer takes mental health as seriously as physical health.
Finance									
credit	OpenML-31	1000	20	7	13	class	30.0	CC BY 4.0	German Credit (Statlog) bank-client financial and personal attributes. *Target = 1*: "good" credit risk (likely to repay); 0: "bad" risk.
audit	UCI / Kaggle	776	26	23	3	label	39.3	CC BY 4.0	Government-firm audit features (financial scores, risk scores, sector). *Target = 1*: firm has fraud risk; 0: no fraud risk.
fraud	Kaggle (neharoychoudhury)	1000*	14	6	8	is_fraud	12.8	Kaggle terms	Credit-card transaction features (merchant, category, amount, location, time). *Target = 1*: transaction is fraudulent; 0: legitimate.

C DATASETS

We use four dataset suites: **(i)** a *core real-world suite* of 24 binary-classification tasks spanning eight knowledge domains (Section C.1); **(ii)** a *new suite* of 8 datasets, one per domain, first released *after* every evaluated LLM's knowledge cutoff (October 2024), used to probe data leakage and out-of-cutoff generalisation (Section C.2); **(iii)** a *synthetic suite* generated from MLP priors at seven controlled difficulty levels (Section C.3); **(iv)** an *LLM-synthetic suite* obtained by prompting Qwen3-8B or GPT-4o-mini to fabricate tabular records matching the schemas of the 24 core datasets (Section C.4). Tables 7, 9 below list per-dataset metadata. Imbalance (% positive) refers to the prevalence of the positive class in the full dataset.

C.1 REAL DATASETS (CORE SUITE)

The core suite covers eight knowledge domains: healthcare, business, people & society, finance, education, software engineering, crimes & justice, and natural science (three datasets per domain). Datasets were selected to span small ($n = 151$, `tae`) to large ($n \approx 49$k, `adult`) sample sizes, low ($d = 4$) to high ($d = 41$) dimensionality, and both balanced ($\sim$50%) and severely imbalanced ($< 10\%$) class distributions. Three originally multi-class tasks (`tae`, `steel`, `san_francisco_crimes`) were collapsed to binary problems by grouping the two most populous classes against the rest. The full per-dataset metadata is given in Table 7.

Table 8: Core real-world suite (continued): the education, software engineering, crimes & justice, and natural science domains. Column definitions as in Table 7.

ID	Source	Size	#Feat.	Num.	Cat.	Target	% Pos.	License	Domain / Task (target = 1 means . . .)
Education									
tae	OpenML-955	151	5	3	2	binaryClass	34.0	Public	University TA evaluations (language, instructor, course, semester, group size). Binarised version: original 3-class target (low/medium/high) re-labelled with the majority class as positive (`P`) and the rest as negative (`N`).
irish	OpenML-451	500	5	2	3	Leaving_Cert	44.4	Research-only[‡]	Irish schoolchildren (age 11) educational-transition records (Greaney & Kelleghan, 1984). *Target = 1*: child took the Leaving Certificate; 0: did not.
student_famsup	UCI-320	1298	32	4	28	target	61.3	CC BY 4.0	Two Portuguese schools (math + Portuguese combined); features include school, demographics, study habits, and grades G1–G3. *Target = 1*: student receives family educational support (`famsup`); 0: does not.
Software Engineering									
pc4	OpenML-1049	1458	37	37	0	defects (c)	12.2	Public (NASA)	Static code metrics of a NASA flight-software module. *Target = 1*: module contains defects.
kc1	OpenML-1067	2109	21	21	0	defects	15.4	Public (NASA)	Static code metrics of a NASA storage-management module. *Target = 1*: module contains defects.
steel	OpenML-1504	1941	33	33	0	Class	65.3	CC BY 4.0	Steel-plate fault sensor readings. *Target = 1*: plate has defects other than the 6 named fault types; 0: no other defects. Originally 7 classes.
Crimes & Justice									
compas	OpenML-42193	1000*	13	5	8	two_year_recid	45.5	MIT	ProPublica COMPAS criminal-defendant features (Broward County). *Target = 1*: defendant re-offended within 2 years.
vote	OpenML-56	435	16	0	16	Class	61.4	CC BY 4.0	1984 US Congressional voting records on 16 key votes. *Target = 1*: representative is a Democrat; 0: Republican.
san_francisco_crimes	Kaggle (SFPD)	1000*	8	2	6	label	32.9	Public (SFPD)	SF Police incident records (date, district, address, coordinates, primary type). *Target = 1*: incident was resolved with an arrest or citation; 0: not resolved with arrest. Originally ∼39 classes.
Natural Science									
biodegr	OpenML-1494	1055	41	41	0	Class	33.7	CC BY 4.0	41 molecular descriptors of chemical compounds. *Target = 1*: ready-biodegradable; 0: not ready-biodegradable.
seismic_bumps	UCI / Kaggle	2584	18	14	4	class	6.6	CC BY 4.0	Sensor readings from a Polish coal mine. *Target = 1*: hazardous state — high-energy seismic bump will occur in the next shift; 0: non-hazardous.
bbbp	MoleculeNet	2039	18	18	0	p_np	76.5	MIT	Precomputed molecular descriptors. *Target = 1*: molecule penetrates the blood–brain barrier; 0: does not penetrate.

C.2 New Suite (post-cutoff datasets)

To probe whether observed zero-shot performance reflects memorisation, we assemble a parallel suite of eight datasets — one per knowledge domain — first released *after* the knowledge cutoff of every LLM evaluated in this paper (October 2024). All datasets are binary-classification tasks that follow the same preprocessing pipeline as the core suite.

Subsampling for large datasets. Several datasets in both the core and new suites contain more than 5,000 instances. To keep the experimental wall-clock and inference cost tractable for closed-source and large open models, we subsample every dataset whose original size exceeds 5,000 down to exactly 1,000 instances. Subsampling is class-stratified (so the positive-class rate reported in Tables 7 and 9 is preserved) and uses a fixed random seed of `42` for full reproducibility. The `Size` columns in both tables therefore report the size used in the experiments, not the source size. Affected datasets are: in the core suite, `telco` (orig. 7,043 → 1,000), `adult` (orig. 48,842 → 1,000), `hiring` (orig. 8,995 → 1,000), `fraud` (orig. 14,446 → 1,000), `san_francisco_crimes` (orig. ∼878,049 → 1,000), `compas` (orig. 5,278 → 1,000), and `bbbp` (orig. 2,039, kept as is). The new suite contains no dataset exceeding 5,000 instances.

Table 9: New suite (post-cutoff): 8 datasets, one per knowledge domain, published after October 2024. `Size`, `#Feat.`, `Num.`, `Cat.`, and `% Pos.` are computed directly from the released CSV files (index columns excluded). The target variable is hand-constructed for each dataset; the semantics of the positive class are described in the *Domain / Task* column. †Mendeley default licence (CC BY 4.0) — applies unless the dataset author explicitly opted for a different one.

ID	Source	Size	#Feat.	Num.	Cat.	Target	% Pos.	License	Domain / Task (target = 1 means . . .)
bank_credit_scoring	Kaggle (kapturovalexander)	1000	15	4	11	target	42.8	Kaggle terms	Finance — bank credit risk assessment. *Target = 1*: client has high credit scoring (credible, likely to repay the loan); 0: low credit scoring.
callcenter	Mendeley 8854gk66td	1020	12	3	9	target	88.0	CC BY 4.0†	Business — Kuwait 151 health hotline caller survey. *Target = 1*: caller was satisfied with the call-centre service; 0: dissatisfied.
postpartum	Mendeley 4nznnrk8cg	800	44	1	43	target	43.8	CC BY 4.0†	Healthcare — hospital female patients, postpartum depression. *Target = 1*: patient has postpartum depression; 0: does not.
machine	Kaggle (mujtabamatin)	1000	5	4	1	target	30.0	Kaggle terms	Software/Engineering — industrial machine sensor data. *Target = 1*: machine is at risk of failure; 0: normal operation.
stars	Kaggle (competition)	1000	6	5	1	target	23.9	Kaggle terms	Natural Science — Hipparcos-style photometric and parallax features. *Target = 1*: giant star; 0: dwarf star.
crimes_arrest	Kaggle (anacpricciardi)	1000	15	8	7	target	17.4	Kaggle terms	Crimes & Justice — Chicago crimes 2025 incident records. *Target = 1*: an arrest was made for this incident; 0: no arrest.
reading	Mendeley 24wf5n2gk5	582	16	0	16	target	61.7	CC BY 4.0†	Education — undergraduate students' reading-habits survey. *Target = 1*: student reads academic books most of the time; 0: reads non-academic books.
extrovert	Kaggle (snigdha247)	1007	10	0	10	target	30.5	Kaggle terms	People & Society — social-interaction preferences and emotional-response survey. *Target = 1*: respondent maintains long-distance friendships through phone calls; 0: uses texting/social media or rarely keeps in touch.

Table 10: MLP-prior difficulty levels.

Difficulty	Layers	Features	Hidden size
MLP-0	0	5	—
MLP-1	1	10	10
MLP-2	2	15	15
MLP-3	3	20	20
MLP-5	5	30	30
MLP-7	7	40	40
MLP-9	9	50	50

C.3 Synthetic Datasets (MLP-prior)

The synthetic suite is generated by sampling random multi-layer perceptrons of controlled depth and width, drawing inputs from $\mathcal{N}(0, I)$, and labelling them by the sign of the network output. Seven difficulty levels are obtained by varying layer count, hidden size, and input dimension; the resulting datasets are used in RQ6 to study how performance scales with task complexity. Table 10 gives the configuration of each level, and Table 11 reports the empirically estimated intrinsic dimensionality, averaged over seeds.

C.4 LLM-synthetic Datasets

For each of the 24 core real-world datasets we generate two LLM-synthetic counterparts by prompting Qwen3-14B and GPT-4o-mini, respectively, to fabricate tabular records matching the original schema and target distribution. The resulting datasets are suffixed `_ls_qwen` and `_ls_gpt` (e.g. `diabetes_ls_qwen`, `diabetes_ls_gpt`). They share the column structure and task description of their real counterparts but contain no leaked instances from the original data; this suite is the substrate for the knowledge-source ablations in RQ2, RQ4, and RQ5.

Generator ↔ backbone pairing. The `_ls_qwen` and `_ls_gpt` variants are not interchangeable evaluation targets: each is paired one-to-one with the model that generated it. The Qwen-generated variants are evaluated only on the two Qwen ICL backbones (Qwen3-8B and Qwen3-14B), and the GPT-generated variants only on GPT-4o-mini. This pairing makes Prompt 3 a clean probe of each

Table 11: Empirical intrinsic dimensionality of the synthetic datasets, averaged across seeds.

Dataset	Avg. intrinsic dim.
mlp0	5.0241
mlp1	9.3174
mlp2	12.8327
mlp3	16.0378
mlp5	21.9469
mlp7	27.4410
mlp9	32.3677

model's *own* internal task knowledge: the in-context examples, decision rules, and inference backbone all come from the same model family, so a non-trivial signal can be attributed to that model's prior knowledge rather than to a distribution mismatch between the generator and the evaluator. Cross-evaluation (e.g. running an `_ls_gpt` dataset on Qwen) is intentionally omitted throughout the appendix.

Three-step generation procedure. For each `(real_dataset, generator_LLM)` pair the synthetic dataset is produced through a chained, three-turn dialogue.[2] The dialogue is structured so that each subsequent turn receives the full text of the preceding answers as context, ensuring that the schema introduced in turn 1 is honoured by the data-generating function in turn 2 and by the labelling rules in turn 3.

- Turn 1 – Schema design.

  The generator is asked to propose a *prototype dataset* from the relevant high-level domain (e.g. *people and society*, *healthcare*) such that the classification task matches that of the real counterpart. The model decides on the number of features, their semantics and admissible ranges/categories.

- Turn 2 – Data-generating function.

  The generator is asked to return a Python function (using only `numpy` and `pandas`) that materialises rows according to the schema of turn 1. Numerical features are typically drawn from uniform or truncated normal distributions over the ranges declared in turn 1; categorical features are drawn uniformly over their declared support.

- Turn 3 – Decision rules.

  The generator is asked for three measurable, threshold-based decision rules that map the features to a binary target label. These three rules form the *ground-truth oracle* used to label rows produced in turn 2: a row is labelled positive if at least one of the three rules fires, and negative otherwise (the labelling logic visible inside the turn-3 code is what is actually executed).

The exact turn-1, turn-2 and turn-3 prompt templates, together with one fully worked example (the `diabetes_ls_qwen` dialogue), are reproduced in Section E.4.

Generation hyperparameters. Both generators are queried with their default decoding configuration (temperature, top-p, and `max_new_tokens` left at the provider's defaults; no custom sampling overrides). We do not tune any decoding hyperparameters per dataset or per turn. For each `(real_dataset, generator_LLM)` pair, we retain all rows produced in turn 2 as the final synthetic dataset (no rejection sampling, no rebalancing). The resulting class imbalance is therefore the natural outcome of the turn-3 oracle applied to the turn-2 distribution.

D SERIALIZATION FORMATS

Each row is rendered into text using one of five serialization formats; the two masked variants are used for the complexity grid.

[2] All three turns share the same system message: *"You are a statistical expert who uses only clearly measurable data."* This wording is meant to nudge the generator away from heavily subjective survey-style features.

- `feat_val`: A simple "Feature = value" listing. Example: "Features are: age = 39, education = Bachelor, gain = 2174. Answer is 0."
- `feat_val_mask`: A "Feature = value" listing with column names masked by proxy names. Example: "Features are: x_1 = 39, x_2 = Bachelor, x_3 = 2174. Answer is: y = 0."
- `markdown`: A Markdown-formatted table.
- `markdown_mask`: A Markdown table with feature (column-name) masking.
- `html`: An HTML-formatted table.

E PROMPTS

We evaluate every model under three prompt regimes, each isolating a different combination of knowledge sources. All three regimes share a common system message that instructs the model to emit a single token, `0` or `1`, with no additional text. The variable component across regimes is a per-dataset *task description* (which embeds the target semantics) and an optional *decision-rule block* (which embeds external expert knowledge synthesised by an auxiliary LLM).

Below we give the template for each regime followed by one fully instantiated example. Full per-dataset task descriptions and decision rules used in the experiments are released with the paper's code repository.

E.1 PROMPT 1 — CONTEXTUAL (FULL TASK DESCRIPTION, NO EXAMPLES)

Template. The contextual prompt describes the task, explains what the input features represent, and defines the meaning of both target labels. The features are then serialised one of five ways (see Section 3).

> *For the given input features ⟨of [entity]⟩ you need to predict ⟨[task in natural language]⟩. Your output should be just a single number representing a binary class prediction: 0 (⟨[negative-class meaning]⟩) or 1 (⟨[positive-class meaning]⟩). Do not predict any other tokens, only 0 or 1.*

Example (`diabetes`).

> For the given input features you need to predict, based on diagnostic measurements, whether a patient has diabetes. Your output should be just a single number representing a binary class prediction: 0 (does not have diabetes) or 1 (has diabetes). Do not predict any other tokens, only 0 or 1.

E.2 PROMPT 2 — CONTEXT-FREE (QUERY ONLY)

Template. The context-free prompt strips every domain hint from the task. The model sees only the serialised features and a generic two-class instruction; no entity, no target semantics, no column meaning. This is the prompt used to isolate the contribution of in-context demonstrations from prior knowledge.

> *For the given input features you need to classify rows into two classes: 0 or 1. Do not predict any other tokens, only 0 or 1.*

Example. Identical for every dataset — the verbatim template above.

E.3 PROMPT 3 — KNOWLEDGE-AUGMENTED (TASK DESCRIPTION + DECISION RULES)

This knowledge-augmented prompt defines the *expert* regime referred to in the main text: it augments the task description with explicit, human-readable decision rules.

Template. The knowledge-augmented prompt extends **Prompt 1** with an additional block listing the *decision rules by which the classes were generated*. These rules are produced by an auxiliary LLM (Qwen3-8B for the `_ls_qwen` suite and DeepSeek for the `_ls_gpt` suite) as a synthetic-data annotation step; they are *not* ground-truth rules for the real datasets and serve as a structured prior of external expert knowledge.

> *⟨Prompt 1 task description⟩ Here are the decision rules by which the classes were generated: ⟨[rule block — typically 3 rules expressed as threshold conditions or short Python functions]⟩*

Example (`diabetes`, decision rules synthesised by Qwen3-8B).

> For the given input features you need to predict, based on diagnostic measurements, whether a patient has diabetes. Your output should be just a single number representing a binary class prediction: 0 (does not have diabetes) or 1 (has diabetes). Do not predict any other tokens, only 0 or 1. Here are the decision rules by which the classes were generated:
>
> **Rule 1 (High Glucose + High BMI → Diabetes Risk).** If `Glucose > 120` AND `BMI > 30`, then `Diabetes = 1`.
>
> **Rule 2 (Age + Family History → Increased Risk).** If `Age > 50` AND `FamilyHistory == 1`, then `Diabetes = 1`.
>
> **Rule 3 (High Insulin + Low Physical Activity → Diabetes Risk).** If `Insulin > 150` AND `PhysicalActivity < 100`, then `Diabetes = 1`.

Coverage of Prompt 3. Prompt 3 is applied *only to the LLM-synthetic suite* (the `_ls_qwen` and `_ls_gpt` variants of the 24 core datasets), because the decision rules are valid only for data generated under those same rules; applying them to the real datasets would introduce label-rule mismatch. This explains why Prompt 3 columns are empty for real and new-suite datasets throughout the per-domain tables in Appendix G.3.

Per-dataset prompt text. Full prompt texts for all 24 core datasets, all 8 new-suite datasets, and all decision-rule blocks (one per dataset × rule source) are included verbatim in the code repository accompanying the paper, under `prompts/prompt1.json`, `prompts/prompt2.json`, and `prompts/prompt3_⟨qwen|gpt⟩.json`.

E.4 Prompts used to generate the LLM-synthetic suite

The LLM-synthetic suite described in Section C.4 is produced through a three-turn chained dialogue. All three turns share the same system message:

> *You are a statistical expert who uses only clearly measurable data.*

The user messages are templated on the real dataset's domain D (e.g. *healthcare*, *people and society*) and the real-task description T (e.g. *predict based on diagnostic measurements whether a patient has diabetes*). The literal strings are as follows.

Turn 1 — Schema design.

> *Generate a prototype dataset from the D domain. Use any number of features, the classification task should be to T.*

Turn 2 — Data-generating function. The verbatim text of the answer to turn 1 (denoted A_1) is appended to the user message before turn 2 is sent:

> *Generate python functions to create a dataset with the same features following the given prototype.*
> A_1

Turn 3 — Decision rules. The verbatim text of A_2 is appended to the user message before turn 3 is sent:

Generate python functions with 3 decision rules for determining a target label based on the features of this D dataset.
A_2

Materialisation. After turn 3 we extract the Python code blocks from A_2 (data-generating function) and A_3 (three decision rules), execute them in a sandboxed interpreter, and assemble the final synthetic table. A row is labelled positive iff at least one of the three turn-3 rules fires and negative otherwise.

Worked example (diabetes, Qwen3-14B). The full transcripts of A_1, A_2 and A_3 for the `diabetes_ls_qwen` dialogue (used as the running illustration in Section E.4) are reproduced verbatim in the supplementary material accompanying the paper. Briefly: turn 1 proposed an 11-feature schema covering standard diabetic-risk markers (`Age`, `BMI`, `Glucose`, `Insulin`, etc.); turn 2 returned a NumPy/Pandas function that draws each feature from a clipped normal or uniform distribution; turn 3 emitted three threshold rules — (i) `Glucose > 120 AND BMI > 30`, (ii) `Age > 50 AND FamilyHistory == 1`, (iii) `Insulin > 150 AND PhysicalActivity < 100` — which are OR-combined into the final binary label.

F Statistical Methodology

F.1 Wilcoxon signed-rank test against the random baseline

For binary classification, random ranking has expected ROC–AUC 0.5. Because the experiment stores aggregate metrics rather than per-example prediction scores, we do not construct an empirical null distribution and do not apply a test that requires individual predictions. Instead, the dataset is the unit of analysis, following the standard multiple-dataset comparison protocol of Demšar (2006).

For model m and classic dataset j, let $\overline{A}_{mj}$ be the zero-shot ROC–AUC averaged over the available independent runs (at most five) in the primary full-context generation configuration. We form the 24 paired differences

$$d_{mj} = \overline{A}_{mj} - 0.5 \tag{1}$$

and test $H_0 : \mathrm{median}_j(d_{mj}) \leq 0$ against the one-sided alternative $H_1 : \mathrm{median}_j(d_{mj}) > 0$ with the Wilcoxon signed-rank test. Zero differences are omitted from the signed-rank statistic in the standard way.

This construction deliberately avoids treating the at most five repeated runs as if they were many independent test examples. The independent units are the benchmark datasets; averaging runs only reduces sensitivity to demonstration sampling within each dataset. For each model we report the median paired difference and the matched-pairs rank-biserial correlation. Since five models are tested against the same null, raw p-values are adjusted with Holm's step-down procedure Holm (1979), controlling the family-wise error rate at $\alpha = 0.05$.

F.2 DeLong's test and Stouffer's method of p-value aggregation

For each dataset and each specific experimental condition (defined by Model, Mode, and a specific k value), the statistical significance of the difference between the k-shot ROC-AUC and the corresponding zero-shot ROC-AUC is assessed. This is performed using **DeLong's test** for two correlated ROC curves DeLong et al. (1988). This non-parametric test is specifically designed to compare AUCs derived from the same set of test instances, making it ideal for this paired, within-dataset comparison. The p-values obtained from DeLong's test across all datasets for a given experimental condition are not independent estimates but are observations of the same underlying k-shot effect across different tasks. To obtain a global assessment of significance for each condition, these p-values are aggregated using Stouffer's method.

G Visual results per research question

This appendix collects the principal figures supporting the research-question analyses of the main text, one block per RQ 1–6. Each block reproduces the headline plot(s) on which the corresponding conclusions rely; the underlying numbers and statistical tests are reported in the main text and in the per-domain tables of Section G.3.

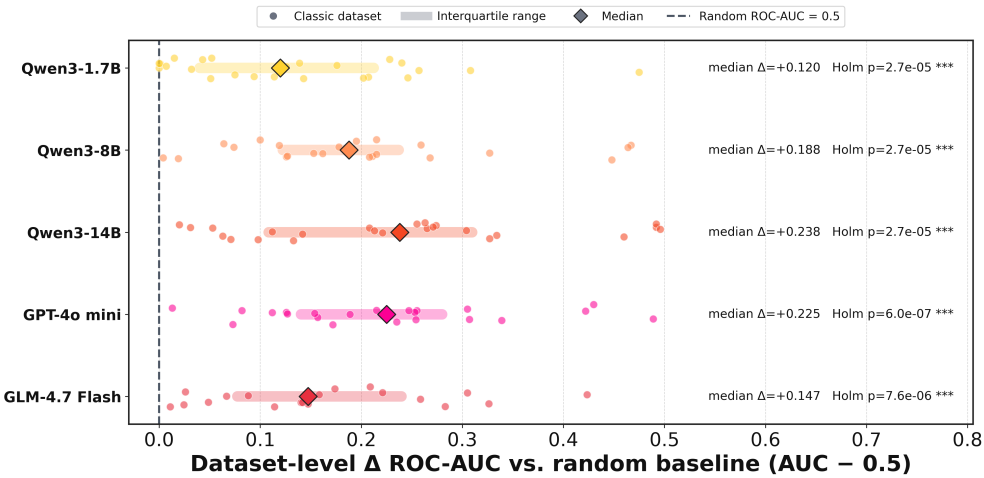

Figure 6: Zero-shot performance relative to random guessing on the 24 classic datasets. Each small point is one dataset-level mean ROC–AUC difference from 0.5; diamonds mark medians and thick segments show interquartile ranges. Reported p-values come from one-sided Wilcoxon signed-rank tests and are Holm-adjusted across the five models. GLM-4.7-Flash uses the available classic-dataset subset.

G.1 RQ 1 — Zero-shot competitiveness against TabPFN-3 Plus

RQ 1 asks whether LLMs are competitive in zero-shot tabular classification. Figure 4 (Appendix A) reports the per-domain comparison against 16-shot TabPFN-3 Plus; the aggregate classic vs. post-cutoff results are reported in Table 12.

Dataset Group	GPT-4o-mini (gen)	Qwen3-1.7B (gen)	Qwen3-8B (gen)	Qwen3-14B (gen)	GLM-4.7 Flash (gen)	TabPFN-3 Plus (16-shot)
All	0.722 ± 0.008	0.605 ± 0.007	0.684 ± 0.001	0.693	0.649 ± 0.028	0.732 ± 0.062
Classic	0.731 ± 0.009	0.608 ± 0.007	0.689	0.699	0.667 ± 0.030	0.736 ± 0.060
New	0.664 ± 0.008	0.587 ± 0.005	0.657 ± 0.001	0.657	0.607 ± 0.023	0.693 ± 0.073

Table 12: Zero-shot LLMs vs. TabPFN-3 Plus (16-shot), broken down by dataset group: *All* (32 datasets), *Classic* (24 established benchmarks), and *New* (8 post-cutoff datasets).

Table 13: Stage-1 recognition outcomes per model (out of 24 datasets), supporting the memorization-probe discussion in Section 4.1.

Outcome	Qwen3-14B	GPT-4o-mini	GLM-4.7 Flash
Leaked	11	3	9
Knows name only	2	9	4
Not recognized	11	12	11

Dedicated LLM-tabular pipelines vs. direct prompting. Table 15 reports the full shot grid for direct prompting against FeatLLM Han et al. (2024) and LLM Trees Knauer et al. (2025) on GPT-4o-mini, supporting the robustness check reported in Section 4.1: FeatLLM stays 0.02–0.03 ROC–AUC lower until nearly converging at 64 shots (0.789 vs. 0.793), and LLM Trees is substantially weaker (0.594) and shot-independent.

G.2 RQ 2 — In-context information vs. prior knowledge

RQ 2 disentangles the contributions of pretrained prior knowledge and in-context information by progressively removing contextual cues from the prompt. Table 16 gives the full per-model breakdown underlying the H2a discussion in Section 5.1; Figure 2 presents the same comparison in the main text.

G.3 RQ 3 — Few-shot effectiveness

RQ 3 quantifies the gains from few-shot demonstrations across $k \in \{0, 4, 8, 16, 32, 64\}$. Table 17 gives the full per-model grid underlying the H2b discussion in Section 5.2; Figure 3 presents the corresponding trajectories in the main text.

Table 14: Stage-2 cell-reconstruction results for datasets flagged at the recognition stage. For each (model, dataset) pair, we report the fraction of probed cells whose value the model reconstructs within the correctness window (*LLM*), the empirical random-baseline accuracy computed from the column's marginal distribution under the same window (*Random*), and the resulting memorization *lift*. Verdict thresholds: lift > 0.20 – *leaked*; > 0.10 – *likely leaked*; > 0.05 – *suspicious*; otherwise – *no leak signal*. Dashes indicate the dataset was not flagged at recognition for that model and therefore not probed.

	Qwen3-14B			**GPT-4o-mini**			
Dataset	LLM	Random	Lift	LLM	Random	Lift	Verdict
adult	0.420	0.385	$+0.035$	0.530	0.385	$+\mathbf{0.145}$	`adult`: likely leaked (GPT)
biodegr	0.390	0.468	-0.078	–	–	–	no leak signal
cancer	0.010	0.382	-0.372	–	–	–	no leak signal
compas	0.420	0.597	-0.177	–	–	–	no leak signal
credit	0.050	0.443	-0.393	–	–	–	no leak signal
diabetes	0.160	0.160	$+0.000$	0.090	0.160	-0.070	no leak signal
san_francisco_crimes	0.140	0.381	-0.241	–	–	–	no leak signal
tae	0.090	0.376	-0.286	–	–	–	no leak signal
telco	0.470	0.414	$+0.056$	–	–	–	`telco`: suspicious (Qwen)
transfusion	0.070	0.235	-0.165	0.040	0.235	-0.195	no leak signal
vote	0.190	0.486	-0.296	–	–	–	no leak signal

Method	0-shot	4-shot	8-shot	16-shot	32-shot	64-shot
Direct prompting	0.728 ± 0.118	0.752 ± 0.143	0.770 ± 0.132	0.774 ± 0.135	0.786 ± 0.135	0.793 ± 0.134
FeatLLM	0.703 ± 0.110	0.735 ± 0.111	0.745 ± 0.126	0.749 ± 0.124	0.766 ± 0.120	0.789 ± 0.127
LLM Trees	0.594 ± 0.078	0.594 ± 0.078	0.594 ± 0.078	0.594 ± 0.078	0.594 ± 0.078	0.594 ± 0.078

Table 15: Dedicated LLM-tabular pipelines vs. direct prompting on the classic suite with GPT-4o-mini, by number of in-context shots. *Direct prompting* is the same setup as Section 4.1/5.2. Cells are mean $\pm$ std ROC–AUC across datasets.

G.4 LLM-SPECIFIC TABULAR BASELINES ON THE NEW-DATASET BUNDLE

To situate our direct in-context approach against other ways of using LLMs for tabular prediction, we additionally compare it with two LLM-specific tabular baselines, **FeatLLM** and **LLM-tree**, on the eight post-cutoff (*new*) datasets introduced in RQ 1. Unlike our setup — which classifies each test row directly from the serialized prompt — these methods use the LLM as an intermediate step: FeatLLM prompts the LLM to extract discriminative decision rules / binary features that a lightweight downstream predictor then aggregates, while LLM-tree elicits a decision-tree-style classifier guided by the LLM. Both baselines use GPT-4o-mini as the backbone. We deliberately restrict this comparison to the *new* datasets, because they are released after the models' knowledge cutoff and thus provide the cleanest, contamination-free setting in which to contrast LLM-based methods.

Table 18 reports zero-shot ROC–AUC per dataset. The direct in-context GPT-4o-mini classifier is competitive with both LLM-specific baselines, leading on several datasets (e.g. `stars`, `reading`, `crimes_arrest`) while trailing on others (e.g. `callcenter`, `postpartum`), and the open-weight Qwen3 models stay in the same range. Table 19 reports the few-shot trajectory averaged over the new datasets; LLM-tree is omitted there because it does not support few-shot prompting. FeatLLM is competitive at low shot counts and is matched or modestly exceeded by the mid-to-large in-context LLMs as the number of shots grows — the same few-shot trend observed on the classic suite (RQ 3) therefore also holds against an LLM-specific baseline on contamination-clean data.

G.5 RQ 4 — CONTEXT CONTRIBUTION ACROSS SHOT REGIMES

RQ 4 extends RQ 2 to the few-shot regime: as the shot count k grows, how much of the headline performance is still attributable to the contextual cues in the prompt? Table 20 gives the full per-shot Δ = *Cols-Context* $-$ *NoCols-NoContext* underlying the H2c discussion in Section 5.3.

Figure 2 in the main text shows the four-way context comparison of RQ 2 at $k = 0$ and $k = 64$; the full per-shot values underlying it, including $k \in \{4, 8, 16, 32\}$, are given in Table 20 above.

Configuration	GPT-4o-mini	Qwen3-1.7B	Qwen3-8B	Qwen3-14B	GLM-4.7 Flash
NoCols-NoContext	0.625 ± 0.013	0.606 ± 0.004	0.579 ± 0.006	0.596 ± 0.002	0.564 ± 0.025
Cols-NoContext	0.682 ± 0.013	0.653 ± 0.003	0.637 ± 0.008	0.678	0.637 ± 0.026
NoCols-Context	0.641 ± 0.013	0.608 ± 0.010	0.624 ± 0.001	0.660	0.589 ± 0.031
Cols-Context	0.728 ± 0.011	0.640 ± 0.007	0.711	0.733	0.677 ± 0.032

Table 16: Context configurations comparison across models (0-shot). *Cols* indicates the presence of column names (schema indicators); *Context* indicates the presence of the natural-language task description.

Model	0-shot	4-shot	8-shot	16-shot	32-shot	64-shot
Qwen3-1.7B	0.635 ± 0.117	0.669 ± 0.122	0.666 ± 0.114	0.684 ± 0.116	0.696 ± 0.120	0.708 ± 0.113
Qwen3-8B	0.700 ± 0.126	0.735 ± 0.138	0.756 ± 0.128	0.769 ± 0.130	0.780 ± 0.126	0.790 ± 0.127
Qwen3-14B	0.733 ± 0.146	0.754 ± 0.134	0.774 ± 0.134	0.784 ± 0.131	0.792 ± 0.132	0.801 ± 0.130
GPT-4o mini	0.725 ± 0.117	0.763 ± 0.132	0.774 ± 0.131	0.779 ± 0.133	0.786 ± 0.135	0.792 ± 0.134
GLM-4.7 Flash	0.667 ± 0.111	0.722 ± 0.130	0.736 ± 0.137	0.748 ± 0.133	0.741 ± 0.117	0.723 ± 0.160
TabPFN	0.500	0.696 ± 0.137	0.721 ± 0.137	0.772 ± 0.142	0.814 ± 0.136	0.845 ± 0.121

Table 17: Mean ROC-AUC $\pm$ standard deviation across the 24 classic datasets as the number of in-context examples increases. TabPFN has no 0-shot inference, so its 0-shot entry is fixed to random ROC-AUC 0.5.

Dataset	GPT-4o-mini	Qwen3-1.7B	Qwen3-8B	Qwen3-14B	FeatLLM	LLM-tree
stars	0.995 ± 0.001	0.616 ± 0.001	0.982	0.990	0.959 ± 0.024	0.631 ± 0.120
machine	0.545 ± 0.016	0.533 ± 0.003	0.536	0.512	0.551 ± 0.009	0.504 ± 0.002
callcenter	0.522 ± 0.010	0.586 ± 0.001	0.510	0.543	0.638 ± 0.052	0.557 ± 0.011
bank_credit_scoring	0.707 ± 0.004	0.527 ± 0.001	0.529 ± 0.001	0.712	0.697 ± 0.028	0.579 ± 0.083
extrovert	0.568 ± 0.001	0.604	0.614	0.627	0.575 ± 0.008	0.548 ± 0.032
reading	0.797 ± 0.003	0.500	0.745 ± 0.002	0.800	0.620 ± 0.029	0.746 ± 0.075
postpartum	0.527 ± 0.009	0.696 ± 0.002	0.753 ± 0.001	0.561	0.731 ± 0.014	0.550 ± 0.034
crimes_arrest	0.650 ± 0.005	0.630 ± 0.013	0.590	0.514	0.619 ± 0.079	0.531 ± 0.011

Table 18: Zero-shot ROC–AUC on the eight post-cutoff *new* datasets: our direct in-context LLM classifiers vs. the LLM-specific tabular baselines FeatLLM and LLM-tree (both run with GPT-4o-mini as backbone). Entries are mean $\pm$ standard deviation.

Model	0-shot	4-shot	8-shot	16-shot	32-shot	64-shot
GPT-4o-mini	0.664 ± 0.008	0.711 ± 0.035	0.720 ± 0.035	0.723 ± 0.039	0.723 ± 0.040	0.730 ± 0.037
Qwen3-1.7B	0.587 ± 0.005	0.602 ± 0.057	0.599 ± 0.054	0.630 ± 0.040	0.637 ± 0.067	0.660 ± 0.038
Qwen3-8B	0.657 ± 0.001	0.695 ± 0.032	0.705 ± 0.038	0.716 ± 0.035	0.723 ± 0.033	0.739 ± 0.027
Qwen3-14B	0.657	0.707 ± 0.044	0.717 ± 0.044	0.731 ± 0.029	0.732 ± 0.031	0.752 ± 0.025
FeatLLM	0.674 ± 0.038	0.668 ± 0.051	0.680 ± 0.054	0.701 ± 0.057	0.703 ± 0.054	0.727 ± 0.049

Table 19: Few-shot ROC–AUC averaged over the *new* datasets vs. shot count, including the FeatLLM baseline (GPT-4o-mini backbone). LLM-tree is omitted because it does not support few-shot prompting.

Together they show that the gap between full-context and minimal-context configurations shrinks substantially with k, indicating partial substitution between in-context information and few-shot demonstrations.

G.6 RQ 5 — Combining ICL entities on LLM-synthetic data

RQ 5 asks whether combining in-context information, few-shot demonstrations, and external (LLM-generated) decision rules improves performance beyond using each source in isolation. We evaluate the combinations on the *LLM-synthetic* suite, where the model has maximal alignment with the dataset's data-generating process. Table 21 gives the full comparison underlying the H3 discussion in Section 6.1, and Figures 7–11 report shot-count curves, model comparison, and per-domain radar charts at 0 and 16 shots.

G.7 RQ 6 — Dataset complexity and the few-shot ceiling

RQ 6 uses the controlled MLP-synthetic complexity grid (Table 10) to delineate the level of intrinsic dataset difficulty at which few-shot demonstrations stop helping. Figure 12 shows the TabPFN-3 Plus baseline on this grid, which serves both as a reference foundation-model and as a sanity check

Model	0-shot	4-shot	8-shot	16-shot	32-shot	64-shot
GPT-4o-mini	0.103 ± 0.017	0.061 ± 0.065	0.062 ± 0.071	0.047 ± 0.067	0.041 ± 0.050	0.032 ± 0.038
Qwen3-1.7B	0.034 ± 0.008	0.019 ± 0.080	0.011 ± 0.100	0.026 ± 0.094	0.005 ± 0.080	0.001 ± 0.077
Qwen3-8B	0.132 ± 0.006	0.062 ± 0.075	0.062 ± 0.074	0.043 ± 0.070	0.040 ± 0.051	0.023 ± 0.038
Qwen3-14B	0.137 ± 0.002	0.060 ± 0.071	0.061 ± 0.069	0.036 ± 0.064	0.033 ± 0.045	0.026 ± 0.036
GLM-4.7 Flash	0.081 ± 0.044	0.018 ± 0.054	0.018 ± 0.068	0.020 ± 0.054	0.011 ± 0.055	0.007 ± 0.052

Table 20: Δ ROC–AUC = *Cols-Context* − *NoCols-NoContext* across shot counts on the classic suite.

Prompting Regime	**Qwen3-1.7B**	**Qwen3-8B**	**Qwen3-14B**	**GPT-4o-mini**	**GLM-4.7 Flash**
Few-shot (16)	$\mathbf{0.620 \pm 0.058}$	0.654 ± 0.043	$\mathbf{0.695 \pm 0.046}$	0.718 ± 0.049	$\mathbf{0.722 \pm 0.046}$
Expert	0.573 ± 0.017	0.646 ± 0.010	0.645 ± 0.002	0.653 ± 0.009	0.637 ± 0.023
Prior	0.581 ± 0.007	**0.667**	0.665	$\mathbf{0.734 \pm 0.006}$	0.661 ± 0.027
Few-shot (16) + expert	0.590 ± 0.058	0.700 ± 0.036	0.719 ± 0.038	0.756 ± 0.042	0.717 ± 0.040
Few-shot (16) + prior	0.644 ± 0.042	0.772 ± 0.041	0.762 ± 0.040	0.795 ± 0.031	0.764 ± 0.041
Expert + prior	$\underline{0.732 \pm 0.006}$	0.825 ± 0.011	$\underline{0.862 \pm 0.002}$	0.779 ± 0.005	0.764 ± 0.027
Few-shot (16) + expert + prior	0.714 ± 0.060	$\underline{0.829 \pm 0.028}$	0.843 ± 0.029	$\underline{0.818 \pm 0.028}$	$\underline{0.771 \pm 0.024}$
TabPFN-3 Plus (16)	0.719 ± 0.058	0.719 ± 0.058	0.719 ± 0.058	0.723 ± 0.064	0.719 ± 0.058

Table 21: Comparison of ROC–AUC between prompting regimes with three in-context learning entities (*few-shot* demonstrations, *prior* natural-language task description, and *expert* LLM-distilled decision rules) on the LLM-synthetic suite. Best single-entity regime per model in **bold**; best overall configuration per model underlined.

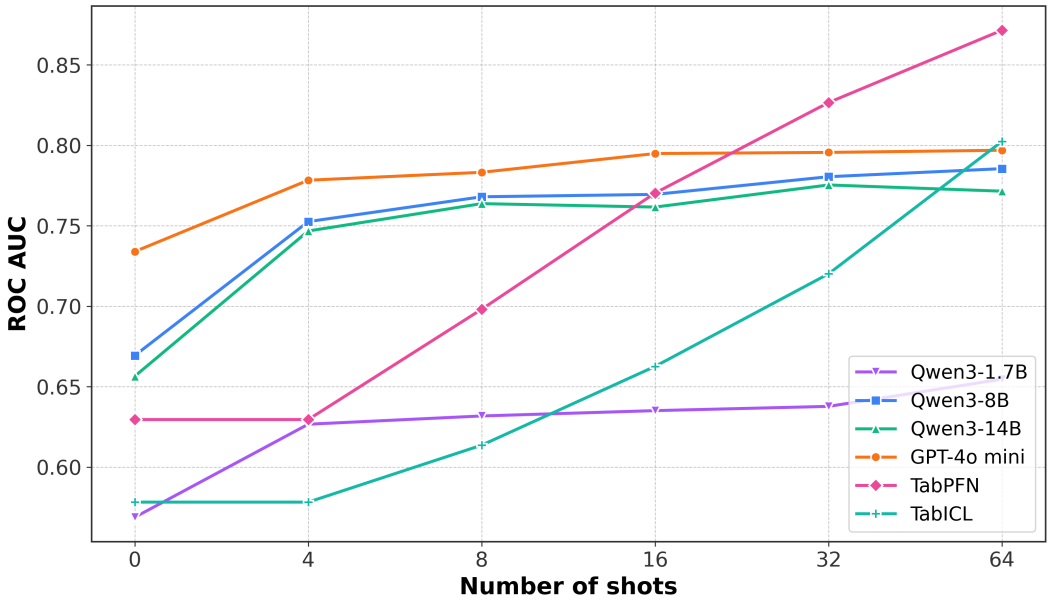

Figure 7: ROC–AUC vs. number of in-context shots on the LLM-synthetic suite. Performance saturates earlier than on real data because the model already aligns with the data-generating distribution at $k = 0$.

Synthetic Dataset	Intrinsic Dim.	Qwen3-14B (0-shot)	GPT-4o-mini (0-shot)	GLM-4.7 Flash (0-shot)
MLP0	4.830	0.602 ± 0.050	0.608 ± 0.083	0.536 ± 0.016
MLP1	9.012	0.561 ± 0.042	0.596 ± 0.063	0.543 ± 0.025
MLP2	12.480	0.545 ± 0.022	0.563 ± 0.020	0.520 ± 0.009
MLP3	15.705	0.578 ± 0.043	0.603 ± 0.064	0.540 ± 0.017
MLP5	21.544	0.534 ± 0.046	0.546 ± 0.033	0.531 ± 0.014
MLP7	27.023	0.541 ± 0.019	0.533 ± 0.014	0.520 ± 0.010
MLP9	31.994	0.516 ± 0.008	0.526 ± 0.013	0.525 ± 0.003

Table 22: Intrinsic dimensionality and zero-shot ROC-AUC on the synthetic MLP suite. Intrinsic dimensionality and model quality are averaged across five seeds per complexity level; uncertainty reports the standard deviation across seeds.

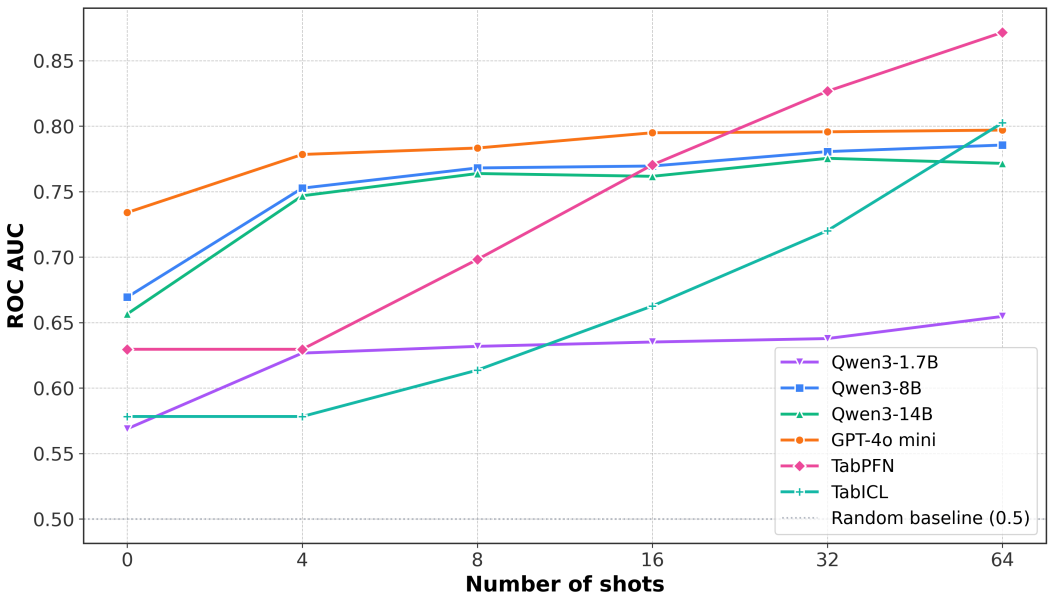

Figure 8: Cross-model comparison on the LLM-synthetic suite, including the foundation baselines TabPFN-3 Plus and TabICL. The relative ranking on LLM-synthetic differs from the ranking on real datasets, reflecting alignment between the generator and its own test distribution.

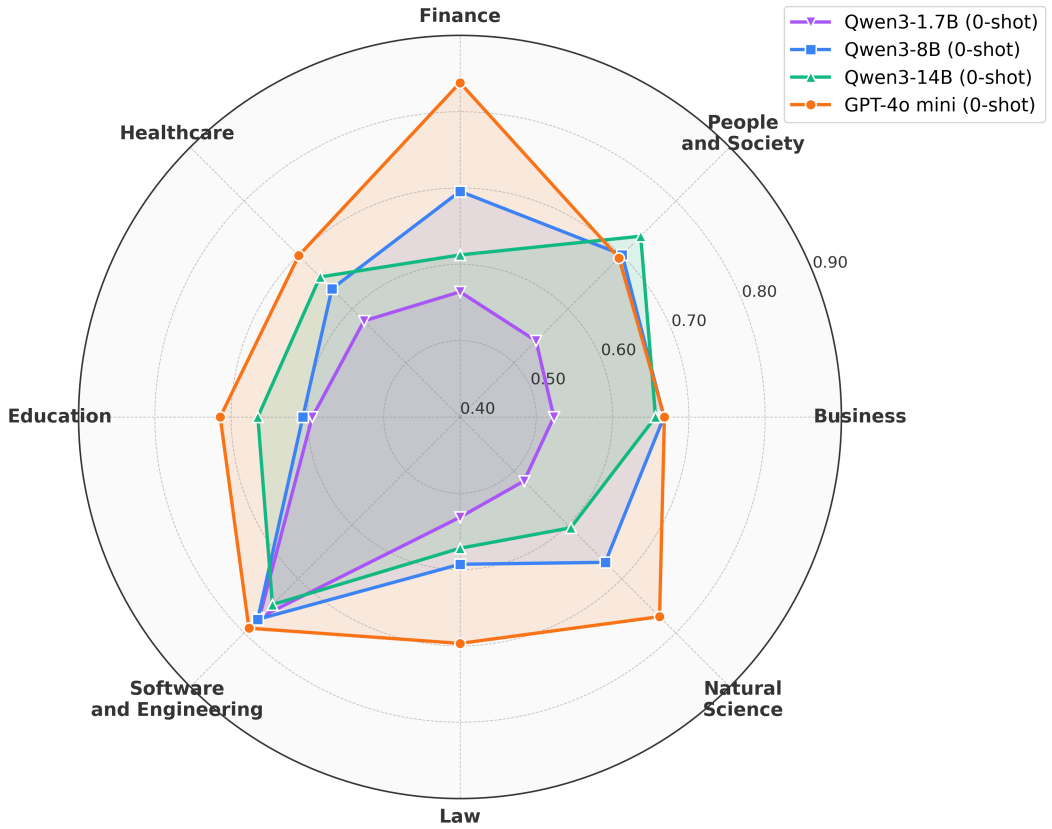

Figure 9: 0-shot per-domain radar on the LLM-synthetic suite. All LLMs reach or exceed the foundation-model trajectories on the synthetic counterparts of their own training distribution.

that the difficulty axis is well-ordered. Figure 13 reports the statistical significance of the per-shot gain over the zero-shot baseline at each MLP level (DeLong's test combined with Stouffer's method, aggregated across five generation seeds per level).

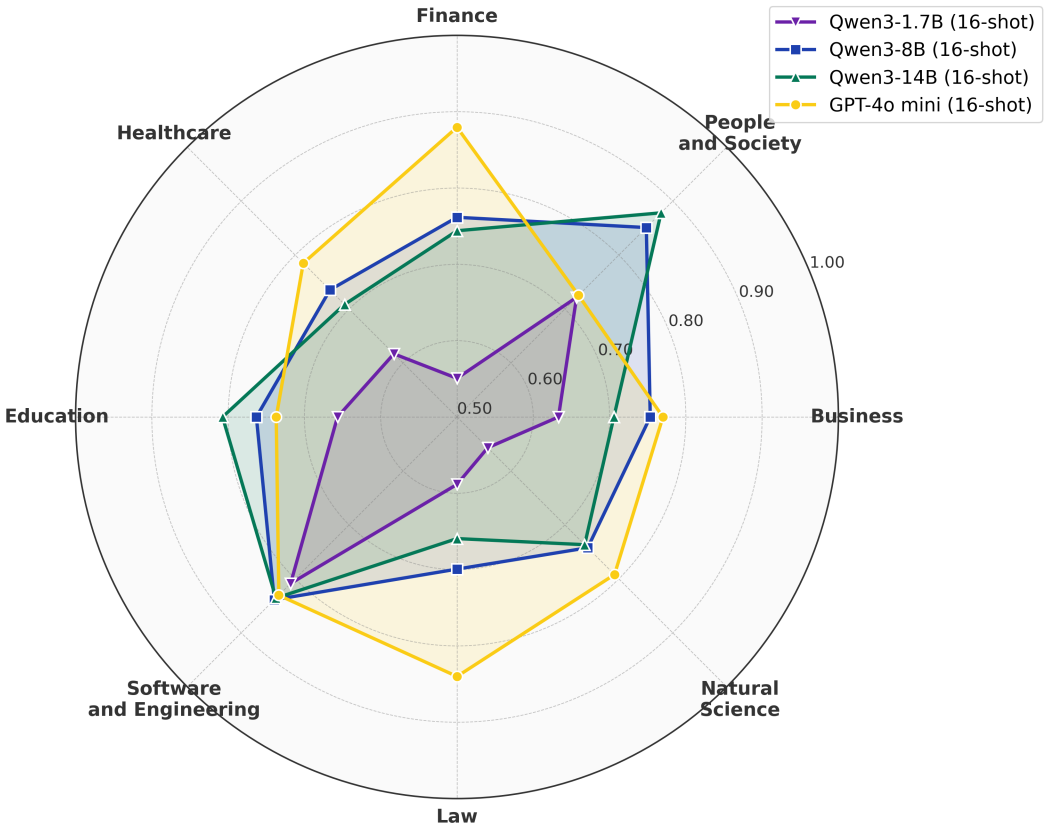

Figure 10: 16-shot per-domain radar on the LLM-synthetic suite. The addition of demonstrations narrows the spread between models; smaller LLMs benefit the most relative to their 0-shot value.

Dataset	GPT-4o-mini			Qwen3-8B			Qwen3-14B			GLM-4.7 Flash		
	0-shot	4-shot	16-shot	0-shot	4-shot	16-shot	0-shot	4-shot	16-shot	0-shot	4-shot	16-shot
MLP0	0.608 ± 0.004	0.645 ± 0.065	0.708 ± 0.055	0.564 ± 0.001	0.639 ± 0.070	0.713 ± 0.059	0.602	0.688 ± 0.088	0.748 ± 0.059	0.536 ± 0.023	0.605 ± 0.057	0.685 ± 0.057
MLP1	0.596 ± 0.007	0.593 ± 0.023	0.578 ± 0.033	0.562 ± 0.004	0.587 ± 0.032	0.568 ± 0.051	0.562	0.584 ± 0.032	0.579 ± 0.053	0.543 ± 0.014	0.555 ± 0.031	0.563 ± 0.039
MLP2	0.563 ± 0.010	0.558 ± 0.023	0.566 ± 0.020	0.547 ± 0.004	0.564 ± 0.022	0.580 ± 0.030	0.535	0.564 ± 0.029	0.577 ± 0.028	0.520 ± 0.011	0.552 ± 0.026	0.550 ± 0.025
MLP3	0.603 ± 0.008	0.569 ± 0.035	0.581 ± 0.026	0.566 ± 0.011	0.576 ± 0.027	0.573 ± 0.027	0.583 ± 0.001	0.587 ± 0.041	0.587 ± 0.031	0.540 ± 0.019	0.558 ± 0.026	0.574 ± 0.030
MLP5	0.546 ± 0.006	0.542 ± 0.014	0.541 ± 0.013	0.527 ± 0.007	0.536 ± 0.016	0.538 ± 0.018	0.541	0.537 ± 0.015	0.543 ± 0.026	0.531 ± 0.012	0.537 ± 0.023	0.536 ± 0.022
MLP7	0.533 ± 0.005	0.538 ± 0.019	0.541 ± 0.012	0.523 ± 0.006	0.534 ± 0.018	0.535 ± 0.018	0.537	0.534 ± 0.017	0.540 ± 0.019	0.520 ± 0.012	0.524 ± 0.016	0.510 ± 0.011
MLP9	0.526 ± 0.007	0.526 ± 0.012	0.525 ± 0.009	0.526 ± 0.009	0.525 ± 0.016	0.529 ± 0.014	0.521	0.528 ± 0.012	0.523 ± 0.011	0.525 ± 0.016	0.528 ± 0.022	0.513 ± 0.005

Table 23: ROC–AUC on MLP-synthetic datasets at $\{0, 4, 16\}$ shots, across model and complexity level. Complexity increases from MLP0 (simplest) to MLP9 (most complex).

H TEXT-TABULAR NICHE STUDY (H1C)

This appendix supports Section 4.3. It gives the full text-tabular dataset descriptions, per-dataset results for all three column-level variants, the complete serialization ablation, and the meta-feature analysis referenced from the main text.

Shots	Dataset View	Qwen3-14B	GPT-4o-mini	GLM-4.7 Flash	TabPFN	CatBoost
0-shot	Full dataset	0.650 ± 0.108	0.677 ± 0.134	—	0.500	0.500
	Text only	0.649 ± 0.105	0.687 ± 0.129	—	0.500	0.500
	Tabular only	0.629 ± 0.110	0.640 ± 0.111	—	0.500	0.500
16-shot	Full dataset	0.719 ± 0.126	0.768 ± 0.129	—	0.719 ± 0.143	0.578 ± 0.010
	Text only	0.725 ± 0.122	0.740 ± 0.132	—	0.539 ± 0.070	0.543 ± 0.027
	Tabular only	0.699 ± 0.131	0.692 ± 0.121	—	0.661 ± 0.122	0.605 ± 0.069

Table 24: Mean ROC-AUC $\pm$ standard deviation across the eight text–tabular datasets for full, text-only, and tabular-only views. Since TabPFN and CatBoost do not support label-free inference, their 0-shot entries are fixed to random ROC-AUC 0.5.

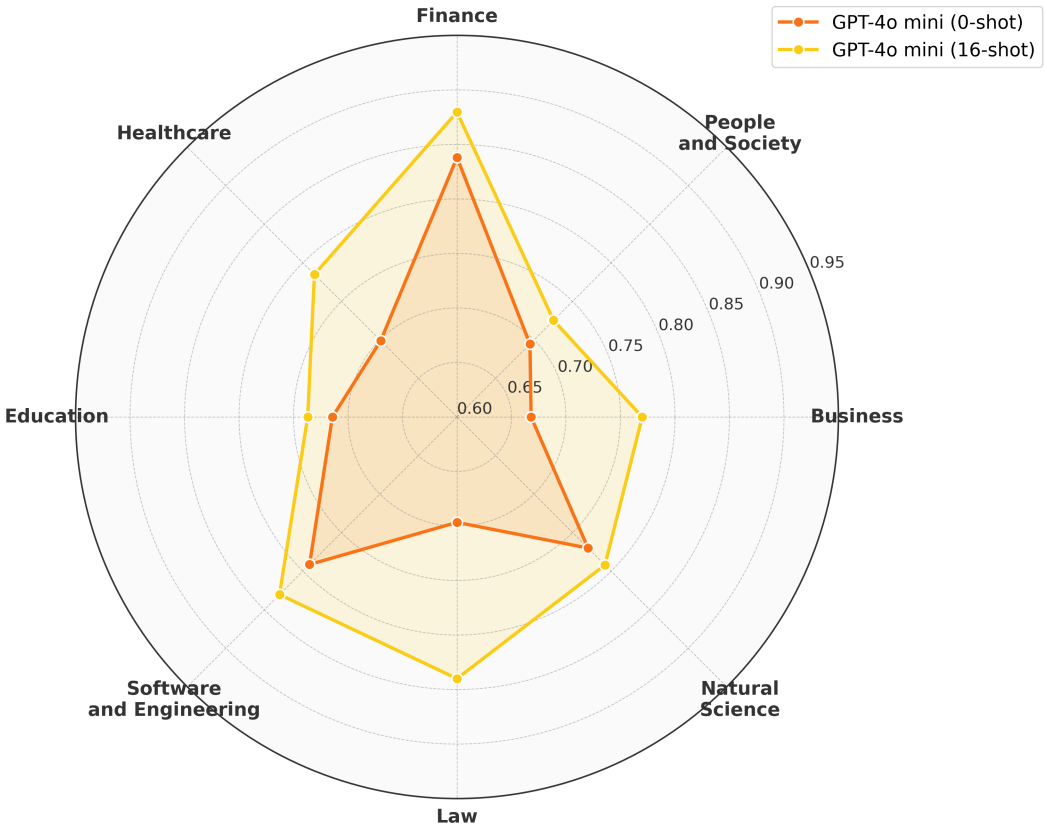

Figure 11: GPT-4o-mini: 0-shot vs. 16-shot per-domain radar on the LLM-synthetic suite. Confirms the same saturation pattern observed on real data, with a smaller absolute gain because the 0-shot baseline is already close to the model's ceiling on its own synthetic distribution.

H.1 TEXT-TABULAR DATASETS

The 8 text-tabular datasets are summarized in Table 25. Each is additionally evaluated as a *text-only* variant (structured columns removed) and a *tabular-only* variant (text columns removed). Datasets larger than 2,000 rows are subsampled to 1,000 rows, preserving class balance.

- **Real / Fake Job Postings**: binary classification of job listings as fraudulent vs. genuine; text columns include the title, description, and requirements.
- **California Real Estate Market 2026**: predict the property type, binarized from the multi-class label; one free-form description column alongside structured listing features. Released in 2026 and therefore unavailable to LLM pretraining as of the models' cutoffs.
- **Wine Tasting Reviews**: binarized wine-quality score from review text and structured wine metadata.
- **Amazon Fine Food Reviews**: binarized review score (low vs. high); text columns include the review body and summary.
- **Fake Reviews Dataset**: classifier of computer-generated vs. genuine product reviews; a single review-text column with one numerical and one categorical feature.
- **PubMed MeSH**: a single MeSH term reduced to a binary topic indicator; abstracts and titles are the text columns.
- **Food Recipes**: binarized recipe category from ingredients, instructions, and recipe metadata.
- **Online Teaching Feedback**: sentiment-derived binary target from student feedback text and lecture metadata.

H.2 PER-DATASET RESULTS

Tables 26, 27, and 28 report per-dataset zero-shot ROC-AUC for the *full*, *text-only*, and *tabular-only* variants respectively, underlying the aggregates in Section 4.3.

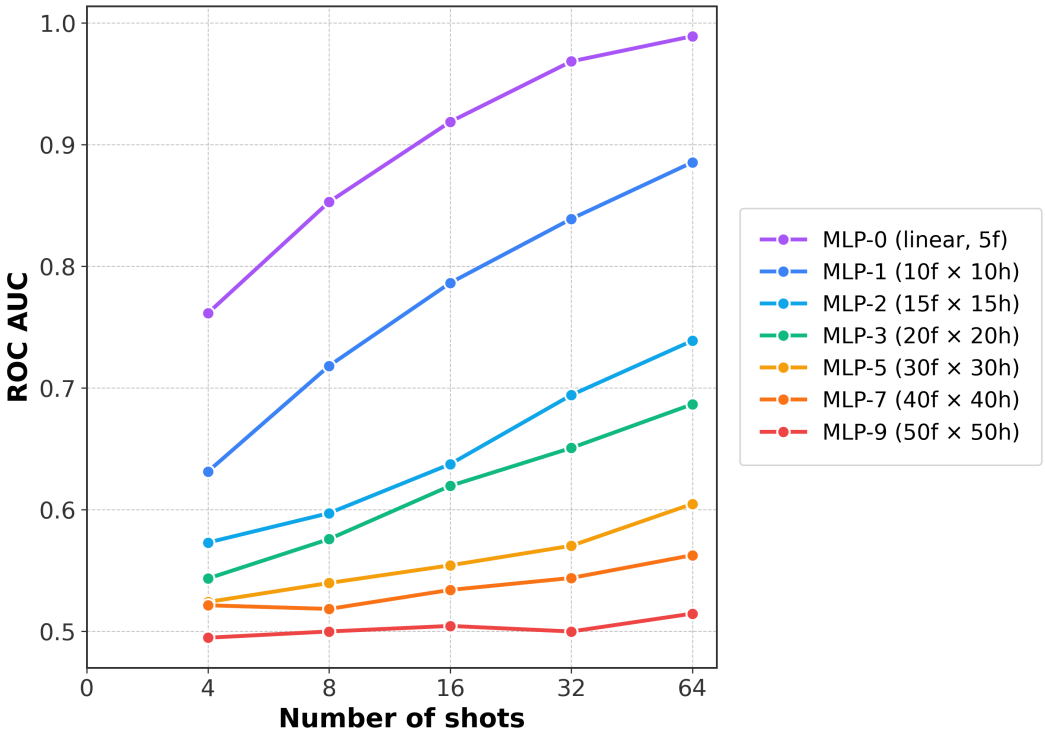

Figure 12: TabPFN-3 Plus ROC–AUC across the seven MLP-prior difficulty levels. The curve flattens as complexity rises, defining the upper end of the grid we use to test LLMs in Section 4.2 of the main text.

Table 25: Text-tabular dataset collection. Datasets larger than 4,000 rows were subsampled to 1,000 stratified by label (marked "curated to 1,000"); smaller datasets were kept at their original size. Column-type counts are reported per the source.

Dataset	Source	#num	#cat	#text	Working size	Original
fake_job_postings	Kaggle	1	7	8	curated to 1,000	18,000
california_price	Kaggle	9	2	1	curated to 1,000	9,500
wine_testing_reviews	Kaggle	2	4	8	curated to 1,000	130,000
reviews_amazon	Kaggle	3	1	2	curated to 1,000	570,000
pubmed	Kaggle	0	4	2	curated to 1,000	50,000
recipes	Kaggle	14	5	9	1,000 (original)	1,000
fake_reviews	Kaggle	1	1	1	curated to 1,000	40,000
online_teaching	Kaggle	5	0	1	1,400 (original)	1,400

H.3 Serialization ablation

Table 29 reports the mean serialization rank (1 = best, 5 = worst) on the combined pool of 32 datasets (24 classical + 8 text-tabular *full*), supporting the H2d discussion in Section 5.4. Tables 30–32 break the same ranking down by pool and column-level variant.

H.4 Memorization check on the text-tabular suite

As with the core suite (Appendix G.1), we cannot fully rule out that LLMs retain some memorized signal about the underlying datasets from pretraining Bordt et al. (2024). We rely on two mitigations: the `california_price` dataset was released in 2026 and is therefore unavailable to any evaluated model's pretraining corpus, and it shows no systematic advantage over the other 7 text-tabular datasets in the main-text comparisons. Combined with the recognition/reconstruction probes already reported for the core suite in Appendix G.1, and with the prior zero-shot-vs-post-cutoff comparison on the core suite (Section 4.1), this suggests the text-tabular niche identified in Section 4.3 reflects a structural property of the column-type mix rather than dataset-specific memorization, though residual per-dataset contamination cannot be excluded given the suite's size ($n = 8$).

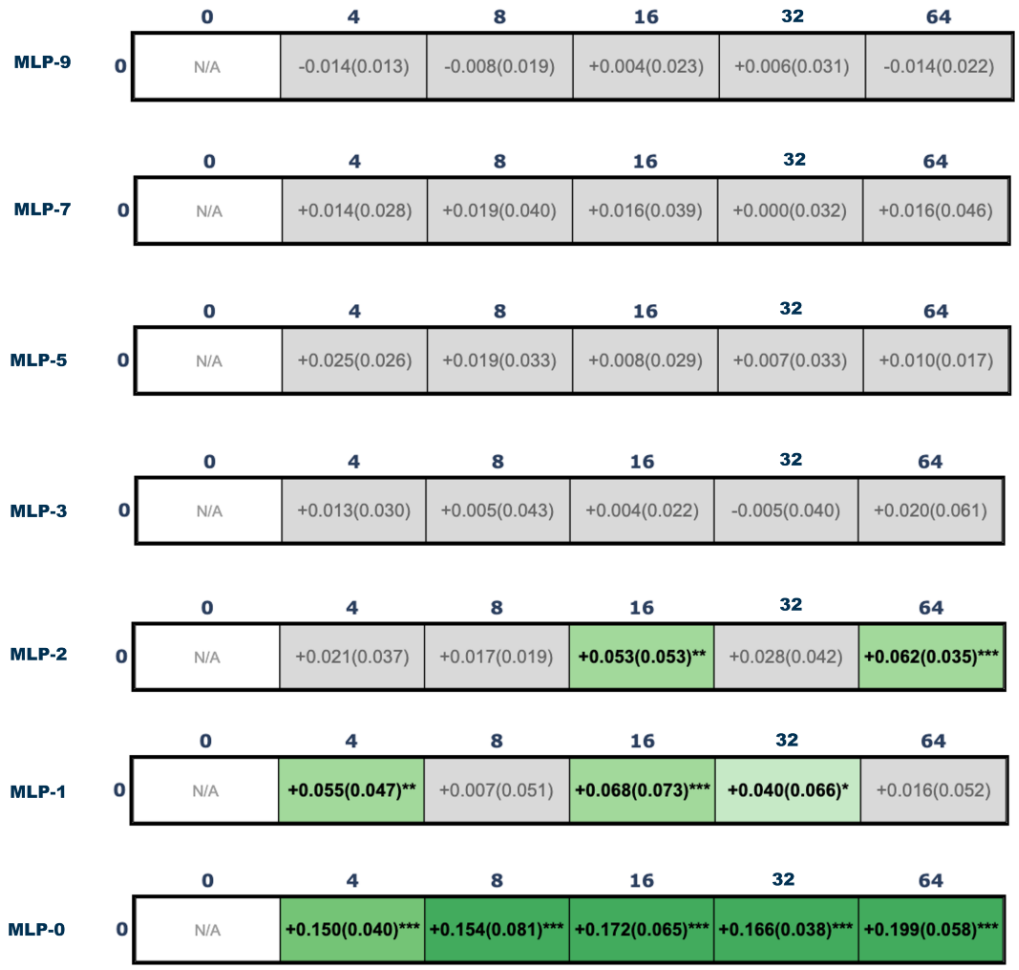

Figure 13: Statistical significance of the per-shot ROC–AUC gain over the zero-shot baseline on the MLP-synthetic grid, computed via DeLong's test aggregated with Stouffer's method across five generation seeds per level. Each cell shows the mean Δ (standard deviation in parentheses); green shading marks cells significant at $p < 0.05$ $(*)$, $p < 0.01$ $(**)$, and $p < 0.001$ $(***)$. Only MLP-0 shows consistent and significant gains; from MLP-1 onward, few-shot improvements are not statistically distinguishable from zero across the family.

Table 26: Per-dataset ROC-AUC on the 8 text-tabular datasets, *full* variant. LLMs zero-shot; TabPFN-2.5/3 Plus at 16 shots; CatBoost at 32 shots (CatBoost was not run at lower shots for text-tabular datasets).

Dataset	Qwen3-1.7B	Qwen3-8B	Qwen3-14B	GPT-4o-mini	TabPFN-2.5 (16)	TabPFN-3 (16)	CatBoost (32)
fake_job_postings	0.536	0.536	0.541	0.623	0.888	0.895	—
california_price	0.527	0.790	0.791	0.796	0.661	0.715	0.698
wine_testing_reviews	0.724	0.788	0.773	0.820	0.685	0.703	0.688
reviews_amazon	0.618	0.690	0.698	0.685	0.674	0.701	—
fake_reviews	0.541	0.643	0.619	0.565	0.505	0.509	0.604
pubmed	0.544	0.526	0.743	0.886	0.674	0.677	0.713
recipes	0.581	0.603	0.517	0.507	0.667	0.666	0.727
online_teaching	0.515	0.525	0.517	0.536	1.000	0.997	0.966
Mean	0.573	0.638	0.650	0.677	0.719	0.733	0.733

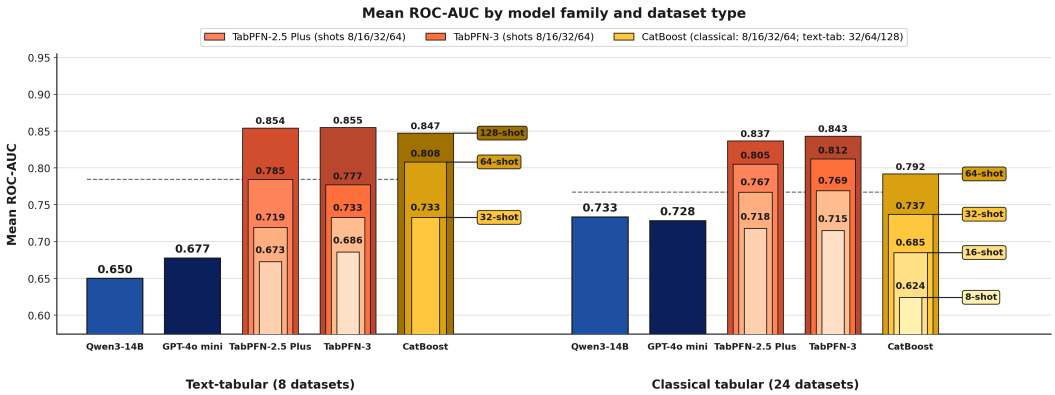

Figure 14: Mean zero-shot ROC-AUC on the 8 text-tabular and 24 classical datasets, LLMs vs. TabPFN-2.5/3 Plus and CatBoost at increasing shot budgets. Dashed reference line marks the label budget at which each baseline family first matches the zero-shot LLMs. Referenced from Section 4.3.

Table 27: Per-dataset ROC-AUC on the 8 text-tabular datasets, *text-only* variant. LLMs zero-shot; TabPFN-2.5/3 Plus at 16 shots; CatBoost at 32 shots.

Dataset	Qwen3-1.7B	Qwen3-8B	Qwen3-14B	GPT-4o-mini	TabPFN-2.5 (16)	TabPFN-3 (16)	CatBoost (32)
fake_job_postings	0.568	0.583	0.520	0.538	0.623	0.667	—
california_price	0.563	0.695	0.736	0.786	0.503	0.502	0.507
wine_testing_reviews	0.562	0.672	0.770	0.805	0.500	0.500	0.553
reviews_amazon	0.607	0.690	0.706	0.699	0.501	0.502	—
fake_reviews	0.535	0.669	0.625	0.609	0.500	0.500	0.610
pubmed	0.529	0.551	0.778	0.904	0.493	0.507	0.609
recipes	0.535	0.574	0.555	0.642	0.691	0.664	0.515
online_teaching	0.508	0.503	0.505	0.510	0.498	0.497	0.501
Mean	0.551	0.617	0.649	0.687	0.539	0.542	0.549

Table 28: Per-dataset ROC-AUC on the 8 text-tabular datasets, *tabular-only* variant. LLMs zero-shot; TabPFN-2.5/3 Plus at 16 shots; CatBoost at 32 shots.

Dataset	Qwen3-1.7B	Qwen3-8B	Qwen3-14B	GPT-4o-mini	TabPFN-2.5 (16)	TabPFN-3 (16)	CatBoost (32)
fake_job_postings	0.538	0.577	0.568	0.669	0.919	0.897	0.841
california_price	0.646	0.799	0.750	0.755	0.646	0.715	0.729
wine_testing_reviews	0.752	0.799	0.821	0.849	0.722	0.709	0.715
reviews_amazon	0.500	0.727	0.726	0.682	0.682	0.677	0.693
fake_reviews	0.507	0.508	0.526	0.522	0.505	0.503	0.504
pubmed	—	—	0.520	0.556	0.667	0.673	0.709
recipes	0.528	0.523	0.562	0.549	0.640	0.640	0.675
online_teaching	0.526	0.545	0.555	0.537	0.508	0.506	0.490
Mean	0.571	0.640	0.629	0.640	0.661	0.665	0.670

Table 29: Mean serialization rank on the combined pool of 32 datasets (24 classical + 8 text-tabular *full*), zero-shot. Lower mean is better; "First (%)" is the share of datasets on which the format wins.

	GPT-4o-mini		Qwen3-14B	
Serialization	Mean	First (%)	Mean	First (%)
markdown	**2.19**	25.0	2.91	15.6
html	2.62	21.9	**2.38**	21.9
feat_val	2.75	**34.4**	2.56	**31.2**
markdown_masked	3.81	6.2	3.31	18.8
feat_val_masked	3.62	12.5	3.81	12.5

Table 30: Mean serialization rank, 24 classical benchmarks only.

	GPT-4o-mini		Qwen3-14B	
Format	Mean	First (%)	Mean	First (%)
feat_val	2.29	45.8	2.25	37.5
feat_val_masked	3.79	8.3	4.04	8.3
html	2.62	20.8	2.25	25.0
markdown	2.08	20.8	2.96	12.5
markdown_masked	4.21	4.2	3.46	16.7

Table 31: Mean serialization rank, 8 text-tabular *text-only* variant.

	GPT-4o-mini		Qwen3-14B	
Format	Mean	First (%)	Mean	First (%)
feat_val	2.75	12.5	2.88	37.5
feat_val_masked	3.88	12.5	3.62	0.0
html	**2.38**	**37.5**	**2.38**	**37.5**
markdown	3.50	12.5	2.25	25.0
markdown_masked	2.50	25.0	3.88	0.0

Table 32: Mean serialization rank, 8 text-tabular *tabular-only* variant.

	GPT-4o-mini		Qwen3-14B	
Format	Mean	First (%)	Mean	First (%)
feat_val	3.00	**37.5**	2.50	**37.5**
feat_val_masked	3.38	12.5	3.38	12.5
html	3.25	12.5	**2.25**	**37.5**
markdown	**2.38**	12.5	3.62	0.0
markdown_masked	3.00	25.0	3.25	12.5