CHARACTERIZING THE SEPARABILITY GAP IN PRIOR-DATA FITTED NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior-data fitted networks (PFNs) are transformers pre-trained on simulated datasets to approximate the posterior predictive distribution (PPD) of a task prior. Evaluating this approximation outside conjugate settings has remained challenging: for the priors of current PFNs the posterior is intractable, and real-world datasets lack ground-truth PPDs. A second difficulty is conceptual, as the standard formalization of PFNs assumes a separable prior, under which the feature distribution and the labeling mechanism are a priori independent. The structural-causal-model priors of current PFNs are non-separable, since features and labels are outputs of one shared random mechanism. We show that this changes the minimizer of the training objective and can invalidate the fixed-query martingale tests used to diagnose Bayesian behavior in PFNs. We characterize the resulting separability gap, show that it affects interpretability and Bayesian optimization, and exploit it to turn PFNs into semi-supervised predictors that use additional unlabeled rows. We characterize the gap of TabICL architectures pre-trained on these priors, test which predictive standard pre-training recovers, and provide insights into how PFNs behave as Bayesian predictors.

1 INTRODUCTION

Tabular foundation models such as TabPFN and TabICL are prior-data fitted networks (PFNs; Grinsztajn et al., 2025; 2026; Hollmann et al., 2023; 2025; Qu et al., 2025; 2026). A PFN is an approximator to a family of posterior predictive distributions (PPDs; Nagler, 2023), usually implemented as a transformer with parameters $\boldsymbol{\theta}$ and pretrained on simulated datasets drawn from a prior Π over tasks. A common data prior choice are structural causal models (SCMs). We use uppercase letters for random data and lowercase letters for their realizations. At inference, the PFN receives an observed context $d_n = \{y_i, \boldsymbol{x}_i\}_{i=1}^n$ and a query feature vector $\boldsymbol{x}$ and returns a predictive distribution $q_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, d_n)$ in its forward pass. For classification, PFNs are pretrained by cross-entropy on simulated contexts and query pairs drawn from the same task. The target is therefore the conditional distribution of the query label given its feature vector and the context (Nagler, 2023, Theorem 2.2). A trained PFN approximates this target with finite capacity and a finite pretraining budget.

While previous research on PFNs has asked *how accurately* a trained network approximates its target PPD, we ask a prior question: *Which predictive distribution does pretraining actually target?* As outlined below, our motivation stems from the (often implicitly made) assumption of *separability* of the data prior, which has so far received too little attention.

Bayesian target Concretely, let $p \sim \Pi$ denote a task distribution, with conditional label distribution $p(y \mid \boldsymbol{x})$. Here, we distinguish between two types of posterior predictives:

$$\begin{aligned} \textit{query-updated predictive:} \quad \pi(y \mid \boldsymbol{x}, d_n) &= \int p(y \mid \boldsymbol{x})\, d\Pi(p \mid d_n, \boldsymbol{x}), \\ \textit{context-updated predictive:} \quad \bar{\pi}(y \mid \boldsymbol{x}, d_n) &= \int p(y \mid \boldsymbol{x})\, d\Pi(p \mid d_n). \end{aligned} \tag{1}$$

The *query-updated predictive* π is the PPD that minimizes pretraining cross-entropy. The *context-updated predictive* $\bar{\pi}$ averages the same label distribution under the task posterior given the context alone. Both predictives condition on the context features and labels, and evaluate the label distribution at $\boldsymbol{x}$. However, only π uses $\boldsymbol{x}$ to update the task posterior.

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

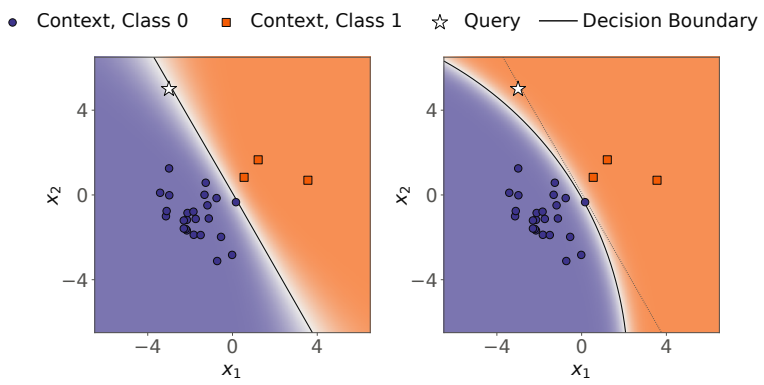

Figure 1: Illustration of the anticausal setting from Schölkopf et al. (2012): Two Gaussian class centers μ_0, μ_1 generate the features from the label, such that the location of an unlabeled point is informative about the labeling mechanism. The mechanism in the left figure does not allow the query's location to update the belief about the centers; the right figure does use the point's position.

Crucially, the **existing literature assumes that PFN pretraining targets the context-updated predictive** $\bar{\pi}$. Nagler (2023) notes in Remark 2.1 that this requires the feature distribution and the labeling mechanism to be independent under the prior, a property we call *separability*, but regards the query-dependent alternative π as unintuitive. This distinction especially matters for the SCM priors used to pretrain state-of-the-art PFNs, as they are typically not separable.

The setting of Figure 1 exemplifies this difference using the anticausal setting from Schölkopf et al. (2012). The task is a pair of class centers μ_0, μ_1 drawn from $\mathcal{N}(0, \tau^2 \boldsymbol{I}_2)$, Y is determined by a fair coin flip, and the features are drawn from that label's Gaussian, $\boldsymbol{X} \mid Y = k \sim \mathcal{N}(\mu_k, \sigma^2 I_2)$. If the context contains many points of class 0 but only a few of class 1, the posterior pins down μ_0 (blue cluster) but leaves μ_1 (red cluster) comparatively uncertain. For a query $\boldsymbol{x}$ far from both clusters (white star), the query-updated predictive π is more likely to assign it label 1 due to the smaller probability of $\boldsymbol{x}$ belonging to features of class 0, while the context-updated predictive $\bar{\pi}$ does not take the feature likelihood into account.

Most current PFNs are trained on non-separable priors like in the previous example. This therefore not only demonstrates that the two models in Eq. (1) lead to different predictive distributions, but also shows that PFNs have a close link to semi-supervised machine learning as they leverage the unlabeled query features. In this paper, we outline how this perspective proves useful for unlocking empirical performance gains as well as obtaining a better understanding of the working mechanisms of PFNs.

Our contributions By studying the difference of predictives in Eq. (1) through various lenses, we make the following contributions:

- We prove that prior non-separability already arises in simple SCMs and that mixing (separable) priors, as in TabPFN, TabICL, and Mitra, preserves separability if and only if the regimes share the same prior over feature distributions or the same prior over conditional label distributions.
- We characterize a *predictive gap* between π and $\bar{\pi}$ at finite context sizes and show that it vanishes for typical queries under posterior concentration. Under misspecification, their common limit can nevertheless have an arbitrarily large *optimality gap* relative to the best supported conditional label law.
- We study the practical consequences for downstream use and evaluation of PFNs, including interpretability, Bayesian optimization, semi-supervised prediction, and martingale checks. We test which type of predictive standard pretraining recovers and whether additional unlabeled rows improve prediction.

2 PROPERTIES OF THE DIFFERENT PREDICTIVES

Model and notation We identify a task with its distribution p, comprising a feature law p_X and a conditional label law $p_{Y|X}$. We use lowercase p for both a probability law and its density or mass function, with the meaning determined by the argument; for example, $p(y \mid \boldsymbol{x}) = p_{Y|X}(y \mid \boldsymbol{x})$ and $p(\boldsymbol{x}) = p_X(\boldsymbol{x})$. The prior Π is a distribution over these task laws, so posterior integrals are taken directly over p. Given a task $p \sim \Pi$, let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ denote the context features and $\boldsymbol{X}$ the query feature. Given a task $p \sim \Pi$, the context rows $(Y_1, \boldsymbol{X}_1), \ldots, (Y_n, \boldsymbol{X}_n)$ and the query row $(Y, \boldsymbol{X})$ are independent and identically distributed (iid) draws from p. Each feature vector is drawn from p_X and its label from $p_{Y|X}$, that is, $\boldsymbol{X}_i \sim p_X$ and $Y_i \mid \boldsymbol{X}_i \sim p(\cdot \mid \boldsymbol{X}_i)$, and likewise for $(Y, \boldsymbol{X})$. The random context is $D_n = (Y_i, \boldsymbol{X}_i)_{i=1}^n$, its realization is $d_n = (y_i, \boldsymbol{x}_i)_{i=1}^n$, and the predictor receives $(\boldsymbol{X}, D_n)$.

The following definition is central in this work.

Definition 1 (Separable prior)**.** Identify each task $p = p_{Y|X}p_X$ with the pair $(p_X, p_{Y|X})$. The prior is *separable* if its induced law on these pairs factorizes as

$$\Pi = \Pi_X \otimes \Pi_{Y|X}, \qquad d\Pi(p) = d\Pi_X(p_X)\, d\Pi_{Y|X}(p_{Y|X}).$$

Here Π_X and $\Pi_{Y|X}$ are the marginal priors over the feature law and conditional label law. Thus these two laws are independent under Π. Otherwise the prior is *non-separable*.

Under separability, the marginal posterior of $p_{Y|X}$ does not depend on the query feature $\boldsymbol{x}$:

$$d\Pi(p_{Y|X} \mid d_n, \boldsymbol{x}) = d\Pi(p_{Y|X} \mid d_n) \propto \prod_{i=1}^n p_{Y|X}(y_i \mid \boldsymbol{x}_i)\, d\Pi_{Y|X}(p_{Y|X});$$

see Proposition A.1 in Appendix A.1. Separability is thus sufficient for equality of the two predictives in Eq. (1), i.e. $\pi = \bar{\pi}$. As shown in the following, a non-separable prior permits $\pi \neq \bar{\pi}$, but does not require it at every context and query. Under a non-separable prior, the context features are also informative about $p_{Y|X}$, and $\bar{\pi}$ uses this information because the posterior $\Pi(\cdot \mid d_n)$ contains their likelihood $\prod_{i=1}^n p_X(\boldsymbol{x}_i)$. The two predictives therefore differ only in the feature likelihood $p_X(\boldsymbol{x})$ of the query.

2.1 SCM PRIORS ARE NON-SEPARABLE

In an SCM prior, a random graph, random node functions, and a random choice of the response node generate the features and the label jointly. The feature distribution and the conditional label distribution are then two functionals of the same random mechanism, and thus we should not expect separability to hold for these priors. In fact, it already fails in simple cases like a linear Gaussian SCM with a random label node (see Proposition A.2).

Mixing multiple generator regimes provides another potential source of non-separability. The original TabPFN prior mixes SCM and Bayesian neural network datasets equally (Hollmann et al., 2023). TabICL mixes neural and tree-based SCM datasets in a $70/30$ ratio (Qu et al., 2025). Mitra mixes SCM and tree-based priors (Zhang et al., 2025b), and Mitra-v2 adds a hybrid SCM to this outer mixture (Tao et al., 2026). TabICLv2 also combines neural, tree, Gaussian-process, and other functions within individual SCMs (Qu et al., 2026). Even if the feature and label laws of these priors were separable within each regime, their mixture is non-separable when both induced laws differ across regimes, as the following result shows (proof in Appendix A.2.1).

Proposition 1. *Let $\Pi_{X,s}$ and $\Pi_{Y|X,s}$ be priors on the induced feature and conditional label laws for $s \in \{0, 1\}$. For $\lambda \in (0, 1)$, set*

$$\Pi = (1 - \lambda)\, \Pi_{X,0} \otimes \Pi_{Y|X,0} + \lambda\, \Pi_{X,1} \otimes \Pi_{Y|X,1}.$$

Then Π is separable if and only if $\Pi_{X,0} = \Pi_{X,1}$ or $\Pi_{Y|X,0} = \Pi_{Y|X,1}$.

A mixture of regimes that differ in both their feature and their label distributions, as in all of the production priors above, is therefore non-separable, whatever the structure within each regime.



Figure 2: Recovery of the query-updated predictive on the non-separable ordered-cluster prior with context length $n = 2$. Standard pretraining follows the identity line, indicating recovery of π, whereas the difference for the decoupled PFN remains near zero, indicating recovery of $\bar{\pi}$. In single-class contexts, both labeled rows come from the same cluster, so the query is evidence about the unobserved center and the difference between π and $\bar{\pi}$ is larger (note the different axis ranges).

2.2 Finite-sample difference

So far, we have shown that PFN pretraining targets π, and that priors of current PFNs are unlikely to be separable, so that we should expect $\pi \neq \bar{\pi}$.

To better understand this difference, we first characterize how observing the query feature changes the predictive at a fixed context, and then examine what remains as the context grows. At finite context sizes, the two predictives differ as follows (proof in Appendix A.3.1).

Corollary 1. *Let $\Pi_n = \Pi(\cdot \mid d_n)$, with $d\Pi_n(p) \propto \prod_{i=1}^{n} p_X(\boldsymbol{x}_i)\, p(y_i \mid \boldsymbol{x}_i) d\Pi(p)$, be the posterior given the context, and let $m(\boldsymbol{x} \mid d_n) = \mathbb{E}_{\Pi_n}[p_X(\boldsymbol{x})]$ be the feature predictive. Then*

$$\Delta_\pi(y \mid \boldsymbol{x}, d_n) := \pi(y \mid \boldsymbol{x}, d_n) - \bar{\pi}(y \mid \boldsymbol{x}, d_n) = \frac{\mathrm{Cov}_{\Pi_n}\big(p(y \mid \boldsymbol{x}),\, p_X(\boldsymbol{x})\big)}{m(\boldsymbol{x} \mid d_n)}, \tag{2}$$

where $\Delta_\pi(y \mid \boldsymbol{x}, d_n)$ denotes the predictive gap between the two predictives.

The *predictive gap* can be interpreted as a correction term of the context-updated predictive to account for the posterior covariance between the label density at $\boldsymbol{x}$ and the conditional feature likelihood $p_X(\boldsymbol{x})$ of the query, which it otherwise ignores. Observing the query feature reweights the posterior toward tasks under which that feature value is likely. The query-updated predictive π therefore assigns more probability to a label than $\bar{\pi}$ exactly when these upweighted tasks also predict that label at the query more strongly. Under a separable prior, the covariance is always zero.

The size of Δ_π is not monotone in n or $m(\boldsymbol{x} \mid d_n)$ in general. A small feature predictive with a small covariance gives a small correction, and a diffuse posterior need not induce a nonzero covariance. It is large where tasks with comparable posterior weight disagree both in their feature likelihood and in their prediction at $\boldsymbol{x}$.

Figure 2 tests which predictive standard PFN pretraining recovers. We pretrain TabICL, denoted by $q_{\boldsymbol{\theta}}(\cdot)$, on a non-separable ordered-cluster prior, whose finite task space makes π and $\bar{\pi}$ exactly computable (Appendix C.1). Under standard pretraining, the Bayes-optimal target is π. In contrast, under a decoupled setting, each query label is drawn from a fresh task $p' \sim \Pi(\cdot \mid d_n)$, so the target is $\bar{\pi}$. At $n = 2$, the deviation of the learned query-updated predictive from the context-updated predictive, $q_{\boldsymbol{\theta}}(1 \mid \boldsymbol{x}, d_n) - \bar{\pi}(1 \mid \boldsymbol{x}, d_n)$, follows the exact difference $\Delta_\pi(1 \mid \boldsymbol{x}, d_n)$ along the identity line. The decoupled PFN's deviation remains near zero. Each PFN thus approximates its own Bayes-optimal target.

2.3 Consistency

The covariance in (2) describes the correction at a given context. Under posterior concentration, this correction vanishes for typical queries as the number of context rows grows. The following result makes this precise. TV and H denote the total variation and Hellinger distances (proof in Appendix A.3.2).

Proposition 2. *Let each task p have joint density $p(y, \boldsymbol{x})$, and let D_n consist of iid rows from the true joint density p_0. Write $\Pi_n = \Pi(\cdot \mid D_n)$; all almost-sure statements refer to the iid context sequence generated by p_0. Suppose there is a joint density $p^* = p^*_{Y|X} p^*_X$ such that, for every $\epsilon > 0$,*

$$\Pi_n\big(\{p : H(p, p^*) > \epsilon\}\big) \to 0 \qquad \textit{almost surely}.$$

Set $\delta_n = \int \mathrm{TV}(p, p^*)\, d\Pi_n(p)$. Then $\delta_n \to 0$ almost surely, and

$$\int \mathrm{TV}\big(\breve{\pi}(\cdot \mid \boldsymbol{x}, D_n), p^*_{Y|X}(\cdot \mid \boldsymbol{x})\big) p^*_X(\boldsymbol{x})\, d\boldsymbol{x} \le 2\delta_n\,, \quad \breve{\pi} \in \{\pi, \bar{\pi}\}.$$

By the triangle inequality, the integrated total variation between π and $\bar{\pi}$ is at most $4\delta_n$. Thus, the query correction in Eq. (2) vanishes for features drawn from p^*_X. If the true feature law satisfies $p_{0,X} \ll p^*_X$, this also gives convergence of all three integrated distances for queries drawn from $p_{0,X}$. When p^* is the unique KL-optimal joint density, both predictives approach its conditional distribution. The query correction therefore vanishes for typical queries as the context grows.

2.4 Misspecification

Although the gap between π and $\bar{\pi}$ vanishes with the sample size, non-separability of the prior can still affect the common limit. To see why, let p_0 denote the true joint density. The joint KL divergence decomposes as

$$D_{\mathrm{KL}}(p_0 \,\|\, p) = D_{\mathrm{KL}}(p_{0,X} \,\|\, p_X) + D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p),$$

where the expected conditional KL divergence,

$$D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p) := \mathbb{E}_{p_{0,X}}\big[D_{\mathrm{KL}}\big(p_0(\cdot \mid \boldsymbol{X}) \,\|\, p(\cdot \mid \boldsymbol{X})\big)\big],$$

measures loss in predictive performance relative to the true mechanism. When the posterior concentrates at a joint KL minimizer p^*, this limiting law minimizes the sum of the two terms. In particular, it can favor a mechanism that predicts labels less accurately if that mechanism better explains the features. This tradeoff disappears under separability, but can otherwise lead to bad models.

Proposition 3. *Let p_0 be the true joint density, and let p^* minimize $D_{\mathrm{KL}}(p_0 \,\|\, p)$ over the support of Π, with $D_{\mathrm{KL}}(p_0 \,\|\, p^*) < \infty$.*

(i) If Π is separable, then $D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p^*) = \inf_{p \in \operatorname{supp} \Pi} D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p)$.

(ii) If Π is non-separable, the equality in (i) need not hold, and the gap can be arbitrarily large.

The proof can be found in Appendix A.4.1. Part (i) shows that a separable prior attains the best conditional label fit available in its support. Thus, the prior's feature marginal Π_X does not affect how well the posterior fits the labels. Under non-separable priors, this is no longer guaranteed. Part (ii) shows that, in this case, a miscalibrated Π_X can make the posterior's label predictions arbitrarily bad. We refer to this as the *optimality gap*.

This tradeoff connects the common limit to the distinction between generative and discriminative estimation under misspecification (Liang & Jordan, 2008): the former optimizes joint fit, whereas the latter targets conditional fit. It is also related to the bias changes through which additional unlabeled data can degrade generative classifiers (Cozman et al., 2003), although the proposition concerns the shared limit with a single query row. If p_0 is a supported joint density, then $D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p^*) = 0$. Thus, more context can make π and $\bar{\pi}$ agree without yielding the best label predictions available under the prior: under misspecification, both can favor better feature fit at the expense of label accuracy.

2.5 What does this tell us?

PFN pretraining targets the query-updated predictive π (Proposition A.1), which coincides with the context-updated predictive $\bar{\pi}$ assumed in the literature when the feature likelihood cancels from the posterior over the labeling mechanism, as it does under separability (Definition 1). However, even

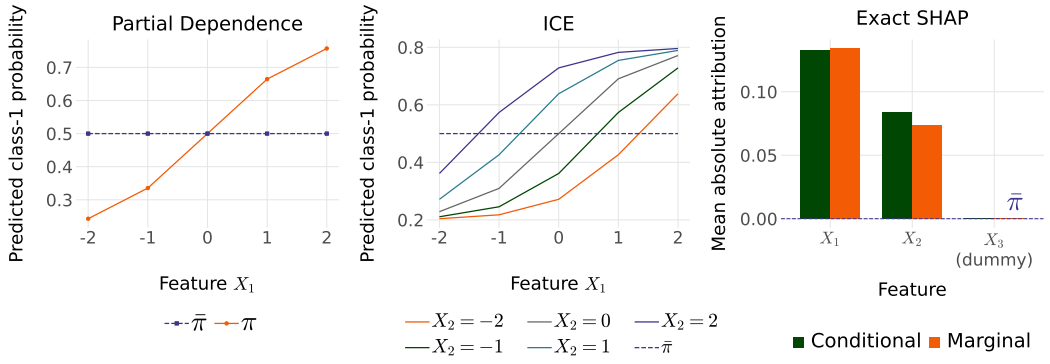

Figure 3: Feature effects under a prior in which every task has a constant class-1 probability c_p that is associated with its feature distribution. There are two tasks, a low-c_p and a high-c_p one. X_1 and X_2 are discrete features shifted toward negative values in the former and toward positive values in the latter, so they indicate the task, while X_3 is independent noise with the same distribution in both tasks. Left: PD of X_1. Middle: ICE curves of X_1 for fixed X_2. Right: mean absolute exact SHAP attributions of π, conditional and marginal. Since no labeling mechanism depends on $\boldsymbol{x}$, $\bar{\pi}$ is constant and all its attributions are zero (dashed). π assigns importance to X_1 and X_2 because they identify the task, and none to the dummy feature X_3.

simple SCM priors do not have this property (Proposition A.2) and neither does a mixture of (potentially separable) priors if the regimes do not share their prior over feature distributions or over labeling mechanisms (Proposition 1). Practically, at finite context sizes, the query can therefore update beliefs about the labeling mechanism, producing a *predictive gap* between what the PFN community reports to target and the query-updated predictive shown to be the actual target (Proposition A.1 and Eq. (2)). Under posterior concentration, this correction vanishes, and both predictives approach the same conditional distribution (Proposition 2). Under misspecification, however, restrictions in the prior's support can make that common limit sacrifice label fit for feature fit (Proposition 3), resulting in an *optimality gap*. Both gaps can be seen as a *separability gap*, causing (potentially arbitrarily large) differences between the predictives under non-separable priors.

For PFN design, the prior's feature generator therefore affects the predictions themselves. A PFN pretrained on an SCM prior infers the labeling mechanism partly from the query's feature values, and it can use additional unlabeled rows the same way (as demonstrated in the following Section 3). Whether this improves or degrades predictive performance on real data depends on whether the prior couples feature and label laws as the data do.

3 Consequences of non-separable priors

We have shown that under a prior with coupled feature and label laws, the query can update the posterior over the labeling mechanism. This has many important consequences.

Interpretability methods Feature effect methods for model interpretability are also applicable to PFNs, where batched inference can bring down the cost of repeated model evaluations (Rundel et al., 2024). Many interpretability methods, such as partial dependence (PD) and individual conditional expectation (ICE) plots vary selected features while averaging over, or fixing, the remaining coordinates (Goldstein et al., 2015). For a PFN approximating π, varying $\boldsymbol{x}$ changes both $p(y \mid \boldsymbol{x})$ and the posterior $\Pi(\cdot \mid \boldsymbol{x}, d_n)$ over tasks (Eq. (2)). The resulting curve therefore combines variation within labeling mechanisms with evidence about which mechanism generated the query. For example, if every task has a constant class probability $c_p := p(1 \mid \boldsymbol{x})$, but c_p is associated with the task's feature distribution, then $\pi(1 \mid \boldsymbol{x}, d_n)$ can vary with $\boldsymbol{x}$ even though $\bar{\pi}(1 \mid \boldsymbol{x}, d_n)$ is constant, as illustrated in Figure 3. This effect can arise within the data support, in addition to the familiar extrapolation problem of PD plots (Apley & Zhu, 2020). The marginalization convention of SHapley Additive exPlanations (SHAP; Lundberg & Lee, 2017) does not remove it either. Conditional SHAP averages missing coordinates under their distribution given the retained coordinates, whereas marginal

SHAP averages under their unconditional distribution (Molnar, 2025). Both still evaluate the same query-updated predictor at each completed input. Thus, these methods can faithfully explain the PFN's predictions while attributing importance to features that identify the task rather than change its conditional label distribution.

PFN for Bayesian optimization PFNs are increasingly used as surrogate models in Bayesian optimization (BO; Müller et al., 2023; Rogers & Ponnada, 2026; Yu et al., 2026). BO needs the predictive distribution of the response y at a candidate $\boldsymbol{x}$ chosen from previous evaluations d_n: $\pi_{\mathrm{BO}}(y \mid \boldsymbol{x}, d_n)$. If each input is selected using the observed history and randomization independent of p, and an evaluation at $\boldsymbol{x}$ follows $p(y \mid \boldsymbol{x})$, this predictive is

$$\pi_{\mathrm{BO}}(y \mid \boldsymbol{x}, d_n) = \int p(y \mid \boldsymbol{x})\, d\Pi_{\mathrm{BO}}(p \mid d_n), \quad \text{where } d\Pi_{\mathrm{BO}}(p \mid d_n) \propto \prod_{i=1}^{n} p(y_i \mid \boldsymbol{x}_i)\, d\Pi(p).$$

Under separability, the feature likelihood $p_X(\boldsymbol{x}) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i)$ in the PFN posterior depends only on p_X (see Proposition A.1 in Appendix A). It cancels when we marginalize over the feature law, so $\pi = \pi_{\mathrm{BO}}$. Therefore, the PFN provides exactly the predictive needed for BO.

Under non-separability, the feature likelihood can also update the labeling mechanism. Standard PFN pretraining thus treats both the previous inputs and the candidate as task-generated evidence, even though the BO algorithm chose them. As a result, its target π is usually different from π_{BO}. Replacing π by $\bar{\pi}$ removes the influence of the new candidate, but not the influence of the history. In either case, a PFN trained on a non-separable prior does not provide the predictive needed for BO.

Amortization gap Following Hülskamp et al. (2026), the amortization gap of a PFN is its excess over the Bayes-optimal predictor: $\Delta_{\boldsymbol{\theta}}(\boldsymbol{x}, d_n) = D_{\mathrm{KL}}\big(\pi(\cdot \mid \boldsymbol{x}, d_n) \,\|\, q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, d_n)\big)$. Since π minimizes the pretraining loss (Proposition A.1), the expected gap equals the excess pretraining loss of the PFN (Appendix B). When analyzing how closely a given PFN approximates its target posterior predictive distribution, as, e.g., in Müller et al. (2022; 2024) and Hülskamp et al. (2026), it must be carefully evaluated which predictive from Eq. (1) is targeted precisely. A reference computed from the context posterior alone, such as an unweighted average over MCMC draws from $\Pi(\cdot \mid d_n)$, estimates $\bar{\pi}$ rather than π. The discrepancy between this reference and the PFN decomposes into

$$q_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, d_n) - \bar{\pi}(y \mid \boldsymbol{x}, d_n) = \big(q_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, d_n) - \pi(y \mid \boldsymbol{x}, d_n)\big) + \Delta_{\pi}(y \mid \boldsymbol{x}, d_n),$$

so any divergence computed against $\bar{\pi}$ is corrupted by the predictive gap of Corollary 1. In the example of Fig. 2, this overstates the gap of the standard PFN by a factor of about 95. When p_X can be evaluated, weighting each draw by the query's feature likelihood removes the bias and can provide the correct reference π. Under a separable prior, no correction is needed since $\pi = \bar{\pi}$.

Martingale checks A martingale check, as employed in the context of PFNs (Falck et al., 2024; Fong et al., 2023; Ng et al., 2026), asks whether generating additional observations $Y_{n+1} \sim \pi(\cdot \mid \boldsymbol{x}_{n+1}, d_n)$ and updating on them (e.g., by updating the context $D_{n+1} = (d_n, (Y_{n+1}, \boldsymbol{x}_{n+1}))$) leaves a prediction at a fixed query unchanged on average. Under a non-separable prior, the predictor treats externally generated features as evidence about the labeling mechanism, so even an exact Bayesian predictor can drift; i.e., in general it holds

$$\mathbb{E}_{Y_{n+1} \sim \pi(\cdot \mid \boldsymbol{x}_{n+1}, d_n)}\big[\pi(A \mid \boldsymbol{x}, D_{n+1})\big] \neq \pi(A \mid \boldsymbol{x}, d_n). \tag{3}$$

The desired equality holds under separability. Without it, the appended feature provides evidence about the task, and the label sampler omits conditioning on the tracked query. Thus, drift can arise without PFN approximation error, although non-separability does not force drift at every query. For the underlying argument, refer to Appendix A.5. Resampling features with a Bayesian bootstrap (Ng et al., 2026; Rubin, 1981) averages these drifts without guaranteeing that they cancel.

To illustrate this failure, we evaluate the exact predictives π and $\bar{\pi}$ of the non-separable ordered-cluster prior (Appendix C.1) on 2000 contexts of size $n = 2$, with eight queries each. Future features that are then appended to the context are fixed at the query (i.e., each new context row is a labeled copy of the query location) or at one context feature selected once per context-query pair. Labels are drawn sequentially from the respective predictive where the query stays fixed throughout, and each appended row updates the context through its joint likelihood. We measure cumulative expected

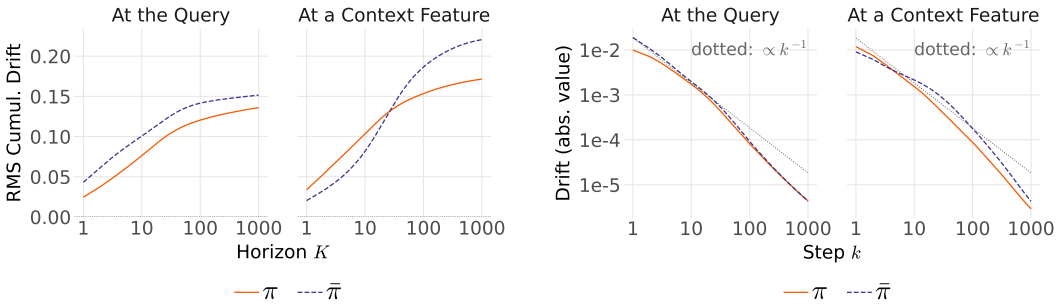

Figure 4: Non-separable priors can induce drift in exact Bayesian predictives. Curves show the RMS of the cumulative expected drift across 2000 contexts of size $n = 2$, with eight queries each, under the ordered-cluster prior. Every future feature is fixed at the query (left) or at one selected context feature (right), and labels are sampled from the predictive being evaluated with absolute conditional one-step drifts in the right figure. No trained network is involved.

drift as the expected class-1 prediction after the appended rows minus its initial value, holding the initial context, query, and repeated future feature fixed. For details, refer to Appendix A.5. The results in Figure 4 show substantial drift in both predictives. This demonstrates that the exact target can drift under the diagnostic's sampling protocol, irrespective of the network's approximation error.

PFNs as semi-supervised predictors By Corollary A.1, PFNs pretrained on non-separable priors are already semi-supervised predictors: π uses the unlabeled query $\boldsymbol{x}$ as evidence about the labeling mechanism. While they are currently limited to only use the additional evidence in the single query row, there is no structural reason not to integrate further unlabeled rows $\boldsymbol{V}$ into the prediction, turning PFNs into full semi-supervised predictors. Schölkopf et al. (2012) argue that this addition of unlabeled data should help especially in anti-causal and confounded tasks, which SCM priors already generate. Hence, integrating such unlabeled rows $\boldsymbol{V}$ into existing training pipelines and architectures should prove beneficial, which we show for both the ordered-cluster prior (see Appendix C.1) and the full-sized production prior of TabICLv2.

As such, we pretrained a novel 8.5M-parameter TabICL variant, *SemiTabICL*, to not just use the unlabeled information in the query $\boldsymbol{x}$, but instead also explicitly allow for the incorporation of an additional set of unlabeled context rows $\boldsymbol{V}$ which are processed exactly as regular context rows, only with an additional learned embedding vector instead of the label embedding (see Appendix C.3). Following Guo et al. (2010), we evaluate SemiTabICL in the standard setting of semi-supervised learning with a low number of labeled training points relative to a large number of unlabeled training points. To this end, we reuse the TALENT benchmark (Liu et al., 2025) and convert it into a benchmark for semi-supervised classification, that we call *semi-TALENT*, by splitting the official training fold of the classification datasets into labeled partitions of $n \in \{16, 32, 64, 96\}$ rows and unlabeled partitions across multiple repetitions. We compare SemiTabICL to a baseline TabICL model pretrained without additional unlabeled context rows, as well as a self-training PFN variant where unlabeled rows predicted with high confidence are iteratively appended with their predicted labels before the prediction on the test set is made (Yarowsky, 1995).

Figure 5 shows the differences in accuracy for these two settings across a range of labeled training rows n and the multiples thereof added as additional unlabeled rows k. SemiTabICL consistently improves upon the baseline by 0.4–1.1 percentage points, while the self-training hurts performance in almost every setting, increasingly so with more labeled rows. Appendix C.3 contains further details on the evaluation as well as another comparison on the UCI datasets used in Guo et al. (2010); Schölkopf et al. (2012). Importantly, when incorporating the capability to use additional unlabeled rows, it must be carefully specified whether these may coincide with the test rows, a point we will cover in the following paragraph.

Evaluation protocols Once PFNs condition on unlabeled rows, the distinction between inductive and transductive inference matters for evaluation (Schölkopf et al., 2012). An inductive model predicts each test row on its own, whereas a transductive one predicts the test rows jointly, so the

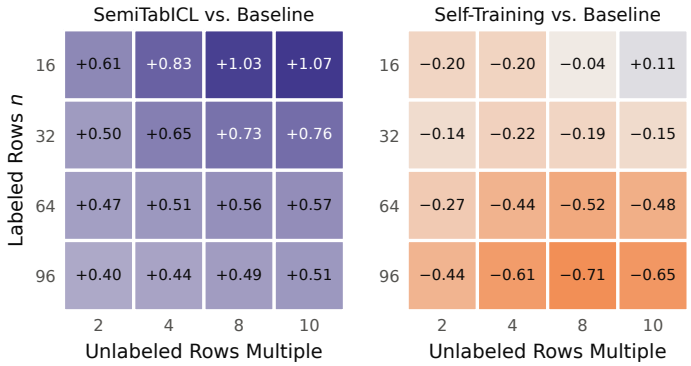

Figure 5: Accuracy gains (percentage points) for semi-supervised classification on the semi-TALENT benchmark for a range of labeled context dataset sizes (n) and multiples thereof of unlabeled context rows. Left: SemiTabICL vs. identical baseline trained and evaluated without unlabeled context rows. Right: Semi-supervised extension of the baseline model via self-training vs. no additional self-training

prediction for one row may depend on the others. Semi-supervised predictions induced by training on non-separable priors or explicitly for SemiTabICL stay inductive as long as unlabeled and other test rows are disjoint. Once they overlap, a method gains information that an inductive competitor does not have, which can lower the Bayes-optimal loss. Current benchmarks serve the test data in different ways, but none of them ensures that the prediction for one test row cannot use the others. No current tabular foundation model is designed to exploit this. Still, several pipelines already let test rows affect each other's predictions through shared preprocessing or compute optimizations (Appendix B). Benchmarks should therefore state whether transductive inference is permitted and, if it is not, enforce invariance to the other test rows throughout preprocessing and prediction.

4 Discussion

This article has highlighted a subtle but important discrepancy between the typical formulation of the posterior predictive distribution in the PFN literature and the actual target of PFN pretraining under non-separable priors. We further showed that this discrepancy has several practical implications for interpretability, Bayesian optimization, martingale checks, and semi-supervised learning. Depending on the application, these effects can be unwanted or beneficial. We argue that the distinction between the query-updated predictive π and the context-updated predictive $\bar{\pi}$ is important when designing PFN pretraining and evaluation protocols, especially in the presence of non-separable priors. Although this difference vanishes under the stated concentration conditions, coupling feature and label distributions can still sacrifice label fit for feature fit under misspecification. Our experiments confirm recovery of the query-updated target and show that explicitly incorporating additional unlabeled rows yields predictive gains with SemiTabICL on real tabular data.

Separable priors avoid the issues discussed in this work, but also forgo potential gains from using unlabeled features. Our analysis already points to useful design principles for such priors. SCM priors can remain the basis for generative modeling of tabular data, but two independent SCM mechanisms should be drawn, one for generating the features and one for generating the label. One could also pretrain on a range of coupling strengths between the mechanisms. These findings make the dependence between feature and label laws a concrete design choice for PFNs and a promising direction for future work.

Limitations The main limitation of this work is that we mainly establish that non-separability *can* have the issues discussed in this work. We do not yet know whether production-grade PFNs benefit or suffer from it in practice, and to what extent. Studying this is challenging, since available PFNs use diverse pretraining priors and many are not open-source. Further, we can only compare to meaningful comparator models after full-scale retraining of the model under a separable sibling of its prior. We hope that our work motivates future research in this direction.

AI USE STATEMENT

In this work, we used generative AI tools to support the implementation of the presented methods in code and to draft and review mathematical proofs. We have not used generative AI tools to formulate research hypotheses and develop the theoretical and conceptual framework. Furthermore, other required disclosures such as data cleaning or data analysis are not applicable to this work. Additionally, we used generative AI tools to assist in the writing and presentation of this manuscript by improving readability and the presentation of the included figures. Furthermore, generative AI tools were used to explore and identify relevant literature in addition to our initial manual survey. We have reviewed all AI-assisted work. AI-generated code was reviewed manually as well as tested in detail. All modifications of the manuscript suggested by AI tools were carefully inspected. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we make our experimental code available in Appendix C.4. The repository contains detailed instructions for reproducing the empirical results presented in this work. For our theoretical contributions, we have included all assumptions, detailed proofs and implications in Appendix A.

REFERENCES

Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4), 2020.

Fabio G Cozman, Ira Cohen, and Marcelo C Cirelo. Semi-supervised learning of mixture models. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 99–106, 2003.

Pierpaolo De Blasi and Stephen G. Walker. Bayesian Asymptotics with Misspecified Models. *Statistica Sinica*, 23(1):169–187, 2013. doi: 10.5705/ss.2010.239.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2026.

Fabian Falck, Ziyu Wang, and Christopher C. Holmes. Is in-context learning in large language models bayesian? A martingale perspective. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 21–27 Jul 2024.

Edwin Fong, Chris Holmes, and Stephen G Walker. Martingale posterior distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 85(5):1357–1391, 11 2023.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. *Advances in Neural Information Processing Systems*, 33:8284–8295, 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional Neural Processes. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1704–1713. PMLR, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard schoelkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-2.5: Advancing the State of the Art in Tabular Foundation Models, 2025.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, Dominik Safaric, Jake Robertson, Benjamin Jäger, Simone Alessi, Adrian Hayler, Vladyslav Moroshan, Lennart Purucker, Philipp Singer, Alan Arazi, Julien Siems, Jan Hendrik Metzen, Georg Grab, Nick Erickson, Siyuan Guo, Eliott Kalfon, Simon Bing, David Salinas, Clara Cornu, Lilly Charlotte Wehrhahn, Diana Kriuchkova, Kursat Kaya, Lydia Sidhoum, Marie Salmon, Jerry Chen, Madelon Hulsebos, Yann LeCun, Samuel Müller, Bernhard schoelkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. TabPFN-3: Technical Report, 2026.

Yuanyuan Guo, Xiaoda Niu, and Harry Zhang. An extensive empirical study on semi-supervised learning. In *2010 IEEE International Conference on Data Mining*, pp. 186–195. IEEE, 2010.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. In *The Eleventh International Conference on Learning Representations*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate Predictions on Small Data with a Tabular Foundation Model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6.

Manuel Hülskamp, Julius Kobialka, Emanuel Sommer, and David Rügamer. On the Uncertainty in Prior-Data Fitted Network Pretraining. In *2nd ICML Workshop on Foundation Models for Structured Data*, 2026.

Dominik Janzing, Lenon Minorics, and Patrick Blöbaum. Feature relevance quantification in explainable AI: A causal problem. In *International Conference on artificial intelligence and statistics*, pp. 2907–2916. PMLR, 2020.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive Neural Processes. In *International Conference on Learning Representations*, 2019.

Jonas Landsgesell, Pascal Knoll, and Tizian Wenzel. ScoringBench: A Benchmark for Evaluating Tabular Foundation Models with Proper Scoring Rules. *arXiv preprint arXiv:2603.29928*, 2026.

Yingcong Li, Xiangyu Chang, Muti Kara, Xiaofeng Liu, Amit Roy-Chowdhury, and Samet Oymak. When and how unlabeled data provably improve in-context learning. *Advances in Neural Information Processing Systems*, 2026.

Percy Liang and Michael I Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *Proceedings of the 25th international conference on Machine learning*, pp. 584–591, 2008.

Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, Huai-Hong Yin, Tao Zhou, Jun-Peng Jiang, and Han-Jia Ye. TALENT: A tabular analytics and learning toolbox. *Journal of Machine Learning Research*, 26, 2025.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Christoph Molnar. *Interpretable Machine Learning*. 3 edition, 2025. ISBN 978-3-911578-03-5. URL https://christophm.github.io/interpretable-ml-book.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers Can Do Bayesian Inference, 2022.

Samuel Müller, Matthias Feurer, Noah Hollmann, and Frank Hutter. Pfns4bo: In-context learning for bayesian optimization. In *International Conference on Machine Learning*, pp. 25444–25470. PMLR, 2023.

Samuel Müller, Noah Hollmann, and Frank Hutter. Bayes' power for explaining in-context learning generalizations. *arXiv preprint arXiv:2410.01565*, 2024.

Thomas Nagler. Statistical Foundations of Prior-Data Fitted Networks. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2023.

Kenyon Ng, Edwin Fong, David Tyler Frazier, Jeremias Knoblauch, and Susan Wei. TabMGP: Martingale Posterior with TabPFN. In *Forty-third International Conference on Machine Learning*, 2026.

Tung Nguyen and Aditya Grover. Transformer Neural Processes: Uncertainty-Aware Meta Learning Via Sequence Modeling. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Lennart Purucker, Andrej Tschalzev, Nick Erickson, Gioia Blayer, David Holzmüller, Alan Arazi, Alexander Pfefferle, Mustafa Tajjar, Gaël Varoquaux, and Frank Hutter. Beyond IID: How General Are Tabular Foundation Models, Really? 2026.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A Tabular Foundation Model for In-Context Learning on Large Data. In *International Conference on Machine Learning*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICLv2: A better, faster, scalable, and open tabular foundation model. In *International Conference on Machine Learning*, 2026.

Theodore Rogers and Srividya Ponnada. Zero-Shot Bayesian Optimization with TabPFN: Competitive with State-of-the-Art without Per-Task Training. In *AutoML 2026 Methods Track*, 2026.

Donald B Rubin. The bayesian bootstrap. *The annals of statistics*, pp. 130–134, 1981.

David Rundel, Julius Kobialka, Constantin von Crailsheim, Matthias Feurer, Thomas Nagler, and David Rügamer. Interpretable machine learning for TabPFN. In *World Conference on Explainable Artificial Intelligence*, pp. 465–476. Springer, 2024.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On Causal and Anticausal Learning. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 2012.

Gowthami Somepalli, Micah Goldblum, Avi Schwarzschild, C Bayan Bruss, and Tom Goldstein. Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.

Omar Swelam, Lennart Purucker, Jake Robertson, Hanne Raum, Joschka Boedecker, and Frank Hutter. Does TabPFN Understand Causal Structures? In *EurIPS 2025 Workshop: AI for Tabular Data*, 2025.

Yefan Tao, Xiyuan Zhang, Xinyi Liu, Boran Han, Danielle Maddix, Haoyang Fang, Zhen Han, Jiading Gai, Xuanqing Liu, Michael Bohlke-Schneider, Yuyang Wang, Gerald Friedland, Kevan Mah, Chris Lee, and Chris Kong. Mitra-v2 technical report, 2026.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995.

Rosen Yu, Cyril Picard, and Faez Ahmed. GIT-BO: High-Dimensional Bayesian Optimization with Tabular Foundation Models. In *International Conference on Learning Representations*, 2026.

Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, et al. Limix: Unleashing structured-data modeling capability for generalist intelligence. *arXiv preprint arXiv:2509.03505*, 2025a.

Xiyuan Zhang, Danielle C. Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W. Mahoney, Cuixiong Hu, Huzefa Rangwala, George Karypis, and Bernie Wang. Mitra: Mixed Synthetic Priors for Enhancing Tabular Foundation Models. In *Advances in Neural Information Processing Systems*, 2025b.

A Further theoretical results and proofs

A.1 Bayes-optimal predictive

Proposition A.1. *Under our data model, the expected log loss over all conditional predictors $q(y \mid \boldsymbol{x}, d_n)$ is minimized, uniquely up to null sets of $(\boldsymbol{X}, D_n)$, by the conditional distribution of Y given $(\boldsymbol{X}, D_n)$. Its density $\pi(y \mid \boldsymbol{x}, d_n)$ averages $p(y \mid \boldsymbol{x})$ under the task posterior*

$$d\Pi(p \mid \boldsymbol{x}, d_n) \propto p_X(\boldsymbol{x}) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i)\, p(y_i \mid \boldsymbol{x}_i)\, d\Pi(p). \tag{4}$$

This is equivalent to π in Eq. (1). Under separability, the marginal posterior of $p_{Y|X}$ is proportional to $\prod_{i=1}^{n} p_{Y|X}(y_i \mid \boldsymbol{x}_i)\, d\Pi_{Y|X}(p_{Y|X})$, and hence does not depend on the query feature $\boldsymbol{x}$. Therefore, the predictive equals $\bar{\pi}(y \mid \boldsymbol{x}, d_n)$ wherever the posteriors are defined.

Proof. Let $\mathfrak{D} = (\mathfrak{X}, D_n)$, where $\mathfrak{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n, \boldsymbol{X})$ and $\mathfrak{Y} = (Y_1, \ldots, Y_n, Y)$, with realizations $\mathfrak{x}$ and $\mathfrak{y} = (y_1, \ldots, y_n, y)$. For any conditional distribution q,

$$\mathbb{E}[-\log q(Y \mid \mathfrak{D})] - \mathbb{E}[-\log \pi(Y \mid \mathfrak{D})] = \mathbb{E}[D_{\mathrm{KL}}(\pi(\cdot \mid \mathfrak{D}) \,\|\, q(\cdot \mid \mathfrak{D}))] \geq 0,$$

where $\pi(\cdot \mid \mathfrak{D})$ is the conditional distribution of Y given $\mathfrak{D}$. Equality holds if and only if $q(\cdot \mid \mathfrak{D}) = \pi(\cdot \mid \mathfrak{D})$ almost surely, proving uniqueness up to null sets of the inputs.

By the independence of the rows given the task p, the joint density at $(\mathfrak{x}, \mathfrak{y})$, conditional on p, is

$$p_X(\boldsymbol{x})\, p(y \mid \boldsymbol{x}) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i)\, p(y_i \mid \boldsymbol{x}_i).$$

Hence, conditioning on $\boldsymbol{X}_i = \boldsymbol{x}_i$, $\boldsymbol{X} = \boldsymbol{x}$, and $Y_i = y_i$, Bayes' rule gives

$$d\Pi(p \mid \boldsymbol{x}, d_n) \propto p_X(\boldsymbol{x}) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i)\, p(y_i \mid \boldsymbol{x}_i)\, d\Pi(p),$$

which is Eq. (4). Moreover,

$$\pi(y \mid \boldsymbol{x}, d_n) = \int p(y \mid \boldsymbol{x})\, d\Pi(p \mid \boldsymbol{x}, d_n),$$

since, conditional on the task p and $\mathfrak{X}$, the distribution of Y is $p(\cdot \mid \boldsymbol{X})$.

Under separability, the prior factorizes as

$$d\Pi(p) = d\Pi_X(p_X)\, d\Pi_{Y|X}(p_{Y|X}).$$

Thus, marginalizing the task posterior over p_X gives

$$\begin{aligned} d\Pi(p_{Y|X} \mid \boldsymbol{x}, d_n) &\propto \left[\int p_X(\boldsymbol{x}) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i)\, d\Pi_X(p_X) \right] \prod_{i=1}^{n} p_{Y|X}(y_i \mid \boldsymbol{x}_i)\, d\Pi_{Y|X}(p_{Y|X}) \\ &\propto \prod_{i=1}^{n} p_{Y|X}(y_i \mid \boldsymbol{x}_i)\, d\Pi_{Y|X}(p_{Y|X}), \end{aligned}$$

because the bracketed factor does not depend on $p_{Y|X}$. Consequently,

$$\pi(y \mid \boldsymbol{x}, d_n) = \int p_{Y|X}(y \mid \boldsymbol{x})\, d\Pi(p_{Y|X} \mid d_n) = \bar{\pi}(y \mid \boldsymbol{x}, d_n).$$

□

As a direct corollary, we can also derive the corresponding quantities when a set of unlabeled features is available.

Corollary A.1. *Let $\boldsymbol{V} = (\boldsymbol{V}_1, \ldots, \boldsymbol{V}_k)$ be additional unlabeled feature rows that, conditional on the task p, are iid draws from p_X and are independent of the context and query row. Let $\boldsymbol{v}$ denote their realizations. Then the expected log loss over all conditional predictors $q(y \mid \boldsymbol{x}, \boldsymbol{v}, d_n)$ is minimized, uniquely up to null sets, by*

$$\pi(y \mid \boldsymbol{x}, \boldsymbol{v}, d_n) = \int p(y \mid \boldsymbol{x})\, d\Pi(p \mid \boldsymbol{x}, \boldsymbol{v}, d_n), \tag{5}$$

where

$$d\Pi(p \mid \boldsymbol{x}, \boldsymbol{v}, d_n) \propto p_X(\boldsymbol{x}) \prod_{j=1}^{k} p_X(\boldsymbol{v}_j) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i) p(y_i \mid \boldsymbol{x}_i)\, d\Pi(p). \tag{6}$$

Under a separable prior, the marginal posterior over $p_{Y|X}$ does not depend on $\boldsymbol{x}$ or $\boldsymbol{v}$, and hence

$$\pi(y \mid \boldsymbol{x}, \boldsymbol{v}, d_n) = \bar{\pi}(y \mid \boldsymbol{x}, d_n).$$

Proof. Conditional on the task p, the labeled context rows, the unlabeled rows, and the query row are independent. Hence, the joint density of the observed features and context labels, conditional on p, is proportional to

$$p_X(\boldsymbol{x}) \prod_{j=1}^{k} p_X(\boldsymbol{v}_j) \prod_{i=1}^{n} p_X(\boldsymbol{x}_i) p(y_i \mid \boldsymbol{x}_i).$$

Bayes' rule therefore gives Eq. (6). Conditional on p and $\boldsymbol{x}$, the query label has distribution $p(\cdot \mid \boldsymbol{x})$, so its conditional distribution given $(\boldsymbol{x}, \boldsymbol{v}, d_n)$ is

$$\int p(y \mid \boldsymbol{x})\, d\Pi(p \mid \boldsymbol{x}, \boldsymbol{v}, d_n).$$

By the same log-loss argument as in Proposition A.1, this conditional distribution is the unique Bayes-optimal predictor up to null sets.

If $\Pi = \Pi_X \otimes \Pi_{Y|X}$, marginalizing the posterior over p_X yields

$$d\Pi(p_{Y|X} \mid \boldsymbol{x}, \boldsymbol{v}, d_n) \propto \prod_{i=1}^{n} p_{Y|X}(y_i \mid \boldsymbol{x}_i)\, d\Pi_{Y|X}(p_{Y|X}),$$

because all factors involving $\boldsymbol{x}$, $\boldsymbol{v}$, and the context feature likelihood depend only on p_X. Thus the unlabeled rows and the query feature do not update the posterior over the labeling mechanism, which gives $\pi(y \mid \boldsymbol{x}, \boldsymbol{v}, d_n) = \bar{\pi}(y \mid \boldsymbol{x}, d_n)$. □

A.2 Non-separability of SCM priors

As discussed before, we should not expect independence to hold for SCM priors. The following shows that it already fails in simple cases such as a linear Gaussian SCM with a random label node.

Proposition A.2. *Let (B, J) be random, where B is an acyclic $d \times d$ matrix and J is a node, with $d \geq 2$. Conditional on (B, J), let $\boldsymbol{Z} = B\boldsymbol{Z} + \boldsymbol{E}$ with $\boldsymbol{E} \sim \mathcal{N}(0, D)$ for a fixed positive diagonal matrix D, and let $Y := Z_J$ and $\boldsymbol{X} := \boldsymbol{Z}_{-J}$. If $\mathrm{Var}(Y \mid \boldsymbol{X}, B, J)$ is nonconstant under the prior on (B, J), the induced prior is non-separable. This holds in particular when $D = I_d$, J is uniform and independent of B, and B has a nonzero edge with positive probability.*

Proof. Let $\Sigma = (I-B)^{-1} D (I-B)^{-T}$. Acyclicity gives $\det(I-B) = 1$ and hence $\det \Sigma = \det D$. Since $\Sigma_{-J,-J}$ is positive definite, the block determinant formula and Gaussian conditioning give

$$\begin{aligned} \det D = \det \Sigma &= \det(\Sigma_{-J,-J}) \big(\Sigma_{JJ} - \Sigma_{J,-J} \Sigma_{-J,-J}^{-1} \Sigma_{-J,J}\big) \\ &= \det(\Sigma_{-J,-J})\ \mathrm{Var}(Y \mid \boldsymbol{X}, B, J). \end{aligned}$$

The feature law determines $\det(\Sigma_{-J,-J})$, and the conditional label law determines $\mathrm{Var}(Y \mid \boldsymbol{X}, B, J)$. If the latter is nonconstant, these laws cannot be independent because the product is fixed.

Now consider the particular case $D = I_d$, with J uniform and independent of B. The conditional variance is

$$\mathrm{Var}(Y \mid \boldsymbol{X}, B, J)^{-1} = (\Sigma^{-1})_{JJ} = 1 + \sum_{k:J\to k} B_{kJ}^2.$$

Every finite acyclic graph has a node with no outgoing edges. If J is such a node, the sum in the last display is empty and the conditional variance is 1. If J instead has a nonzero outgoing edge $J \to k$, it holds $B_{kJ} \neq 0$ and, thus,

$$\mathrm{Var}(Y \mid \boldsymbol{X}, B, J) = \left(1 + \sum_{l:J\to l} B_{lJ}^2\right)^{-1} \leq (1 + B_{kJ}^2)^{-1} < 1.$$

On the event that B has a nonzero edge, both types of node exist. Since J is uniform and independent of B, each type is selected with conditional probability at least $1/d$. Since the event has positive probability, the conditional variance is nonconstant under the prior. □

A.2.1 Proof of Proposition 1

Proposition 1 states that even for separable priors, mixtures are non-separable when both induced laws differ across regimes.

Proof. The marginals of Π are

$$\Pi_X = (1-\lambda)\Pi_{X,0} + \lambda\Pi_{X,1},$$
$$\Pi_{Y|X} = (1-\lambda)\Pi_{Y|X,0} + \lambda\Pi_{Y|X,1},$$

which implies

$$\Pi - \Pi_X \otimes \Pi_{Y|X} = \lambda(1-\lambda)(\Pi_{X,1} - \Pi_{X,0}) \otimes (\Pi_{Y|X,1} - \Pi_{Y|X,0}).$$

Since $\lambda(1-\lambda) > 0$, the right-hand side vanishes iff $(\Pi_{X,1} - \Pi_{X,0}) \otimes (\Pi_{Y|X,1} - \Pi_{Y|X,0}) = 0$. A tensor product of two finite signed measures is zero iff at least one factor is zero. Hence Π is separable iff $\Pi_{X,0} = \Pi_{X,1}$ or $\Pi_{Y|X,0} = \Pi_{Y|X,1}$. □

A.3 Consistency and the finite-sample difference

A.3.1 Proof of Corollary 1

Proof. By Eq. (4) and the definition of Π_n, the posterior given the query has density $p_X(\boldsymbol{x})/m(\boldsymbol{x} \mid d_n)$ relative to Π_n, so $\pi(y \mid \boldsymbol{x}, d_n) = \mathbb{E}_{\Pi_n}[p(y \mid \boldsymbol{x})\, p_X(\boldsymbol{x})]/\mathbb{E}_{\Pi_n}[p_X(\boldsymbol{x})]$, and $\mathbb{E}[AB] = \mathbb{E}[A]\mathbb{E}[B] + \mathrm{Cov}(A, B)$. □

A.3.2 Proof of Proposition 2

Nagler (2023, Theorem 3.1) shows for classification with iid rows that $\bar{\pi}$ converges to the conditional of the KL-optimal distribution in the prior's support. The proof is based on the task posterior Π_n formed from the joint likelihood, and it uses only that Π_n concentrates in Hellinger distance. In the non-separable setting, the query feature itself changes the task posterior. The following proposition shows that concentration of the posterior over joint task laws is sufficient for convergence of both π and $\bar{\pi}$, in both classification and regression problems. Sufficient KL-support and Hellinger-cover conditions for a unique KL-optimal density are given by De Blasi & Walker (2013, Corollary 1) and used in the PFN setting by Nagler (2023, Appendix A.2).

Recall that, for two densities m and p with respect to a common dominating measure μ, the total variation and Hellinger distances are defined as

$$\mathrm{TV}(m, p) = \frac{1}{2}\int |m - p|\, d\mu, \qquad H(m, p) = \left(\int (\sqrt{m} - \sqrt{p})^2\, d\mu\right)^{1/2}.$$

Proof. Since $\mathrm{TV}(m,p) \le H(m,p)$ and $\mathrm{TV}(m,p) \le 1$, we get

$$\begin{aligned}\delta_n &= \int \mathrm{TV}(p,p^*)\, d\Pi_n(p) \\ &= \int_{\{H(p,p^*)\le\epsilon\}} \mathrm{TV}(p,p^*)\, d\Pi_n(p) + \int_{\{H(p,p^*)>\epsilon\}} \mathrm{TV}(p,p^*)\, d\Pi_n(p) \\ &\le \epsilon + \Pi_n\big(\{p : H(p,p^*) > \epsilon\}\big).\end{aligned}$$

Taking $n \to \infty$ followed by $\epsilon \to 0$ gives $\limsup_{n\to\infty} \delta_n = 0$ almost surely.

For convergence of the two predictives π and $\bar{\pi}$, we first show that the integrated total variation of the conditional distributions is bounded by the total variation of the joint distributions. If m and p are joint densities on $\mathcal{Y} \times \mathcal{X}$, the product of the label and feature spaces, with feature marginals m_X and p_X, then

$$\begin{aligned}&\int \mathrm{TV}\big(m(\cdot \mid \boldsymbol{x}), p(\cdot \mid \boldsymbol{x})\big) p_X(\boldsymbol{x})\, d\boldsymbol{x} \\ &= \frac{1}{2} \iint \big|m(y \mid \boldsymbol{x}) p_X(\boldsymbol{x}) - p(y,\boldsymbol{x})\big|\, dy\, d\boldsymbol{x} \\ &\le \frac{1}{2} \iint \big|m(y \mid \boldsymbol{x}) p_X(\boldsymbol{x}) - m(y,\boldsymbol{x})\big|\, dy\, d\boldsymbol{x} + \frac{1}{2} \iint \big|m(y,\boldsymbol{x}) - p(y,\boldsymbol{x})\big|\, dy\, d\boldsymbol{x} \\ &= \mathrm{TV}(m_X, p_X) + \mathrm{TV}(m,p) \\ &\le 2\,\mathrm{TV}(m,p). \end{aligned} \tag{7}$$

Now let $m_n(y,\boldsymbol{x}) = \int p(y,\boldsymbol{x})\, d\Pi_n(p)$ be the posterior predictive joint density of one further row, with feature marginal $m_{n,X}$. Because the rows are conditionally iid, $\pi(y \mid \boldsymbol{x}, D_n) = m_n(y,\boldsymbol{x})/m_{n,X}(\boldsymbol{x})$. Applying Eq. (7) to m_n, gives

$$\begin{aligned}\int \mathrm{TV}\big(\pi(\cdot \mid \boldsymbol{x}, D_n), p^*(\cdot \mid \boldsymbol{x})\big) p_X^*(\boldsymbol{x})\, d\boldsymbol{x} &\le 2\,\mathrm{TV}(m_n, p^*) \\ &= 2\,\mathrm{TV}\left(\int p(y,\boldsymbol{x})\, d\Pi_n(p),\, p^*\right) \\ &\le 2 \int \mathrm{TV}(p,p^*)\, d\Pi_n(p) \\ &= 2\delta_n,\end{aligned}$$

The second inequality uses convexity of total variation. This proves the first bound.

For the second bound, recall that $\bar{\pi}(\cdot \mid \boldsymbol{x}, D_n)$ is the posterior mixture of $p(\cdot \mid \boldsymbol{x})$. Convexity of total variation applied to this mixture gives, for every $\boldsymbol{x}$,

$$\mathrm{TV}\big(\bar{\pi}(\cdot \mid \boldsymbol{x}, D_n), p^*(\cdot \mid \boldsymbol{x})\big) \le \int \mathrm{TV}\big(p(\cdot \mid \boldsymbol{x}), p^*(\cdot \mid \boldsymbol{x})\big)\, d\Pi_n(p).$$

Integrating over $p_X^*(\boldsymbol{x})$, changing the order of integration, and Eq. (7) give

$$\begin{aligned}\int \mathrm{TV}\big(\bar{\pi}(\cdot \mid \boldsymbol{x}, D_n), p^*(\cdot \mid \boldsymbol{x})\big) p_X^*(\boldsymbol{x})\, d\boldsymbol{x} &\le \int \left[\int \mathrm{TV}\big(p(\cdot \mid \boldsymbol{x}), p^*(\cdot \mid \boldsymbol{x})\big) p_X^*(\boldsymbol{x})\, d\boldsymbol{x}\right] d\Pi_n(p) \\ &\le 2 \int \mathrm{TV}(p,p^*)\, d\Pi_n(p) \\ &= 2\delta_n.\end{aligned}$$

□

A.4 Misspecification

A.4.1 Proof of Proposition 3

Proof. (i) Under separability, the support of the induced pairs $(p_X, p_{Y|X})$ is the Cartesian product $\mathrm{supp}(\Pi_X) \times \mathrm{supp}(\Pi_{Y|X})$. For every supported mechanism p, it then contains the pair $(p_X^*, p_{Y|X}^*)$.

Joint optimality of p^* and the KL chain rule therefore give

$$D_{\mathrm{KL}}(p_{0,X} \,\|\, p^*_X) + D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p^*) \le D_{\mathrm{KL}}(p_{0,X} \,\|\, p^*_X) + D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p).$$

The feature divergence $D_{\mathrm{KL}}(p_{0,X} \,\|\, p^*_X)$ is finite by the assumption on p^*. Canceling it and taking the infimum over p proves (i).

(ii) We construct an explicit model to verify the claim. Suppose X and Y are independent $\mathcal{N}(0,1)$ variables under p_0, and let $t > 0$ be arbitrary. Consider a prior with positive mass on only the following two task densities $p^{(A)}$ and $p^{(B)}$:

$$\begin{aligned} p^{(A)}: &\quad X \sim \mathcal{N}(\sqrt{2(t+1)}, 1), && Y \mid X \sim \mathcal{N}(0,1), \\ p^{(B)}: &\quad X \sim \mathcal{N}(0,1), && Y \mid X \sim \mathcal{N}(\sqrt{2t}, 1). \end{aligned}$$

By the standard formula for KL divergence between Gaussians,

$$D_{\mathrm{KL}}(\mathcal{N}(\mu_0, \sigma_0^2) \,\|\, \mathcal{N}(\mu_1, \sigma_1^2)) = \frac{1}{2}\left(\frac{(\mu_0 - \mu_1)^2}{\sigma_1^2} + \frac{\sigma_0^2}{\sigma_1^2} - 1 - \log\left(\frac{\sigma_0^2}{\sigma_1^2}\right)\right),$$

the feature and conditional KL divergences are

$$\begin{aligned} \big(D_{\mathrm{KL}}(p_{0,X} \,\|\, p^{(A)}_X), D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p^{(A)})\big) &= (t+1, 0), \\ \big(D_{\mathrm{KL}}(p_{0,X} \,\|\, p^{(B)}_X), D^{Y|X}_{\mathrm{KL}}(p_0 \,\|\, p^{(B)})\big) &= (0, t). \end{aligned}$$

Thus, $p^{(B)}$ uniquely minimizes the joint KL divergence, while its conditional KL exceeds the supported minimum by t. The prior contains both the true feature law and the true conditional label law, but not their pairing p_0. The prior is therefore non-separable, and the penalty is unbounded as $t \to \infty$. □

A.5 Martingale checks

Fix a label event $A \subseteq \mathcal{Y}$, a query $\boldsymbol{x}$ and a future feature $\boldsymbol{x}_{n+1}$. The diagnostic draws $Y_{n+1} \sim q(\cdot \mid \boldsymbol{x}_{n+1}, d_n)$, appends $(Y_{n+1}, \boldsymbol{x}_{n+1})$ to the context, and compares the expected updated prediction at $\boldsymbol{x}$ with its initial value. For the exact target, the tower property gives

$$\mathbb{E}_{Y_{n+1} \sim \pi(\cdot \mid \boldsymbol{x}_{n+1}, \boldsymbol{x}, d_n)}\big[\pi(A \mid \boldsymbol{x}, (d_n, (Y_{n+1}, \boldsymbol{x}_{n+1})))\big] = \pi(A \mid \boldsymbol{x}, \boldsymbol{x}_{n+1}, d_n).$$

Unlike the diagnostic in Eq. (3), this identity conditions the label sampler on the tracked query and retains the additional feature evidence on the right-hand side. The feature arguments represent distinct rows even when their values coincide. Sampling the entire future row from its joint posterior predictive conditional on $(\boldsymbol{x}, d_n)$ recovers zero drift by another application of the tower property.

Cumulative drift in the experiments. For $\breve{\pi} \in \{\pi, \bar{\pi}\}$, fix an initial context d_n, a query $\boldsymbol{x}$, and a repeated future feature $\boldsymbol{x}^\star$. Starting from $D_n = d_n$, generate additional rows sequentially according to

$$Y_{n+k+1} \mid D_{n+k} \sim \breve{\pi}(\cdot \mid \boldsymbol{x}^\star, D_{n+k}), \qquad D_{n+k+1} = \big(D_{n+k}, (Y_{n+k+1}, \boldsymbol{x}^\star)\big).$$

Each appended row updates the context through its joint likelihood. For the label event $A = \{1\}$, the cumulative expected drift after K appended rows is

$$\Delta_K = \mathbb{E}[\breve{\pi}(1 \mid \boldsymbol{x}, D_{n+K}) \,|\, d_n, \boldsymbol{x}, \boldsymbol{x}^\star] - \breve{\pi}(1 \mid \boldsymbol{x}, d_n).$$

where the expectation is over future labels generated from $\breve{\pi}$. The feature $\boldsymbol{x}^\star$ is either the query itself or one context feature selected uniformly once per context-query pair and held fixed throughout the rollout. The cumulative-drift panel of Fig. 4 reports the root mean square of these differences across $M = 2000 \cdot 8 = 16000$ context-query pairs:

$$\mathrm{RMS}_{\breve{\pi}}(K) = \left[\frac{1}{M}\sum_{i=1}^{M}\big(\Delta_K^{(i)}\big)^2\right]^{1/2},$$

where $\Delta_K^{(i)}$ denotes Δ_K evaluated for the ith context–query pair and its selected future feature. Thus, the expectation over future labels is taken before squaring, and the square root returns the

statistic to probability units. Squaring prevents positive and negative expected drifts from cancelling across pairs, while giving larger drifts more weight. The statistic measures the cumulative expected change from the initial prediction; it does not sum squared one-step changes.

For binary labels at a repeated feature, the predictive after k appended rows depends on their labels only through the number of class-1 outcomes. We therefore propagate the probabilities of the $k+1$ possible counts, integrating over all future label sequences without Monte Carlo sampling.

B Remarks and further details

Details on the amortization gap Following Hülskamp et al. (2026), we use *amortization gap* for the total excess of a PFN over the Bayes-optimal predictor, covering both the limits of the model class and those of finite pretraining. Pointwise, it is $\Delta_{\boldsymbol{\theta}}(\boldsymbol{x}, d_n) = D_{\mathrm{KL}}\big(\pi(\cdot \mid \boldsymbol{x}, d_n) \,\|\, q_{\boldsymbol{\theta}}(\cdot \mid \boldsymbol{x}, d_n)\big)$. During pretraining, the context size $N \sim \Pi_N$ is drawn independently of the task $p \sim \Pi$. For the random-size context $D_N = (Y_i, \boldsymbol{X}_i)_{i=1}^N$, write $\mathcal{L}(\boldsymbol{\theta}) = -\mathbb{E}[\log q_{\boldsymbol{\theta}}(Y \mid \boldsymbol{X}, D_N)]$ and $\mathcal{L}^* = -\mathbb{E}[\log \pi(Y \mid \boldsymbol{X}, D_N)]$, where $\mathbb{E} = \mathbb{E}_{\Pi_N}\mathbb{E}_{\Pi}$ averages over context sizes, tasks, and data. Since π minimizes the pretraining loss (Proposition A.1), its expectation equals $\mathcal{L}(\boldsymbol{\theta}) - \mathcal{L}^*$ with $\mathcal{L}^* = -\mathbb{E}_{\Pi_N}\mathbb{E}_{\Pi}[\log \pi(Y \mid \boldsymbol{X}, D_N)]$. Both statements require π as the reference, though under a non-separable prior, a reference computed from the context posterior alone yields $\bar{\pi}$ instead. An example would be MCMC draws from $\Pi(\cdot \mid d_n)$ that are not reweighted by the query's feature likelihood $p_X(\boldsymbol{x})$. The discrepancy to this reference then decomposes as

$$q_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, d_n) - \bar{\pi}(y \mid \boldsymbol{x}, d_n) = \big(q_{\boldsymbol{\theta}}(y \mid \boldsymbol{x}, d_n) - \pi(y \mid \boldsymbol{x}, d_n)\big) + \Delta_\pi(y \mid \boldsymbol{x}, d_n),$$

so the predictive gap Δ_π (Corollary 1) is added on top of the approximation error, and any divergence or loss difference computed against $\bar{\pi}$ mixes the two. On real data, labels do not follow π, so even with the correct reference the loss difference is no longer a divergence and can be negative (cf. Hülskamp et al. (2026)).

Counterexample for martingale zero-drift under non-separable priors Consider two equally likely tasks on $X \in \{0, 1, 2\}$ with feature probabilities $(0.5, 0.4, 0.1)$ and $(0.5, 0.1, 0.4)$, and class 1 probabilities at $X = 0$ of 0.2 and 0.8, respectively. At $X = 1, 2$, let both tasks assign class 1 probability 0.5. This prior is non-separable, but fixing the query and all future features at zero yields a martingale: their common feature likelihood 0.5 cancels from every update, leaving ordinary Bayesian label updates. Non-separability therefore permits drift without requiring it at every query.

The two terms can also cancel while both are nonzero. With no initial context, take two equally likely tasks on $X \in \{0, 1\}$ with $p_X(0)$ equal to 0.8 and 0.2, and deterministic labels $Y = 1$ and $Y = 0$, respectively. At query and future feature $x = 0$, the initial class 1 probability is $4/5$ and conditioning on both feature rows gives $16/17$. The sampled label identifies the task, so the feature-update and sampling terms are $12/85$ and $-12/85$: the net drift is zero.

Transductive channels in current benchmarks and models Currently, the handling of test data differs substantially between different benchmark suites as well as between different models. Below we summarize important aspects with regard to possible transductive channels in some of the most popular benchmarks and models contained in them:

- *TabArena* (Erickson et al., 2026) passes the full test fold to the model. It has no written rule on transductive use. Its only check is a smoke test that compares the prediction for a single row alone and within the batch, and methods can be exempted from it.
- *TALENT* (Liu et al., 2025) processes test minibatches in its neural-model evaluation routine, so a prediction can depend on the composition of the batch.
- *ScoringBench* (Landsgesell et al., 2026) passes all test features jointly to its predictive-distribution interface.
- *BeyondArena* (Purucker et al., 2026) explicitly documents transductive preprocessing for label-per-group tasks. It computes feature summaries within each test group and attaches them to that group's rows (Purucker et al., 2026, Appendix C.2). This reflects tasks whose prediction unit is naturally a group and does not by itself make comparisons unfair.

On the model side, the commonly used attention masks, for example in TabPFN and TabICL, prevent direct attention between test rows. TabICL also excludes test rows from the shared context of its column embedding by default and includes them only with `embed_with_test=True`. While these restrictions blocks large interactions, they do not make the complete pipeline independent of the other test rows:

- The standard forward path of TabPFN-v2.5 computes constant-feature masks and feature-group normalization parameters from the combined training and test input.
- TALENT's bundled LimiX implementation (Zhang et al., 2025a) performs feature filtering, categorical-type inference, and ordinal encoding on the combined training and test features.
- TALENT's default SAINT configuration enables intersample attention, so its predictions explicitly depend on the other rows in the test minibatch (Somepalli et al., 2021).
- Proprietary models served through an API, such as TabPFN-Thinking, receive the whole test fold server-side and can only be probed as a black box for batch invariance.

In total, an evaluation protocol should specify the prediction unit, which unlabeled observations are available and when, and how each method uses them, including any dependence on the composition of the test batch. Per-row prediction calls enforce invariance only if preprocessing and retained state have not already used other test rows. When transductive inference is permitted, all methods should receive the same information, although they need not exploit it identically.

C Experimental details

C.1 Ordered-Cluster Prior

Prior We introduce the *ordered-cluster prior* that lets us study our theoretical claims empirically, where we pretrain TabICL models from scratch on this custom prior. It is a finite non-separable classification prior over task distributions $p_{a,b}$, where the centers $a < b$ are chosen uniformly from the 36 ordered pairs on the grid $\{-4, \dots, 4\}$. For each row, the class is drawn uniformly, with class 0 generating features from $\mathcal{N}(a, 0.5^2)$ and class 1 from $\mathcal{N}(b, 0.5^2)$. Because the same cluster centers determine both the feature distribution and the labeling mechanism, an unlabeled query feature can provide information about its label. More precisely, conditioning on a query $\boldsymbol{x}$ changes the label prediction exactly when its likelihood under the posterior tasks is correlated with their class probability at $\boldsymbol{x}$ (see Eq. (2)). This effect is particularly pronounced when the context contains only one class. In this case, the corresponding center is observed, while the other remains uncertain, so the query location helps identify the missing center. This is also the effect illustrated in Fig. 1. The finite task space makes both the query-updated predictive π and the context-updated predictive $\bar{\pi}$ exactly enumerable, providing a controlled setting in which to determine which predictive standard pretraining recovers.

Fig. 2: We train a 2.6M parameter version of TabICL on the non-separable ordered-cluster prior using two different query-labeling schemes: standard pretraining and a decoupled-query control, denoted `decoupled`. In both schemes, we sample a latent task p and use it to generate the context d_n and query features $\boldsymbol{x}$. Under standard pretraining, the query labels are generated from the same task p, so their conditional distribution is the query-updated predictive $\pi(y \mid \boldsymbol{x}, d_n)$. Under the decoupled setting, we instead draw a new task p' from the context posterior $\Pi(\cdot \mid d_n)$, without conditioning on the query features, and generate each query label from $p'(\cdot \mid \boldsymbol{x})$. Conditional on the context, this decouples the query feature from the task used to generate its label. The resulting marginal label distribution is therefore the context-updated predictive $\bar{\pi}$ rather than $\pi(y \mid \boldsymbol{x}, d_n)$. The two schemes thus receive identical contexts and query features but have different Bayes-optimal targets: π for standard pretraining and $\bar{\pi}$ for the decoupled setting. At context size $n = 2$, the figure plots the network difference $q_1 - \bar{\pi}_1$ against the exact target difference $\pi_1 - \bar{\pi}_1$ (predictives for class 1), averaged over three training seeds after 50 000 optimizer steps. For each query both coordinates are differences against the same baseline: the exact difference between the two predictives of Eq. (1), $\pi(1 \mid \boldsymbol{x}, d_n) - \bar{\pi}(1 \mid \boldsymbol{x}, d_n)$ (horizontal), and the network's own difference $q_{\boldsymbol{\theta}}(1 \mid \boldsymbol{x}, d_n) - \bar{\pi}(1 \mid \boldsymbol{x}, d_n)$ (vertical); a network at π therefore lies on the dashed diagonal and one at $\bar{\pi}$ on the dotted horizontal. Queries are grouped into 20 bins of equal width in the exact difference, with the bin edges running from its 0.5th to its 99.5th percentile; bins holding fewer than 20 queries are dropped,

and each marker averages both coordinates over its bin, with the band giving one standard deviation across the three training seeds. Standard pretraining closely follows the identity line, indicating recovery of π, whereas the decoupled setting remains near zero, indicating recovery of $\bar{\pi}$. This means that pretraining on a non-separable prior actually approximates the query-updated predictive π of Eq. (1) and in this experiment, using the wrong reference can make the apparent approximation error roughly 95 times larger than the true one.

The two panels differ for a simple reason. In a single-class context, all labeled examples come from one cluster, so the location of the other cluster remains uncertain. The query feature is then valuable unlabeled evidence about that unseen cluster and can substantially change the prediction, producing a large difference between π and $\bar{\pi}$. In a two-class context, examples from both clusters already reveal both centers and the query adds little new information, so the target difference, and hence the plotted range, is much smaller (y-axis).

Training details All networks are TabICL architectures (Qu et al., 2025) trained from scratch on the prior described above and consist of a column embedding, row interaction, and an in-context transformer. Two sizes, 2.6M parameters (embedding 64, blocks 2/2/4) and 8.1M (embedding 96, blocks 2/2/6). We use the Muon optimizer (Jordan et al., 2024) with standard cross-entropy loss, learning rate $8 \cdot 10^{-4}$, cosine schedule with restarts and 2% warmup, gradient clipping at 10, a batch size of 64 datasets, and 50 000 optimizer steps, with one pass over freshly simulated datasets.

Semi-supervised gains on the ordered-cluster prior On the ordered-cluster prior, the semi-supervised target $\pi(y \mid \boldsymbol{x}, \boldsymbol{v}, d_n)$ of Corollary A.1 can be computed exactly, which lets us check whether a trained PFN realizes the gain from unlabeled rows. We insert k unlabeled rows $\boldsymbol{v}$ from the generating task between the context and the queries. We pretrain TabICL with access to them either through the in-context transformer or through the column embedding (`embed_with_test`), and train a control with $\boldsymbol{v}$ masked. At $n = 2$, the unlabeled rows lower the Bayes-optimal loss by 11%, 34% and 38% for $k = 1, 8, 32$. For $k \geq 8$ and $n \leq 4$, the expected divergence $D_{\mathrm{KL}}(\pi(\cdot \mid \boldsymbol{x}, \boldsymbol{v}, d_n) \,\|\, q_{\boldsymbol{\theta}})$ of the in-context network stays below 2% of this gain, and that of the column-embedding network below 5%. The masked control stays at π. Both networks approximate their own targets closely, so their loss difference, which matches the gain to within 0.2%, reflects the larger information set rather than better approximation. For a more thorough experiment regarding the use of additional unlabeled queries, refer to Appendix C.3.

C.2 Feature effects under a prior with constant class probability

Prior Figure 3 is computed exactly on a finite non-separable classification prior with two tasks p_s, $s \in \{-1, 1\}$, each with prior probability $1/2$. Under p_s the three features are independent. X_1 and X_2 take values in $\{-2, \ldots, 2\}$ with $p_s(X_1 = t) \propto \exp\{-(t - 0.75s)^2/2\}$ and $p_s(X_2 = t) \propto \exp\{-(t - 0.5s)^2/2\}$, and X_3 is uniform on $\{-1, 1\}$. The label does not depend on the features, $Y \mid \boldsymbol{X} \sim \mathrm{Bernoulli}(c_s)$ with $c_{-1} = 0.2$ and $c_1 = 0.8$. Every task therefore has a constant conditional label distribution, whereas X_1 and X_2 carry information about the task and hence about c_s. X_3 has the same distribution under both tasks and serves as a negative control. All 50 feature combinations have positive probability under both tasks, so every input evaluated by PD, ICE and SHAP lies in the support.

Predictives Let $w = \Pi(s = 1 \mid d_n)$. By the symmetry of the grid, the normalizing constants cancel and $\log\{r_{p_1}(\boldsymbol{x})/r_{p_{-1}}(\boldsymbol{x})\} = 1.5x_1 + x_2$, so that

$$\pi(1 \mid \boldsymbol{x}, d_n) = 0.2 + 0.6\,\sigma\big(\mathrm{logit}\, w + 1.5x_1 + x_2\big), \qquad \bar{\pi}(1 \mid \boldsymbol{x}, d_n) = 0.2 + 0.6\, w,$$

where σ is the logistic function. The figure uses the context d_n with $n = 2$ rows, $\boldsymbol{x}_1 = (-1, 1, -1)$ with $y_1 = 0$ and $\boldsymbol{x}_2 = (1, -1, 1)$ with $y_2 = 1$. Both the feature and the label likelihood ratios of this context equal one, so $w = 1/2$ and $\bar{\pi}(1 \mid \boldsymbol{x}, d_n) = 0.5$ for all $\boldsymbol{x}$. The context was chosen for illustration, not drawn from the prior.

Explanations All methods are applied to $f(\boldsymbol{x}) = \pi(1 \mid \boldsymbol{x}, d_n)$ and to $\bar{\pi}(1 \mid \boldsymbol{x}, d_n)$ with the same background distribution, the feature predictive $B = m(\cdot \mid d_n)$. In this case it is the equal-weight mixture of the two feature distributions. All quantities are finite sums over the grid, and no network

is trained. The partial dependence of X_1 is $\mathrm{PD}_1(t) = \mathbb{E}_B[f(t, X_2, X_3)]$, with (X_2, X_3) drawn from their joint distribution under B. The ICE curves are $t \mapsto f(t, x_2, x_3)$ for $x_2 \in \{-2, \dots, 2\}$ (Goldstein et al., 2015). Since f does not depend on x_3, each x_2 yields a single curve. Exact Shapley values are obtained by enumerating all 2^3 coalitions at each of the 50 grid points for two value functions. The marginal game uses $v_{\boldsymbol{x}}(S) = \mathbb{E}_B[f(\boldsymbol{x}_S, \boldsymbol{X}_{\bar{S}})]$, where $\boldsymbol{X}_{\bar{S}}$ follows its joint marginal under B. The conditional game uses $v_{\boldsymbol{x}}(S) = \mathbb{E}_B[f(\boldsymbol{X}) \mid \boldsymbol{X}_S = \boldsymbol{x}_S]$ (Janzing et al., 2020). The right panel reports $\mathbb{E}_{\boldsymbol{X}\sim B}|\phi_j(\boldsymbol{X})|$, which for π equals 0.134 and 0.073 for X_1 and X_2 under the marginal game and 0.133 and 0.084 under the conditional game. Because $\bar{\pi}$ is constant, its PD and ICE curves are constant and its Shapley values vanish under both games. The attribution of X_3 under π is zero in both games as well, since f does not depend on x_3 and X_3 is independent of (X_1, X_2) under B.

C.3 SemiTabICL: Architectural details and evaluation

Architecture SemiTabICL uses the three-stage TabICL architecture with a column embedding (2 blocks, embedding dimension 128, 128 inducing points), a row interaction (2 blocks, 3 CLS tokens) and an in-context transformer (6 blocks, width $3 \cdot 128 = 384$), each with 8 attention heads, for 8.5M parameters in total. Every dataset is laid out as $[\, d_n \mid \boldsymbol{v} \mid \text{queries} \,]$ with n labeled, k unlabeled and m query rows. Wherever TabICL attends to the training rows, i.e. in the inducing-point attention of the column embedding and in the in-context transformer, the attention context consists of the first $n + k$ rows, so labeled and unlabeled rows attend to each other and queries attend to both. Unlabeled rows carry a single learned vector in place of the label embedding. The loss is the cross-entropy on the query rows only. The baseline TabICL is identical except for this vector and never sees unlabeled rows.

Prior Both models are trained on the unchanged graph-SCM prior of TabICLv2 (Qu et al., 2026) (random DAGs with 2–32 nodes, 1–100 features, up to 10 classes, predictability filters). The unlabeled rows are further samples of the same SCM whose labels are dropped. For SemiTabICL, every dataset has such a pool: the labeled rows make up 30% of a sequence length drawn log-uniformly from $[14, 334]$, so $n \in [4, 100]$ and $m \approx 2.3\, n$, and the pool has $k = \mathrm{round}(r\, n)$ rows with r log-uniform on $[1, 9]$. The number of classes is capped at $\max(2, \lfloor \min(n, m)/2 \rfloor)$. The median synthetic dataset has $n = 20$, $k = 59$ and $m = 47$ rows. The baseline is trained on the same distribution of (n, m) without a pool.

Training We run pretraining for 30 000 optimizer steps, 1% warmup, with bfloat16 precision, and batches of 64 datasets in groups of 16 that share their sequence length and pool size. Each model trains in about two hours on four NVIDIA H200 GPUs.

Compared methods All methods use an ensemble of 8 members and fit preprocessing on the labeled rows only, and all see identical splits. We compare SemiTabICL with its additional unlabeld context rows (*SemiTabICL*), SemiTabICL with a the unlabeled context whose columns are permuted independently, which keeps the marginals but destroys the joint structure (*shuffled pool*), the baseline TabICL on the labeled rows (*TabICL*), and self-training (ST) with scikit-learn's `SelfTrainingClassifier` (Pedregosa et al., 2011; Yarowsky, 1995) (confidence threshold 0.75, at most 10 rounds) around TabICL and around a gradient-boosted tree classifier (with default settings). As an upper bound, the oracle gives TabICL the pool together with its labels. The purpose of the shuffled unlabeled context is to assess to what extent possible gains from the additional unlabeled context come from only refining the marginal distributions per column vs. the model actually learning to leverage the full distribution of the additional data.

semi-TALENT We use the 150 classification datasets of TALENT (Liu et al., 2025) with at most 100 features and 10 classes, excluding the subset reserved for validation during pretraining. From the official training split, we draw a stratified labeled sample of $n \in \{16, 32, 64, 96\}$ rows and a pool of $k = r\, n$ further rows with $r \in \{2, 4, 8, 10\}$, where pools of one labeled sample are nested. Each budget is repeated five times and evaluated on the official test split. On average over datasets, TabICL reaches 72.6% accuracy, SemiTabICL 73.3% and the oracle 77.9%.

semi-UCI Following Guo et al. (2010), we use their 26 UCI datasets with raw features and repeat a 4-fold stratified cross-validation 10 times. In each outer training fold, 10% of the rows are labeled and the remainder forms the pool, so that $k = 9n$, and the held-out fold is the test set. Two datasets with more than 10 classes (letter, vowel) and one with a preprocessing failure (sick) are excluded, leaving 23 datasets. Grouped by the categories of Schölkopf et al. (2012), the structural gain is larger on the 11 anticausal or confounded datasets (+0.70 pp) than on the three causal ones (+0.30 pp), in line with their hypothesis. With only three causal datasets, however, the difference is not significant (exact permutation test, $p = 0.42$).

Table 1: Paired comparisons of SemiTabICL and its controls. Δacc is the difference in accuracy (percentage points) and ΔNLL the reduction in log loss (nats), both positive when the first method is better. For each dataset, differences are averaged over all its splits; for semi-TALENT, these pool the full 4×4 grid of $n \in \{16, 32, 64, 96\}$ and $r \in \{2, 4, 8, 10\}$ with five repetitions each. Reported values are means of these per-dataset differences. For semi-TALENT, win is the share of the 150 datasets with a positive difference, and p is a two-sided Wilcoxon signed-rank test of whether the per-dataset differences are centred at zero.

	semi-TALENT						semi-UCI	
Comparison	Δacc	win	p	ΔNLL	win	p	Δacc	ΔNLL
SemiTabICL vs. TabICL	+0.63	0.81	5e−16	+0.015	0.85	6e−19	+0.48	+0.014
SemiTabICL vs. shuffled	+0.66	0.67	1e−10	+0.017	0.65	2e−7	+0.47	+0.017
Shuffled vs. TabICL	−0.03	0.59	0.56	−0.001	0.63	0.06	+0.01	−0.003
TabICL ST vs. TabICL	−0.33	0.41	3e−4	−0.440	0.03	6e−26	−0.53	−0.393
GBDT ST vs. GBDT	+0.45	0.77	3e−14	−0.577	0.01	2e−26	+0.46	−0.452
Oracle vs. TabICL	+5.18	1.00	2e−26	+0.125	1.00	2e−26	+5.33	+0.120

C.4 Software and computational environment

Our experiments were implemented on top of the official TabICL software repository (Qu et al., 2025; 2026) based on `pytorch` (Paszke et al., 2019). TabICL pretraining runs were carried out on NVIDIA H200 GPUs with a full pretraining run with up to 50k optimizer steps taking less than 3 hours of wall-clock time for both the 2.6M parameter model as well as the 8.1M and 8.5M parameter model.

We make our code available at: https://anonymous.4open.science/r/SeparabilityGap-ICLR2027-52F0/README.md

D Related work

Semi-supervised learning (in ICL) Schölkopf et al. (2012) point out the importance of analyzing the causal mechanisms underlying datasets for supervised learning and highlight under what conditions additional unlabeled data should prove useful with semi-supervised approaches. Li et al. (2026) show, from an expressivity perspective, that linear attention needs depth to exploit unlabeled in-context rows under a binary Gaussian-mixture prior, which is a special case of what we call a non-separable prior. Though they mention TabPFN as an application, the question studied is orthogonal to ours: we ask whether a network pretrained on a non-separable prior natively exploits additional unlabeled rows, as its objective says it should, whereas they wrap an off-the-shelf TabPFN in an external pseudo-labeling loop and show that self-training can improve predictive performance. Swelam et al. (2025) study whether the internal representations of TabPFN contain causal information and how well these can be repurposed for causal discovery by training an additional decoder. Their positive findings provide adjacent evidence that predictive pretraining yields feature representations that not only identify the latent mechanism $p(y \mid \boldsymbol{x})$ but also $p(\boldsymbol{x})$ as required for the query-updated predictive of Eq. (4).

Bayesian targets and direct reference comparisons. Müller et al. (2022) introduce PFNs and show that prior-data NLL equals the expected KL to the PPD up to the Bayes-optimal loss, so held-out NLL measures the amortization gap whenever that loss is known, which is measured for a GP

with fixed hyperparameters and a Gaussian target whose mean is linear in the context labels. In these settings the PFN approaches the exact NLL with more pretraining. Furthermore, Müller et al. (2024) extends the analysis by enumerating finite libraries of one-dimensional step, sine, and linear functions and show on selected datasets that predictive means follow the exact posterior, including non-obvious generalizations. This setup also separates failures of the posterior from failures of the network, such as an inability to count repeated observations. For intractable priors, the existing comparisons rank a PFN against NUTS, variational inference, or MLE-II (Müller et al., 2022), primarily designed as speed comparisons and are very informative as such, but a Monte Carlo reference whose NLL never falls below the PFN's cannot locate the Bayes-optimal loss, so the gap under a general prior has not been measured. The later literature has largely turned to predictive performance on real benchmarks (Grinsztajn et al., 2025; Hollmann et al., 2025; Qu et al., 2025; 2026), which establishes that PFNs predict well but not what they compute.

Neural processes Neural processes (NPs) share the PFN goal of learning to predict from a context set across a distribution of tasks. In particular, PFNs are closely related to transformer NPs, a transformer-based extension of conditional NPs (Nagler, 2023; Nguyen & Grover, 2022). Conditional NPs directly parameterize predictive distributions (Garnelo et al., 2018a), while latent NPs introduce a shared latent variable to model dependencies between predictions (Garnelo et al., 2018b). Attentive neural processes use query-specific context representations (Kim et al., 2019), and transformer neural processes formulate prediction through likelihood-based sequence modeling (Nguyen & Grover, 2022). Closely related to our target-identification question, Foong et al. (2020) show, under their stated conditions, that the population maximum-likelihood objective recovers the ground-truth stochastic process's prediction map. This supports interpreting learned predictions as approximations to conditional distributions, without guaranteeing exact Bayesian inference for a finite trained network.