EVIDENCE IS NOT EXCHANGEABLE: QUERY-CONDITIONED EVIDENCE ROUTING FOR TABULAR FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tabular foundation models predict from labeled examples placed in their context. In long-tailed classification, an additional labeled example has no fixed value: the same evidence can repair one query and interfere with another. Evidence selection is therefore a query-conditioned decision, yet existing selection rules often assign candidates a shared value or retrieve them by representation similarity. We formalize evidence value as a query-candidate utility $U(q, e)$, whose query-specific rankings create regret for global selection. QCUR estimates this utility for a frozen backbone from the query representation and the candidate-induced posterior response, then routes one additional example to each query. Across three tabular foundation-model backbones, nine datasets, and three random seeds, QCUR yields a 4.5% relative improvement in mean Accuracy over backbone-only inference, together with a 4.3% relative improvement in Macro-F1 and a 1.1% relative improvement in Macro ROC-AUC. QCUR also exceeds the strongest protocol-matched context-selection baseline by a 3.1% relative improvement in mean Accuracy under the same one-evidence budget. Query-specific ranking inversions, measurable global-selection regret, and repair-dominant prediction changes show that evidence utility varies across queries. Query-conditioned evidence valuation provides a portable principle for improving frozen tabular foundation models.

1 INTRODUCTION

Tabular foundation models (TFMs) turn labeled context into a predictor. From the prior-data fitted network view of approximate Bayesian inference (Müller et al., 2022), TabPFN, TabICL, and TabDPT have extended this interface across tabular regimes (Hollmann et al., 2023; 2025; Qu et al., 2025; Ma et al., 2025). Each inserted context example acts directly on the posterior. This is especially consequential in long-tailed classification, where the least represented classes offer the smallest margin for harmful intervention (Cui et al., 2019; Kang et al., 2020; Menon et al., 2021).

Existing context-selection methods already recognize that examples should be chosen deliberately. They optimize latent coverage, local relevance, distributional fit, labeling value, or model-aware set quality (Thomas et al., 2024; Ipas et al., 2026; Chen et al., 2026; Zeng et al., 2026; Treerath & Pittorino, 2026). These objectives answer which evidence is representative, related, or useful on average. Even a query-aware retriever usually treats relevance as the selection target. The frozen predictor exposes a more primitive quantity: the signed change caused by placing one candidate beside one receiving query. This leads to the question at the center of this paper: *is evidence value a property of the evidence item, or of the evidence–query pair?*

We begin with the frozen backbones themselves. We append the same candidate to a fixed long-tailed context and measure the posterior change over held-out queries. The same item can increase the correct-class probability for one query and decrease it for another, and the preferred candidate reverses across queries. Figure 1 presents the motivation: evidence ranking must depend on the receiving query. Across the full study, 44.9% of candidate-pair rankings invert relative to the post-hoc mean-utility ordering; the post-hoc best fixed candidate disagrees with the query-wise oracle for 93.2% of queries. The training-side V choice leaves 5.22 probability points of post-hoc utility regret.

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.

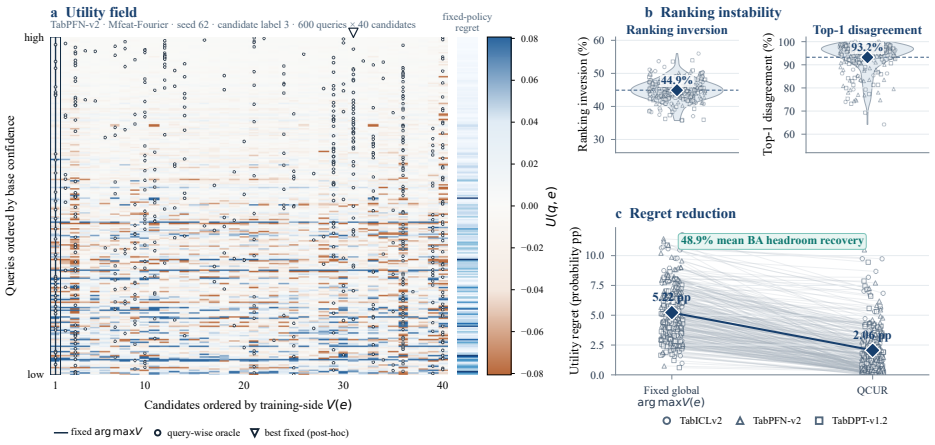

Figure 1: Motivation: query-dependent utility, ranking reversals, and within-unit utility regret; the displayed recovery is measured in BA.

Throughout, the intervention class c is fixed before held-out evaluation, independently of q and y_q; QCUR routes within its targeted pool $\mathcal{E}_c$.

These observations identify the underlying problem: evidence is not exchangeable across queries. We formalize each candidate insertion as an intervention. Let $U(q, e)$ denote its change in correct-class probability for query q, and let $V(e)$ denote any value that omits the receiving query. A query-independent selector cannot recover the query-wise utility optimum when no single candidate is optimal for almost every query. Proposition 1 characterizes this limitation; changes among lower-ranked candidates alone need not incur selection regret.

QCUR estimates $\widehat{U}(q, e)$ from the frozen model's query representation and the local posterior response induced by the candidate, then routes the candidate with largest predicted benefit.

Across three model families, nine public datasets, and three seeds, QCUR raises mean Accuracy, Macro-F1, and Macro ROC-AUC over backbone-only inference by 4.5%, 4.3%, and 1.1% relative, and Accuracy over the strongest external protocol adaptation by 3.1%. Our contributions are: (1) we identify evidence non-exchangeability in targeted context intervention and characterize the regret of global valuation; (2) we introduce a query-conditioned utility router and bound its regret by utility-estimation error; and (3) we validate the principle across three backbone families and 27 backbone–dataset cells, with 48.9% mean per-unit recovery of fixed selection's BA headroom and repair-dominant prediction changes.

2 Related Work

Tabular foundation models. Prior-data fitted networks cast in-context prediction as approximate Bayesian inference over task distributions (Müller et al., 2022). TabPFN and TabPFN-v2 developed this interface for accurate small-data prediction (Hollmann et al., 2023; 2025); TabICL scales synthetic pretraining through row-context architectures (Qu et al., 2025; 2026); TabDPT scales real-data pretraining (Ma et al., 2025; Hosseinzadeh et al., 2026); and MixturePFN couples local prompting with task adaptation (Xu et al., 2024). All expose the frozen context-conditioned prediction interface required by QCUR.

Example selection in in-context learning. Demonstration labels, order, diversity, and calibration can all change an in-context predictor (Brown et al., 2020; Min et al., 2022; Lu et al., 2022; Zhao et al., 2021; Levy et al., 2023). This motivated query-aware retrieval and active selection (Liu et al., 2022; Rubin et al., 2022; Margatina et al., 2023). QCUR changes the target of this decision from semantic relevance to the signed posterior effect of the example.

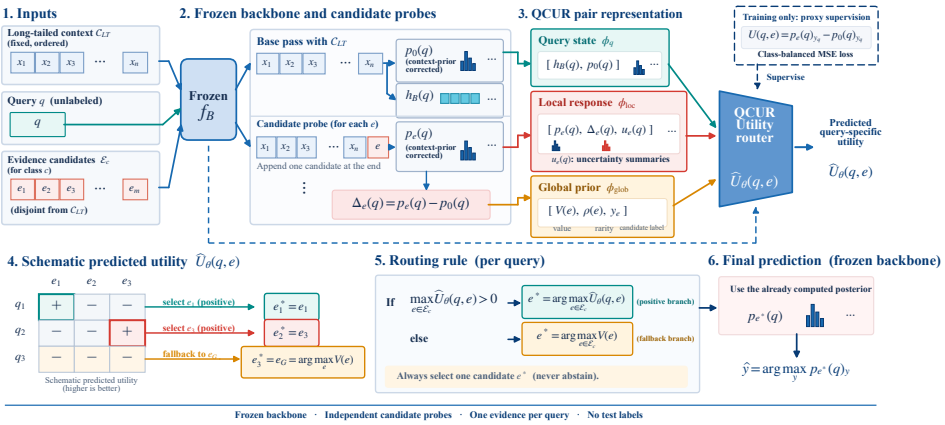

Figure 2: QCUR estimates query–candidate utility from the frozen backbone's query representation, candidate-local posterior response, and a training-side global prior.

Selecting and optimizing tabular context. LoCalPFN uses query-local retrieval and task fine-tuning (Thomas et al., 2024); LUCoS learns a coverage-oriented latent geometry (Ipas et al., 2026); VIP-COp performs model-aware black-box optimization (Chen et al., 2026); Tab-AICL selects along an active trajectory (Treerath & Pittorino, 2026); and DCS combines density-ratio matching with diversity (Zeng et al., 2026). QCUR isolates a different decision: which intervention benefits the receiving query. This parallels the distinction between prediction-level influence and average data value (Koh & Liang, 2017; Ghorbani & Zou, 2019), but acts directly on a frozen in-context predictor through signed utility $U(q, e)$.

3 Method

For a protocol-specified target class, QCUR evaluates a frozen TFM under the long-tailed context and each append-at-end candidate. A shared router combines the backbone-native query representation, candidate-local posterior response, and training-side global prior to estimate query–candidate utility. It routes the best positive-utility candidate and otherwise uses the global fallback. The protocol is fixed throughout all experiments. Figure 2 shows how frozen responses become query-conditioned decisions.

3.1 Query-conditioned evidence valuation

Let B denote a frozen tabular foundation-model backbone, and let $C_{\mathrm{LT}} = ((x_j, y_j))_{j=1}^{n}$ be its ordered long-tailed context over K classes. For a query q, the backbone-only posterior is

$$p_0(q) = f_B(q \mid C_{\mathrm{LT}}) \in \Delta^{K-1}. \tag{1}$$

For each tail class c, a frozen candidate pool $\mathcal{E}_c$ contains labeled evidence items $e = (x_e, y_e)$ with $y_e = c$ that are disjoint from C_{LT}. Appending one candidate to the end of the context gives

$$p_e(q) = f_B(q \mid C_{\mathrm{LT}} \oplus e), \qquad \Delta_e(q) = p_e(q) - p_0(q), \tag{2}$$

where $\oplus$ denotes append-at-end insertion. We call $\Delta_e(q)$ the candidate-local functional response: it records how the frozen predictor changes when e is introduced, for the query currently being evaluated.

The index c denotes the protocol-specified tail-class intervention unit. It is fixed before evaluation, independent of q and y_q, and shared by all selectors. Each unit evaluates the same class-balanced held-out population.

All posterior features use the same class-prior correction. If $\bar{p}_B(q \mid C)$ is the backbone's native posterior and $\pi_k(C) = n_k(C)/|C|$ is the empirical context prior, we use

$$p_k(q \mid C) = \frac{\bar{p}_{B,k}(q \mid C)/\pi_k(C)}{\sum_{j=1}^{K} \bar{p}_{B,j}(q \mid C)/\pi_j(C)}. \tag{3}$$

The count vector is updated after appending e, so both p_0 and p_e are corrected against the context that produced them.

The value of evidence is defined by its effect on the correct-class probability. For a labeled training-side proxy query (q, y_q),

$$U(q, e) = p_e(q)_{y_q} - p_0(q)_{y_q}. \tag{4}$$

A positive value moves probability mass toward the proxy query's true class; a negative value moves it away. This probabilistic target is evaluated through the final hard label and Accuracy at deployment. At test time y_q is unavailable. QCUR therefore learns an estimator $\widehat{U}_\theta(q, e)$ from label-free query–candidate features and routes evidence according to this estimate.

Because every class is represented in the frozen context, the correction in Eq. (3) is well defined and the realized utility is bounded by $-1 \le U(q, e) \le 1$; the full bounds for posterior responses and derived features are recorded in the supplement. The router is an unconstrained regression estimate; the stated bounds apply to the target.

3.2 A QUERY–CANDIDATE REPRESENTATION

QCUR represents each pair by $h_B(q)$, the full posterior triple (p_0, p_e, Δ_e), their three candidate-class coordinates, the values and changes of entropy H and top-two margin m, $(V(e), \rho(e))$, and a one-hot candidate label. In this order the vector has dimension $d_B + 4K + 11$; its coordinate-wise expression and bounds are in the supplement.

The two candidate-level scalars provide a weak global reference. We split the proxy population into five fixed class-stratified folds $\mathcal{S}_1, \ldots, \mathcal{S}_5$ and compute a fold-aggregated marginal prior. Here $\mathrm{BA}_{\mathcal{S}_r}$ denotes the macro-average recall on fold $\mathcal{S}_r$, so each class contributes equally to the candidate-level prior.

$$V(e) = \mathrm{median}_{r \in \{1,\ldots,5\}} \left[\mathrm{BA}_{\mathcal{S}_r}(C_{\mathrm{LT}} \oplus e) - \mathrm{BA}_{\mathcal{S}_r}(C_{\mathrm{LT}}) \right]. \tag{5}$$

Within each $\mathcal{E}_c$, $\rho(e) \in [0, 1]$ is the normalized descending rank of $V(e)$, with one assigned to the highest-ranked candidate. Both quantities are computed with the current frozen backbone and training-side proxy labels. They form a candidate-quality prior refined by the query-dependent terms. The proxy population used for V and router fitting is disjoint from held-out evaluation.

3.3 UTILITY REGRESSION

For one dataset, backbone, and data seed, we form the training pairs

$$\mathcal{P} = \left\{ (q, e) : q \in \mathcal{S}_{\mathrm{proxy}},\ e \in \textstyle\bigcup_c \mathcal{E}_c \right\}. \tag{6}$$

One router is trained for the state and shared across its tail classes. The router is a two-hidden-layer regressor,

$$\widehat{U}_\theta(q, e) = W_3\ \mathrm{GELU}(W_2\ \mathrm{LN}(\mathrm{GELU}(W_1 \phi_B(q, e) + b_1)) + b_2) + b_3, \tag{7}$$

with hidden widths 256 and 128. The first hidden activation is normalized by LayerNorm. We fit θ to the target in Eq. (4) by class-balanced weighted mean squared error,

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{P}|} \sum_{(q,e) \in \mathcal{P}} w_{y_q} \big(\widehat{U}_\theta(q, e) - U(q, e) \big)^2, \qquad w_k \propto \frac{1}{N_k}, \tag{8}$$

where N_k counts proxy training pairs with query label k, and the weights are normalized to have mean one over $\mathcal{P}$. Training uses AdamW for 20 epochs with learning rate 10^{-3}, weight decay 10^{-4}, batch size 2048, and router seed 3401. The backbone remains in inference mode and none of its parameters are updated.

3.4 ROUTING AND PREDICTION

For a test query q in the protocol-specified tail-class intervention unit c, QCUR scores every $e \in \mathcal{E}_c$. Let

$$\widetilde{e}(q, c) = \arg\max_{e \in \mathcal{E}_c} \widehat{U}_\theta(q, e), \qquad e_{\mathrm{G}}(c) = \arg\max_{e \in \mathcal{E}_c} V(e). \tag{9}$$

The routed evidence is

$$e^*(q,c) = \begin{cases} \widetilde{e}(q,c), & \max_{e\in\mathcal{E}_c} \widehat{U}_\theta(q,e) > 0, \\ e_{\mathrm{G}}(c), & \text{otherwise.} \end{cases} \tag{10}$$

Ties are resolved by the frozen ascending candidate identifier. The final prediction is read directly from the selected candidate-conditioned posterior,

$$\widehat{y}(q,c) = \arg\max_k p_{e^*(q,c)}(q)_k. \tag{11}$$

QCUR adds one labeled evidence item. Gold test labels are used for scoring and analysis. Candidate-conditioned probes score the frozen pool and are included in the efficiency accounting.

Algorithm 1 isolates this central operation for a single query. The proxy construction and router fitting that produce θ, V, and ρ are performed once per experimental state.

Algorithm 1 Atomic query-conditioned routing with a frozen utility threshold and global fallback.

Require: Query q; unit c; ordered context C_{LT}; pool $\mathcal{E}_c$; frozen f_B, h_B; trained θ; V, ρ
Ensure: Routed evidence e^* and prediction $\widehat{y}$

$p_0 \leftarrow f_B(q \mid C_{\mathrm{LT}})$; $z \leftarrow h_B(q \mid C_{\mathrm{LT}})$
for $e \in \mathcal{E}_c$ **do**
  $p_e \leftarrow f_B(q \mid C_{\mathrm{LT}} \oplus e)$
  $\phi_e \leftarrow \phi_B(q, e; z, p_0, p_e, V(e), \rho(e))$
  $s_e \leftarrow \widehat{U}_\theta(\phi_e)$
end for
$\widetilde{e} \leftarrow \textsc{FirstArgMax}((s_e)_{e\in\mathcal{E}_c})$ ▷ ascending frozen ID
if $s_{\widetilde{e}} > 0$ **then**
  $e^* \leftarrow \widetilde{e}$
else
  $e^* \leftarrow \textsc{FirstArgMax}((V(e))_{e\in\mathcal{E}_c})$
end if
return e^*, $\widehat{y} = \arg\max_k p_{e^*}[k]$

Backbone portability. The formulation requires only posteriors under C_{LT} and $C_{\mathrm{LT}} \oplus e$ plus the backbone's final query state before its readout. Changing the backbone changes these outputs and the input dimension; the utility, router, training schedule, threshold, prior, and evidence budget remain fixed.

4 Theoretical Analysis

The query dependence in Eq. (4) induces a precise gap between evidence that is best on average and evidence selected after observing the query. Fix an intervention unit c, let $q \sim \mathcal{D}_c$, and write $M(q) = \max_{e\in\mathcal{E}_c} U(q,e)$. The strongest query-independent selector chooses

$$e_{\mathrm{avg}} \in \arg\max_{e\in\mathcal{E}_c} \mathbb{E}_q[U(q,e)]. \tag{12}$$

Proposition 1 (Regret of global evidence valuation)**.** *For a finite nonempty candidate pool, define*

$$R_{\mathrm{global}} = \mathbb{E}_q[M(q)] - \max_{e\in\mathcal{E}_c} \mathbb{E}_q[U(q,e)]. \tag{13}$$

Then $0 \le R_{\mathrm{global}} \le 1$*. Furthermore,* $R_{\mathrm{global}} = 0$ *if and only if some fixed candidate is pointwise optimal for* $\mathcal{D}_c$*-almost every query. Thus* $R_{\mathrm{global}} > 0$ *whenever no single candidate is almost surely optimal.*

Proof sketch. Pointwise non-negativity gives the lower bound, cancellation of $p_0(q)_{y_q}$ gives the upper bound in $[0,1]$, and zero expectation of a nonnegative gap is equivalent to pointwise optimality almost surely. Full details are in the supplement.

The regret in Proposition 1 is measured in probabilistic utility U against the post-hoc best fixed candidate. The deployable fallback in Eq. (10) instead maximizes the training-side prior V, a fold-aggregated Balanced-Accuracy change. Section 5.5 separates these fixed choices and reports their hard-prediction outcomes.

Utility estimation gives the complementary routing guarantee. If $\epsilon(q) = \max_e |\widehat{U}_\theta(q, e) - U(q, e)|$, the estimated-utility maximizer has pointwise regret at most $\min\{1, 2\epsilon(q)\}$. On QCUR's fallback event the regret is at most one, yielding the corresponding expected bound by conditioning on that event. The statement and proof are in the supplement.

5 Experiments

5.1 Experimental Setup

We evaluate three frozen tabular foundation-model backbones: Original TabICLv2, a scalable column–row–context model (Qu et al., 2026); TabPFN-v2, a prior-data fitted transformer trained on synthetic tasks (Hollmann et al., 2025); and TabDPT-v1.2, pretrained on real tabular data (Ma et al., 2025; Hosseinzadeh et al., 2026). They provide distinct architectures and pretraining sources under a common in-context interface. Each backbone–dataset pair uses the fixed seeds $\{42, 52, 62\}$.

The benchmark is a public collection of nine OpenML datasets (Vanschoren et al., 2013). Mfeat-Fourier, Optdigits, Pendigits, Satimage, and Segment are image- or shape-derived, often high-dimensional recognition tasks; Vehicle and Wall-Robot are sensor or physical-state datasets with overlapping local signals; Connect-4 is structured discrete data and CoverType is a larger environmental multi-class problem. These groups vary the source of evidence usefulness across representation, measurement, and scale.

We compare QCUR with the frozen Backbone and four external approaches, used as one-evidence protocol adaptations: LUCoS (latent unsupervised context selection) (Ipas et al., 2026), VIP-COp (model-aware context optimization) (Chen et al., 2026), LoCalPFN (query-local retrieval) (Thomas et al., 2024), and DCS (density-ratio selection with diversity constraints) (Zeng et al., 2026). The adaptations retain LUCoS's pretrained selector, VIP-COp's 100-round coalition search, LoCalPFN's query-local retrieval without its task fine-tuning, and DCS's density–coverage ranking, respectively. All select from the same frozen candidate pool for the same target backbone, context, intervention unit, and held-out queries. Preprocessing, proxy folds, insertion order, and prior correction are fixed; QCUR fits its router on labeled training-side proxy queries, independently of held-out labels. Its candidate-conditioned backbone probes and router fitting are included in the runtime accounting. The pools contain 40–200 candidates per state, with 412–6,859 proxy and 91–2,000 held-out queries.

We report Accuracy, Macro-F1, Macro ROC-AUC, negative log-likelihood (NLL), and worst-class recall (WCR), the minimum recall over classes. Results are averaged over the three seeds within each backbone–dataset cell, with the 27 cells weighted equally.

5.2 Main Results

Table 1 reports five predictive metrics in aggregate and for each frozen backbone family.

Relative to the frozen Backbone, QCUR improves Accuracy by 4.51%, Macro-F1 by 4.33%, and Macro ROC-AUC by 1.12%, while reducing NLL by 9.67% and increasing WCR by 8.09%. It is the strongest aggregate method on every reported metric. Paired bootstrap intervals over the 27 backbone–dataset cells put its Accuracy advantage over Backbone at 2.66–4.04 percentage points. The supplement reports all 81 seed-level Accuracy results and paired comparisons for every method. WCR rises on TabICLv2 and TabPFN-v2; TabDPT-v1.2 retains the highest QCUR Accuracy, F1, AUC, and lowest NLL among its methods, with WCR 0.4856.

VIP-COp is the strongest external adaptation in aggregate Accuracy and on each individual backbone. QCUR improves over it by 3.14% overall, with a paired 95% interval of 1.84–2.89 Accuracy points and relative gains of 3.57%, 3.86%, and 1.98% on TabICLv2, TabPFN-v2, and TabDPT-v1.2, respectively. After aggregating seeds within each backbone–dataset cell, QCUR exceeds Backbone on 100.0% of cells and VIP-COp on 96.3% of cells. Against the best external baseline selected sep-

arately for each cell, QCUR improves Accuracy on 92.6% of cells. Across these distinct pretraining sources and architectures, the routing rule remains unchanged; its router is fitted separately for each backbone–dataset–seed state.

Table 1: Predictive performance across three frozen tabular foundation-model backbones and nine datasets.

Backbone	Method	Accuracy ↑	Macro-F1 ↑	Macro-AUC ↑	NLL ↓	WCR ↑
Overall						
	Backbone	0.7386	0.7323	0.9128	0.6482	0.4885
	DCS	0.7423	0.7363	0.9138	0.6397	0.4981
	LUCoS	0.7448	0.7389	0.9147	0.6329	0.5010
	LoCalPFN	0.7458	0.7375	0.9169	0.6286	0.4786
	VIP-COp	0.7484	0.7427	0.9171	0.6172	0.5084
	QCUR (Ours)	**0.7719**	**0.7641**	**0.9230**	**0.5855**	**0.5280**
Original TabICLv2						
	Backbone	0.7534	0.7479	0.9197	0.6174	0.5048
	DCS	0.7564	0.7513	0.9211	0.6133	0.5180
	LUCoS	0.7587	0.7540	0.9222	0.6019	0.5240
	LoCalPFN	0.7583	0.7525	0.9237	0.6063	0.4922
	VIP-COp	0.7611	0.7564	0.9249	0.5897	0.5228
	QCUR (Ours)	**0.7883**	**0.7830**	**0.9340**	**0.5577**	**0.5443**
TabPFN-v2						
	Backbone	0.7320	0.7234	0.9117	0.6714	0.4695
	DCS	0.7397	0.7317	0.9128	0.6545	0.4882
	LUCoS	0.7411	0.7329	0.9139	0.6497	0.4882
	LoCalPFN	0.7422	0.7333	0.9155	0.6399	0.4831
	VIP-COp	0.7454	0.7378	0.9168	0.6310	0.5025
	QCUR (Ours)	**0.7741**	**0.7661**	**0.9209**	**0.5852**	**0.5539**
TabDPT-v1.2						
	Backbone	0.7305	0.7257	0.9070	0.6558	0.4910
	DCS	0.7307	0.7260	0.9076	0.6512	0.4883
	LUCoS	0.7347	0.7297	0.9081	0.6470	0.4907
	LoCalPFN	0.7369	0.7267	0.9114	0.6397	0.4606
	VIP-COp	0.7386	0.7341	0.9094	0.6308	**0.5000**
	QCUR (Ours)	**0.7533**	**0.7431**	**0.9141**	**0.6137**	0.4856

5.3 Ablation Study

We isolate the added information from the backbone-native query representation and candidate-local functional response on a separately fitted, 12-state TabPFN-v2 diagnostic suite (CoverType, Optdigits, Pendigits, and Segment; three seeds each). Global-only, Query-only, Functional-only, and Full QCUR form a 2×2 feature design. Functional-only retains query-dependent posteriors; removing the native query representation does not remove all query information. We also compare Full QCUR with Globalized QCUR, which uses the same learned features and utility model but makes one shared decision, and with a response-shuffled control that applies a deterministic within-candidate permutation to the functional-response block on proxy and held-out queries, leaving query-side features and candidate priors attached to their original queries. Table 2 gives the component and correspondence evidence for this mechanism.

Table 2: Factorial ablation and correspondence controls on the TabPFN-v2 diagnostic suite.

Variant	Accuracy ↑	Macro-F1 ↑	WCR ↑	NLL ↓
Global-only	0.7894	0.7840	0.5573	0.5745
Query-only	0.7870	0.7815	0.5466	0.5711
Functional-only	0.8032	0.7989	0.5856	0.5394
Full QCUR	**0.8187**	**0.8147**	**0.6251**	**0.5068**
Globalized QCUR	0.7859	0.7800	0.5350	0.5783
Response shuffled	0.7882	0.7832	0.5562	0.5702

Functional-only reaches 0.8032 Accuracy, compared with 0.7894 for Global-only and 0.7870 for Query-only, so the functional response supplies the larger main effect. Full QCUR reaches 0.8187 and improves over Functional-only in every diagnostic state; the $Q \times F$ interaction contributes a further 1.78 Accuracy points beyond the additive effects. Globalizing the learned decision reduces Accuracy to 0.7859, and shuffling the response correspondence reduces it to 0.7882. The gain reflects the native query state together with candidate-local responses. Shuffling the response block also disrupts its algebraic relationship to the unchanged query-side posterior.

5.4 Hyperparameter Sensitivity

We evaluate the frozen QCUR configuration on the same 12-state TabPFN-v2 diagnostic suite while varying router capacity and learning rate, the fallback threshold, and the available fraction of the frozen candidate pool. Table 3 is the sensitivity evidence for this operating regime; it reports the range over each sweep, with router width and learning rate varied jointly. The reference configuration is retained throughout the paper (widths 256/128, learning rate 10^{-3}, threshold 0, and the full pool), regardless of post-hoc variation.

Table 3: QCUR sensitivity to router capacity, optimization, fallback threshold, and candidate availability.

Sweep	Accuracy ↑	Macro-F1 ↑	WCR ↑
Capacity and learning rate	0.8163–0.8219	0.8125–0.8186	0.6143–0.6259
Fallback threshold	0.8165–0.8201	0.8120–0.8160	0.6151–0.6279
Candidate-pool fraction	$0.8064 \rightarrow 0.8187$	$0.8017 \rightarrow 0.8147$	$0.5940 \rightarrow 0.6251$

Accuracy spans 0.8163–0.8219 across capacity and learning rate and 0.8165–0.8201 across fallback thresholds, with 0.8187 at the frozen threshold of 0. Positive thresholds invoke the global fallback more often. As candidate availability grows, Accuracy rises smoothly from 0.8064 to 0.8187, showing stable router behavior and an interpretable dependence on evidence coverage.

5.5 Mechanistic Analysis: Evidence Is Not Exchangeable

We measure realized utility over 81 backbone–dataset–seed states, 261 tail-class units, and 10,401 candidate-level records. The post-hoc oracle selects $\arg\max_e U(q, e)$ using test labels, then evaluates its hard predictions; it is not a hard-label-optimal BA oracle. Utility regret uses this oracle, while BA recovery averages each unit's fraction of observed BA headroom recovered over the training-side V choice. Figure 3 shows the corresponding posterior responses.

Panel A shows response fields ordered by utility; panel B shows response reversals across queries and candidates. Panel C compares predicted and realized utility, with routed actions concentrated in the positive region. Labels enter only the retrospective diagnostics.

Across 261 units, 44.91% of candidate-pair orders invert against their post-hoc mean-utility ordering. With a fixed 0.01 true-class-probability margin, 19.66% of all query–candidate-pair comparisons still invert (by backbone in the supplement). The post-hoc best fixed candidate disagrees with the query-wise oracle on 93.24% of queries. The training-side V choice leaves 5.22 probability points of utility regret versus QCUR's 2.06; even the post-hoc best fixed candidate leaves 4.68 points. QCUR's mean per-unit BA headroom recovery is 48.90%; it repairs 3.49% and harms 1.39% of predictions. Using the same cached posteriors, union-pool QCUR reaches 0.7940 Accuracy across all 81 states. Within fixed pools, QCUR reaches 0.8111, above entropy reduction (0.7883), base-class probability gain (0.7806), and margin gain (0.7746). The same-probe mean posterior ensemble reaches 0.7845 without a router; it combines posteriors rather than selecting one evidence. These values use 261-unit aggregation. An optional no-append action on nonpositive predicted scores changes the fixed-pool value from 0.8111 to 0.8123, leaving the base context intact on 13.6% of decisions using the original scores and cached posteriors. The main-table 0.7719 instead equally weights 27 backbone–dataset cells after averaging three seeds, hence its different denominator.

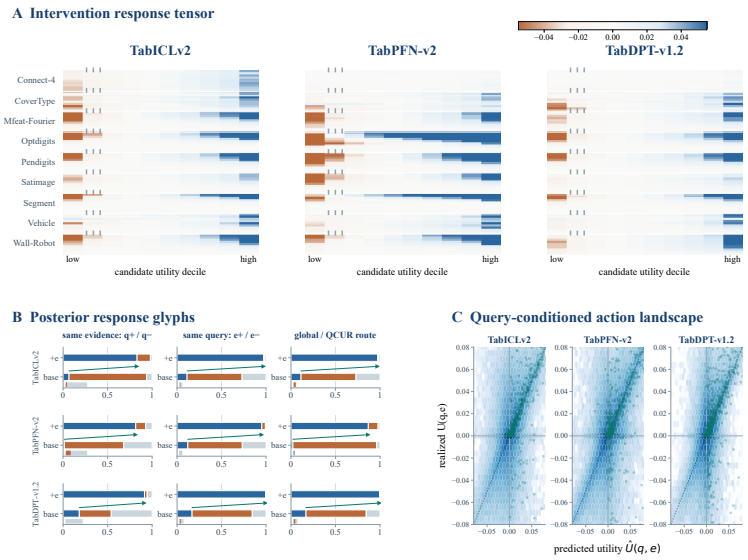

Figure 3: Counterfactual posterior responses and query-conditioned routing.

Efficiency. Across 81 states on an RTX 4060 Laptop GPU, QCUR requires 100.9/66.1/63.4 seconds per state for TabICLv2/TabPFN-v2/TabDPT-v1.2, versus 4,266/12,440/2,618 seconds for VIP-COp's native 100-round search; candidate-conditioned probes are included in these totals, and full accounting is in the supplement.

Table 4: QCUR runtime and observed native VIP-COp search cost.

Backbone	QCUR states	Calls	Pairs (M)	Router s/state	QCUR s/state	VIP-COp s/state
TabICLv2	27	6,934	16.53	59.7	100.9	4,266
TabPFN-v2	27	6,934	16.53	31.4	66.1	12,440
TabDPT-v1.2	27	6,934	16.53	41.6	63.4	2,618

6 Discussion

Ranking reversals persist beyond near ties, and even the post-hoc best fixed candidate leaves measurable utility regret. Full QCUR improves on the functional-only and query-only variants: the native query state helps interpret the candidate-induced response rather than serving only as a relevance score. The measured response includes the class-prior correction defined in the intervention.

QCUR learns a smooth correct-class-probability utility, while Accuracy, Macro-F1, and WCR evaluate hard predictions. The repair–harm analysis and TabDPT-v1.2 WCR show why gains in one need not imply gains in every outcome. The no-append diagnostic further shows value in deciding whether to intervene, alongside which candidate to select. Here the target class is specified and one row is appended; the union-pool result extends candidate choice, while multiple-row interactions remain a separate problem. Labeled proxy training and per-candidate probing are the costs of this formulation, with the latter included in Table 4.

7 Conclusion

Evidence value in frozen TFMs depends on the receiving query. QCUR estimates this conditional utility from the query state and candidate-induced posterior response, improving prediction across three backbones and nine datasets. The observed fixed-policy regret and its recovery connect a measurable selection problem to a simple routing rule for targeted context intervention.

AI USE STATEMENT

Generative AI tools assisted with language editing and restructuring portions of the Abstract, Introduction, Methods, Experiments, and Discussion. The authors manually revised and reviewed the AI-assisted material and take full responsibility for the entire manuscript, including its claims, analyses, citations, and figures.

REFERENCES

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://arxiv.org/abs/2005.14165.

Yilong Chen, Xueying Ding, and Leman Akoglu. Vip-cop: Context optimization for tabular foundation models. arXiv:2605.12904, 2026. URL https://arxiv.org/abs/2605.12904.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9260–9269, 2019. doi: 10.1109/CVPR.2019.00949.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2242–2251, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2207.01848.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025. doi: 10.1038/s41586-024-08328-6.

Rasa Hosseinzadeh, Alex Labach, Zexin Xue, Shuyi Han, Valentin Thomas, and Anthony L. Caterini. Tabdpt-turbo: Efficient in-context learning for tabular prediction, 2026. URL https://arxiv.org/abs/2608.01400.

Oroel Ipas, Guillermo Gomez-Trenado, Rocío Romero-Zaliz, and Isaac Triguero. Lucos: Latent unsupervised context selection for tabular foundation models. arXiv:2605.27254, 2026. URL https://arxiv.org/abs/2605.27254.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1910.09217.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1885–1894, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 1401–1422, 2023. doi: 10.18653/v1/2023.acl-long.78.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out*, pp. 100–114, 2022. doi: 10.18653/v1/2022.deelio-1.10.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pp. 8086–8098, 2022. doi: 10.18653/v1/2022.acl-long.556.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Hamidreza Kamkari, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maksims Volkovs. Tabdpt: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=pIZxEOZCId.

Katerina Margatina, Timo Schick, Nikolaos Aletras, and Jane Dwivedi-Yu. Active learning principles for in-context learning with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5011–5034, 2023. doi: 10.18653/v1/2023.findings-emnlp.334.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2007.07314.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11048–11064, 2022. doi: 10.18653/v1/2022.emnlp-main.759.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2112.10510.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2502.05564.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026. URL https://arxiv.org/abs/2602.11139. Published at ICML 2026.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2655–2671, 2022. doi: 10.18653/v1/2022.naacl-main.191.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval and fine-tuning for in-context tabular models. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=LN5X5rdlb3.

Wilailuck Treerath and Fabrizio Pittorino. Active in-context learning for tabular foundation models. arXiv:2603.27385, 2026. URL https://arxiv.org/abs/2603.27385.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013. doi: 10.1145/2641190.2641198.

Derek Xu, Olcay Cirit, Reza Asadi, Yizhou Sun, and Wei Wang. Mixture of in-context prompters for tabular PFNs. arXiv:2405.16156, 2024. URL https://arxiv.org/abs/2405.16156.

Jingyuan Zeng, Ming Zeng, Jianghong Guo, Chuanxian Jiang, and Yafen Feng. Dcs: Diversity-constrained density-ratio selection for test-time context optimization of tabular foundation models under covariate shift. GitHub repository, 2026. URL https://github.com/mingyi0818/DCS-Tabular-CovariateShift. Accessed September 23, 2026.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12697–12706, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.