ORDINALICL: MAKING TABULAR FOUNDATION MODELS ORDINAL-NATIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Ordinal targets arise in ratings, disease stages, and educational attainment: their categories are ordered, but the distances between them are unknown. General tabular foundation models usually ignore this structure and predict nominal classes. We ask a controlled question: if the backbone, trainable capacity, data, initialization, optimizer, and compute are held fixed, does giving the model an ordinal target interface help? We introduce **OrdinalICL**, which represents a declared variable-cardinality scale by ordered thresholds and returns a coherent class distribution with exact reversal symmetry. A TabICLv2 ordinal adapter and a compact model trained from scratch first establish that the interface learns and uses order. The original survey identifies TabDPT-Turbo v1.2 as the best-performing surveyed backbone that supports an exactly matched training experiment. We keep this backbone unchanged and attach equally sized prediction-correction modules that differ only in target geometry: the control uses class identities; OrdinalICL uses rank coordinates and ordered boundary decisions. Two independently prespecified evaluations on unseen synthetic mechanisms favor the ordinal module for every seed, class count, mechanism, and coverage stratum. Across a unified 40-family real benchmark, the final model also improves both its matched nominal control and released TabDPT, with family-level intervals excluding zero. The synthetic experiments provide the architecture test, while the real-data analysis measures transfer beyond the synthetic generator. Under matched training conditions, explicitly modeling target order improves TabDPT on ordinal problems.

1 INTRODUCTION

Many important tabular prediction problems have ordered outcomes, such as disease stage, educational attainment, consumption frequency, ratings, and risk levels. Their category order has meaning, but their numerical spacing does not. Relabeling `low, medium, high` as `0, 10, 11` should preserve the scientific problem; swapping `low` and `medium` should not.

Here, the ordered scale belongs to the target categories. A Trustpilot star rating, workplace score, or clinical self-rating places each observation on such a scale. The predictors can be numerical or categorical and may also include representations of other modalities, such as text from a separately validated encoder.

Ordinal prediction is related to, but different from, learning to rank. An ordinal predictor assigns each observation a probability distribution over a fixed scale, such as one to five stars. The meaning of a four-star prediction does not depend on which other observations happen to be present. A ranking model instead orders a set of candidates relative to one another, usually within a query or group; it need not assign calibrated probabilities on an absolute category scale. Ranking methods therefore use pairwise or listwise objectives and are evaluated on the quality of the resulting order, whereas ordinal prediction uses category-level outcomes and proper probabilistic scores. Both exploit order, but they solve different prediction problems.

Current tabular foundation models (TFMs) amortize learning over synthetic or real tables and provide strong classification and regression when only a small labeled context is available (Hollmann et al. 2025; Qu et al. 2025, 2026; Grinsztajn et al. 2026). Applying them to an ordinal target typically requires an external formulation: ignore order and classify, impose equal category spacing and regress, fit `K-1` binary reductions, or discretize a continuous predictive distribution. The first

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.

discards adjacency, the second changes under a monotone recoding, the third uses the TFM as an unchanged component, and the fourth remains a regression-pretrained wrapper. None teaches the shared TFM a variable-cardinality ordinal task across tables.

This paper tests whether a tabular foundation model can learn one ordinal interface that transfers across tables with different category counts. The study proceeds in three stages. We first implement the same ordinal interface as a TabICLv2 adapter and as a small model trained from scratch. We then compare these models with released TFMs on a unified 40-family benchmark. TabDPT-Turbo v1.2 has the lowest mean in the original survey and exposes the trainable internals needed for the final controlled study. We build OrdinalICL–TabDPT and compare it with an equal-capacity nominal adaptation of the same backbone. Our contributions are:

1. We distinguish nominal, ordinal, and metric targets by the relabelings that preserve their meaning. Nominal labels admit arbitrary permutations. Ordinal labels admit only order-preserving renamings. Metric targets also specify distances. These symmetries yield practical tests of order use.
2. We introduce OrdinalICL, which encodes an ordered label by the thresholds it exceeds, jointly predicts monotone cumulative probabilities, gives equal training weight to each table, and guarantees that predictions transform correctly when the scale is reversed.
3. We compare two equally sized prediction-correction modules attached to the same unchanged TabDPT-Turbo v1.2. The control uses nominal class evidence; OrdinalICL uses ranks and ordered-boundary decisions. Data, initialization, optimization, and compute are identical within each paired seed.
4. We report the original model survey that selects TabDPT-Turbo v1.2, then test the matched ordinal-versus-nominal contrast twice on disjoint synthetic mechanisms and on a unified 40-family real benchmark.

We first establish that OrdinalICL learns and uses category order. The matched-backbone experiment then tests whether it improves TabDPT from only 32 labeled examples per new table. The ordinal correction is consistent across unseen synthetic mechanisms and the unified 40-family real benchmark.

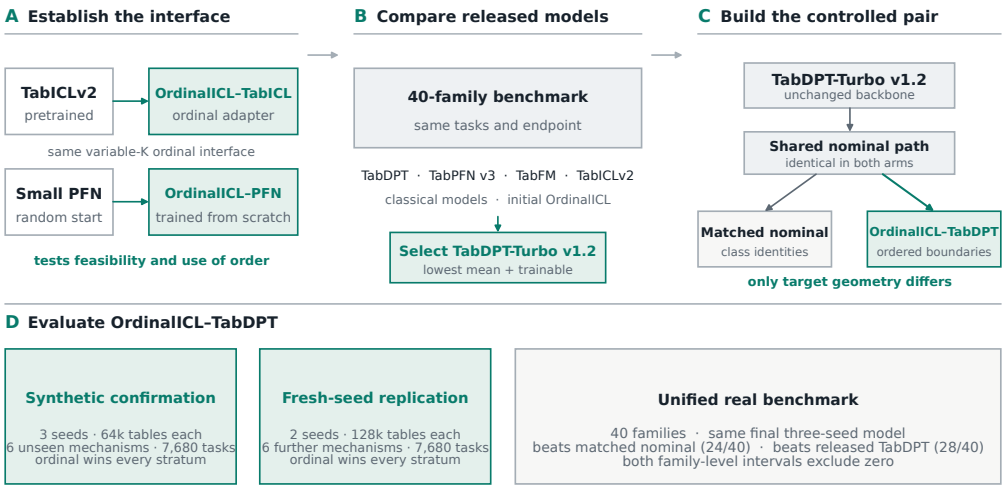

Figure 1. Study logic and model lineage. (A) Adapter and scratch-PFN models establish the interface. (B) The original survey selects TabDPT-Turbo v1.2. (C) The matched models share backbone, capacity, and training, differing only in target geometry. (D) Two synthetic holdouts and the unified real benchmark evaluate the final model.

2 ORDERED TARGETS AND PRIOR WORK

2.1 WHAT AN ORDINAL MODEL SHOULD PRESERVE

An ordinal task is unchanged when its category names are replaced without changing their order. For example, renaming `low, medium, high` as `0, 10, 11` does not change the task, even though the numerical gaps have changed. It yields four requirements for a model that uses ordinal information:

1. **Invariance to order-preserving recodings.** Replacing category names by any strictly increasing set of codes should not change the predicted probabilities, apart from renaming their output coordinates.
2. **Equivariance to reversal.** If the scale is presented in the opposite direction, such as `high, medium, low`, the output probabilities should be reversed in exactly the same way and should otherwise remain unchanged.
3. **Sensitivity to the declared order.** If category identities, features, targets, and model capacity are held fixed but the declared order is scrambled, the predictions should change. A model that is unaffected by this intervention is not using the supplied order.
4. **Retention of unobserved categories.** Suppose the declared scale has five levels but the labeled context contains no three-star example. The model should still return a five-category distribution with three stars between two and four. The task declaration must determine the scale, including categories absent from the context.

The first two requirements describe symmetries of the prediction problem. The last two become controlled tests of the implementation: one tests whether the model uses order, and the other tests whether it respects the complete declared scale under sparse observations.

2.2 WHY NOMINAL CLASSIFICATION AND RANK REGRESSION ARE INSUFFICIENT

A nominal classifier receives category identities but no relation between them. If all permutations of those identities are equally valid, the model has no information with which to distinguish an adjacent pair, such as `low` and `medium`, from a nonadjacent pair, such as `low` and `high`. This is a direct consequence of permutation symmetry, formalized in Proposition 1 of the supplement.

Treating ordinal labels as regression targets has the opposite problem: it adds distances that the task did not specify. Under squared error, codes `0, 1, 2` and `0, 10, 11` generally produce different optimal predictions, although both encode the same order. Propositions 1 and 2 establish these limitations independently of the neural architecture.

OrdinalICL therefore receives the declared order explicitly. A label is represented by the thresholds it exceeds, together with a mask identifying which thresholds belong to the declared scale. This representation retains all K output categories even when one is absent from the context. The decoder jointly predicts nonincreasing cumulative probabilities; taking adjacent differences then produces nonnegative class probabilities that sum to one. Ranked probability score (RPS) is a strictly proper scoring rule for this distribution (Epstein 1969; Gneiting and Raftery 2007).

The coherence guarantee is marginal: each query observation receives a valid categorical distribution. Cross-query coherence is outside the model definition. A recent empirical study reports failures of marginalization and factorization consistency in both classification and regression TFMs (Klötergens et al. 2026). Our formal results concern the ordinal representation and decoder; predictive performance is evaluated separately in Sections 4 and 5.

The decoder can represent exactly any distribution in which every category has positive probability. It can approximate a distribution assigning exactly zero probability to a category as closely as desired, while finite numerical outputs are clamped and renormalized when necessary. The model also handles a coarser ordinal scale in the natural way: combining adjacent categories amounts to removing the threshold between them and adding their probabilities. A coarsened task has fewer thresholds and a different RPS, so its score is not directly comparable with the score on the original scale. Each dataset therefore retains its declared categories throughout evaluation.

2.3 Relation to existing methods

Classical proportional-odds models learn an ordinal decision rule for one dataset (McCullagh 1980). CORAL is a neural ordinal regression method that converts a K-category problem into $K-1$ rank-consistent binary threshold decisions (Cao et al. 2020). CORN and ADABORD provide related neural and boosting approaches (Shi et al. 2023; Ayllón-Gavilán et al. 2026). Classifier Pooling offers model-agnostic cumulative and hierarchical reductions (Rotenberg et al. 2026). Each method requires a separate fit for a new dataset.

Other recent work develops task-specific ordinal representations and losses. TabCGOK models retrieval and relations between ordered classes; GoR and DiffoR use discrete and continuous generation; ConOrd uses rank-sensitive contrastive learning; and OCE modifies cross-entropy using ordinal costs (Luo et al. 2024; Ma, Zhou, et al. 2026; Ma, Wang, et al. 2026; Lee et al. 2026; Dvora et al. 2026). OrdinalICL differs in learning a shared interface for different category counts and applying it in context to previously unseen tables.

Several studies are closer at the foundation-model level. TabPFN-MT pretrains a target encoder and shared decoder for multiple classification targets, and FlexTab supports modular pretrained decoders (Cureton and Armanfard 2026; Polewczyk et al. 2026). Nori-30M provides a released wrapper for categorical or ordinal targets: it integrates a regression-pretrained TFM's predictive distribution over cells around the declared ordered levels (Synthefy 2026). They supply pretrained target encoders, modular decoders, or wrappers. OrdinalICL trains a shared variable-cardinality ordinal interface across tables.

TabFM is a recent zero-shot TFM trained with synthetic tables, while TabDPT-Turbo pretrains on real tables (Kong and Das 2026; Hosseinzadeh et al. 2026). Their released classifiers are general nominal predictors with nominal target representations. We include both in the model survey, averaging their predictions over the declared category order and its reversal.

Learning-to-rank is a neighboring but distinct setting. Pretrained tabular models have been studied for query-grouped ranking, while R1-Ranker is an LLM-based reasoning framework evaluated across recommendation, routing, and passage-ranking tasks (Hou et al. 2024; Feng et al. 2025). As discussed in the introduction, OrdinalICL instead predicts a calibrated distribution on a fixed ordinal scale for each observation.

To the best of our knowledge, based on a literature review completed in September 2026, no released tabular in-context model had been trained across heterogeneous tables with a shared interface for ordinal targets whose number of categories varies by table. Our contribution is this shared variable-cardinality ordinal interface.

3 OrdinalICL

Model design. A cumulative wrapper preserves unseen-table ICL but needs `K-1` backbone calls and learns no ordinal parameters across tables. A conventional ordinal learner fits anew per dataset. In contrast, OrdinalICL uses one shared variable-K decoder, receives declared order before query prediction, learns ordinal parameters across tables, and returns coherent class mass in one joint backbone pass per orientation. Its defining contribution is to integrate declared order into the shared context representation and learn a single ordinal interface across heterogeneous tables.

Table 1. Comparison with cumulative reduction.

Property	Generic cumulative TFM	OrdinalICL
Order enters shared context	No	Yes
Ordinal parameters learned across tables	No	Yes
One joint decoder shared across declared K	No	Yes
Backbone calls per orientation	$K-1$	One joint call
Coherent class mass	After reduction or repair	By construction

3.1 DECLARED-SCALE INTERFACE

For category rank y, declared count K, and $j = 0, \ldots, K_{max} - 2$, the order branch receives

$$t_j(y) = \mathbf{1}[y > j], \qquad m_j(K) = \mathbf{1}[j < K - 1],$$

plus centered rank $2y/(K-1) - 1$. For each context label, the input records three pieces of information separately: the number of categories K, the label's identity, and its position in the declared order. This separation lets the destroyed-order control change only the stated order and lets the matched one-hot control use class identity at the same capacity. OrdinalICL–TabICL and OrdinalICL–PFN encode labels with the threshold indicators t_j, validity mask m_j, and centered rank; class identity is used only by the matched controls. The $K - 1$ cumulative coordinates are sufficient and dimension-minimal for an interior K-class distribution. Merging adjacent categories simply removes their separating threshold. The declared K is never inferred from the largest observed context rank.

3.2 COHERENT THRESHOLD DECODER AND LOSS

Each query representation interacts with a context summary and explicit threshold tokens encoding relative threshold position and scale. In the initial implementations, raw scores are transformed so logits are nonincreasing; the final model obtains the same monotonicity through the sequential factorization described below. Hence

$$1 \geq q_0 \geq \cdots \geq q_{K-2} \geq 0, \qquad q_j = P(Y > j).$$

Adjacent differences give nonnegative class probabilities summing to one. Invalid padded thresholds are masked. For table b, query i, and valid threshold j, training minimizes

$$\mathcal{L}_b = \frac{1}{n_b(K_b - 1)} \sum_{i,j} (\hat{q}_{bij} - q_{bij})^2.$$

With finite real-valued parameters, the decoder can represent any distribution whose category probabilities are strictly positive. In float32, sigmoid cancellation or saturation can round a valid mass to zero; clamping and renormalization preserve coherence, while the optional BCE factor retains gradient under saturated-logit stress. Tables 2–4 evaluate predictive accuracy separately from this coherence guarantee.

The normalization gives every table equal weight, regardless of its number of ranks. Targets can be hard indicators or empirical multi-rater cumulative probabilities. Averaging judge-level RPS differs from fitting their empirical soft cumulative target only by a prediction-independent disagreement constant. Ordinal reversal preserves RPS exactly; OrdinalICL directly optimizes the Reynolds-averaged distribution.

3.3 INITIAL IMPLEMENTATIONS

OrdinalICL–TabICL is a parameter-efficient adaptation of the released 27.56M-parameter TabICLv2 backbone. Its ordinal adapter trains 4.83M parameters for 20,000 steps and ensembles three predetermined initializations. **OrdinalICL–PFN** is a compact prior-fitted network trained from random initialization, with width 64, two column blocks, two row blocks, six in-context blocks, and 3.51M trainable parameters. Both implementations are trained on the same family of synthetic ordered tasks and use the same target interface, decoder, objective, and inference procedure. The initial constructions test whether a shared variable-K ordinal interface learns order and transfers across tables.

3.4 FINAL MODEL AND MATCHED CONTROL

The main architectural comparison uses the best-performing accessible backbone in our original survey, **TabDPT-Turbo v1.2**. The resulting ordinal model is **OrdinalICL–TabDPT**. We freeze all 63.51M backbone parameters and a shared 424,129-parameter nominal skip adapter. Each arm then

adds a 424,129-parameter correction module with identical trainable tensor shapes. The nominal module uses class-identity lookup tokens and free class evidence. The ordinal module uses smooth rank tokens and a sequential ordinal decoder. At boundary j, it uses the conventional risk set, consisting only of context rows whose labels have reached that level, $Y \geq j$, and estimates

$$c_j(x) = P(Y > j \mid Y \geq j, x), \qquad q_j(x) = \prod_{r=0}^{j} c_r(x).$$

The cumulative product is monotone by construction, and adjacent differences give a valid class distribution. Both modules start with exactly zero effect relative to the common path. Their paired initialization tensors, realized training tables, loss, optimizer, schedule, precision, and compute are identical. The matching attributes within-seed differences to target geometry while holding model scale and training budget fixed.

All OrdinalICL implementations use exact two-orientation Reynolds inference. For a forward prediction $p_f(y \mid x, D_c)$, we reverse every context rank within its declared scale, predict again, map the result back, and average:

$$p_{\mathrm{R}}(y \mid x, D_c) = \tfrac{1}{2}\left[p_f(y \mid x, D_c) + p_f(K - 1 - y \mid x, RD_c)\right].$$

Reynolds averaging guarantees reversal equivariance algebraically. The two orientations are packed into one batched backbone invocation, with roughly twice the example volume of an unsymmetrized forward path.

3.5 Initial feasibility comparisons

We train each implementation with three seeds for 20,000 steps. All synthetic effects use paired realized tables and a 10,000-resample table bootstrap. OrdinalICL–TabICL is compared with its initialization, a model with the same architecture and parameter count but a scrambled order channel, and the released TabICLv2 cumulative wrapper. OrdinalICL–PFN is compared with its random initialization, a corresponding scrambled-order model, a parameter-matched one-hot model, and the same cumulative wrapper. We also measure reversal error and variation across training seeds.

The destroyed-order controls preserve the architecture, parameter count, features, labels, and training budget, but scramble the declared category order. Comparing them with OrdinalICL measures whether the order information is useful. The PFN study also includes a matched one-hot control that replaces the ordinal code with a category-identity representation of nearly equal capacity; this checks whether the result can be explained by a larger target encoder.

The controls identify four effects: learning relative to initialization, access to the correct order, ordinal encoding at matched capacity, and performance relative to the released cumulative wrapper. The synthetic prior varies class count from 2 to 10 and includes missing internal ranks, covariate and mechanism shift, hard or soft ordinal targets, and factors held disjoint between training and validation. Its role is to stress-test the construction across controlled mechanisms; the 40 real families evaluate external transfer.

4 Experimental design

4.1 Synthetic evaluation

The synthetic generator produces tables with 1–12 features, 2–10 declared classes, 32 context rows, and 64-row packed sequences. It varies latent ordinal mechanisms, feature geometry, imbalance, noise, missing context ranks, and distribution shift. The initial interface study uses three predetermined training seeds and 1,536 paired evaluation tables per comparison; its intervals are paired 10,000-resample table bootstraps. The released TabICLv2 cumulative wrapper is the shared predictive reference. The separate seeds and evaluation sets for the final matched study are specified below.

4.2 UNIFIED EXTERNAL BENCHMARK

The unified real benchmark comprises 40 independent dataset families and 46 scored views. The tasks span product and media ratings, public reviews, health and social surveys, educational outcomes, and temporal or entity-cold transfer. Repeated country folds, product categories, related releases, and survey rounds are grouped into one inferential family. Every view provides four 32-row contexts; 36 views use 512 queries and ten use 320. Predictors exclude target and post-outcome information, and temporal or entity features use only information available before the predicted event.

4.3 MODEL SURVEY AND BACKBONE CHOICE

The model survey compares OrdinalICL–TabICL and OrdinalICL–PFN with released TabICLv2, TabPFN v2.6 and v3, TabFM, TabDPT-Turbo v1.2, tuned LightGBM (Ke et al. 2017), proportional odds, and CORAL. Every survey score uses the same 40 families and 46 views, selected rows, features, contexts, queries, declared category order, and failure rule. General classifiers are evaluated through either a cumulative wrapper or their stronger nominal head with exact two-orientation Reynolds averaging. The survey selects the backbone. The final matched experiment supplies the formal architectural comparison.

TabDPT-Turbo v1.2 has the lowest mean in the original survey and exposes the internals required for an exactly paired training path. Performance and experimental controllability determine the backbone choice. Table 4 also includes the later TabDPT v1.3 and TabPFN 3.5 releases, evaluated on the same 40 families with the primary OrdinalICL model already fixed. These additions do not enter the original model-selection comparisons.

4.4 MATCHED-BACKBONE PROTOCOL

The architecture-controlled study freezes the TabDPT-Turbo v1.2 backbone and shared nominal path described in Section 3. Two independently prespecified confirmation studies use disjoint synthetic mechanisms absent from training. The first trains three paired seeds on 64,000 tables per arm and evaluates 7,680 held-out tasks. The replication changes no architecture or optimization rule, doubles training to 128,000 tables, uses two fresh paired seeds, and evaluates 7,680 tasks from six further mechanisms. All checkpoints are fixed before their evaluation tasks are generated. The final 128,000-table model averages three seeds and is evaluated on all 40 real families. No real row enters model training.

The endpoint is threshold-mean normalized RPS, RPS divided by $K - 1$, with query-population reweighting. We average over contexts, then views, then give each dataset family equal weight. Lower is better. Intervals use 10,000 bootstrap resamples of dataset families, and exact sign-flip tests are adjusted by Holm's method across the ten initial-model survey comparisons. A survey comparison passes when its mean improvement is at least 0.005, the upper 95% confidence bound is below zero, and the adjusted exact test is below 0.05. Failed cells are retained at the metric's worst value, one. For the matched synthetic confirmations, the prespecified minimum ordinal-minus-nominal gains are 0.00015 and 0.00020 nRPS, together with an upper hierarchical interval below zero and favorable subgroup directions. The unified 40-family final-model analysis uses the same family bootstrap and exact sign-flip test for its two paired comparisons. The survey threshold is larger because it compares entire models; the matched thresholds measure one controlled architectural change to a shared prediction path.

5 RESULTS

5.1 SYNTHETIC EVIDENCE

Initial interface checks. Table 2 shows that both implementations learn to use category order and improve over the cumulative TabICLv2 reference. Destroying the order-bearing channel degrades both models by more than 0.04 nRPS. The adapter also improves on its initialization, while the compact PFN learns substantially from random initialization and matches the capacity of its parameter-matched one-hot model.

Table 2. Does the model learn and use order? Δ is comparator minus OrdinalICL nRPS, so positive values favor OrdinalICL. Intervals are paired-table 95% bootstrap intervals over 1,536 tables.

Model	Comparator	Δ nRPS ↑	95% interval
OrdinalICL–TabICL	initialization	+0.0100	[+0.0092, +0.0109]
OrdinalICL–TabICL	destroyed order	+0.0408	[+0.0386, +0.0430]
OrdinalICL–TabICL	TabICLv2 cumulative	+0.0382	[+0.0365, +0.0400]
OrdinalICL–PFN	random initialization	+0.0452	[+0.0431, +0.0474]
OrdinalICL–PFN	destroyed order	+0.0437	[+0.0414, +0.0461]
OrdinalICL–PFN	matched one-hot model	0.0000	[-0.0003, +0.0003]
OrdinalICL–PFN	TabICLv2 cumulative	+0.0396	[+0.0378, +0.0414]

Across the common 1,536-table synthetic evaluation set, absolute mean soft nRPS was 0.1280 for OrdinalICL–TabICL, 0.1266 for OrdinalICL–PFN, and 0.1662 for the released wrapper. Reversal and Reynolds-identity errors are below 10^{-6} for both models. The PFN and one-hot means are equal to four decimals, supporting the capacity match. Scrambling the declared order degrades the PFN by 0.0437 nRPS, showing that the model uses that order.

Final matched-backbone confirmations. Table 3 gives the main architecture result. The ordinal module improves on its matched nominal control on the first synthetic holdout and again in an independent doubled-compute replication. Because only target geometry differs within each pair, this matching isolates the contribution of the ordinal architecture.

Table 3. OrdinalICL–TabDPT versus its matched nominal control on held-out synthetic tasks. Effects are ordinal minus nominal soft nRPS; lower is better. Intervals resample both training seeds and mechanism groups.

Evaluation	Paired training	Tasks	Δ nRPS	Hierarchical 95% interval
Confirmation	3 seeds × 64k	7,680	-0.000454	[-0.000624, -0.000293]
Doubled-compute replication	2 fresh seeds × 128k	7,680	-0.000575	[-0.000736, -0.000418]

Every training seed, class count, unseen mechanism, seed–mechanism cell, and context-coverage stratum favors the ordinal arm in both studies. All prespecified gates pass, and an independent 41-check audit reproduces the replication from raw records. On the first holdout, OrdinalICL–TabDPT also improves on the unchanged released TabDPT by 0.004909 nRPS, with hierarchical 95% interval [-0.005735, -0.004111] for ordinal minus released.

5.2 Unified 40-family real benchmark

Table 4 compares the ten original candidates and two late releases on all 40 families. TabDPT-Turbo v1.2 led the original survey and supplies the accessible internals needed for the controlled experiment. The final panel evaluates the three-seed OrdinalICL–TabDPT, its matched nominal control, and the released backbone on the same families.

Table 4. Unified 40-family model survey and final TabDPT experiment. Scores are equal-family mean population-reweighted threshold-mean nRPS; lower is better. Both panels use all 40 families and 46 views.

Method	Mean nRPS	Median family nRPS
A. Common survey and late-release additions		
TabPFN 3.5 cumulative[‖]	**0.1485**	0.1472
TabDPT-Turbo v1.2 nominal[†]	0.1486	**0.1468**
TabDPT-Turbo v1.3 nominal[‖]	0.1492	0.1471
TabPFN v3 cumulative	0.1501	0.1487
OrdinalICL–TabICL (adapter)	0.1505	0.1472
OrdinalICL–PFN (from scratch)	0.1517	0.1517
TabPFN v2.6 cumulative[§]	0.1528	0.1512
TabFM nominal[‡]	0.1530	0.1526

Method	Mean nRPS	Median family nRPS
Proportional odds	0.1617	0.1622
Tuned LightGBM cumulative	0.1627	0.1540
TabICLv2 cumulative	0.1676	0.1723
CORAL	0.2166	0.2030
B. Final TabDPT experiment (unified 40 families)		
OrdinalICL–TabDPT (ours)*	**0.1469**	**0.1453**
Matched nominal TabDPT*	0.1471	0.1457
Released TabDPT-Turbo v1.2*	0.1490	0.1468

†Panel A uses the common survey wrapper and inference settings; Panel B uses the final controlled experiment's shared inference settings. ‡Evaluation is subject to the model's research-use license. §Previous-generation TabPFN. *Final 128,000-table configuration averaged over three paired seeds. All comparisons within each panel use identical contexts and feature permutations. ‖Late releases; excluded from the original selection comparisons.

Both initial OrdinalICL implementations outperform TabICLv2, tuned LightGBM, proportional odds, and CORAL under the stated decision rule, but are statistically indistinguishable from TabPFN v3 after Holm correction. TabDPT leads all 40 leave-one-family-out rankings of the original roster and remains first after excluding families containing a failure. Full contrasts are in the supplement.

Across all 40 families, mean nRPS is 0.146894 for OrdinalICL–TabDPT, 0.147059 for matched nominal, and 0.148968 for released TabDPT. Ordinal minus matched nominal is -0.0001649 [-0.0003002, -0.0000277], exact $p = 0.0249$, with 24/40 wins. Ordinal minus released TabDPT is -0.0020735 [-0.0033762, -0.0009739], $p = 0.00025$, with 28/40 wins. Every training-seed mean favors ordinal over its matched nominal counterpart. The family-level intervals exclude zero for both comparisons.

Late-release evaluations. Released after the primary study and manuscript draft were complete, TabPFN 3.5 has the lowest survey mean; its difference from TabDPT v1.2 is not statistically significant. Our fixed OrdinalICL–TabDPT outperforms TabPFN 3.5 on 30/40 families (mean nRPS difference -0.00164, 95% CI [-0.00251, -0.00073]). Separately, applying the unchanged training recipe to TabDPT v1.3 reproduces the ordinal-over-nominal gain on both synthetic holdouts and the 40 real families. Details are in the appendix.

6 Discussion and limitations

OrdinalICL gives a tabular foundation model a shared variable-cardinality ordinal interface with coherent probabilities and exact reversal symmetry. Attached to TabDPT, this interface beats an equal-capacity nominal module under matched backbone, data, initialization, optimization, and compute. The advantage holds across two independent synthetic holdouts and the unified 40-family real benchmark, and reproduces with the TabDPT v1.3 backbone. In these matched comparisons, target order improves TabDPT without increasing model size or training compute.

The study evaluates one context size, one endpoint, and a finite collection of sources. Three benchmark families appear in TabDPT's published pretraining list (Ma et al. 2025). This overlap may inflate TabDPT's absolute scores on those families. It does not invalidate the matched ordinal-versus-nominal experiment: both arms use the identical pretrained backbone, so pretraining exposure is shared across the pair. Their difference isolates the effect of changing the target interface within that backbone.

Ethics Statement

The real-data experiments reuse public or research-access datasets; we collected no new human-subject data. Identifiers such as respondent, user, and business IDs were not used as predictive features. The artifact reports dataset-family aggregate scores and distributes no predictions for individual records. It includes adapted ESS records under CC BY-NC-SA 4.0 and omits Yelp records under Yelp's academic-use terms. Real-data results are reported at the dataset-family level.

AI Use Statement

We used generative AI, notably OpenAI Codex, to help guide the research; generate, collect, and prepare data; formalize concepts and mathematical arguments; design and implement methods and experiments; analyze results; identify relevant literature and fill gaps in the authors' knowledge; and assist with writing and editing. The human authors take responsibility for the final content, including all text, claims, code, and artifacts produced with AI assistance.

Reproducibility Statement

The appendix describes the datasets, preprocessing, model, training procedure, evaluation, and proofs. Code, configurations, and the final learned modules are provided in the anonymous supplementary material.

References

Ayllón-Gavilán, Rafael, Francisco José Martínez-Estudillo, David Guijo-Rubio, César Hervás-Martínez, and Pedro A. Gutiérrez. 2026. "ADABORD: A Novel AdaBoost Approach for Ordinal Classification." *arXiv Preprint arXiv:2607.21003*. https://arxiv.org/abs/2607.21003.

Cao, Wenzhi, Vahid Mirjalili, and Sebastian Raschka. 2020. "Rank Consistent Ordinal Regression for Neural Networks with Application to Age Estimation." *Pattern Recognition Letters* 140: 325–31. https://doi.org/10.1016/j.patrec.2020.11.008.

Cureton, Cormac, and Narges Armanfard. 2026. "TabPFN-MT: A Natively Multitask in-Context Learner for Tabular Data." *arXiv Preprint arXiv:2605.20234*. https://arxiv.org/abs/2605.20234.

Dvora, Tal, Rotem Haba, and Gonen Singer. 2026. "Deep Neural Networks with Ordinal Loss for Medical Applications." *arXiv Preprint arXiv:2606.25769*. https://arxiv.org/abs/2606.25769.

Epstein, Edward S. 1969. "A Scoring System for Probability Forecasts of Ranked Categories." *Journal of Applied Meteorology* 8 (6): 985–87. https://doi.org/10.1175/1520-0450(1969)008%3C0985:ASSFPF%3E2.0.CO;2.

Feng, Tao, Zhigang Hua, Zijie Lei, et al. 2025. "R1-Ranker: Teaching LLM Rankers to Reason." *arXiv Preprint arXiv:2506.21638*. https://arxiv.org/abs/2506.21638.

Gneiting, Tilmann, and Adrian E. Raftery. 2007. "Strictly Proper Scoring Rules, Prediction, and Estimation." *Journal of the American Statistical Association* 102 (477): 359–78. https://doi.org/10.1198/016214506000001437.

Grinsztajn, Léo, Klemens Flöge, Oscar Key, et al. 2026. "TabPFN-3: Technical Report." *arXiv Preprint arXiv:2605.13986*. https://arxiv.org/abs/2605.13986.

Hollmann, Noah, Samuel Müller, Lennart Purucker, et al. 2025. "Accurate Predictions on Small Data with a Tabular Foundation Model." *Nature* 637 (8045): 319–26. https://doi.org/10.1038/s41586-024-08328-6.

Hosseinzadeh, Rasa, Alex Labach, Zexin Xue, Shuyi Han, Valentin Thomas, and Anthony L. Caterini. 2026. "TabDPT-Turbo: Efficient in-Context Learning for Tabular Prediction." *arXiv Preprint arXiv:2608.01400*. https://arxiv.org/abs/2608.01400.

Hou, Charlie, Kiran Koshy Thekumparampil, Michael Shavlovsky, Giulia Fanti, Yesh Dattatreya, and Sujay Sanghavi. 2024. "Pretrained Deep Models Outperform GBDTs in Learning-to-Rank Under Label Scarcity." *Transactions on Machine Learning Research*. https://openreview.net/forum?id=093Q9VxaWt.

Ke, Guolin, Qi Meng, Thomas Finley, et al. 2017. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *Advances in Neural Information Processing Systems* 30. https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

Klötergens, Christian, Vijaya Krishna Yalavarthi, Lars Schmidt-Thieme, and Tom Hanika. 2026. "Do Tabular Foundation Models Agree with Themselves?" *arXiv Preprint arXiv:2608.06004*. https://arxiv.org/abs/2608.06004.

Kong, Weihao, and Abhimanyu Das. 2026. *Introducing TabFM: A Zero-Shot Foundation Model for Tabular Data*. Google Research blog. https://research.google/blog/introducing-tabfm-a-zero-shot-foundation-model-for-tabular-data/.

Lee, Chaewon, BeomJun Shim, Kwang Pyo Choi, and Chang-Su Kim. 2026. "Contrastive Order Learning: A General Framework for Ordinal Regression." *Proceedings of the 43rd International Conference on Machine Learning*, Proceedings of machine learning research, vol. 306. https://icml.cc/virtual/2026/poster/61309.

Luo, Zhengdong, Abibulla Atawulla, Fengyi Yang, et al. 2024. "TabCGOK: Intra-Class Groups Retrieval and Inter-Class Ordinal Knowledge Augmented Network for Ordinal Tabular Data Prediction." *ECAI 2024 – 27th European Conference on Artificial Intelligence*, Frontiers in artificial intelligence and applications, vol. 392: 2242–49. https://doi.org/10.3233/FAIA240746.

Ma, Hongxu, Lin Wang, Chenghou Jin, et al. 2026. "DiffoR: A Unified Continuous Generative Framework for Universal Ordinal Regression." *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining v.2*, 3597–608. https://doi.org/10.1145/3770855.3818149.

Ma, Hongxu, Han Zhou, Kai Tian, et al. 2026. "GoR: A Unified and Extensible Generative Framework for Ordinal Regression." *International Conference on Learning Representations* 2026: 122256–82. https://proceedings.iclr.cc/paper_files/paper/2026/hash/c6d8301d2235036e1d23081252158a79-Abstract-Conference.html.

Ma, Junwei, Valentin Thomas, Rasa Hosseinzadeh, et al. 2025. "TabDPT: Scaling Tabular Foundation Models on Real Data." *Advances in Neural Information Processing Systems* 38: 172692–722. https://doi.org/10.52202/085713-5748.

McCullagh, Peter. 1980. "Regression Models for Ordinal Data." *Journal of the Royal Statistical Society: Series B (Methodological)* 42 (2): 109–27. https://doi.org/10.1111/j.2517-6161.1980.tb01109.x.

Polewczyk, Marek, Maximilian Schambach, Marco Spinaci, Sam Thelin, and Johannes Höhne. 2026. "FlexTab: A Flexible Encoder-Decoder Architecture for in-Context Learning Across Diverse Tabular Tasks." *arXiv Preprint arXiv:2606.30336*. https://arxiv.org/abs/2606.30336.

Prior Labs. 2026. *TabPFN-3.5*. Official model release notes. https://docs.priorlabs.ai/changelog/tabpfn-3.5.

Qu, Jingang, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2025. "TabICL: A Tabular Foundation Model for in-Context Learning on Large Data." *Proceedings of the 42nd International Conference on Machine Learning*, Proceedings of machine learning research, vol. 267: 50817–47. https://proceedings.mlr.press/v267/qu25d.html.

Qu, Jingang, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2026. "TabICLv2: A Better, Faster, Scalable, and Open Tabular Foundation Model." *arXiv Preprint arXiv:2602.11139*. https://arxiv.org/abs/2602.11139.

Rotenberg, Noam H., Andreia V. Faria, and Brian Caffo. 2026. "Classifier Pooling for Modern Ordinal Classification." *arXiv Preprint arXiv:2603.17278*. https://arxiv.org/abs/2603.17278.

Shi, Xintong, Wenzhi Cao, and Sebastian Raschka. 2023. "Deep Neural Networks for Rank-Consistent Ordinal Regression Based on Conditional Probabilities." *Pattern Analysis and Applications* 26 (3): 941–55. https://doi.org/10.1007/s10044-023-01181-9.

Synthefy. 2026. *Synthefy Nori: Tabular Foundation Model for Regression*. V. 0.18.0. Released August. https://github.com/Synthefy/synthefy-nori/releases/tag/synthefy-nori-v0.18.0.

Supplementary material

This supplement documents the benchmark, matched experiment, formal results, and implementation.

A Unified external benchmark

The benchmark contains 40 independent dataset families and 46 scored views. Related releases and categories are grouped within one family. The artifact includes adapted ESS records under CC BY-NC-SA 4.0 but excludes Yelp records because Yelp's academic-use agreement bars third-party sharing. It instead provides verified Yelp acquisition and preprocessing instructions; original terms govern both datasets.

A.1 Source inventory

Source family	Levels	Views
Afrobarometer R9	5	3
Amazon Reviews 2023	5	2
BookCrossing	10	1
BRFSS 2024	5	1
CSES Module 5	4	3
Druglib	5	1
ESS Rounds 10–11	5	1
Eurostat Personal Well-being 2024	10	1
Goodreads UCSD	5	1
Google Local Vermont	5	1
GSS 2024	3	1
HINTS 6 to 7	5	1
IMDb Large Movie Review	8	1
INE Living Conditions 2025	10	1
MEPS 2023	5	1
MovieLens	5–10	2
NFCS 2024	10	1

Source family	Levels	Views
NHANES general health	5	1
NHIS 2024	5	1
NSFG 2022–2023	5	1
OCLAR	5	1
OpenReview ICLR 2024	6	1
OULAD	3	1
PIAAC 2023 US	10	1
POLNES 2023	4	1
RelBench RateBeer	10	1
SCF 2022	4	1
SHED 2025	4	1
Stack Overflow 2020	5	1
Student Performance Portuguese	5	1
UCI Cardiotocography	3	1
UCI Diabetes 130-US Hospitals	3	1
UCI Drug Consumption	7	1
UCI Restaurant Consumer	3	1
UCI Room Occupancy	4	1
UCI SUPPORT2	4	1
UCI Turkiye Student Evaluation	5	1
UCI Wine Quality	5	1
Yelp Open Dataset	5	1
YRBSS 2023	5	1

The artifact records exact versions, acquisition, support, features, and splits. Survey-weighted views admit only finite, strictly positive final weights before support or selection: MEPS uses HC-251 `PERWT23F`, and GSS uses `WTSSPS` without unit-weight substitution. Documentation URLs and hashes are included.

Amazon categories and MovieLens releases are averaged within their respective families. Wine scores 3–4 and 8–9 are pooled into ordered tail levels to avoid a five-row extreme query level.

A.2 Initial-model comparisons on the real benchmark

These comparisons use all 40 real families. Differences are OrdinalICL model minus reference nRPS; negative favors OrdinalICL. The decision requires the effect, interval, and ten-way Holm conditions jointly.

OrdinalICL–TabICL.

Reference	Δ nRPS	95% interval	Exact p	Holm p	W/T/L	Decision
TabICLv2	-0.0171	[-0.0211, -0.0129]	< 0.0001	< 0.0001	34/0/6	pass
TabPFN-v3	+0.0004	[-0.0016, +0.0027]	0.6910	0.6910	24/0/16	fail
LightGBM	-0.0122	[-0.0242, -0.0049]	< 0.0001	< 0.0001	32/0/8	pass
PO	-0.0112	[-0.0147, -0.0079]	< 0.0001	< 0.0001	36/0/4	pass
CORAL	-0.0661	[-0.0857, -0.0500]	< 0.0001	< 0.0001	40/0/0	pass

OrdinalICL–PFN.

Reference	Δ nRPS	95% interval	Exact p	Holm p	W/T/L	Decision
TabICLv2	-0.0159	[-0.0201, -0.0116]	< 0.0001	< 0.0001	34/0/6	pass
TabPFN-v3	+0.0016	[-0.0006, +0.0038]	0.1663	0.3327	19/0/21	fail
LightGBM	-0.0110	[-0.0227, -0.0037]	0.0002	0.0006	27/0/13	pass
PO	-0.0101	[-0.0137, -0.0065]	< 0.0001	< 0.0001	35/0/5	pass
CORAL	-0.0649	[-0.0847, -0.0489]	< 0.0001	< 0.0001	40/0/0	pass

A.3 SOURCE-FAMILY SCORES

Scores first average the four contexts within a view and repeated views within a source. The machine-readable supplement contains all 400 family–method scores for the original ten-model survey, 80 additional late-release scores, and the final matched-model scores for all 40 families.

A.4 RETAINED FAILURES

Four of the original survey's 1,840 model-context evaluations failed and receive nRPS one under the common rule. CORAL fails on one IMDb, one MovieLens, and one NFCS context because its internal validation split is smaller than the number of observed classes. LightGBM fails on one Turkiye Student Evaluation context because an internal validation target contains one class. TabDPT leads the original roster after excluding all four affected families; TabPFN 3.5 leads the expanded survey.

A.5 FINAL MATCHED EXPERIMENT: REPRODUCIBILITY RECORD

The final experiment follows directly from the survey: TabDPT-Turbo v1.2 provides both the lowest original-survey mean and a checkpoint whose internals permit an exactly paired adaptation. Every run uses the upstream `tabdpt1_2.safetensors` checkpoint (SHA-256 `0668022 0fd66c4524051706b98c1c659a674d19d3a766cd0bb276505e99faccd`). The precursor OrdinalICL–TabICL and OrdinalICL–PFN results are retained above because they establish that the ordinal interface works with both a pretrained backbone and a model trained from scratch.

A.5.1 MATCHED ARCHITECTURE

Both final arms freeze the 63,507,472-parameter TabDPT-Turbo v1.2 backbone and the same 424,129-parameter nominal path. Each adds a 424,129-parameter module with identical trainable tensor shapes. The nominal control uses category-identity tokens and free class evidence. OrdinalICL–TabDPT uses rank coordinates, ordered risk sets, and a cumulative decoder. Within a paired seed, both modules start from identical tensors, consume the same realized training tables, and share the objective, optimizer, schedule, precision, and compute. Dataset-specific routing, blending, or test-time fitting is not used.

A.5.2 SYNTHETIC CONFIRMATIONS

The first confirmation trains three paired seeds on 64,000 synthetic tables per arm and evaluates 7,680 tasks from six mechanisms absent from training. Ordinal-minus-nominal nRPS is -0.0004540, with hierarchical 95% interval [-0.0006238, -0.0002933]. The independent replication changes no architecture or optimization choice, doubles training to 128,000 tables, and evaluates two fresh seed pairs on 7,680 tasks from six further unseen mechanisms. Its effect is -0.0005747 [-0.0007364, -0.0004185]. Every seed, class count, mechanism, seed–mechanism cell, and context-coverage stratum favors ordinal in both studies. The replication audit reproduces 41/41 checks from 23,040 raw paired records.

A.5.3 UNIFIED REAL EVALUATION

The final 128,000-table model averages three fixed training seeds and is scored on all 40 eligible real families. Each family receives equal weight after averaging contexts and repeated views. All 576 newly required cells for the two fresh seeds on the original survey families succeeded; no family or method was removed. Mean nRPS is 0.146894 for OrdinalICL–TabDPT, 0.147059 for matched nominal, and 0.148968 for released TabDPT. Ordinal minus nominal is -0.0001649 [-0.0003002, -0.0000277], exact $p = 0.0249$, with 24/40 wins. Ordinal minus released is -0.0020735 [-0.0033762, -0.0009739], $p = 0.00025$, with 28/40 wins. All three training-seed means favor ordinal over nominal.

All ten survey models were evaluated on the same 40 eligible families. The original 30-family predictions were reused without recomputation; the ten additional families use the same model

configurations and evaluation rule. Family inclusion followed source validity and task eligibility. A schema compatibility repair mapped three legacy field names and empty secondary-context fields to the current loader. It changed no rows, features, contexts, queries, weights, models, or metrics. Of 400 added model-context evaluations, 399 succeed; the sole CORAL failure is retained at nRPS one.

The anonymous artifact includes the protocols, hashes, model and evaluation code, compact family-level results, independent audits, and the exact first-party learned modules used by the final matched experiments. Third-party checkpoints, source-restricted rows, and row-level predictions are omitted. Their hashes are recorded in the artifact.

A.6 TabDPT v1.3 cross-release replication

TabDPT v1.2 was released on 17 June 2026. TabDPT v1.3 was released on 8 September 2026, after completion of the original survey and v1.2 experiments. The addendum uses official source commit `97e5494431e9527c7edb31cb4dcfc5f00b232fdf` and checkpoint SHA-256 `97dc3b60bfad6b42ec1a07b7e121d86b0fc7c9fd67c8d2eaac3d815da197eacb`. Released v1.3 has mean nRPS 0.149199 on the common survey wrapper. Its difference from released v1.2 is +0.000566 [-0.000168, +0.001340], exact $p = 0.1566$; lower is better.

We then retrain the unchanged final recipe on the v1.3 backbone. The common nominal path and each of three paired nominal or ordinal correction modules see 128,000 synthetic tables. Within every seed pair, initialization, realized tables, objective, optimizer, schedule, parameter count, and update count match exactly. The first synthetic holdout gives ordinal minus nominal -0.000957, with hierarchical 95% interval [-0.001201, -0.000646]; the independent holdout gives -0.000926 [-0.001179, -0.000671]. All 16 cardinality results, 12 mechanism results, six seed results, and four coverage-stratum results favor ordinal.

On the unified real benchmark, the v1.3 ordinal mean is 0.147098 versus 0.147527 for its matched nominal control and 0.149199 for released v1.3. Ordinal minus nominal is -0.0004287 [-0.0006121, -0.0002465], exact $p = 0.0000455$, with 30/40 wins. Ordinal minus released v1.3 is -0.0021007 [-0.0033738, -0.0009891], $p = 0.000407$, with 29/40 wins. The v1.3 ordinal model differs from the final v1.2 ordinal model by +0.0002039 [-0.0002400, +0.0006629], $p = 0.391$. With the same evaluation suite, the unchanged recipe reproduces the ordinal advantage on the newer backbone.

A.7 TabPFN 3.5 late-release evaluation

TabPFN 3.5 was released on 15 September 2026, after the primary experiments and manuscript draft were complete (Prior Labs 2026). We evaluated its base checkpoint on the unchanged 40-family benchmark, with our model and training recipe already fixed. The evaluation uses `tabpfn==9.0.0` and `tabpfn-v3.5-20260909.safetensors`, upstream revision `06bf2ba35c80a92a3b9abb436b99cf49e7a0365e`. As for TabPFN v3, inference uses eight estimators per cumulative binary classifier and two-orientation Reynolds averaging. Contexts, queries, feature encoding, model seeds, evaluation weights and failure rules are unchanged; other options use the release defaults. All 184 cells succeed. Fast, Plus, Thinking and native multiclass inference are outside this comparison.

TabPFN 3.5 improves on v3 on 29/40 families: mean nRPS difference -0.001562, family-bootstrap 95% CI [-0.002556, -0.000580], Holm-adjusted $p = 0.0104$ across its ten comparisons with the original roster. This effect is smaller than the survey's 0.005 practical threshold. Its mean, 0.148535, is slightly below TabDPT v1.2's 0.148633, but their difference is not statistically significant (-0.000098 [-0.001556, 0.001253], exact $p = 0.895$). TabPFN 3.5 leads 33/40 leave-one-family-out rankings and the 36-family complete-case analysis; TabDPT v1.2 leads the remaining seven leave-one-family-out rankings.

The final OrdinalICL–TabDPT retains a lower mean of 0.146894. OrdinalICL minus TabPFN 3.5 is -0.001641 [-0.002509, -0.000733], with 30/40 wins and exact $p = 0.00107$. This is a descriptive comparison of fixed models under their respective inference settings: the final model averages three training-seed scores with four feature ensembles, while the survey uses eight-estimator wrappers. It supplies a current reference on an already observed benchmark; the matched synthetic experiments remain the architecture test.

B FORMAL RESULTS

We give four results supporting the representation and loss used by OrdinalICL. Let the declared scale be $\mathcal{Y}_K = \{0, \dots, K-1\}$, with cumulative probabilities $q_j = P(Y > j)$ and targets $z_j(y) = \mathbf{1}\{y > j\}$.

B.1 PROPOSITION 1: NOMINAL PERMUTATION SYMMETRY CANNOT IDENTIFY ORDINAL ADJACENCY

Let $A(\{a, b\})$ be any adjacency decision constructed from a target whose only supplied structure is K distinct nominal identities. Relabeling equivariance requires the induced relation to be invariant under joint relabeling:

$$A(\{\pi(a), \pi(b)\}) = A(\{a, b\}) \qquad \text{for every } \pi \in S_K.$$

For $K \geq 3$, the symmetric group acts transitively on unordered pairs of distinct labels. Hence A must be constant on all such pairs and cannot distinguish an adjacent pair from a nonadjacent pair. An order-bearing input or another symmetry-breaking structure is necessary.

Permutation equivariance permits distinct class representations. The proposition concerns identification of adjacency without declared order; feature-based predictive regularities remain available.

B.2 PROPOSITION 2: SCALAR RANK REGRESSION IS CODING-DEPENDENT

Under squared loss and numeric codes $a_0 < \cdots < a_{K-1}$, the conditional Bayes predictor is $\sum_k p_k a_k$. After a strictly increasing recoding g, it becomes $\sum_k p_k g(a_k)$. Except when g is affine on the support, this is not generally $g(\sum_k p_k a_k)$. For example, masses $(1/2, 0, 1/2)$ on codes $(0, 1, 2)$ yield mean 1, whereas the increasing recoding $(0, 10, 11)$ yields 5.5. The ordinal problem is unchanged, but the squared-loss target and distances have changed.

Scalar ranks can remain a useful approximation, but their interpretation assumes metric spacing absent from the ordinal scale.

B.3 PROPOSITION 3: THE DECODER DEFINES A COHERENT ORDINAL DISTRIBUTION

Let $1 \geq q_0 \geq \cdots \geq q_{K-2} \geq 0$. Set

$$p_0 = 1 - q_0, \qquad p_k = q_{k-1} - q_k \ (1 \leq k \leq K-2), \qquad p_{K-1} = q_{K-2}.$$

Every mass is nonnegative and the sum telescopes to one. Conversely, $q_j = \sum_{k>j} p_k$, so the mapping is one-to-one. Coherence follows from a joint cumulative vector. Independent binary fits require a repair or joint parameterization to guarantee monotonicity.

OrdinalICL sets $\ell_0 = r_0$ and $\ell_j = r_0 - \sum_{s=1}^{j} \mathrm{softplus}(r_s)$, then uses $q_j = \mathrm{sigmoid}(\ell_j)$. Finite parameters therefore produce strictly decreasing cumulatives and positive class masses. Conversely, every strictly positive categorical distribution can be represented: compute its cumulatives, take their logits, and choose each raw gap through the inverse softplus. Distributions with zero-mass categories can be approached arbitrarily closely.

B.4 PROPOSITION 4: CUMULATIVE RPS IS STRICTLY PROPER AND REVERSAL-INVARIANT

Use the threshold-mean ranked probability score

$$L(q, Y) = \frac{1}{K-1} \sum_{j=0}^{K-2} (q_j - z_j(Y))^2.$$

Let $q_j^* = P(Y > j)$. Taking expectation threshold by threshold gives

$$E[L(q, Y)] = \frac{1}{K-1} \sum_j \{(q_j - q_j^*)^2 + q_j^*(1 - q_j^*)\}.$$

The second term is independent of q, and the first is zero only at $q = q^*$. Because the cumulative-to-mass map is injective, the full ordinal class distribution is uniquely recovered. Thus cumulative RPS is strictly proper.

Under scale reversal, class mass becomes $p'_k = p_{K-1-k}$, and cumulative probabilities satisfy $q'_j = 1 - q_{K-2-j}$. The hard or soft target transforms in the same way. Therefore

$$(q'_j - z'_j)^2 = (q_{K-2-j} - z_{K-2-j})^2.$$

Summing merely reindexes the thresholds, so RPS is unchanged. The symmetry requires joint reversal of prediction and target; reversing only one is an intervention that should generally change the score.

C Implementation details

C.1 Selected declared-scale interface

Each context label carries the declared K, its identity, and its order rank through an invertible packed integer. The selected models use K to construct nine validity-mask coordinates and the order rank to construct nine thermometer coordinates plus one centered-rank coordinate. Because the mask uses declared K, a category retains its threshold even when it is absent from the context. The identity path is inactive except in the matched controls.

For one orientation, the threshold decoder emits $K - 1$ nonincreasing cumulatives, which adjacent differences convert into class probabilities. Let f denote this predictor. OrdinalICL averages its prediction with the mapped-back prediction from the rank-reversed table:

$$p_{\mathrm{R}}(y \mid x, D_c) = \frac{1}{2}\left[f(y \mid x, D_c) + f(K - 1 - y \mid x, RD_c)\right].$$

Reversing the input exchanges the two summands, so the average is reversal-equivariant independently of the learned weights.

C.2 OrdinalICL–TabICL

OrdinalICL–TabICL averages three 20,000-step continuations of the released TabICLv2 backbone (27,556,859 total; 4,830,209 trainable parameters). Controls use the pretraining initialization or scramble only the order field.

C.3 OrdinalICL–PFN

OrdinalICL–PFN trains from random initialization for 20,000 steps: width 64, two column blocks, two row blocks, six in-context blocks, and four attention heads (3,511,243 total; 3,507,721 active trainable parameters). Its control with a scrambled order channel is parameter matched; the one-hot control has 3,504,841 active parameters.

C.4 Shared optimization and inference contract

Both implementations use three seeds and AdamW (3×10^{-4} learning rate, 10^{-4} weight decay), 1,000 warm-up steps, unit gradient clipping, batches of eight 64-row tables with 32-row contexts, at most 12 features, and $2 \le K \le 10$. Training minimizes $B^{-1} \sum_b \mathcal{L}_b$, using the per-table loss in Section 3.2 with exact Reynolds-averaged cumulatives and soft cumulative targets.