On the Rotation-Equivariance Geometry of Tabular Foundation Models

Mert Ogul [1]

Abstract

Tree-based models often outperform deep tabular models on benchmarks where features carry domain meaning, a phenomenon attributed to axis-alignment in the feature representation. We study when tabular foundation model architectures preserve or break feature-rotation symmetry under the orthogonal group $O(d)$, distinguishing fixed-parameter equivariance from class-level closure. First, PFN architectures with a row-affine encoder and a d-blind trunk and head are class-level closed under $O(d)$: rotations can be absorbed by reparameterising the first encoder layer, and the orbit-averaged predictor is $O(d)$-invariant. Second, population-level nonlinear column tokenisers satisfying an explicit witness condition are generically non-equivariant in the analytic-genericity sense. Third, this token-level obstruction propagates to predictor-level rotation variance only under an explicit downstream witness-separation condition. On 9 strict-Grinsztajn binary numerical tasks and 6 architectures under one training-data/evaluation-seed configuration, the observed $\mathrm{rotation\_std}$ diagnostic separates a low-std comparison group from ColumnPFN-style models on every task, with medians 0.0012–0.0025 versus 0.031–0.044. Because the ColumnPFN-style evaluation resamples sub-contexts with seeds coupled to rotation index, these magnitudes are evidence consistent with the taxonomy, not a pure rotation-only causal estimate. We also report a Monte Carlo oracle-Bayes-floor estimate for the synthetic priors used here (≈ 0.39 BCE). One fixed $3\times$ capacity recipe does not close the remaining gap.

[1]Eindhoven University of Technology. Correspondence to: Mert Ogul <m.ogul@student.tue.nl>.

Accepted to the 1st Workshop on Combining Theory and Benchmarks, CTB@ICML 2026, Seoul, South Korea.

1. Introduction

Tree-based models routinely outperform deep models on tabular benchmarks where the feature axes carry semantic content (Grinsztajn et al., 2022). Tree splits are axis-aligned. Rotating the feature space destroys the information they exploit. Rotation-invariant deep models are in this sense *too symmetric*: they refuse to use a structural property the data has. For tabular foundation models (TFMs) (Hollmann et al., 2023; 2025; Qu et al., 2025) pretrained on synthetic priors and deployed on real data, the locus of the axis-alignment inductive bias is an architectural choice. Whether and how a TFM respects $O(d)$ feature-rotation symmetry is the subject of this paper.

The TFM landscape. TabPFN v1 (Hollmann et al., 2023) tokenises each row by a shared linear projection. TabPFN v2 (Hollmann et al., 2025) uses per-cell tokens with random feature-identifier resampling at inference. TabICL/v2 (Qu et al., 2025; 2026) compute per-cell tokens through a hypernetwork conditioned on column statistics. Our proposed ColumnPFN takes the population-level extreme: one nonlinear token per feature axis, computed from all N rows. EquiTabPFN (Arbel et al., 2026) treats target-permutation but not feature rotation. To our knowledge no published paper proves a rotation-invariance or -variance property of any TFM architecture.

Contributions. **(1) Symmetry taxonomy.** Theorems 1–3 separate three levels of statement: a row-affine reparameterisation-closure baseline for PFN-style models, a generic token-level non-equivariance result for population-level nonlinear column tokenisers satisfying an explicit witness condition, and a conditional predictor-level propagation result requiring downstream witness separation. **(2) Operational empirical evidence.** Across 9 strict-Grinsztajn binary tasks and 6 architectures (MLP, XGBoost, FT-Transformer (row-affine), TabPFN-v2, ColumnPFN, scalar ablation), the reported $\mathrm{rotation\_std}$ diagnostic separates the low-std comparison models from the ColumnPFN-style models on every task ($\mathbf{9/9}$ separations, median separation factor $\mathbf{12}\times$). Because ColumnPFN-style evaluation resamples sub-contexts across rotations, we interpret this as evidence consistent with the taxonomy rather than as a rotation-only causal

estimate. **(3) Bayes-floor methodology.** A Monte Carlo estimator quantifies the irreducible (oracle) binary-cross-entropy of stochastic-labeller priors (mixture floor ≈ 0.39 BCE for both priors used here). Trained models plateau ≈ 0.22 above this oracle floor, and one fixed $3\times$ capacity recipe does not close the gap.

2. Theory

A *PFN context* is a triple $\mathcal{C} = (X, y, x_\star)$ with $X \in \mathbb{R}^{N\times d}$, $y \in \mathcal{Y}^N$, $x_\star \in \mathbb{R}^d$. The orthogonal group $G = O(d)$ acts on $\mathcal{C}$ by right multiplication $g \cdot \mathcal{C} := (Xg, y, x_\star g)$. Strictly this is a right action (notation we retain for compactness). $O(d)$ is unimodular, so left- and right-Haar measures coincide and the integration arguments below are unaffected. A predictor f is $O(d)$-*invariant* if $f(g \cdot \mathcal{C}) = f(\mathcal{C})$ for all g and $\mathcal{C}$. We model a PFN as a composition of layers acting on representations $X \in \mathbb{R}^{N\times d'}$, classified into row-wise, column-wise, cross-row, cross-column, and pool-then-mix primitives. Formal definitions are deferred to Appendix A. The results below operate at different levels. Theorem 1 is a class-level closure baseline: it shows what row-affine PFN families can absorb by reparameterising the first encoder layer, not that a fixed checkpoint is invariant. Theorem 2 is the main token-level non-equivariance result. Theorem 3 then gives a conditional route from token-level non-equivariance to predictor-level rotation variance, requiring the downstream stack not to collapse the witness-separated token configurations.

Class-level closure baseline. The literal conjecture that any row-wise transformer with a shared linear projection is exactly $O(d)$-equivariant for fixed parameters is false (counterexample in Appendix A). The useful baseline is instead class-level closure: closure under rotation up to a parameter reparameterisation of the first row-affine encoder layer.

Definition 1 (Linear-projection encoder, d-blind trunk)**.** The row-encoder class $\mathcal{F}_{\mathrm{row}}^{(d,d_{\mathrm{model}})}$ consists of maps $\phi(x, y) = \eta(xW + b, y)$ for some $W \in \mathbb{R}^{d\times d_{\mathrm{model}}}$, $b \in \mathbb{R}^{d_{\mathrm{model}}}$, and measurable η. A parameterised map M_θ is d-*blind* if no parameter tensor's shape formally depends on d and no operation in M_θ indexes into a d-axis of its input. (Numerical coincidences such as a hyperparameter d_{model} equalling d do not affect this structural definition.)

Theorem 1 (Class-level $O(d)$-closure)**.** *Suppose the row- and query-encoders lie in $\mathcal{F}_{\mathrm{row}}^{(d,d_{\mathrm{model}})}$, the trunk and head are d-blind, and no feature-identity or feature-position embedding is added before the encoders. Then for every θ and every $g \in O(d)$ there exists a parameter $\theta^{(g)}$ (changing only the encoder first-layer weights) such that $f_{\theta^{(g)}}(g \cdot \mathcal{C}) = f_\theta(\mathcal{C})$ for every $\mathcal{C}$. Under any isotropic prior over $\mathcal{C}$ and any convex loss, the orbit-averaged predictor $\bar{f}$ is $O(d)$-invariant and dominates f_θ in expected risk (along the lines of Elesedy & Zaidi, 2021).*

Proof in Appendix A.

Generic non-equivariance of column tokens. A column encoder $\mathrm{Enc}_\eta : \mathbb{R}^N \times \mathcal{Y}^N \to \mathbb{R}^k$ produces a population-level token per axis: $T_\eta(X, y) = (\mathrm{Enc}_\eta(X_{:,1}, y), \ldots, \mathrm{Enc}_\eta(X_{:,d}, y))$. We say T_η is $O(d)$-equivariant if there is a continuous representation $\rho : O(d) \to GL_{kd}(\mathbb{R})$, independent of X, y, and η, with $T_\eta(Xg, y) = \rho(g)\, T_\eta(X, y)$ for all X, y, g. A subset of $\mathbb{R}^p$ is *generic* if its complement is contained in the zero set of a non-identically-zero real-analytic function on $\mathbb{R}^p$. By (Mityagin, 2015) every such zero set is closed, nowhere dense, and Lebesgue-null.

Theorem 2 (Generic non-equivariance)**.** *Let $\{\mathrm{Enc}_\eta\}_{\eta\in\mathbb{R}^p}$ be permutation-invariant in the row index, with $\eta \mapsto \mathrm{Enc}_\eta(c, y)$ real-analytic. Assume the* witness condition*: there exist contexts $X_1, X_2 \in \mathbb{R}^{N\times d}$ and a constant y_0 with $T_\eta(X_1, y_0) = T_\eta(X_2, y_0)$ for* every *$\eta \in \mathbb{R}^p$, together with a rotation $g^* \in O(d)$ that mixes columns non-trivially and some $\eta_1 \in \mathbb{R}^p$ for which $T_{\eta_1}(X_1 g^*, y_0) \neq T_{\eta_1}(X_2 g^*, y_0)$. Then the equivariant locus $\{\eta \in \mathbb{R}^p : T_\eta$ is $O(d)$-equivariant$\}$ is closed, nowhere dense, and Lebesgue-null in $\mathbb{R}^p$.*

The proof (Appendix A) hinges on the witness pair: by hypothesis $A(\eta) := T_\eta(X_1, y_0) - T_\eta(X_2, y_0) \equiv 0$ on $\mathbb{R}^p$, while $B(\eta) := T_\eta(X_1 g^*, y_0) - T_\eta(X_2 g^*, y_0)$ is real-analytic in η and non-identically-zero (it does not vanish at η_1). Equivariance forces $E \subseteq \{B = 0\}$, and the Mityagin lemma (Mityagin, 2015) delivers measure-zero. The witness condition is strictly stronger than non-affineness in c: a quadratic-symmetric family such as $\mathrm{Enc}_\eta(c) = \eta(\sum_i c_i)^2$ is non-affine but fails the witness condition on the standard pair (both witness columns sum to zero), whereas encoder families containing a second-moment-sensitive setting satisfy it. ColumnPFN-class GELU encoders fall in this latter family (Lemma 2 in Appendix A).

Minimal architectural conditions.

Definition 2 (Conditions A1–A4)**. (A1) Axis-first factorisation:** a column-wise primitive applied before any cross-column primitive produces $Z \in \mathbb{R}^{N'\times d}$ with $Z_{:,j}$ depending on X only through $X_{:,j}$ and y. **(A2) Non-commuting axis summary:** the per-column map is nonlinear in c (or genuinely label-conditioned). **(A3) Cross-axis interaction:** a downstream cross-column primitive does not factor diagonally over column slots. **(A4) No full $O(d)$-symmetrisation:** no internal layer Haar-averages over $O(d)$ on any internal representation. Symmetrisation over a strict subgroup (e.g. $\mathrm{Sym}(d)$) does not count.

Architecture	A1	A2	A3	A4	Predicted regime	rot_std (median)
TabPFN v1 (Hollmann et al., 2023)	✗	–	–	–	class invariant	–
MLP per-dataset	✗	–	–	–	class invariant	0.0012
FT-Transformer (row-affine) (Gorishniy et al., 2021)	✗	–	–	–	class invariant†	0.0021
TabPFN v2 (Hollmann et al., 2025)	(✓)	✓	✓	(✓)	marginally variant‡	0.0013
TabICL/v2 (Qu et al., 2025; 2026)	(✓)	✓	✓	(✓)	marginally variant‡	–
ColumnPFN (this paper)	✓	✓	✓	✓	strongly variant	0.0311
Scalar ablation (this paper)	✓	✓	✓	✓	strongly variant	0.0441
XGBoost (axis-aligned)	n/a	n/a	n/a	n/a	outside framework§	0.0025

Table 1. Architectural taxonomy along A1–A4 for PFN and neural architectures. Parentheses denote weak satisfaction (A1 at the per-cell rather than per-column level, A4 via random feature-identifier resampling rather than full Haar averaging). † "FT-Transformer (row-affine)" refers to the variant with tokeniser $\phi(x) = xW + b$ followed by a row-wise transformer. Class-invariance follows from Thm. 1 only under this configuration. The standard FT-Transformer with per-feature scalar embeddings and cross-feature attention indexes a d-axis and is outside the scope of Thm. 1. ‡ "Marginally variant" is a heuristic taxonomic label outside the scope of Thm. 3's clauses (i)–(ii), motivated by partial satisfaction of A1/A4 and by random feature-identifier resampling at inference (a strict-subgroup approximate symmetrisation). § XGBoost is included for empirical reference: axis-aligned tree splits do not fit the A1–A4 framework, and a low $\mathrm{rotation\_std}$ here reflects uniform AUROC degradation across rotations rather than architectural invariance. The reported $\mathrm{rotation\_std}$ medians are empirical diagnostics from §3. They should not be read as theorem consequences for XGBoost or TabPFN-v2.

Theorem 3 (Conditional propagation and invariant constructions)**.** *Let $f_{\beta,\eta} = F_\beta \circ T_\eta$ be a PFN built from the primitives of Appendix A, where T_η is the first axis-first column tokeniser and F_β is the downstream stack from tokens to prediction.* (i) *Suppose A1–A4 all hold and T_η satisfies the witness condition of Theorem 2. Assume in addition that the downstream stack is* witness-separating*: for a witness X_1, X_2, y_0, g^*, the output-difference map*

$$D_\beta(\eta) := F_\beta(T_\eta(X_1 g^*, y_0)) - F_\beta(T_\eta(X_2 g^*, y_0))$$

is real-analytic in η and is not identically zero. Then the predictor-equivariant tokeniser locus is contained in the zero set $\{D_\beta = 0\}$, hence is Lebesgue-null and non-generic. Equivalently, $f_{\beta,\eta}$ is rotation-variant for generic η. (ii) *If any of A1–A4 fails, an $O(d)$-invariant predictor is realisable in the same architectural family: ¬A1 by orbit-averaging (Theorem 1), ¬A2 because the rotation passes linearly through the column-wise primitive, ¬A3 by choosing a column-symmetric aggregation, ¬A4 by Haar averaging.*

Proof in Appendix A. Clause (i) is a conditional propagation statement: tokeniser-level non-equivariance need not survive arbitrary downstream processing. Clause (ii) is constructive and architecture-internal: it does not assert that an arbitrary instance violating Ak is itself invariant, only that the family admits an invariant construction. This is what Table 1 reports.

3. Empirical evidence under an operational rotation protocol

We evaluate the architectural distinction of Theorems 2–3 on the strict-Grinsztajn binary subset (Grinsztajn et al., 2022) (numerical-only, study-337, 9 tasks). Per (task, model) we draw 10 deterministic Haar rotations and report the operational diagnostic $\mathrm{rotation\_std}_{t,a} = \mathrm{std}\{\mathrm{AUROC}(g_r; t, a)\}_{r=1}^{10}$. Six architectures are evaluated: per-task 3-layer MLP, XGBoost, FT-Transformer (row-affine) (Gorishniy et al., 2021), TabPFN-v2 (Hollmann et al., 2025) ($\mathsf{n\_estimators} = 8$), and two PFNs retrained on a scaled-geometry stochastic-labeller mixture prior (N up to 1024, P up to 32) after a pre-registered initial study B-failed under context-length OOD (Appendix C): a population-level column-token PFN ("ColumnPFN") and a scalar-pooled column-encoder ablation. The two pretrained models are evaluated with sub-context sampling matched to the scaled-pretraining geometry ($K = 50$ stratified draws). This diagnostic is not a fully disentangled estimate of pure rotation sensitivity for ColumnPFN and Scalar. For those models, the sub-context RNG seed includes the rotation index (Appendix E), so the measured $\mathrm{rotation\_std}$ combines feature-rotation effects with sub-context-resampling variance. We did not run fixed-subcontext or zero-rotation controls in this version. Configurations and per-task numbers are in Appendices D–E.

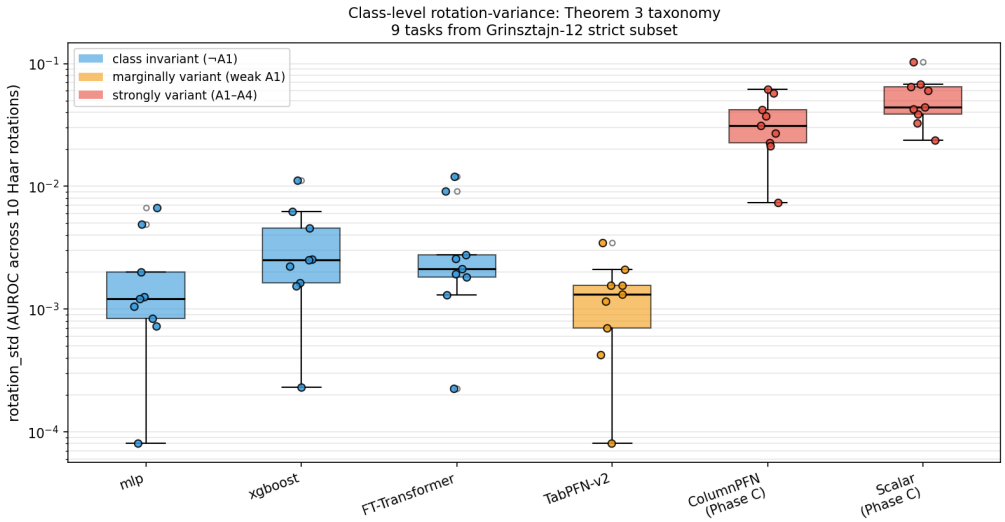

Figure 1. Observed rotation_std diagnostic on 9 strict-Grinsztajn binary tasks (log scale). For ColumnPFN and Scalar, each rotation is evaluated by averaging over $K = 50$ sub-context draws, and the sub-context RNG seed includes the rotation index. Their spread therefore reflects both feature-rotation effects and sub-context-resampling variance. Per-architecture median (range): MLP 0.0012 (0.0001–0.0067), XGBoost 0.0025 (0.0002–0.0112), FT-Transformer (row-affine) 0.0021 (0.0002–0.0119), TabPFN-v2 0.0013 (0.0001–0.0035), ColumnPFN 0.0311 (0.0074–0.0620), Scalar 0.0441 (0.0239–0.1037). The low-std comparison group is separated from the ColumnPFN-style group on $9/9$ tasks under this operational diagnostic.

Bayes-floor and capacity probe. A Monte Carlo estimator (Appendix B) computes the *oracle* irreducible binary cross-entropy of a stochastic-labeller prior by resampling labels at fixed features and scores ($K = 200$ resamples per task, 2,000 tasks per geometry). This is the aleatoric component, $H(Y \mid X, S)$, and lower-bounds any finite-context Bayes risk. Both priors used in this paper have oracle mixture floor $\approx \mathbf{0.39}$ BCE. Small-scale and scaled pretraining checkpoints both plateau $\approx \mathbf{0.22}$ BCE above their respective oracle floors. A $3\times$ capacity probe under a fixed training recipe ($d_{\text{model}} = 192$, 6 column layers, ~ 2.5M parameters, batch size 4 down from 8, 50,000 steps, same lr/seed) reaches eval BCE $\mathbf{0.626}$ versus the standard run's 0.6096 at the matched 50,000-step horizon, so this single-recipe scale-up does not close the gap (Appendix D). Capacity *relevance* is not ruled out under matched or retuned schedules. Only this specific recipe fails to help.

Operational separation. Define the low-std comparison set $\mathcal{C}_{\text{low}} = \{\text{MLP}, \text{XGB}, \text{FT}, \text{TabPFN-v2}\}$ and the ColumnPFN-style set $\mathcal{C}_{\text{col}} = \{\text{ColumnPFN}, \text{Scalar}\}$. On every one of the 9 tasks, $\max_{a \in \mathcal{C}_{\text{low}}} \text{rotation\_std}_{t,a} < \min_{a \in \mathcal{C}_{\text{col}}} \text{rotation\_std}_{t,a}$ ($\mathbf{9/9}$ separations, per-task separation factor $1.9\times$ on `eye_movements` to $37\times$ on `pol`, median $\mathbf{12}\times$). The two groups differ by approximately an order of magnitude in median observed rotation_std: 0.0012–0.0025 for the low-std comparison group versus 0.031–0.044 for the ColumnPFN-style group. XGBoost falls in $\mathcal{C}_{\text{low}}$ by metric, not by mechanism: tree splits are axis-aligned, and a generic rotation degrades AUROC uniformly across rotations rather than producing rotational variance (Table 1 caption). This is empirical evidence consistent with the predicted architectural separation under the reported operational diagnostic, not a pure rotation-only causal estimate.

Exploratory within-group ordering. Within $\mathcal{C}_{\text{col}}$, Scalar has larger observed rotation_std than ColumnPFN on $8/9$ tasks (paired Wilcoxon signed-rank test[1] two-sided $p = 0.0195$, one-sided $p = 0.0098$, mean $\bar{\delta} = +0.0189$). The theory does not predict an ordering within $\mathcal{C}_{\text{col}}$, and this single-seed direction-of-effect should be treated as exploratory.

TabPFN-v2 anomaly. TabPFN-v2 is heuristically classified in our taxonomy as marginally variant (A1 weak at the per-cell level, A4 weak via random feature-identifier resampling) but empirically falls in the low-std comparison group. The official inference path (Hollmann et al., 2025) ensembles predictions over randomly-permuted-and-sign-flipped feature dimensions, an approximate symmetrisation over a finite axis-aligned subgroup of $O(d)$ that we conjecture suppresses rotation_std below what weak A1 alone would predict. Thus "marginally variant" is a heuristic taxonomy label, not a prediction of a large observed rotation_std for TabPFN-v2.

[1] `scipy.stats.wilcoxon`, `mode='auto'`, SciPy 1.17.1, for $n = 9$ with no ties or zero deltas, this is the exact null distribution.

4. Discussion and limitations

What the theory says. Theorems 1–3 separate class-level closure, token-level non-equivariance, and conditional predictor-level propagation. Theorem 1 is a closure baseline, not a fixed-parameter invariance theorem (Proposition 1 rules out the literal fixed-parameter claim). Theorem 2 is the generic tokeniser-level obstruction. Theorem 3 propagates that obstruction to the full predictor only under the stated downstream witness-separation condition. These theorems do not predict $\mathrm{rotation\_std}$ magnitudes or within-group orderings. The empirical study should therefore be read as evidence consistent with the predicted architectural separation under the reported protocol, not as a theorem-level verification of pure rotation sensitivity.

Bayes-floor gap. The single-recipe capacity probe does not support one specific scale-up recipe as a remedy for the ≈ 0.22 BCE gap, but it does not rule out capacity relevance under retuned schedules. The MC estimator measures only the aleatoric component of the prior. Finite-context PFN inference additionally carries an epistemic component (uncertainty over the latent task at $N \leq 1024$ in-context rows), so the achievable finite-context Bayes risk can exceed the oracle floor. The observed gap may therefore reflect finite-context epistemic uncertainty alongside potential optimisation shortfalls, not saturation at the oracle floor. The observed low-std/high-std separation is measured at each architecture's reported checkpoint and does not depend on closing this gap.

Eval-protocol matching. ColumnPFN-class architectures use $K = 50$ stratified sub-context draws per rotation while the row-wise baselines use full-data inference. The sub-context RNG seed is keyed by rotation index, so each rotation draws fresh sub-contexts: the reported $\mathrm{rotation\_std}$ for ColumnPFN and Scalar confounds rotation sensitivity with sub-context-resampling variance. The current experiment shows a large separation in the full operational diagnostic (median ratio $\geq 10\times$, $9/9$ separation), but it does not decompose that separation into rotation-only and resampling components. Three controls would disentangle these components and are left to future work: *(i)* dropping `rotation_index` from the seed so sub-contexts are held fixed across rotations, *(ii)* a zero-rotation $K = 50$ control measuring the resampling-variance floor at $g = I$, and *(iii)* a K-sweep verifying band stability as resampling noise shrinks.

Scope. All six architectures use a single training-data seed per task and a single eval-seed configuration. The within-group direction-of-effect is single-seed and warrants replication. We therefore do not claim seed-stability, broad benchmark coverage, or immediate extension of the empirical findings to multiclass, regression, or mixed categorical/numerical settings. Extending the tokeniser-side analysis beyond the binary numerical setting is left to future work.

References

Arbel, M., Salinas, D., and Hutter, F. Equitabpfn: A target-permutation equivariant prior fitted networks. 2026. URL https://arxiv.org/abs/2502.06684.

Elesedy, B. and Zaidi, S. Provably strict generalisation benefit for equivariant models. 2021. URL https://arxiv.org/abs/2102.10333.

Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. volume abs/2106.11959, 2021. URL https://arxiv.org/abs/2106.11959.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on tabular data? 2022. URL https://arxiv.org/abs/2207.08815.

Hollmann, N., Müller, S., Eggensperger, K., and Hutter, F. Tabpfn: A transformer that solves small tabular classification problems in a second. 2023. URL https://arxiv.org/abs/2207.01848.

Hollmann, N., Müller, S., Purucker, L., et al. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Mityagin, B. The zero set of a real analytic function. 2015. URL https://arxiv.org/abs/1512.07276.

Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. Tabicl: A tabular foundation model for in-context learning on large data. 2025. URL https://arxiv.org/abs/2502.05564.

Qu, J., Holzmüller, D., Varoquaux, G., and Morvan, M. L. Tabiclv2: A better, faster, scalable, and open tabular foundation model. 2026. URL https://arxiv.org/abs/2602.11139.

A. Full proofs of Theorems 1–3

A.1. Architectural primitives and basic definitions

Definition 3 (Architectural primitives)**.** A layer is one of:

(1) **Row-wise:** $f : \mathbb{R}^{d'} \to \mathbb{R}^{d''}$ applied per row, $X'_{i,:} = f(X_{i,:})$.

(2) **Column-wise:** $g : \mathbb{R}^N \times \mathcal{Y}^N \to \mathbb{R}^{N''}$ applied per column, $X'_{:,j} = g(X_{:,j}, y)$.

(3) **Cross-row:** $\Phi : \mathbb{R}^{N\times d'} \to \mathbb{R}^{N''\times d'}$, permutation-equivariant in rows and acting identically across columns (i.e. $\Phi(X)_{:,j}$ depends only on $X_{:,j}$).

(4) **Cross-column:** $\Psi : \mathbb{R}^{N\times d'} \to \mathbb{R}^{N\times d''}$ whose output column j depends nontrivially on more than one input column.

(5) **Pool-then-mix:** $H : \mathbb{R}^{N\times d'} \to \mathbb{R}^{d''}$ collapsing the row dimension before the head's output.

Definition 4 (Rotation invariance and equivariance)**.** A predictor $f : \mathcal{C} \mapsto \mathcal{H}$ is $O(d)$-*invariant* if $f(g \cdot \mathcal{C}) = f(\mathcal{C})$ for all $g \in O(d)$ and all $\mathcal{C}$. A hidden representation $h : \mathcal{C} \mapsto \mathbb{R}^m$ is $O(d)$-*equivariant* if there exists a continuous representation $\rho : O(d) \to GL_m(\mathbb{R})$ – depending on neither $\mathcal{C}$ nor the parameters θ of h – such that $h(g \cdot \mathcal{C}) = \rho(g)\, h(\mathcal{C})$ for all $g, \mathcal{C}$.

The non-dependence of ρ on $\mathcal{C}$ is the substantive part of equivariance. An "X-dependent representation" is not a representation in any useful sense, since it commutes with no symmetry of the architecture and cannot be implemented by any fixed downstream layer.

A.2. The literal fixed-parameter claim is too strong

Proposition 1 (Disproof of fixed-parameter equivariance)**.** *There exist row-wise PFN parameters θ and a rotation $g \in O(d)$ such that no scalar representation $\rho(g)$ independent of $\mathcal{C}$ satisfies $f_\theta(g \cdot \mathcal{C}) = \rho(g)\, f_\theta(\mathcal{C})$ for all $\mathcal{C}$.*

Counterexample. Take $d=2$, label space trivial, row encoder $\phi_\theta(x) = \mathrm{ReLU}(w^\top x)$ with $w = e_1$, identity head, and g the rotation by $90°$. Then $\phi_\theta(x) = \mathrm{ReLU}(x_1)$ and $\phi_\theta(xg) = \mathrm{ReLU}(x_2)$. For $x = (1, -1)$, $\phi_\theta(x) = 1$ and $\phi_\theta(xg) = 0$. For $x = (-1, 1)$ the values exchange. No scalar $\rho(g)$ relates these for both x simultaneously. □

A.3. Theorem 1: class-level $O(d)$-closure

Assumption 1. No feature-identity or feature-position embedding is added to the rows of X or to $x_\star$ before the encoders are applied.

Assumption 2. $\phi_\theta \in \mathcal{F}_{\mathrm{row}}^{(d,d_{\mathrm{model}})}$ and $\phi_\theta^q \in \mathcal{F}_{\mathrm{row}}^{(d,d_{\mathrm{model}})}$.

Assumption 3. The trunk Ψ_θ and head are d-blind.

Definition 5 (Rotation reparameterisation)**.** For $g \in O(d)$ and $\theta = (W, b, \theta_\eta, W^q, b^q, \theta_{\eta^q}, \theta_\Psi, \theta_{\mathrm{Head}})$, define

$$\theta^{(g)} = (g^{-1}W,\ b,\ \theta_\eta,\ g^{-1}W^q,\ b^q,\ \theta_{\eta^q},\ \theta_\Psi,\ \theta_{\mathrm{Head}}).$$

Only the encoder first-layer weights change.

Proof of Theorem 1. Step 1 (row tokens). By Assumption 2, $\phi_\theta(x, y) = \eta(xW + b, y)$. Under $\theta \mapsto \theta^{(g)}$ the encoder weight is $g^{-1}W$ and (θ_η, b) are preserved. For row $X_i g$ of $g \cdot \mathcal{C}$,

$$\phi_{\theta^{(g)}}(X_i g, y_i) = \eta((X_i g)g^{-1}W + b, y_i) = \eta(X_i W + b, y_i) = \phi_\theta(X_i, y_i),$$

using $gg^{-1} = I$ ($g \in O(d)$). The same calculation gives $\phi^q_{\theta^{(g)}}(x_\star g) = \phi^q_\theta(x_\star)$.

Step 2 (trunk and head). By Definition 5, $\Psi_{\theta^{(g)}} = \Psi_\theta$ and $\mathrm{Head}_{\theta^{(g)}} = \mathrm{Head}_\theta$ as functions. By Step 1, the input tuple to Ψ on $g \cdot \mathcal{C}$ under $\theta^{(g)}$ equals the input tuple to Ψ on $\mathcal{C}$ under θ. Same function on same input, same output, so $f_{\theta^{(g)}}(g \cdot \mathcal{C}) = f_\theta(\mathcal{C})$.

Symmetrised predictor. Continuity on the compact group $O(d)$ gives existence of $\bar{f}(\mathcal{C}) := \int_{O(d)} f_{\theta^{(g)}}(\mathcal{C})\, d\mu(g)$. Theorem 1 rewrites as $f_{\theta^{(g)}}(\mathcal{C}') = f_\theta(g^{-1} \cdot \mathcal{C}')$, so $\bar{f}$ averages the original predictor over the orbit. Left-invariance of Haar measure delivers $O(d)$-invariance. Jensen's inequality combined with isotropy of the prior delivers the risk bound along the lines of (Elesedy & Zaidi, 2021, Thm. 5). □

A.4. Theorem 2: generic non-equivariance

Lemma 1 (Witness pair)**.** *Let $d=2$, $N=3$, $k=2$, and $\mathrm{Enc}(c) = (\sum_i c_i, \sum_i c_i^2)$ (label-free). Let g be the 45° rotation in $O(2)$, and take*

$$X_1 = \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 0 & 0 \end{pmatrix}, \qquad X_2 = \begin{pmatrix} 1 & -1 \\ -1 & 1 \\ 0 & 0 \end{pmatrix}.$$

Then $T(X_1) = T(X_2) = (0, 2, 0, 2)$ but $T(X_1 g) = (0, 4, 0, 0) \neq (0, 0, 0, 4) = T(X_2 g)$. No fixed $\rho(g)$ relates these.

Proof. Each column of X_2 is a row-permutation (swap of rows 1, 2) of the corresponding column of X_1, and Enc is permutation-invariant in the row index, so $T(X_1) = T(X_2)$. After rotation, the cross-column term $p = \sum_i c_{1,i} c_{2,i}$ enters the second-moment coordinate of T, and $p(X_1) = 2 \neq -2 = p(X_2)$, so $T(X_1 g) \neq T(X_2 g)$. If a fixed $\rho(g)$ realised equivariance, $T(X_1) = T(X_2)$ would force $T(X_1 g) = T(X_2 g)$ – contradiction. □

Proof of Theorem 2. Step A. The witness condition supplies $X_1, X_2, y_0, g^*, \eta_1$. Define

$$A(\eta) = T_\eta(X_1, y_0) - T_\eta(X_2, y_0), \quad B(\eta) = T_\eta(X_1 g^*, y_0) - T_\eta(X_2 g^*, y_0).$$

Both are real-analytic on $\mathbb{R}^p$ valued in $\mathbb{R}^{kd}$.

Step B. The witness condition's first clause states $T_\eta(X_1, y_0) = T_\eta(X_2, y_0)$ for every η, so $A \equiv 0$. The second clause supplies η_1 with $T_{\eta_1}(X_1 g^*, y_0) \neq T_{\eta_1}(X_2 g^*, y_0)$, i.e. $B(\eta_1) \neq 0$. Hence B is a non-identically-zero real-analytic function on $\mathbb{R}^p$.

Step C. If T_η is equivariant, the necessary condition $T_\eta(X_1, y) = T_\eta(X_2, y) \Rightarrow T_\eta(X_1 g, y) = T_\eta(X_2 g, y)$ holds. Combined with $A \equiv 0$, this gives the equivariant locus $E \subseteq \{B = 0\}$. By (Mityagin, 2015), the zero set of a non-identically-zero real-analytic function on $\mathbb{R}^p$ is closed, nowhere dense, and Lebesgue-null. □

Remark 1 (Lemma 1 as a witness pair). The pair (X_1, X_2) in Lemma 1 satisfies the witness condition's first clause for any row-permutation-invariant Enc_η: each column of X_2 is a row-permutation of the corresponding column of X_1, so $\mathrm{Enc}_\eta(X_{1,:,j}, y_0) = \mathrm{Enc}_\eta(X_{2,:,j}, y_0)$ columnwise and hence $T_\eta(X_1, y_0) = T_\eta(X_2, y_0)$ for every η. The witness condition is therefore reduced to producing some η_1 at which $T_{\eta_1}(X_1 g^*, y_0) \neq T_{\eta_1}(X_2 g^*, y_0)$, which the sum-of-squares example in Lemma 1 exhibits and Lemma 2 below verifies for the ColumnPFN encoder family.

Remark 2 (Witness condition vs. non-affineness). The witness condition is strictly stronger than non-affineness in c. The family $\mathrm{Enc}_\eta(c) = \eta(\sum_i c_i)^2$ is real-analytic, permutation-invariant, and non-affine for $\eta \neq 0$, but every witness column from Lemma 1 sums to zero, so $A \equiv B \equiv 0$ on this family and the obstruction argument does not apply. The witness condition holds for the standard pair whenever the encoder family separates $(\sqrt{2}, -\sqrt{2}, 0)$ from $(0, 0, 0)$ at some parameter. In particular for any encoder sensitive to the second moment $\sum_i c_i^2$. ColumnPFN-class encoders are MLPs over $\sum_i c_i^2$ and related second-moment statistics, and satisfy the witness condition.

Remark 3 (Activation-class scope). For an MLP Enc_η with real-analytic activations (tanh, sigmoid, GELU), $\eta \mapsto \mathrm{Enc}_\eta(c, y)$ is real-analytic globally on $\mathbb{R}^p$ and Theorem 2 applies verbatim. ColumnPFN-class encoders use GELU and fall in this case. For piecewise-polynomial activations (ReLU, leaky ReLU), Enc_η is real-analytic only on each open semi-algebraic region of parameter space, and the single-witness obstruction can vanish identically on regions whose activation pattern places the witness pair into a "dead" subgraph. A region-by-region argument requires a witness pair tailored to each activation pattern. We do not make that formal claim here.

Proposition 2 (Token-to-output witness separation)**.** *The following proposition supplies the downstream non-collapse condition used in Theorem 3. Let T_η be a column tokeniser satisfying the conclusion of Theorem 2, and let $f_{\beta,\eta} = F_\beta \circ T_\eta$ be the composed predictor. For a witness X_1, X_2, y_0, g^*, define*

$$D_\beta(\eta) := F_\beta(T_\eta(X_1 g^*, y_0)) - F_\beta(T_\eta(X_2 g^*, y_0)).$$

If D_β is real-analytic and not identically zero, then the predictor-equivariant tokeniser locus is contained in $\{D_\beta = 0\}$, so predictor-level non-equivariance holds for generic η. A sufficient condition is that F_β be injective on the relevant witness-token image at some parameter value. Conversely, if F_β collapses the witness images to the same output, token-level non-equivariance need not survive to the predictor. Such collapse can, for example, arise from a column-symmetric aggregation.

Lemma 2 (ColumnPFN encoder satisfies the witness condition)**.** *Let* $\mathrm{Enc}_\eta^{\mathrm{ColPFN}}(c, \hat{y}) := \mathrm{MLP}_\eta\big(\big[\, c\,;\, \hat{y}\,;\, \mu(c)\,;\, \sigma^2(c)\,;\, c \odot \hat{y}\,\big]\big)$ *where* $\mu(c) = N^{-1}\sum_i c_i$ *and* $\sigma^2(c) = N^{-1}\sum_i c_i^2 - \mu(c)^2$ *are the empirical mean and variance of the column,* $[\,\cdot\,]$ *denotes concatenation and* MLP_η *is a feed-forward network with at least one hidden layer and a non-affine real-analytic activation (GELU). For the standard pair* (X_1, X_2) *of Lemma 1 and* $g^* = 45°$ *rotation in the first two columns,* $\mathrm{Enc}_\eta^{\mathrm{ColPFN}}$ *satisfies the witness condition of Theorem 2.*

Proof. The first clause holds by Remark 1: every column-symmetric statistic in $\mathrm{Enc}^{\mathrm{ColPFN}}$ is invariant to row permutations. For the second clause, take η_1 with identity first-layer weights on the σ^2 summary input. Then $\mathrm{Enc}_{\eta_1}^{\mathrm{ColPFN}}\big((\sqrt{2}, -\sqrt{2}, 0)^\top, y_0\big)$ has variance coordinate $\sigma^2 = \frac{4}{3}$, while $\mathrm{Enc}_{\eta_1}^{\mathrm{ColPFN}}\big((0,0,0)^\top, y_0\big)$ has variance coordinate 0. These two columns appear after rotation in $X_1 g^*$ and $X_2 g^*$ respectively (Lemma 1), so the column tokens differ at η_1 and hence $T_{\eta_1}(X_1 g^*, y_0) \neq T_{\eta_1}(X_2 g^*, y_0)$. □

The Scalar ablation, which pools the column encoder to a single scalar via a sum-of-squares MLP, satisfies the witness condition by the same argument (the variance coordinate alone suffices).

A.5. Theorem 3: minimal A1–A4 conditions

Proof of Theorem 3. Clause (i), all of A1–A4 hold. By A1, the architecture has a column-wise primitive applied before any cross-column primitive, producing axis-first tokens T_η. By A2 and the witness condition assumed in the theorem (verified for ColumnPFN-class encoders in Lemma 2), Theorem 2 gives a Lebesgue-null non-generic equivariant locus for T_η. The additional witness-separation assumption in Theorem 3 states that the composed output difference $D_\beta(\eta)$ is real-analytic and not identically zero. If the full predictor were $O(d)$-equivariant at such an η, the witness argument would force $D_\beta(\eta) = 0$. Hence the predictor-equivariant tokeniser locus is contained in the zero set of a non-identically-zero real-analytic function and is Lebesgue-null and non-generic. A3 supplies the architectural cross-column interaction needed for such a downstream witness separation to be available, but A3 alone is not an injectivity assumption. By A4, no internal layer Haar-averages over $O(d)$, so the witness separation is not erased by full symmetrisation.

Clause (ii), recoverability under each failure case. The four cases construct an $O(d)$-invariant predictor in the architectural family, not a claim that arbitrary instances of the family are themselves invariant.

Case ¬A1. If no column-wise primitive precedes the cross-column primitives, the architecture is built from row-wise, cross-row, and pool-then-mix primitives only. Class-level closure (Theorem 1) applies and the orbit-averaged predictor is $O(d)$-invariant.

Case ¬A2. If the per-column map is linear, $g(c, y) = M_y c$, then $g(X g_{\mathrm{rot}}, y) = M_y X g_{\mathrm{rot}} = g(X, y) g_{\mathrm{rot}}$. The rotation passes through the column-wise primitive unchanged. The resulting representation can be processed by a d-blind row-wise/pool-then-mix tail, reducing to the ¬A1 case.

Case ¬A3. If the cross-column primitive factors diagonally as $\sum_j \pi(g(X_{:,j}, y))$ for some shared π, choose $\pi \circ g$ to compute an $O(d)$-joint invariant of the columns (e.g. the Frobenius norm $\sum_j \|X_{:,j}\|_2^2 = \|X\|_F^2$). The resulting predictor is exactly $O(d)$-invariant.

Case ¬A4. If an internal layer Haar-averages over $O(d)$, the layer's output is $O(d)$-invariant: under the right action $g_0 \cdot z = z g_0$ and using unimodularity of $O(d)$,

$$L(g_0 \cdot z) = \int_{O(d)} m(g \cdot (g_0 \cdot z))\, d\mu(g) = \int_{O(d)} m((g_0 g) \cdot z)\, d\mu(g) = L(z),$$

by left-invariance of μ (equivalently, right-invariance, as $O(d)$ is unimodular). Composition with the rest of the architecture inherits invariance. □

B. Bayes-floor methodology

Definition. Let Π be a stochastic-labeller prior over binary tasks: each sample comprises a latent task variable s (the underlying score mechanism, e.g. the SCM weights or tree-stump split) and an observation x. Let $p(y \mid x, s)$ be the labelling distribution *given the latent task* s. The *oracle (aleatoric) binary cross-entropy* under Π is

$$\mathrm{BCE}^\star(\Pi) := \mathbb{E}_{(s,x)\sim\Pi}[H(p(y \mid x, s))], \quad H(q) = -q \log q - (1-q)\log(1-q).$$

This is the expected per-example labeller entropy *conditioned on the true task* – i.e. the irreducible noise an oracle that knew s would incur. It lower-bounds the BCE achievable by any predictor at any compute budget. The achievable *finite-context* Bayes risk at N in-context rows additionally includes the epistemic uncertainty over s given a length-N context, and is strictly larger than $\mathrm{BCE}^\star(\Pi)$. A trained model whose held-out BCE sits substantially above $\mathrm{BCE}^\star(\Pi)$ may have not fit the prior, may be encountering finite-context epistemic uncertainty, or both. Note that the MC estimator below correctly fixes both `task.X` and `task.s` across resamples and varies only the labeller stochasticity, computing $H(p(y \mid x, s))$ rather than the strictly-zero-shot $H(p(y \mid x))$.

Monte Carlo estimator (pseudo-Python). We estimate the mixture Bayes floor by Monte Carlo sampling over the labeller stochasticity, fixing features and the underlying latent score across resamples.

```
def mc_irreducible_bce(task, K=200, seed=0):
    """Per-task irreducible BCE under stochastic labelling.

    task.X    : (N, P) design matrix, FIXED across resamples
    task.s    : (N,)   score values, FIXED across resamples
    task.label_fn(s, rng) : returns y in {0,1}^N, may be
                            stochastic (logistic) or
                            deterministic (threshold).
    """
    rng = np.random.default_rng(seed)
    p_hat = np.zeros(task.X.shape[0])
    for k in range(K):
        y_k = task.label_fn(task.s, rng)
        p_hat += y_k
    p_hat /= K
    eps = 1e-6
    p_hat = np.clip(p_hat, eps, 1.0 - eps)
    H = -(p_hat * np.log(p_hat) +
          (1 - p_hat) * np.log(1 - p_hat))
    return H.mean()
```

Methodological note (transferable). Under the MC estimator above with $K = 200$ resamples and 2,000 tasks per geometry, the mixture floor is $\mathbf{0.395}$ for the small-scale prior (fixed $N = 64$, $P = 10$) and $\mathbf{0.383}$ for the scaled-pretraining prior (variable N, P). This subtlety is transferable to other PFN papers that use threshold-labelled or labeller-mixture synthetic priors: the irreducible BCE depends on the labeller's stochasticity profile across the prior's full support, not on the labeller's stochasticity given a single realisation.

Family	Tasks	Mean MC floor	Std (across tasks)
Small-scale prior (fixed $N = 64$, $P = 10$)			
SCM	762	0.392	0.056
tree_stump	826	0.397	0.066
rotated_hard_negative	412	0.398	0.065
Mixture floor		**0.395**	
Scaled-pretraining prior (variable N, P)			
SCM	821	0.364	0.062
tree_stump	780	0.395	0.067
rotated_hard_negative	399	0.397	0.066
Mixture floor		**0.383**	

Table 2. Monte Carlo Bayes-floor estimates ($K = 200$ label resamples per task, 2,000 tasks per geometry) for the two pretraining priors used in this paper. Both priors have mixture floor close to 0.39 BCE. Trained models in small-scale and scaled pretraining plateau ≈ 0.22 BCE above their respective floors.

Quantitative implication. The small-scale pretraining plateau of 0.638 (ColumnPFN) sits $0.638 - 0.395 = \mathbf{0.243}$ BCE above the small-scale oracle floor. The scaled-pretraining plateau of 0.6059 sits $0.6059 - 0.383 = \mathbf{0.223}$ BCE above the scaled-pretraining oracle floor. Because the MC estimator measures only the aleatoric component of the prior, the achievable finite-context Bayes risk at $N \leq 1024$ in-context rows is strictly above the oracle floor: a perfectly trained finite-context model would still incur the epistemic uncertainty over the latent task. These gaps therefore reflect the expected epistemic uncertainty of finite-context inference, alongside potential residual optimisation shortfalls, rather than true saturation at the oracle prior's irreducible noise.

C. Small-scale pretraining and pre-registered initial rotation experiment

Small-scale pretraining configuration. Pretraining prior is a stochastic-labeller mixture $\Pr[\text{SCM}] = 0.4$, $\Pr[\text{tree\_stump}] = 0.4$, $\Pr[\text{rotated\_hard\_negative}] = 0.2$ at fixed $N = 64$, $P = 10$, single_eval_pos $= 32$ (binary classification). Both architectures: $50{,}000$ steps, batch size 8, $\text{lr} = 3 \times 10^{-4}$, seed 0, single A100-80GB. ColumnPFN: $d_{\text{model}} = 128$, num_heads $= 8$, num_column_layers $= 4$, set_encoder_depth $= 2$, $912{,}130$ parameters. Scalar ablation: matched-capacity at scalar_mlp_hidden $= 2{,}048$, $915{,}202$ parameters (within 0.34% of ColumnPFN).

Pretraining plateaus. ColumnPFN: eval BCE $\mathbf{0.638}$. Scalar: $\mathbf{0.640}$. Train–eval gap is 0.002 across the run for both. An LR sweep over $\{1 \times 10^{-4}, 3 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}\}$ at step $5{,}000$ gives ColumnPFN eval BCE in $[0.6521, 0.6550]$ (range 0.0029) and Scalar in $[0.6417, 0.6551]$ (range 0.0134, the wider band driven by slower escape from initialisation at $\text{lr} = 1 \times 10^{-4}$).

Model	LR	best eval BCE	final eval BCE	step
ColumnPFN	1e-04	0.6550	0.6550	5000
ColumnPFN	3e-04	0.6547	0.6547	5000
ColumnPFN	1e-03	0.6521	0.6521	5000
ColumnPFN	3e-03	0.6544	0.6544	5000
Scalar	1e-04	0.6550	0.6550	5000
Scalar	3e-04	0.6417	0.6417	5000
Scalar	1e-03	0.6454	0.6454	5000
Scalar	3e-03	0.6550	0.6551	5000

Table 3. Small-scale pretraining LR sweep at 5,000 steps. The plateau is broadly consistent across the tested $30\times$ LR range: ColumnPFN spans 0.0029 BCE (a flat plateau), while Scalar spans 0.0134 in a U-shape with the optimum at $\text{lr} = 3 \times 10^{-4}$ (eval BCE 0.6417) and similar high BCE at the extremes (0.6550 at $\text{lr} = 1 \times 10^{-4}$ and 0.6551 at $\text{lr} = 3 \times 10^{-3}$). The broad consistency on ColumnPFN is evidence that the plateau is a learning rather than an LR-tuning artifact. Scalar's wider band, and specifically its weaker $\text{lr} = 1 \times 10^{-4}$ result, indicates the model is still actively traversing the early loss landscape at that LR rather than resting on a converged plateau. Full $50{,}000$-step runs at $\text{lr} = 3 \times 10^{-4}$ reach 0.638 (ColumnPFN) and 0.640 (Scalar).

The OOD confound. Applying the small-scale pretraining checkpoints (pretrained at $N = 64$, single_eval_pos $= 32$) to the Grinsztajn study-337 strict-numerical 9-task subset, the unrotated ColumnPFN AUROC averages $\mathbf{0.611}$, and $\mathbf{3/9}$ tasks fall below random (AUROC < 0.5). The sub-context evaluator passes up to $N = 1024$ rows per forward pass (matched to the eventual scaled-pretraining geometry). Pretraining at 32 in-context rows versus eval at up to 1024 in-context rows is a $32\times$ context-length shift. No candidate single_eval_pos appearing in the small-scale prior is within an order of magnitude of the deployment context length. Pretraining $P = 10$ vs. evaluation $P \in [7, 419]$ is up to $42\times$ feature-count OOD on the broader Grinsztajn-12 set. Under these conditions per-task AUROC variance is dominated by prediction instability rather than by rotation-induced signal. The scaled pretraining of Appendix D (N up to 1024) was designed to cover the deployment context-length regime.

Pre-registered hypotheses for the initial rotation experiment.

- **H1** (architectural rotation-variance): ColumnPFN exhibits a paired AUROC drop > 0.005 relative to the unrotated baseline, larger than Scalar's drop, with paired Wilcoxon $p < 0.10$.
- **H2** (TFM control): ColumnPFN's AUROC drop on TabPFN-rotation-sensitive tasks exceeds TabPFN-v2's drop on the same tasks by mean ≥ 0.005.
- **H3** (performance floor): ColumnPFN within 0.05 unrotated AUROC of Scalar.

Outcomes (pre-registered B-fail). H1 **rejected**: paired Wilcoxon signed-rank, $\Delta = -\mathbf{0.0037}$, $p = \mathbf{0.79}$ (sign reversed, far from significance). H2 **rejected**: $\Delta = -\mathbf{0.035}$, $p = \mathbf{0.29}$ (ColumnPFN's rotation drop is *smaller* than TabPFN-v2's, contradicting H2's prediction that ColumnPFN should drop more). H3 **supported**: $\Delta = +\mathbf{0.008}$. Pre-registered verdict: **B-fail**. The diagnosis (OOD confound) was reached after the verdict was recorded and is reported as a diagnosis, not a recovery. This motivated the scaled pretraining of Appendix D, which retrains the architectures on a prior whose context-length distribution covers the Grinsztajn-9 deployment regime.

Pre-registration document. The pre-registration document fixes the H1–H3 thresholds, the statistical tests (paired Wilcoxon $p < 0.10$, mean difference threshold 0.005 for H1, mean drop comparison restricted to TabPFN-rotation-sensitive tasks for H2, 0.05 AUROC tolerance for H3), and the decision rubric (B-pass, B-mixed, B-mixed-2, B-fail). It also pre-registers the compute environment: a hard CUDA device check, a per-model device self-check banner, the TabPFN ensemble budget ($n_{\text{estimators}} = 8$), and the multi-seed inference protocol. All thresholds and decision rules were fixed before any rotation experiment was run.

D. Scaled pretraining and capacity-probe configurations

Scaled-pretraining configuration. Same prior mixture and labellers as the small-scale pretraining. Context length is randomised: $N \sim \mathsf{Cat}(\{128, 256, 512, 768, 1024\}; (0.1, 0.2, 0.4, 0.2, 0.1))$. Feature count is uniform over $\{6, 8, 10, 12, 16, 20, 24, 32\}$. The in-context split position single_eval_pos is sampled per task uniformly over the candidate set $\{N/4, N/2, 3N/4\}$ subject to a ≥ 32 rows-per-side constraint (full sampling logic below). Both architectures: 100,000 steps, batch size 8, $\text{lr} = 3 \times 10^{-4}$, seed 0, parameter counts unchanged from the small-scale pretraining.

Sample-geometry sampling logic (verbatim).

```
def sample_geometry(rng):
    N = rng.choice([128, 256, 512, 768, 1024],
                   p=[0.1, 0.2, 0.4, 0.2, 0.1])
    P = rng.choice([6, 8, 10, 12, 16, 20, 24, 32])
    candidates = [N // 4, N // 2, (3 * N) // 4]
    valid = [c for c in candidates
             if c >= 32 and (N - c) >= 32]
    if valid:
        single_eval_pos = rng.choice(valid)
    else:
        # Defensive fallback. Unreachable for the candidate
        # geometries above: even at N=128 the candidates
        # {32, 64, 96} all satisfy the >=32 constraint on
        # both sides. Kept in case the candidate set is
        # tightened in future runs.
        single_eval_pos = N // 2
    return N, P, single_eval_pos
```

The fallback path is unreachable for any geometry $N \in \{128, 256, 512, 768, 1024\}$ used here. For $N = 128$, the candidates $\{32, 64, 96\}$ all satisfy $\text{candidate} \geq 32$ and $N - \text{candidate} \geq 32$. The branch is retained defensively in case the candidate set is changed.

Scaled-pretraining plateaus. ColumnPFN: best eval BCE $\mathbf{0.6059}$ at step 100,000, midpoint 0.6096 at step 50,000, second-half improvement $\Delta_{\text{eval}} = -0.0037$. Scalar: best $\mathbf{0.6105}$ at step 100,000, midpoint 0.6119, $\Delta_{\text{eval}} = -0.0014$ (Figure 2).

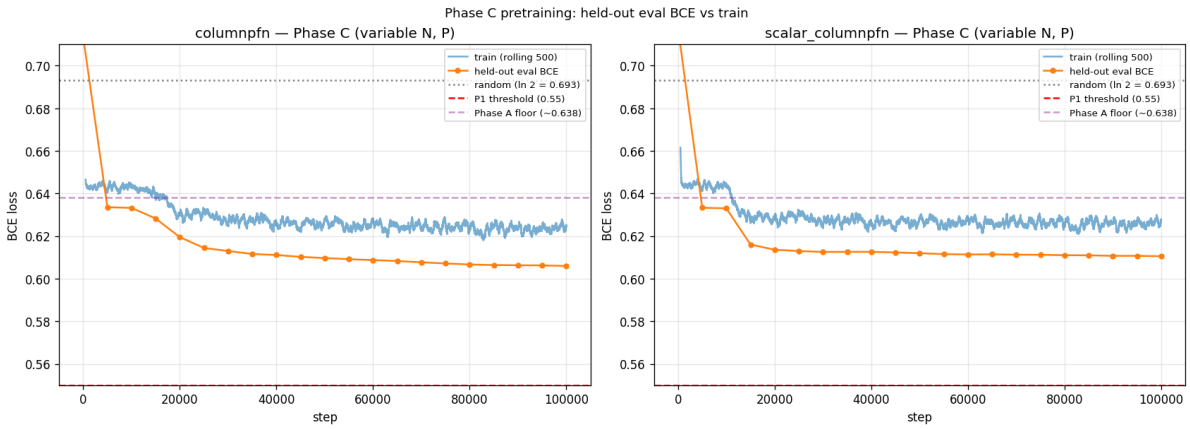

Figure 2. Scaled pretraining (variable N, P) over 100,000 steps: rolling-500 train BCE (blue) and held-out eval BCE (orange) for ColumnPFN (left) and Scalar (right). Both plateau ≈ 0.22 BCE above the scaled-pretraining MC mixture floor 0.383. The Scalar plateau is 0.0046 higher than ColumnPFN's at step 100,000. The pre-registered P1 threshold 0.55 and the small-scale pretraining plateau ~ 0.638 are shown as reference lines.

Pre-registered decision gate (verbatim).

```
ColumnPFN:
  best eval BCE = 0.6059 at step 100000
  final eval BCE = 0.6059
  midpoint (50k) eval BCE = 0.6096
  improvement (final - midpoint) = -0.0037

Scalar:
  best eval BCE = 0.6105 at step 100000
  final eval BCE = 0.6105
  midpoint (50k) eval BCE = 0.6119
  improvement (final - midpoint) = -0.0014

P1 (best <= 0.55):                FAIL  (best = 0.6059)
P2 (|col - scalar| <= 0.03):      PASS  (gap = 0.0046)
P3 (improvement <= -0.005):       FAIL  (worst impr = -0.0014)

VERDICT: FAIL -- criteria failed: P1, P3.
```

Capacity probe (negative result). A 3× larger ColumnPFN ($d_{\text{model}} = 192$, num_column_layers $= 6$, set_encoder_depth $= 2$, $\sim$2.5M parameters, batch size 4 down from 8 due to activation memory at larger N, 50,000 steps, same prior, same lr, same seed) terminates at eval BCE **0.626**, **worse than the 912k-param standard scaled-pretraining run** (0.6059 at 100k, 0.626 already at $\approx$ 20k steps in the small run). Capacity is not the bottleneck within the 50,000-step horizon.

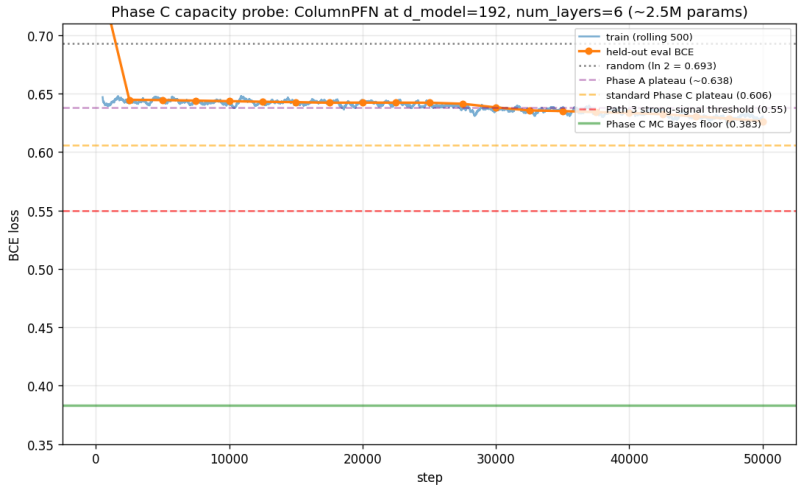

Figure 3. Capacity probe: ColumnPFN at $d_{\text{model}} = 192$, num-layers $= 6$ ($\sim$ 2.5M parameters), 50,000 steps. Eval BCE plateaus at ~ 0.626, worse than the 912k-param standard scaled-pretraining run plateau (0.606). Reference lines: random ($\ln 2$), small-scale pretraining plateau (~ 0.638), standard scaled-pretraining plateau (0.606), pre-registered P1 threshold (0.55), scaled-pretraining MC Bayes floor (0.383).

Future-work attribution. Architecture-orthogonal axes plausibly responsible for the residual **0.223** BCE Bayes-floor gap: LR schedule (cosine vs. constant), warm-up duration, optimiser choice (β_2, EMA), training horizon ($> 10^5$ steps), and curriculum on (N, P). None were varied in this paper. Closing the gap is left to future work.

E. Rotation experiment protocol details

Task	MLP	XGB	FT-Tr	TabPFN-v2	ColumnPFN	Scalar
credit	0.0010	0.0062	0.0026	0.0016	0.0419	0.0387
electricity	0.0020	0.0016	0.0018	0.0013	0.0620	0.0677
pol	0.0001	0.0002	0.0002	0.0001	0.0074	0.1037
house_16H	0.0012	0.0015	0.0019	0.0007	0.0576	0.0605
MagicTelescope	0.0008	0.0025	0.0021	0.0012	0.0375	0.0647
bank-marketing	0.0012	0.0025	0.0028	0.0015	0.0311	0.0428
eye_movements	0.0049	0.0112	0.0119	0.0035	0.0226	0.0239
Diabetes130US	0.0007	0.0022	0.0013	0.0004	0.0270	0.0329
Bioresponse	0.0067	0.0046	0.0091	0.0021	0.0213	0.0441
median	0.0012	0.0025	0.0021	0.0013	0.0311	0.0441

Table 4. **Per-task rotation_std** (AUROC standard deviation across 10 Haar rotations) for 6 architectures on the strict-Grinsztajn 9-task subset. The low-std comparison group (MLP, XGBoost, FT-Transformer (row-affine), TabPFN-v2) is separated from the ColumnPFN-style group (ColumnPFN, Scalar) on every task: $\max_{\mathcal{C}_{\text{low}}} \text{rotation\_std} < \min_{\mathcal{C}_{\text{col}}} \text{rotation\_std}$ on $9/9$ tasks. ColumnPFN and Scalar are both scaled-pretraining checkpoints evaluated with sub-context sampling matched to the scaled-pretraining geometry ($K = 50$ averaging draws), so their rotation_std includes both rotation effects and sub-context-resampling variance.

Per-task within-class deltas. $\delta_t = \text{rotation\_std}_{t,\text{Scalar}} - \text{rotation\_std}_{t,\text{ColumnPFN}}$ per-task: `credit` -0.0033, `electricity` $+0.0056$, `pol` $+0.0963$, `house_16H` $+0.0029$, `MagicTelescope` $+0.0271$, `bank-marketing` $+0.0117$, `eye_movements` $+0.0013$, `Diabetes130US` $+0.0059$, `Bioresponse` $+0.0228$. Mean $\bar{\delta} = +\mathbf{0.0189}$, Scalar $>$ ColumnPFN on $\mathbf{8/9}$ tasks (only `credit` reverses).

Hypothesis-test protocol. All hypothesis tests in this paper use `scipy.stats.wilcoxon` with `mode='auto'` and SciPy version 1.17.1. For $n = 9$ paired samples with no ties or zero deltas, `mode='auto'` resolves to the exact null distribution, so the reported p-values are exact finite-sample p-values rather than asymptotic-normal approximations.

Sub-context sampler (matched to scaled-pretraining geometry). For each (task, ColumnPFN-class model, rotation $r \in \{1, \ldots, 10\}$, eval seed s), the sub-context evaluator does the following:

(1) Draw $K = 50$ stratified sub-contexts. Each sub-context samples $N \in \{128, 256, 512, 768, 1024\}$ from the scaled-pretraining distribution (probabilities $0.1, 0.2, 0.4, 0.2, 0.1$), then samples N rows from the task's training split with class proportions matched to the task to within ± 1 row per class.

(2) For each sub-context, fit the model on the sub-context, predict probabilities on the full test split, accumulate.

(3) Average probabilities across the K draws and compute AUROC on the full test split using the averaged probabilities.

(4) Apply rotation g_r to features (training, sub-context, and test rows simultaneously) and repeat to obtain $\mathrm{AUROC}(g_r)$.

(5) Define $\mathrm{rotation\_std} := \mathrm{std}\{\mathrm{AUROC}(g_r)\}_{r=1}^{10}$.

The full RNG seed for the per-(task, rotation, eval-seed) draws is `np.random.SeedSequence([subcontext_seed, task_id, rotation_index+1, eval_seed])`, ensuring exact reproducibility. Because the seed includes `rotation_index`, the reported ColumnPFN/Scalar $\mathrm{rotation\_std}$ changes both the rotation and the sampled sub-contexts across r. The current protocol does not decompose these two sources of variability.

Architectural classification by A1–A4 (justification).

- **MLP (per-task).** Each row is a flat vector input to a shared MLP. No column-specific representation. A1 fails. Class invariant by Theorem 1.
- **XGBoost.** Tree splits are axis-aligned in feature space. XGBoost is not a member of the A1–A4 architectural taxonomy (no column-token representation). It is listed in Table 1 as "outside framework" rather than as a class-invariant TFM. Its low empirical $\mathrm{rotation\_std}$ (0.0025 median) reflects *uniform* AUROC degradation under generic rotation (rotated splits no longer align with feature axes, so the rotated predictor degrades to a low but consistent AUROC across rotations) rather than architectural invariance. The $\mathcal{C}_{\mathrm{inv}}$ vs. $\mathcal{C}_{\mathrm{var}}$ contrast in §3 therefore relies on $\mathrm{rotation\_std}$ as a diagnostic of *rotational variance*, not as a measure of true invariance.
- **FT-Transformer (row-affine).** The variant evaluated here uses a per-row affine tokeniser $\phi(x) = xW + b$ followed by a row-wise transformer. Under this configuration A1 fails (no population-level column statistics in the tokeniser) and class invariance follows from Theorem 1. The empirical $\mathrm{rotation\_std}$ (0.0021 median) is consistent with this. The standard FT-Transformer of (Gorishniy et al., 2021) uses per-feature scalar embeddings followed by attention across the resulting feature tokens, which indexes a d-axis and does not satisfy the d-blindness hypothesis of Theorem 1. We do not evaluate that variant here, and use the qualifier "(row-affine)" in Table 1 and Table 4 to flag the ablation.
- **TabPFN-v2.** Per-cell tokens with random feature-identifier resampling and alternating-axis attention. A1 holds at the per-cell rather than per-column level (each cell is a function of $X_{i,j}$ rather than $X_{:,j}$), and A4 is approximately satisfied via random feature-identifier resampling at inference, which approximates strict-subgroup symmetrisation. Heuristically classified in our taxonomy as marginally variant. Empirically falls in the class-invariant band – marginally variant is a taxonomic label, not a binary verdict from Theorem 3.
- **ColumnPFN.** Population-level column tokens followed by cross-column attention. A1, A2, A3, A4 all hold strictly. Strongly variant.
- **Scalar ablation.** Identical architecture to ColumnPFN except the column encoder is collapsed to a scalar pooling of column statistics. A1, A2, A3, A4 all hold strictly. Strongly variant.