# AMORTIZING LABEL SHIFT FOR IN-CONTEXT TABULAR PREDICTION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Prior Fitted Networks have emerged as a disruptive paradigm enabling the pre-training of Tabular Foundation Models (TFMs) entirely on synthetic data. While these models have proven successful across a broad variety of real-world predictive tasks, they still make strong assumptions about the train and test distributions being fixed. This assumption is often unrealistic, as many applications exhibit substantial distribution shifts. In particular, a drawback of current TFMs for real-world deployment is their vulnerability to label shift, where class proportions change between train and test time. In this work, we introduce ShiftICL, the first Tabular Foundation Model that explicitly amortizes label-shift adaptation during pre-training. ShiftICL is trained on synthetic tasks spanning a broad range of label distributions and shift severities, enabling it to adapt its predictive distribution to previously unseen label shifts without retraining or access to test labels. Across a large collection of 209 OpenML tabular datasets under controlled label shift, ShiftICL substantially improves predictive accuracy and calibration over existing state-of-the-art methods tailored for distribution drifts, with gains becoming increasingly pronounced as the shift grows.

## 1 INTRODUCTION

Tabular Foundation Models (TFMs) have gained considerable popularity in recent years due to their ability to perform in-context learning of classification and regression tasks with no task-specific gradient updates. These models have demonstrated state-of-the-art performance in a broad repertoire of tasks and datasets(Erickson et al., 2026), alongside widespread adoption across industries and scientific applications (Grinsztajn et al., 2026). However, they inherit a well known challenge of model deployment: assuming the training and test sets come from the same distribution is usually a strong hypothesis.

A more realistic scenario involves a *distribution shift* between training and test data. For example, a flu diagnostic model might be trained on hospital records collected during a summer period of low disease prevalence. When deployed during a winter outbreak, it may encounter a substantially larger proportion of patients with flu. If we assume that the distribution of symptoms conditional on disease status remains approximately unchanged, this is an example of *label shift*: the label distribution $p(\mathrm{y})$ changes between training and test data, while the class-conditional distribution $p(\mathbf{x} \mid \mathrm{y})$ remains fixed (Lipton et al., 2018). The same symptoms can then imply different probabilities of flu, even though their distribution within each class has not changed. When class proportions change at deployment, even a model that accurately recovers the training posterior need not provide the right predictive distribution at test time. Classical approaches estimate changes in class proportions from unlabeled test data and use those estimates to correct classifiers (Saerens et al., 2002; Lipton et al., 2018; Garg et al., 2020). For TFMs, Lee et al. (2026) propose test-time posterior adjustments that reweight predicted class probabilities without changing the architecture or requiring additional training.

These approaches motivate a complementary question central to this work: *can label-shift adaptation be learned during pre-training, rather than applied as a separate correction to an existing predictor?* In this paper, we frame prediction under label shift as an in-context Bayesian inference problem. As in existing label shift correction methods, the key observation is that the unlabeled test features provide evidence about the change in class proportions(Lee et al., 2026; Lipton et al., 2018).



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.

This information, together with the labeled training set, can identify the test predictive distribution under certain conditions(Garg et al., 2020). While existing methods rely on post-hoc adjustments to an existing predictor, we investigate whether this adaption can be learned during pre-training. Following Müller et al. (2022), we train ShiftICL to approximate the corresponding Bayesian posterior predictive distribution using synthetic episodes. Within each episode, a stochastic shift kernel varies class proportions between datasets while preserving their class-conditional distributions. This incorporates label-shift adaptation directly into pre-training, allowing ShiftICL to adjust its predictions without access to test label information and in a single forward pass.

In the overall, our contributions are the following. We introduce ShiftICL, a tabular foundation model explicitly pre-trained to adapt to changing class proportions. ShiftICL can handle datasets suffering from different levels of label shift and produce adapted predictions in a single forward pass with no test labels, parameter updates, or separate post-hoc correction mechanism. To the best of our knowledge, this is the first formulation of prediction under label shift as approximate Bayesian inference. Bulding on existing results on label shift identifiability, we give sufficient conditions for the test predictive distribution to be identifiable from the training set law and unlabeled set marginal. We also characterize the asymptotic behavior of the corresponding Bayesian posterior predictive distribution and show that, under certain conditions, it recovers the true test conditional distribution. Finally, we empirically demonstrate improvements in predictive accuracy and calibration across a broad collection of real-world tabular datasets with controlled label shifts.

## 2 PRELIMINARIES

We follow the convention that $\mathrm{x}$ and $\mathrm{y}$ denote random variables, while $\boldsymbol{x}$ and $y$ denote their realizations. Let $\mathcal{X}$ denote the feature domain and $\mathcal{Y} = \{1, \dots, K\}$ the label space. We consider a labeled training, or *context*, dataset $\mathcal{D}_n \in (\mathcal{X} \times \mathcal{Y})^n$ and an unlabeled test, or *query*, set $\boldsymbol{X}_{1:m} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)$. In the conventional pointwise formulation, a pretrained in-context model returns class probabilities $q(y \mid \boldsymbol{x}_j, \mathcal{D}_n)$ without task-specific parameter updates (Hollmann et al., 2023).

Let $\phi$ denote the latent data-generating task. Under label shift, the context and query distributions share the same class-conditional law $p_\phi(\mathbf{x} \mid \mathrm{y})$ but may have different label marginals. More specifically, we consider joint law factorizations of the form:

$$\begin{aligned} p_{\mathrm{ctx}}(\mathbf{x}, \mathrm{y}) &= \pi_{\mathrm{ctx}}(\mathrm{y})\, p_\phi(\mathbf{x} \mid \mathrm{y}), \\ p_{\mathrm{qry}}(\mathbf{x}, \mathrm{y}) &= \pi_{\mathrm{qry}}(\mathrm{y})\, p_\phi(\mathbf{x} \mid \mathrm{y}), \end{aligned} \tag{1}$$

where in general $\pi_{\mathrm{qry}} \neq \pi_{\mathrm{ctx}}$. We further assume $\mathrm{supp}(\pi_{\mathrm{qry}}) \subseteq \mathrm{supp}(\pi_{\mathrm{ctx}})$, so that no query class lies outside the context distribution's support, and no access to query labels at inference time.

### 2.1 RELATED WORK

#### 2.1.1 LABEL SHIFT CORRECTION

Many algorithms that address label shift rely on correction mechanisms. In general, they aggregate the information given by the unlabeled test set to adjust the model's predictions (Saerens et al., 2002; Lipton et al., 2018; Azizzadenesheli et al., 2019; Garg et al., 2020). More recently, Lee et al. (2026) introduced DistPFN, a mechanism that is specifically tailored for TFMs and that does not require any parameter updates. This method avoids estimating the query set prior constructing a test-time adjustment from the empirical context prior and model predictions. In its default implementation, DistPFN averages the model predictions over the full query set, although an instance-wise variant is also discussed. This average provides a useful signal to correct for the label shift. However, this quantity is in general different from the true query class proportions, even for an exact context posterior. Our focus is instead on learning this adaptation during pre-training explicitly, so that query-population information enters the predictive model itself.

#### 2.1.2 PRIOR-FITTED NETWORKS AND TABULAR FOUNDATION MODELS

Müller et al. (2022) introduce Prior-Data Fitted Networks (PFNs), which approximate Bayesian posterior prediction through pre-training on synthetic tasks. This framework led to TFMs, whose

first example, TabPFN, was introduced in the seminal work of Hollmann et al. (2023). TabPFN applies prior fitting to tabular predictive tasks using synthetic priors that incorporate structural causal models. Subsequent TabPFN and TabICL models broaden task coverage and improve scalability through changes to their architectures, synthetic priors, and pre-training procedures (Hollmann et al., 2025; Qu et al., 2025; 2026). A complementary direction is represented by TabDPT, which pre-trains in-context tabular predictors on real datasets (Ma et al., 2025).

Other extensions tailor prior fitting to specific observation mechanisms or distribution changes. Drift-Resilient TabPFN models temporal changes through a secondary SCM that varies the primary SCM's parameters, and conditions prediction on supplied temporal domain indices (Helli et al., 2024). SurvivalPFN applies prior fitting to right-censored survival data (ang Qi et al., 2026). Our work studies label-shift adaptation without temporal indices, combining a shifted task prior with transductive query-set conditioning.

## 3 AMORTIZING LABEL SHIFT

The goal of this work is to extend the pre-training mechanism of TFMs to explicitly account for label shift. More specifically, we aim to enable in-context learning when the label distribution of the context may differ from that of the query set. Following the notation introduced in the previous section, we consider a labeled context set $\mathcal{D}_n$ of $n$ rows and an unlabeled query set $\boldsymbol{X}_{1:m}$ of $m$ rows $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, whose latent labels $y_1, \ldots, y_m$ belong to one of $K$ classes.

We assume the label-shift setting defined in Section 2: the label marginal may differ between context and query, while the class-conditional distribution $p_\phi(\boldsymbol{x} \mid y)$ remains fixed. We further assume $\mathrm{supp}(\pi_{\mathrm{qry}}) \subseteq \mathrm{supp}(\pi_{\mathrm{ctx}})$, so that the query contains no classes outside the source support.

Let $\Phi$ denote a prior over latent source tasks, with $\phi \sim \Phi$. We model the unknown query label distribution as a latent variable

$$\pi_{\mathrm{qry}} \sim \Pi(\cdot \mid \phi),$$

where $\Pi$ is a shift kernel over target label distributions conditional on the source task.

Under the label-shift factorization, marginalizing the unobserved query labels gives

$$p_{\phi,\pi_{\mathrm{qry}}}(\boldsymbol{x}) = \sum_y \pi_{\mathrm{qry}}(y)\, p_\phi(\boldsymbol{x} \mid y)\,.$$

Assuming the query rows are conditionally i.i.d. given $(\phi, \pi_{\mathrm{qry}})$,

$$p_{\phi,\pi_{\mathrm{qry}}}(\boldsymbol{X}_{1:m}) = \prod_{j=1}^{m} p_{\phi,\pi_{\mathrm{qry}}}(\boldsymbol{x}_j)\,.$$

Garg et al. (2020) give conditions under which $\pi_{\mathrm{qry}}$ is identifiable from the population query distribution. In particular, for fixed $\phi$, if the class-conditional distributions $\{p_\phi(\cdot \mid y)\}_{y=1}^K$ are linearly independent, then distinct query label distributions induce distinct query marginals. The context distribution alone is generally insufficient to determine the query predictive distribution. For fixed $\phi$, two distinct query label distributions $\pi_{\mathrm{qry}} \neq \pi'_{\mathrm{qry}}$ leave the context distribution unchanged, while generally inducing different predictive distributions

$$p_{\phi,\pi_{\mathrm{qry}}}(y \mid \boldsymbol{x}) \neq p_{\phi,\pi'_{\mathrm{qry}}}(y \mid \boldsymbol{x}).$$

A direct consequence of the identifiability result of Garg et al. (2020) is that, under label shift, support inclusion, and linear independence of the class-conditionals, the pair of population laws

$$p_{\mathrm{ctx}}(\mathbf{x}, \mathrm{y}) \qquad \text{and} \qquad p_{\phi,\pi_{\mathrm{qry}}}(\mathbf{x})$$

uniquely determines the query predictive distribution $p_{\phi,\pi_{\mathrm{qry}}}(\mathrm{y} \mid \mathbf{x})$. This idea is formalized in the following statement:

**Corollary 3.1** (Predictive identifiability under label shift)**.** *Let $(\phi, \pi_{\mathrm{qry}})$ and $(\phi', \pi'_{\mathrm{qry}})$ be two latent pairs satisfying*

$$\mathrm{supp}(\pi_{\mathrm{qry}}) \subseteq \mathrm{supp}(\pi^{\phi}_{\mathrm{ctx}}), \qquad \mathrm{supp}(\pi'_{\mathrm{qry}}) \subseteq \mathrm{supp}(\pi^{\phi'}_{\mathrm{ctx}}),$$

*and suppose that the class-conditional distributions* $\{p_\phi(\cdot \mid y)\}_{y \in \mathrm{supp}(\pi^\phi_{\mathrm{ctx}})}$ *are linearly independent.*

*If* $p_\phi(\boldsymbol{x}, y) = p_{\phi'}(\boldsymbol{x}, y)$ *and* $p_{\phi,\pi_{\mathrm{qry}}}(\boldsymbol{x}) = p_{\phi',\pi'_{\mathrm{qry}}}(\boldsymbol{x})$*, then*

$$p_{\phi,\pi_{\mathrm{qry}}}(y \mid \boldsymbol{x}) = p_{\phi',\pi'_{\mathrm{qry}}}(y \mid \boldsymbol{x})$$

*for* $p_{\phi,\pi_{\mathrm{qry}}}$*-almost every* $\boldsymbol{x} \in \mathcal{X}$ *and every* $y \in \mathcal{Y}$*.*

Corollary 3.1 is a population-level statement. In finite samples, however, the model does not observe the query marginal itself, but only samples from it. A conventional pointwise TFM predicting $y_i$ from $(\mathcal{D}_n, \boldsymbol{x}_i)$ therefore observes only a single draw from the target feature distribution, which generally leaves substantial uncertainty about the latent query prior $\pi_{\mathrm{qry}}$. By conditioning instead on $\boldsymbol{X}_{1:m}$, the model can aggregate evidence about the common target distribution across query rows. This motivates our transductive formulation: the query set acts as a finite-sample proxy for the population-level information that makes the target predictive identifiable.

Since both the source task $\phi$ and target label distribution $\pi_{\mathrm{qry}}$ are latent, prediction marginalizes uncertainty over both. The resulting Bayesian posterior predictive distribution (PPD) is

$$q^{\mathrm{B}}_{n,m}(y_i \mid \boldsymbol{X}_{1:m}, \mathcal{D}_n) = \iint p_{\phi,\pi_{\mathrm{qry}}}(y_i \mid \boldsymbol{x}_i)\, p(\phi, \pi_{\mathrm{qry}} \mid \mathcal{D}_n, \boldsymbol{X}_{1:m})\, d\phi\, d\pi_{\mathrm{qry}}. \tag{2}$$

Since $\boldsymbol{x}_i$ is the $i$-th row of $\boldsymbol{X}_{1:m}$, it need not appear separately in the conditioning arguments. We use Equation equation 2 as the Bayesian prediction target of our method. Following Müller et al. (2022), we amortize this predictor with a PFN trained on synthetic episodes drawn from the hierarchical task prior described above.

In the remainder of this section, we describe how we instantiate this prior in practice (Section 3.1), how we modify the PFN architecture to enable transductive query-set inference (Section 3.2), and finally establish consistency of the Bayesian PPD (Section 3.3).

### 3.1 Pre-training prior.

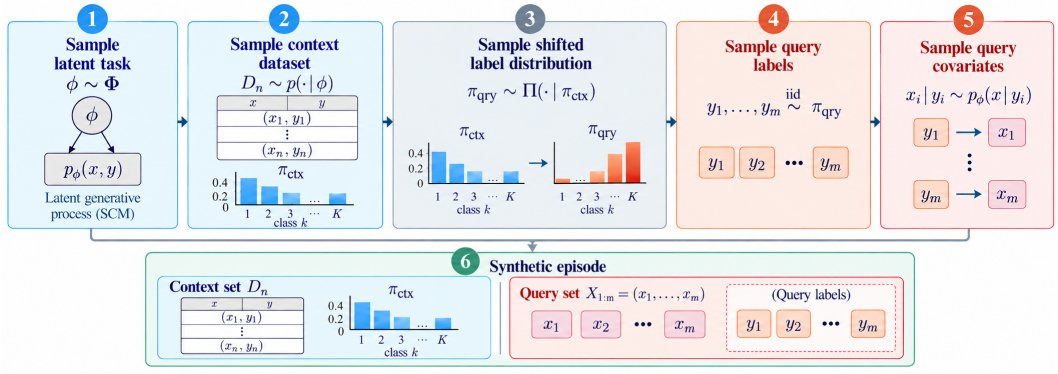



Figure 1: Synthetic data generation process under label shift.

Amortizing equation 2 hinges on a synthetic generation process that accounts for label shift, in order to generalize to such real-world scenarios. Figure 1 presents a schematic summary of our data generation process: we first sample a latent task and a context dataset, then a label distribution that is potentially different from that of the context, and finally, a query set whose class-conditional distribution is shared with the context set.

Let $\Phi$ denote a prior over latent data-generating processes. Following Hollmann et al. (2023), we instantiate $\Phi$ using a prior over Structural Causal Models (SCMs). In this context, sampling a task amounts to first drawing an SCM as $\phi \sim \Phi$. We can then sample a dataset $\mathcal{D}_n \sim p(\cdot \mid \phi)$ by injecting noise in the SCM's nodes and propagating the signal through its causal edges. In this work we leverage the `NanoTabICL` package (soda-inria, 2026) which includes a simplified implementation of TabICL's prior (Qu et al., 2025; 2026).

To incorporate label shift we perturb the class proportions of the context label distribution, with these perturbations being more or less pronounced so as to represent a broad range of shift severity. We generate candidate target label distributions by perturbing the context prior in log-odds space. For a task with $K$ classes, we use class $K$ as an arbitrary reference class[1] and define the log-ratio map

$$\eta : \mathrm{int}(\Delta^{K-1}) \to \mathbb{R}^{K-1}, \qquad \eta(\pi)_k = \log \frac{\pi(k)}{\pi(K)}, \quad k = 1, \ldots, K-1.$$

This gives a $K-1$ dimensional unconstrained representation of the simplex, with inverse

$$\eta^{-1}(z) = \mathrm{softmax}([z, 0]) \,,$$

where $[z, 0]$ concatenates a value 0 with the vector $z$. We then obtain a candidate query prior by adding a random perturbation $\omega \in \mathbb{R}^{K-1}$ to the context log-odds,

$$\eta(\pi_{\mathrm{qry}}) = \eta(\pi_{\mathrm{ctx}}) + \omega, \qquad \text{or equivalently} \qquad \pi_{\mathrm{qry}} = \eta^{-1}(\eta(\pi_{\mathrm{ctx}}) + \omega) \,. \tag{3}$$

Thus, the shift kernel operates additively in unconstrained log-odds coordinates while ensuring that the resulting query prior remains a valid probability distribution.

Once $\pi_{\mathrm{qry}}$ is obtained, query observations are generated as

$$\mathrm{y}_j \overset{\mathrm{iid}}{\sim} \pi_{\mathrm{qry}}, \qquad \mathbf{x}_j \mid \mathrm{y}_j \sim p_\phi(\mathbf{x} \mid \mathrm{y}_j) \,.$$

Thus, context and query share the same class-conditional distribution, while their label marginals may differ.

The distribution of the perturbation $\omega$ determines the family of label shifts seen during pre-training. We sample perturbations with varying magnitude and structure, including dense changes affecting all class log-odds, sparse changes affecting only a subset of classes, and occasional larger shifts. Importantly, the magnitude of a change in label proportions is not by itself a measure of adaptation difficulty: whether correcting the prior improves prediction also depends on the underlying class-conditional distributions. We therefore generate several candidate query priors for each task and use a lightweight surrogate to estimate their recoverable headroom, i.e. the improvement available from replacing the context prior with the candidate query prior while holding the class-conditionals fixed. Training episodes are selected to cover a broad range of such headroom values, while explicitly retaining no-shift episodes. Appendix A.2 gives the complete shift-kernel distribution, surrogate criterion, and curriculum parameters.

### 3.2 Architecture

To approximate the Bayesian predictor in equation 2, we enable each prediction to incorporate evidence from the full query set, yielding a transductive predictor. This does not require full query-to-query attention. Instead, we introduce a small inducing bottleneck in the backbone's row-interaction stage. Learned inducing vectors summarize the query representations, which are themselves conditioned on the labeled context. Writing $T_\psi(\mathcal{D}_n, \boldsymbol{X}_{1:m})$ for the resulting context-dependent summary, the predictor takes the form

$$q_\psi(y_i \mid \boldsymbol{x}_i, \mathcal{D}_n, T_\psi(\mathcal{D}_n, \boldsymbol{X}_{1:m})) \,.$$

Defined like this, the summary is invariant to permutations of the query rows. Appendix A.3 details the architecture.

The resulting predictor $q_\psi$ is trained across synthetic episodes to minimize the average query negative log-likelihood,

$$\mathcal{L}(\psi) = -\mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} \log q_\psi(y_i \mid \boldsymbol{x}_i, \mathcal{D}_n, T_\psi(\mathcal{D}_n, \boldsymbol{X}_{1:m}))\right],$$

where the expectation is over the pre-training episode distribution described above. Query labels are used only to evaluate this loss and are never provided to the forward pass. At inference time, ShiftICL receives only the labeled context $\mathcal{D}_n$ and unlabeled query set $\boldsymbol{X}_{1:m}$, and produces all query predictions in a single forward pass. In this work we parameterize ShifICL using TabPFNv3 architecture Grinsztajn et al. (2026). We provide more details in Appendix C.1.

[1] The choice of reference class is arbitrary.

### 3.3 CONSISTENCY OF THE BAYESIAN PPD

The network $q_\psi$ is trained to approximate the Bayesian PPD from equation 2. Defining $q_\psi^{(i)} = q_\psi(y_i \mid \boldsymbol{X}_{1:m}, \mathcal{D}_n)$, $q_{\mathrm{B}}^{(i)} = q_{n,m}^{\mathrm{B}}(y_i \mid X_{1:m}, D_n)$ and $p_\star^{(i)} = p_{\phi^\star,\pi_{\mathrm{qry}}^\star}(y_i \mid x_i)$, its error can be decomposed into two terms:

$$|q_\psi^{(i)} - p_\star^{(i)}| \le \underbrace{|q_\psi^{(i)} - q_{\mathrm{B}}^{(i)}|}_{\text{amortization error}} + \underbrace{|q_{\mathrm{B}}^{(i)} - p_\star^{(i)}|}_{\text{Bayesian estimation error}} . \tag{4}$$

Assuming the amortization error vanishes as $n, m \to \infty$, it suffices to establish consistency of the Bayesian PPD to bound the total error in equation 4. To build the consistency argument we need to account for the fact that even infinitely many pairs of context and unlabeled query sets may not identify a unique latent task $(\phi, \pi_{\mathrm{qry}})$. In other words, different pairs can induce the same observable laws. We therefore group pairs that induce the same observable distributions into equivalence classes. We formalize these ideas below.

Two latent pairs $(\phi, \pi_{\mathrm{qry}})$ and $(\phi', \pi'_{\mathrm{qry}})$ are observationally equivalent, written

$$(\phi, \pi_{\mathrm{qry}}) \sim (\phi', \pi'_{\mathrm{qry}}),$$

if they induce the same labeled context distribution and the same unlabeled query feature distribution:

$$p_\phi^S(\mathbf{x}, \mathrm{y}) = p_{\phi'}^S(\mathbf{x}, \mathrm{y}), \qquad p_{\phi,\pi_{\mathrm{qry}}}^X(\mathbf{x}) = p_{\phi',\pi'_{\mathrm{qry}}}^X(\mathbf{x}).$$

We write $[\phi, \pi_{\mathrm{qry}}]$ for the corresponding equivalence class. Let $(\Phi, \Pi)$ denote the joint prior induced by

$$\phi \sim \Phi, \qquad \pi_{\mathrm{qry}} \mid \phi \sim \Pi(\cdot \mid \phi),$$

where $\Pi$ is a fixed shift kernel. The following definition requires the target posterior to depend on the latent pair only through its observational equivalence class.

**Definition 3.2** (Shift-identifiable prior)**.** *We say that the prior $(\Phi, \Pi)$ is* shift-identifiable *if there exists a measurable map*

$$F : \mathcal{X} \times \mathcal{Y} \times \mathcal{Q} \to [0, 1],$$

*where $\mathcal{Q}$ denotes the space of observational equivalence classes, such that, for $(\Phi, \Pi)$-almost every $(\phi, \pi_{\mathrm{qry}})$ and almost every $\boldsymbol{x} \in \mathcal{X}$,*

$$p_{\phi,\pi_{\mathrm{qry}}}(y \mid \boldsymbol{x}) = F(\boldsymbol{x}, y, [\phi, \pi_{\mathrm{qry}}])$$

*for all $y \in \mathcal{Y}$.*

In other words, latent pairs that the observations cannot tell apart must agree on the target posterior, up to null sets. When they disagree, even infinitely many observations cannot resolve which prediction is correct. The Bayesian predictor therefore averages their target posteriors according to the prior conditioned on their equivalence class. For an observational equivalence class $[\phi, \pi_{\mathrm{qry}}]$, we write this average as:

$$\mathrm{M}(\boldsymbol{x}, y, [\phi, \pi_{\mathrm{qry}}]) \coloneqq \mathbb{E}_{(\Phi,\Pi)}\left[ p_{\phi',\pi'_{\mathrm{qry}}}(y \mid \boldsymbol{x}) \,\middle|\, [\phi', \pi'_{\mathrm{qry}}] = [\phi, \pi_{\mathrm{qry}}] \right] .$$

We now state the formal convergence proposition.

**Proposition 3.3** (Consistency up to shift-identifiability)**.** *Assume the Bayesian model is well specified and satisfies the standard measurability and posterior-consistency conditions for the observational equivalence classes induced by $(\mathcal{D}_n, \boldsymbol{X}_{1:m})$. Then, for prior-almost every generating pair $(\phi^\star, {\pi_{\mathrm{qry}}}^\star)$ and any fixed query index $i$,*

$$q_{n,m}^{\mathrm{B}}(y_i \mid \boldsymbol{X}_{1:m}, \mathcal{D}_n) \xrightarrow[n,m\to\infty]{\text{a.s.}} \mathrm{M}(\boldsymbol{x}_i, y_i, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]) .$$

*If the target predictive distribution is shift-identifiable at $\boldsymbol{x}_i$, then*

$$\mathrm{M}(\boldsymbol{x}_i, y_i, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]) = p_{\phi^\star,{\pi_{\mathrm{qry}}}^\star}(y_i \mid \boldsymbol{x}_i),$$

*so the Bayesian query-set predictor is consistent for the true target posterior. A PFN that exactly amortizes the Bayesian posterior predictive inherits the same limit.*

Intuitively, this proposition shows that additional observations resolve uncertainty between distinguishable tasks, while uncertainty within an equivalence class remains governed by the prior. Under shift-identifiability, this remaining uncertainty has no effect on prediction, and the Bayesian PPD converges to the true target posterior. The proof is given in Appendix B.

# 4 EXPERIMENTS

This experimental section is organized around three questions whose goal is to investigate whether ShiftICL improves over the baseline models under controlled label shift. First, we investigate whether our model retains a higher predictive accuracy across different levels of label shift severity, and for different families of label shift. Second, we evaluate whether amortizing the adaptation explicitly at the level of the predictive distribution yields better calibrated predictions than post-hoc label shift corrections.

Finally, we study to what extent our model exploits population-level information provided by the query set during transductive inference. In particular, we are interested in providing evidence that can distinguish a genuine mechanism that leverages the unlabeled query set information from a model that has only learned a correction rule. Through an ablation study comparing ShiftICL with a Base model trained on synthetic shifted episodes, we empirically demonstrate that a transductive formulation leads to a stronger performance, both in terms of accuracy and calibration metrics. The details of this ablation can be found in Appendix C.2.

We provide the full code base of this work in the following repository: `https://anonymous.4open.science/r/shifticl-88D2/`.

## 4.1 EVALUATION PROTOCOL

**Datasets.** We use a cache of 253 tabular classification datasets from OpenML (Bischl et al., 2025) and evaluate all 209 datasets supported by the model: at most 50 encoded features and 10 classes, ensuring that a minimum number of samples per class are observed in both context and query reservoirs. For each of the five evaluation seeds, we split the rows into disjoint reservoirs within classes. Sampling within a class is independent of its features, so the intervention changes class proportions without deliberately changing the class-conditional distributions. Finite samples of these distributions need not coincide. Preprocessing is fitted on the context and applied to the query set.

**Shift families.** We first follow the inverse-frequency construction used by the DistPFN evaluator: we resample the context with class weights proportional to $N_k^{-\beta}$, where $N_k$ is the pooled count of class $k$, and retain the query reservoir. We use $\beta \in \{0, 0.1, 0.5, 1, 2, 5\}$ and cap both context and query at 512 rows. Here $\beta = 0$ balances the context; it does not generally make the source and target priors equal.

We also evaluate three other forms of label shift with a fixed context. For the Dirichlet family, we draw an endpoint $q \sim \mathrm{Dirichlet}(\widehat{\pi}_{\mathrm{ctx}})$, whose total concentration is one. We resample the endpoint until $\mathrm{TV}(q, \widehat{\pi}_{\mathrm{ctx}}) \geq 0.4$, then set

$$\pi_{\mathrm{qry}}(d) = \widehat{\pi}_{\mathrm{ctx}} + \frac{d}{\mathrm{TV}(q, \widehat{\pi}_{\mathrm{ctx}})}(q - \widehat{\pi}_{\mathrm{ctx}}), \qquad d \in \{0, 0.1, 0.2, 0.3, 0.4\},$$

where $\mathrm{TV}(\pi, \pi') = \frac{1}{2}\sum_k |\pi(k) - \pi'(k)|$. Thus $d$ is the requested distance between the source and target priors, rather than a fraction of an endpoint whose distance varies across tasks.

For the spike and swap families, we use an interpolation protocol

$$\pi_{\mathrm{qry}}(t) = (1 - t)\widehat{\pi}_{\mathrm{ctx}} + tq, \qquad t \in \{0, 0.25, 0.5, 0.75, 1\}.$$

A spike places all mass on a uniformly chosen class while a swap exchanges the probabilities of two uniformly chosen classes. The context and endpoint remain fixed across levels within each dataset and seed. Each episode contains 512 context and 512 query rows. At level zero, query class counts match the context exactly. At positive levels, they are sampled from the requested target prior, so their realized distance has finite-sample variation.

**Baseline models.** Base and ShiftICL use the same TabPFNv3 backbone and a synthetic task generator adapted from the NanoTabICL prior[2]. Base is trained without label shift and predicts each query from its features and the labeled context. Instead, ShiftICL adds the query set mechanism of Section 3.2, with two inducing tokens per stage-3 block, and is trained across a range of shift severities,

[2] `https://github.com/soda-inria/nanotabicl`

including no shift. Both were trained from scratch for 5,000 optimizer updates on approximately 2.5 million syntetic datasets. Their comparison therefore includes both the architectural change and the training distribution. More details about the pre-training procedure and configurations can be found in Appendix C.1.

DistPFN reweights Base's predictions using the mean predicted query class probabilities, and DistPFN-T temperature-scales this estimate before reweighting. Oracle uses the same correction with target proportions computed from withheld query labels. This benchmark is not a theoretical upper-bound but the Bayesian optimal PPD computed with privileged information. Therefore, it is not a competing baseline. Although Drift-Resilient TabPFN was designed for temporal distribution shift rather than label shift, we include it as a relevant point of comparison since it is a pretrained tabular model in which adaptation to distribution shift is learned during pretraining. The shift mechanism this model amortizes is different than the one addressed in this work. We use the model's official implementation and open weights[3]. Since our episodes have no natural time index, we supply domain index 0 for context rows and 1 for query rows.

**Metrics.** We report accuracy, negative log-likelihood divided by $\log K$ for a $K$-class task (normalized NLL), and expected calibration error (ECE) with ten equal-width confidence bins. We average over datasets within each seed, then report the mean and standard error across five seeds. These error bars describe variability from evaluation sampling, not training-seed or dataset-population uncertainty.

### 4.2 Results

Figure 2 reports performance under the inverse-frequency shifts used by the DistPFN evaluator. ShiftICL is increasingly robust as the shift becomes more severe. Its accuracy decreases only moderately with $\beta$, while all non-oracle baselines deteriorate substantially; at $\beta = 5$, ShiftICL reaches roughly $0.72$ accuracy compared with $0.66$ for the strongest competing baseline. The same pattern is more pronounced for probabilistic performance: ShiftICL's normalized NLL and ECE degrade only modestly as the shift grows and remain substantially closer to the oracle, whereas the competing methods become progressively less calibrated. These results show that the learned adaptation remains effective in regimes where the evaluated post-hoc corrections deteriorate.

Figure 3 broadens the comparison to Dirichlet, spike, and swap shifts. At zero or very weak shift, all models are comparable. Drift-Resilient TabPFN exhibits a small advantage that can be probably attributed to the fact that it is a larger model than the all its competitors. As the source and query priors separate, ShiftICL becomes the strongest non-oracle method under the larger shifts, with particularly large gains for concentrated spike shifts and substantial improvements under both Dirichlet and swap perturbations. Its calibration follows the same pattern, remaining comparatively stable while the competing methods degrade with shift severity. Taken together, these experiments suggest that amortizing adaptation during pre-training provides substantially greater robustness to label shift while maintaining a strong performance in tasks with zero or weak label shift.

## 5 Conclusions and Limitations

Pre-trained exclusively on synthetic episodes, ShiftICL is a prior-fitted network that corrects for label shift in a single forward pass. Its premise is simple: under label shift, the unlabeled query set carries information about the target label distribution. This information, that standard in-context learners discard, our model learns to exploit. By conditioning on a summary of the query set, and by meta-training over a prior spanning a broad range of shift types and severities, ShiftICL transfers to real data without any adaptation step. It outperforms all baselines with a gap that widens as the shift grows. Furthermore, its calibration stays close to the oracle's, while methods that rescale the output probabilities degrade steadily. Our ablations further confirm that these gains stem from conditioning on the query set rather than from the shifted prior alone: ShiftICL outperforms a Base model pre-trained on shifted episodes bot in terms of accuracy and calibration metrics.

[3] https://github.com/automl/Drift-Resilient_TabPFN.git

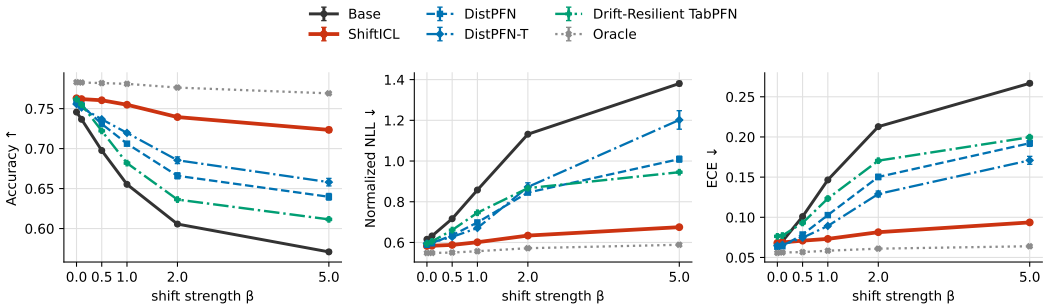



Figure 2: Inverse-frequency label shifts on 209 datasets. Curves show accuracy, normalized NLL, and ECE, with standard errors across five sampling seeds. All methods receive the same episodes. The oracle uses privileged query-label proportions.

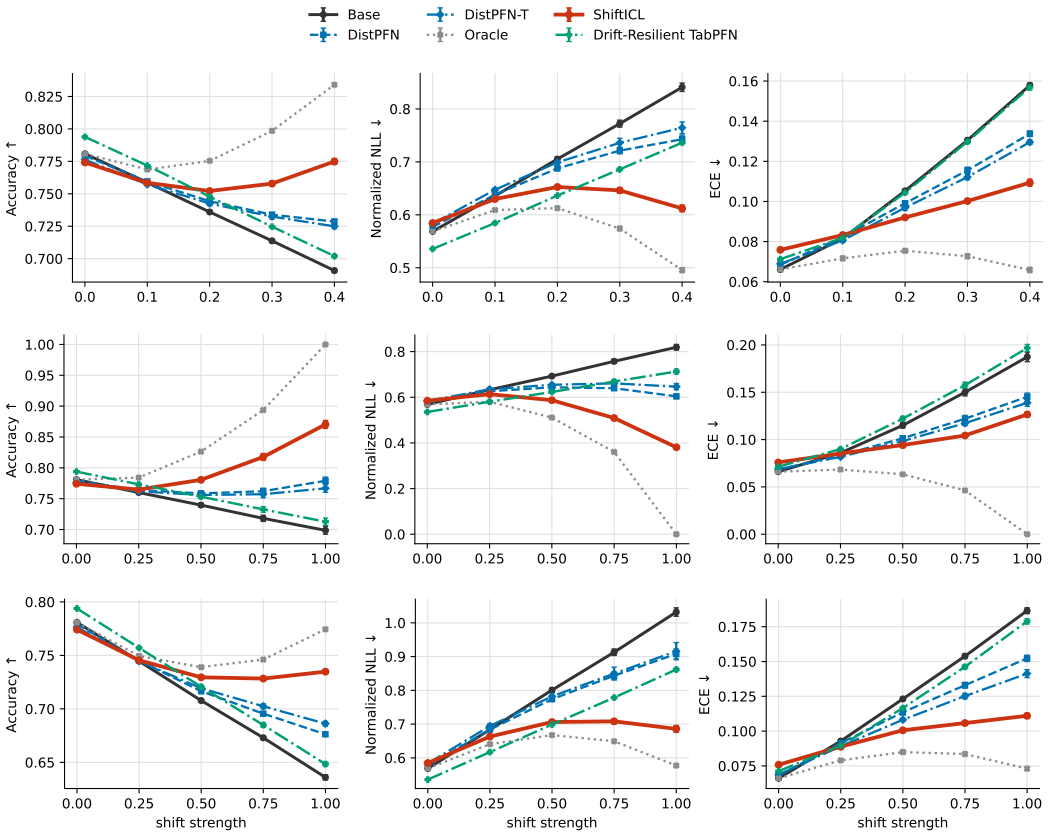



Figure 3: Target-prior shifts on the same 209 datasets: Dirichlet (top), spike (middle), and swap (bottom). The context is fixed across strengths. $t = 0$ matches its empirical label proportions. Points average datasets within each of five seeds and bars show the standard error across seeds. The strength parameter is specific to these constructions and is not numerically comparable to $\beta$.

**Limitations and future work.** Our model is compact: roughly 8 million parameters, trained on tasks of up to 50 features and 1024 rows. Scaling up both the model and the datasets it supports is a natural next step. More fundamentally, ShiftICL assumes that context and query share their class-conditional distributions. Some assumption of this kind is needed for the target label distribution to be recoverable from unlabeled data at all, and any method tailored to a particular kind of drift rests on one. However, our transductive formulation is not specific to label shift, and extending the prior to other types of distribution shift is a promising avenue for future work.

### AI USE STATEMENT

In this work, we used AI coding agents to devise an implementation plan and putting together the code base. For writing, we mainly use AI tools as a copy editing tool: grammar checks, improve clarity of certain passages, remove redundant words, suggest a draft for putting ideas in order, etc. (for example, with a prompt along the lines of "this paragraph needs a smoother transition between the first and second sentences, suggest an improvement"). For the Related Work section, LLMs helped with the literature review. Some implementation details for Appendix A.2 and A.3, and experimental details for Appendix C.1 (such as Table2) were drafted with the coding agent that was used for assisting with the development of the code base (for example "generate a paragraph detailing the architecture configuration of the checkpoints that generated the results to improve reproducibility, and summarize the details in a Table"). We also used AI tools to verify the claims and proofs of this work (for example, "I need to apply the Doob's consistency argument, are my hypothesis correct and well stated?"). Even though we manually conducted an extensive literature review on the subject, we used LLMs for ensuring this work is original and that no similar proposal has been overlooked.

We have not used generative AI tools for coming up with the ideas presented here, generate full sections or large blocks of text with no human input, generate the title, or name the method. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

All details regarding the transductive architecture and synthetic episode generation are detailed in Appendix A.2 and Appendix A.3. The architecture and pre-training details are in Appendix C.1. The code base can be found at `https://anonymous.4open.science/r/shifticl-88D2/`.

## REFERENCES

Shi ang Qi, Vahid Balazadeh, Michael Cooper, Russell Greiner, and Rahul G. Krishnan. SurvivalPFN: Amortizing survival prediction via in-context bayesian inference, 2026. URL https://arxiv.org/abs/2605.15488.

Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl0r3R9KX.

Bernd Bischl, Giuseppe Casalicchio, Taniya Das, Matthias Feurer, Sebastian Fischer, Pieter Gijsbers, Subhaditya Mukherjee, Andreas C Müller, László Németh, Luis Oala, Lennart Purucker, Sahithya Ravi, Jan N van Rijn, Prabhant Singh, Joaquin Vanschoren, Jos van der Velde, and Marcel Wever. Openml: Insights from 10 years and more than a thousand papers. *Patterns*, 6(7):101317, 2025. doi: 10.1016/j.patter.2025.101317. URL https://www.cell.com/patterns/fulltext/S2666-3899(25)00165-5.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. URL https://openreview.net/forum?id=jZqCqpCLdU.

Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary C. Lipton. A unified view of label shift estimation. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3249–3260, 2020.

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 Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-3: Technical report, 2026. URL https://arxiv.org/abs/2605.13986.

Kai Helli, David Schnurr, Noah Hollmann, Samuel Müller, and Frank Hutter. Drift-resilient tabPFN: In-context learning temporal distribution shifts on tabular data. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=p3tSEFMwpG.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations 2023*, 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. URL https://www.nature.com/articles/s41586-024-08328-6.

Seunghan Lee et al. Mitigating label shift in tabular in-context learning via test-time posterior adjustment. In *International Conference on Machine Learning (ICML)*, 2026.

Zachary C Lipton, Yu-Xiang Wang, and Alexander J Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3122–3130. PMLR, 07–15 Jul 2018. URL https://proceedings.mlr.press/v80/lipton18a.html.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse C. Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L. Caterini, and Maks Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-5748. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/fc0e3f908a2116ba529ad0a1530a3675-Abstract-Conference.html.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net.

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. 2026.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

soda-inria. NanoTabICL: A minimal TabICLv2 implementation. https://github.com/soda-inria/nanotabicl, 2026. GitHub repository, accessed September 15, 2026.

# A Further method details

## A.1 Proof of Corollary 3.1

**Corollary A.1** (Predictive identifiability under label shift)**.** *Let* $(\phi, \pi_{\mathrm{qry}})$ *and* $(\phi', \pi'_{\mathrm{qry}})$ *be two latent pairs satisfying*

$$\mathrm{supp}(\pi_{\mathrm{qry}}) \subseteq \mathrm{supp}(\pi_{\mathrm{ctx}}), \qquad \mathrm{supp}(\pi'_{\mathrm{qry}}) \subseteq \mathrm{supp}(\pi'_{\mathrm{ctx}}),$$

*and suppose that the class-conditional distributions* $\{p_\phi(\cdot \mid y)\}_{y \in \mathrm{supp}(\pi^{\phi}_{\mathrm{ctx}})}$ *are linearly independent.*

*If* $p_\phi(\boldsymbol{x}, y) = p_{\phi'}(\boldsymbol{x}, y)$ *and* $p_{\phi,\pi_{\mathrm{qry}}}(\boldsymbol{x}) = p_{\phi',\pi'_{\mathrm{qry}}}(\boldsymbol{x})$*, then*

$$p_{\phi,\pi_{\mathrm{qry}}}(y \mid \boldsymbol{x}) = p_{\phi',\pi'_{\mathrm{qry}}}(y \mid \boldsymbol{x})$$

*for* $p_{\phi,\pi_{\mathrm{qry}}}$*-almost every* $\boldsymbol{x} \in \mathcal{X}$ *and every* $y \in \mathcal{Y}$*.*

*Proof.* The labeled context distribution $p_\phi(\boldsymbol{x}, y)$ determines both the context label marginal $\pi^\phi_{\mathrm{ctx}}$ and, on its support, the class-conditional distributions $p_\phi(\boldsymbol{x} \mid y)$. Under the assumed support condition, these are therefore known for every class that can occur in the query distribution.

By the identifiability result of Garg et al. (2020), linear independence of the class-conditional distributions implies that the mixture map

$$\pi \longmapsto \sum_y \pi(y) p_\phi(\boldsymbol{x} \mid y)$$

is injective. Hence the observable query marginal $p_{\phi,\pi_{\mathrm{qry}}}(\boldsymbol{x})$ uniquely determines $\pi_{\mathrm{qry}}$. The target posterior then follows from Bayes' rule,

$$p_{\phi,\pi_{\mathrm{qry}}}(y \mid \boldsymbol{x}) = \frac{\pi_{\mathrm{qry}}(y) p_\phi(\boldsymbol{x} \mid y)}{\sum_k \pi_{\mathrm{qry}}(k) p_\phi(\boldsymbol{x} \mid k)},$$

for $p_{\phi,\pi_{\mathrm{qry}}}$-almost every $\boldsymbol{x}$. □

### A.2 Synthetic prior implementation details

#### A.2.1 Constructing training episodes.

For each synthetic task, we generate a large labeled pool from a sampled SCM and divide each class into disjoint context and query reservoirs. We first draw a context label prior $\pi_{\mathrm{ctx}} \sim \mathrm{Dirichlet}(1.5\,\mathbf{1}_K)$, where $\mathbf{1}_K$ is the $K$-dimensional vector with all entries set to 1. This way, classes are treated symmetrically while allowed to have different proportions. To construct the context, we reserve two examples per class to ensure support, then draw the remaining labels from $\pi_{\mathrm{ctx}}$ and sample their features from the corresponding class reservoirs. The generator therefore knows $\pi_{\mathrm{ctx}}$ explicitly, even though the observed context proportions generally differ from it. We use this drawn prior to define the shift kernel and select a query prior $\pi_{\mathrm{qry}}$. Finally, we draw query labels from $\pi_{\mathrm{qry}}$ and their features from the matching query reservoirs. Both reservoirs originate from the same SCM, so the shift is introduced through the label-sampling probabilities.

#### A.2.2 Modeling different kinds of label shift.

We incorporate different types of label shift by modeling different perturbation patterns. We randomize both the magnitude and the pattern of these changes. For each candidate prior, we independently draw

$$\Sigma = s^2(I + BB^\top), \qquad s \sim \mathrm{LogUniform}(0.1, 2), \qquad B \in \mathbb{R}^{(K-1)\times 2}, \qquad B_{ij} \overset{\mathrm{iid}}{\sim} \mathcal{N}(0, 1/2).$$

Here, $s$ controls the typical perturbation size, while $\Sigma$ allows both independent and correlated changes in class log-odds. Given $g \sim \mathcal{N}(0, \Sigma)$, the shift kernel draws

$$\omega = \begin{cases} g, & \text{with prob. } 0.45, \\ b \odot g, & \text{with prob. } 0.45, \\ ag, & \text{with prob. } 0.10, \end{cases} \qquad b_k \overset{\mathrm{iid}}{\sim} \mathrm{Bernoulli}(0.5), \qquad a = \begin{cases} 4, & \text{with prob. } 0.10, \\ 1, & \text{otherwise.} \end{cases}$$

The mixture covers dense perturbations, perturbations restricted to subsets of log-odds, and occasional amplified jumps, occurring in $1\%$ of candidate draws. A masked coordinate retains its odds relative to the reference class, although its probability may change through normalization. These candidates span different shift patterns, but a larger perturbation need not produce a harder task: its effect depends on the class-conditionals. We account for this distinction when selecting training episodes using the headroom criterion described next.

### A.2.3 A CURRICULUM OVER LABEL SHIFT SEVERITY

Diverse label proportions do not necessarily yield diverse adaptation problems: the benefit of correcting a shift also depends on the class-conditionals. We therefore guide episode selection by *recoverable headroom*, an inexpensive estimate of the accuracy gain available from correcting the class prior.

For each task, we fit a Quadratic Discriminant Analysis (QDA) based surrogate to the synthetic pool and hold its estimated class-conditionals fixed. Let $g_\pi$ denote the resulting classifier with prior $\pi$, and let $\widehat{\mathrm{Acc}}_\pi(g)$ denote its empirical per-class accuracies on the pool, weighted by $\pi$. For each of $C$ candidate target priors, we compute

$$\widehat{\Gamma}(\pi_{\mathrm{ctx}}, \pi_{\mathrm{qry}}) = \widehat{\mathrm{Acc}}_{\pi_{\mathrm{qry}}}(g_{\pi_{\mathrm{qry}}}) - \widehat{\mathrm{Acc}}_{\pi_{\mathrm{qry}}}(g_{\pi_{\mathrm{ctx}}}).$$

Both classifiers are evaluated under the same target mixture, so their difference isolates the estimated benefit of replacing the source prior with the target prior. This is a surrogate measure of adaptation opportunity, rather than a guarantee of the PFN's achievable gain. We retain candidates whose target-prior QDA accuracy exceeds the target majority- class baseline by a margin $\epsilon$:

$$\mathcal{F} = \left\{ c : \widehat{\mathrm{Acc}}_{\pi_{\mathrm{qry}}{}^{(c)}}(g_{\pi_{\mathrm{qry}}{}^{(c)}}) > \max_k \pi_{\mathrm{qry}}{}^{(c)}(k) + \epsilon \right\}.$$

With probability $p_0$, we explicitly generate a no-shift episode by setting $\pi_{\mathrm{qry}} = \pi_{\mathrm{ctx}}$. Otherwise, we sample a desired headroom and choose the closest feasible candidate:

$$\Gamma^\star \sim \mathrm{Beta}(2,1), \qquad c^\star = \arg\min_{c \in \mathcal{F}} \left| \widehat{\Gamma}(\pi_{\mathrm{ctx}}, \pi_{\mathrm{qry}}{}^{(c)}) - \Gamma^\star \right|, \qquad \pi_{\mathrm{qry}} = \pi_{\mathrm{qry}}{}^{(c^\star)}.$$

If no candidate is feasible, we also fall back to the source prior. This procedure favors shifts with greater recoverable headroom while retaining explicit no-shift episodes. The realized headroom depends on the candidates available for each task and therefore does not itself follow the sampled Beta distribution.

## A.3 DETAILS ON THE TRANSDUCTIVE ARCHITECTURE

Let $C^\ell$ and $Q^\ell$ denote the context and query representations entering stage-3 block $\ell$. Each block has $r$ learned inducing vectors $I^\ell \in \mathbb{R}^{r \times d}$, which attend to the query representations to form an episode-dependent summary:

$$H^\ell = \mathrm{Attn}_\ell\left(I^\ell, Q^\ell, Q^\ell\right).$$

The summary is recomputed at every block and uses that block's existing attention projections. Since the query representations are themselves conditioned on the labeled context, the summary can capture features of the query distribution relative to the current task. Context rows continue to attend only to context rows, whereas query rows attend jointly to the context and the summary:

$$C^{\ell+1} = \mathrm{Block}_\ell\left(C^\ell; C^\ell\right), \qquad Q^{\ell+1} = \mathrm{Block}_\ell\left(Q^\ell; [C^\ell, H^\ell]\right).$$

Here, the semicolon specifies the key/value set; the notation suppresses the backbone's normalization, attention projections, residual connections, and feed-forward updates. All valid query rows contribute to the summary, including the row being classified.

The model performs a single forward pass, with no query labels supplied. Its retrieval decoder remains unchanged: each final query representation predicts a label by attending exclusively to labeled context representations. Preprocessing statistics and the feature-distribution encoder likewise use context rows only, so context representations remain independent of the query set. Padding is excluded from attention, and attention-length scaling uses the number of valid keys.

Communication between query rows is therefore mediated entirely through the inducing summaries, adding $O(mr)$ attention interactions per block rather than $O(m^2)$ direct query self-attention. At inference, summarization is invariant to query order and predictions are equivariant to it. Predictions nevertheless depend on the membership and size of the supplied query set, which forms part of the model's conditioning information.

Table 1: Main notation.

| SYMBOL | DESCRIPTION |
|---|---|
| $\Phi$ | Prior over latent tasks |
| $\phi \sim \Phi$ | Latent task sampled from the task prior |
| $p_\phi(\mathrm{x}, \mathrm{y})$ | Context joint distribution induced by $\phi$ |
| $\pi_{\mathrm{ctx}}$ | Context label marginal, $\pi_{\mathrm{ctx}}(y) = p_\phi(\mathrm{y} = y)$ |
| $\Pi(\cdot \mid \phi)$ | Shift kernel over target label distributions |
| $\pi_{\mathrm{qry}} \sim \Pi(\cdot \mid \phi)$ | Latent target label distribution |
| $P$ | Joint prior over $(\phi, \pi_{\mathrm{qry}})$ induced by $\Phi$ and $\Pi$ |
| $\mathcal{D}_n$ | Labeled source context $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ |
| $\boldsymbol{X}_{1:m}$ | Unlabeled target set $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ observed by the model |
| $p_{\phi,\pi_{\mathrm{qry}}}(\mathbf{x})$ | Query feature marginal, $\sum_{k=1}^K \pi_{\mathrm{qry}}(k) p_\phi(\mathbf{x} \mid \mathrm{y} = k)$ |
| $p_{\phi,\pi_{\mathrm{qry}}}(\mathrm{y} \mid \boldsymbol{x})$ | Query predictive distribution under $(\phi, \pi_{\mathrm{qry}})$ |
| $[\phi, \pi_{\mathrm{qry}}]$ | Observational equivalence class of $(\phi, \pi_{\mathrm{qry}})$ |
| $q^{\mathrm{B}}_{n,m}$ | Bayesian query-set posterior predictive |
| $M(\boldsymbol{x}, y, [\phi, \pi_{\mathrm{qry}}])$ | Prior-average predictive over an equivalence class |

## B Consistency of the Bayesian PPD

We now study the consistency of the Bayesian PPD in equation 2. This result alone does not establish consistency of the trained network. Rather, under the conditions stated below, the Bayesian predictor converges as $n, m \to \infty$ to the predictive distribution associated with the observational equivalence class of the generating process. Under shift-identifiability, this limit coincides with the true query posterior. Table 1 contains all necessary notation.

**Proposition B.1** (Consistency up to shift-identifiability)**.** *Assume the Bayesian model is well specified and satisfies the standard measurability and posterior-consistency conditions for the observational equivalence classes induced by $(\mathcal{D}_n, \boldsymbol{X}_{1:m})$. Then, for prior-almost every generating pair $(\phi^\star, {\pi_{\mathrm{qry}}}^\star)$ and any fixed query index $i$,*

$$q^{\mathrm{B}}_{n,m}(y_i \mid \boldsymbol{X}_{1:m}, \mathcal{D}_n) \xrightarrow[n,m\to\infty]{\text{a.s.}} \mathrm{M}(\boldsymbol{x}_i, y_i, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]) \,.$$

*If the target predictive distribution is shift-identifiable at $\boldsymbol{x}_i$, then*

$$\mathrm{M}(\boldsymbol{x}_i, y_i, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]) = p_{\phi^\star, {\pi_{\mathrm{qry}}}^\star}(y_i \mid \boldsymbol{x}_i),$$

*so the Bayesian query-set predictor is consistent for the true target posterior. A PFN that exactly amortizes the Bayesian posterior predictive inherits the same limit.*

*Proof.* We follow an argument analogous to that of SurvivalPFN.

Recall that the labeled source context $\mathcal{D}_n$ consists of $n$ i.i.d. observations from $p_\phi(\mathbf{x}, \mathrm{y})$. To generate the target set, we first sample

$$\pi_{\mathrm{qry}} \sim \Pi(\cdot \mid \phi),$$

followed by

$$\mathrm{y}_j \overset{\text{iid}}{\sim} \pi_{\mathrm{qry}}, \qquad \mathbf{x}_j \mid \mathrm{y}_j \sim p_\phi(\mathbf{x} \mid \mathrm{y}_j), \qquad j = 1, \ldots, m.$$

The rows $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ of $\boldsymbol{X}_{1:m}$ are realizations of these target feature variables. This construction ensures that source and target share the same class-conditional law $p_\phi(\mathbf{x} \mid \mathrm{y})$.

After marginalizing the unobserved target labels, each target feature has distribution

$$p_{\phi,\pi_{\mathrm{qry}}}(\mathbf{x}) = \sum_{y \in \mathcal{Y}} \pi_{\mathrm{qry}}(y)\, p_\phi(\mathbf{x} \mid \mathrm{y} = y).$$

Let $N = \min(n, m)$. Conditional on $(\phi, \pi_{\mathrm{qry}})$, pair the first $N$ source observations with the first $N$ target features. The resulting sequence is i.i.d. with distribution

$$p_\phi(\mathbf{x}, \mathrm{y}) \otimes p_{\phi,\pi_{\mathrm{qry}}}(\mathbf{x}).$$

The pairing is purely notational: there is no association between the source and target observations at the same index.

The observational equivalence relation $[\phi, \pi_{\mathrm{qry}}]$ is precisely the one induced by this observable distribution. Hence, under the usual measurability conditions required for Doob consistency, applied under the joint prior induced by

$$\phi \sim \Phi, \qquad \pi_{\mathrm{qry}} \sim \Pi(\cdot \mid \phi),$$

we obtain

$$\mathbb{E}[\mathrm{M}(\boldsymbol{x}_i, y_i, [\phi, \pi_{\mathrm{qry}}]) \mid \mathcal{D}_N, \boldsymbol{X}_{1:N}] \xrightarrow[N\to\infty]{\text{a.s.}} \mathrm{M}(\boldsymbol{x}_i, y_i, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]) \tag{5}$$

for prior-almost every generating pair $(\phi^\star, {\pi_{\mathrm{qry}}}^\star)$.

It remains to extend the argument from the diagonal sequence $N = N$ to arbitrary $n, m \to \infty$. Since $N = \min(n, m)$, we have $N \to \infty$ whenever $n, m \to \infty$, and

$$\sigma(\mathcal{D}_N, \boldsymbol{X}_{1:N}) \subseteq \sigma(\mathcal{D}_n, \boldsymbol{X}_{1:m}).$$

The observational equivalence class is measurable with respect to the limiting sigma-field generated by the diagonal sequence. It is therefore also measurable with respect to the limiting sigma-field generated by the full observations. The same Doob/martingale-consistency argument thus applies as $n, m \to \infty$.

We now show that the conditional expectation appearing above is the Bayesian target posterior predictive. By definition of M,

$$\begin{aligned}
&\mathbb{E}[\mathrm{M}(\boldsymbol{x}_i, y_i, [\phi, \pi_{\mathrm{qry}}]) \mid \mathcal{D}_n, \boldsymbol{X}_{1:m}] \\
&= \mathbb{E}\big[\mathbb{E}\big[p_{\phi,\pi_{\mathrm{qry}}}(y_i \mid \boldsymbol{x}_i) \mid [\phi, \pi_{\mathrm{qry}}]\big] \,\big|\, \mathcal{D}_n, \boldsymbol{X}_{1:m}\big] \\
&\overset{(i)}{=} \mathbb{E}\big[\mathbb{E}\big[p_{\phi,\pi_{\mathrm{qry}}}(y_i \mid \boldsymbol{x}_i) \mid [\phi, \pi_{\mathrm{qry}}], \mathcal{D}_n, \boldsymbol{X}_{1:m}\big] \,\big|\, \mathcal{D}_n, \boldsymbol{X}_{1:m}\big] \\
&\overset{(ii)}{=} \mathbb{E}\big[p_{\phi,\pi_{\mathrm{qry}}}(y_i \mid \boldsymbol{x}_i) \mid \mathcal{D}_n, \boldsymbol{X}_{1:m}\big] \\
&= q^{\mathrm{B}}_{n,m}(y_i \mid \boldsymbol{X}_{1:m}, \mathcal{D}_n).
\end{aligned}$$

Here, $(i)$ follows because the observed data are conditionally independent of $(\phi, \pi_{\mathrm{qry}})$ given their observational equivalence class, and $(ii)$ follows from the tower property.

Combining this identity with Equation equation 5 yields

$$q^{\mathrm{B}}_{n,m}(y_i \mid \boldsymbol{X}_{1:m}, \mathcal{D}_n) \xrightarrow[n,m\to\infty]{\text{a.s.}} \mathrm{M}(\boldsymbol{x}_i, y_i, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]).$$

Thus the Bayesian posterior predictive converges to the prior average of the target predictive over the observational equivalence class of the generating latent pair.

Finally, if $(\Phi, \Pi)$ is shift-identifiable, then $p_{\phi,\pi_{\mathrm{qry}}}(y \mid \boldsymbol{x})$ is constant over each observational equivalence class in the support of $(\Phi, \Pi)$. Hence

$$\mathrm{M}(\boldsymbol{x}, y, [\phi^\star, {\pi_{\mathrm{qry}}}^\star]) = p_{\phi^\star,{\pi_{\mathrm{qry}}}^\star}(y \mid \boldsymbol{x}).$$

□

# C Additional experimental details

## C.1 Model and pretraining

Base and ShiftICL use the same implementation of TabPFNv3 architecture, trained from scratch. Table 2 summarizes the details of the training configurations for both models. Its three stages encode features across examples, aggregate features within each example, and mix context and query rows. The retrieval decoder uses labeled context rows only. ShiftICL differs from Base only in the final row-mixing stage: each of its blocks uses two learned inducing tokens to summarize the unlabeled query set. These are distinct from the 64 inducing tokens in TabPFNv3's feature encoder.

Pretraining samples the number of features from $\{8, 16, 32, 50\}$ and the number of classes from $\{2, \ldots, 10\}$. Episodes contain 256, 512, or 1024 rows, with the context fraction chosen from

Table 2: Mid-sized TabPFNv3 (approximately 8 million parameters) configuration used for Base and ShiftICL. Settings are shared unless noted.

| COMPONENT | SETTING |
|---|---|
| Embedding width; block depths | 96; $(2, 2, 6)$ |
| Attention heads; context-label tokens | $(4, 4, 8)$; 4 |
| Feature-group size; backbone inducing points | 3; 64 |
| Target-set inducing points | 2 (ShiftICL only) |
| Test-time key/value heads | 1 |
| Maximum classes; feature choices | 10; $\{8, 16, 32, 50\}$ |
| Rows; context fractions | $\{256, 512, 1024\}$; $\{0.4, 0.5, 0.6\}$ |
| Pool size; minimum rows per class | 1024; 8 in pool, 2 in context |
| Context-prior concentration | 1.5 |
| Shift curriculum | One-step, decoupled; 64 candidates (ShiftICL) |
| No-shift probability; feasibility margin | 0.12; 0.02 (ShiftICL) |
| Optimizer; peak learning rate | Schedule-free AdamW; $5 \times 10^{-4}$ |
| Warmup; weight decay; gradient clipping | 1000 updates; 0; norm 1 |
| Effective batch | 512 episodes per update |
| Base batch decomposition | 8/GPU $\times$ 4 GPUs $\times$ 16 accumulation |
| ShiftICL batch decomposition | 16/GPU $\times$ 4 GPUs $\times$ 8 accumulation |
| Updates used; random seed | 5000 (2.56M episodes); 0 |

$\{0.4, 0.5, 0.6\}$. Base sees unshifted episodes; ShiftICL sees episodes from the label-shift prior described in the main text. Both use 512 synthetic tasks per optimizer update and schedule-free Adam with learning rate $5 \times 10^{-4}$, 1,000 warmup updates, and gradient clipping at norm one. No additional feature noise is applied.

### C.2 Is a transductive formulation justified?

The purpose of this ablation is to separate the contribution of the unlabeled query set from that of the weights of the model at inference time. In other words, we study whether our transductive formulation brings benefits to the predictive task in terms of accuracy and normalized NLL. To do that, we trained two small backbones, a Base model and a ShiftICL model. Both have approximately 1 million parameters and were trained using the same hyperparameters (see Table 3) for all metrics to be comparable. Crucially, both Base and ShiftICL were trained on the *same synthetic shifted episodes*.

We follow a standard procedure to construct synthetic classification tasks. Each task is a sixteen-covariate Gaussian mixture comprising 10,240 samples and number of classes $K \in \{2, \ldots, 10\}$. Each pool is drawn once and partitioned into disjoint context and query splits. The source prior is fixed to the uniform distribution over the $K$ classes, so that the number of classes is the only quantity that alters it, and the target prior is constructed to sit at a prescribed total-variation distance from it. We sample a direction on the simplex from a Dirichlet distribution with concentration $0.3$, which favours near-degenerate priors, retain the first draw whose distance from the source exceeds the requested level, and interpolate linearly between the source and that endpoint. We sample context and target sets of size 256 each for predicting the target.

Figures 4 and 5 summarize the results for Accuracy and Normalized NLL, averaged over 20 tasks. Black lines see no query set: a Base mode model trained on unshifted episodes degrades steadily with the shift, while one trained on shifted episodes stays flat. In contrast, ShiftICL recovers almost all the oracle advantage for $K \leq 5$ in terms of accuracy and trails the oracle in terms on Normalized NLL for $K \leq 7$. We also compare the ShiftICL checkoint with the query set conditioning disabled. In all cases, the transductive ShiftICL model outperforms all other variants for moderate to severe label shifts, with a growing gap in favor of ShiftICL.

Table 3: Tiny-scale training configuration. Base and ShiftICL share all settings except the episode distribution and target-set attention.

| COMPONENT | SETTING |
|---|---|
| Embedding width; block depths | 64; $(1, 1, 2)$ |
| Attention heads; context-label tokens | $(2, 2, 2)$; 2 |
| Feature-group size; backbone inducing points | 2; 16 |
| Target-set inducing points | 1 (ShiftICL only) |
| Maximum classes; feature choices | 5; $\{4, 8, 10\}$ |
| Rows; context fractions | $\{256, 512\}$; $\{0.4, 0.5, 0.6\}$ |
| Pool size; minimum rows per class | 1024; 8 in pool, 2 in context |
| Context-prior concentration | 1.5 |
| Shift curriculum | One-step, decoupled; 64 candidates (ShiftICL) |
| No-shift probability; feasibility margin | 0.12; 0.02 (ShiftICL) |
| Optimizer; peak learning rate | Schedule-free AdamW; $5 \times 10^{-4}$ |
| Warmup; weight decay; gradient clipping | 1000 updates; 0; norm 1 |
| Effective batch | 16/GPU $\times$ 4 GPUs $\times$ 8 $=$ 512 |
| Training budget; seed | 5000 updates (2.56M episodes); 0 |

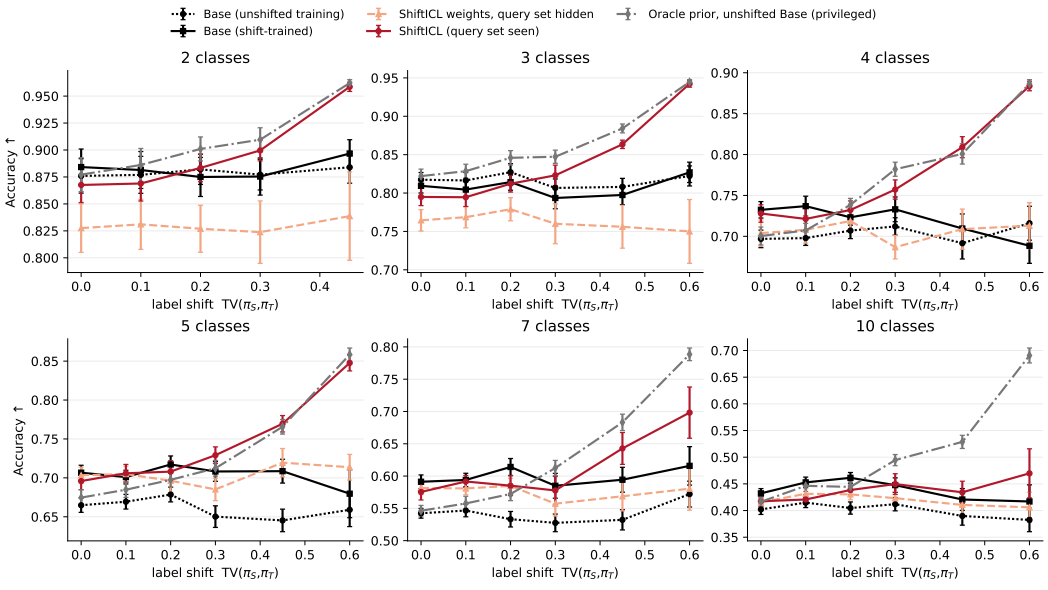



Figure 4: Accuracy under controlled label shift, by number of classes. Each panel shows the mean over 20 synthetic tasks with 256 context and 256 target rows, against the total-variation distance between the source and target priors.

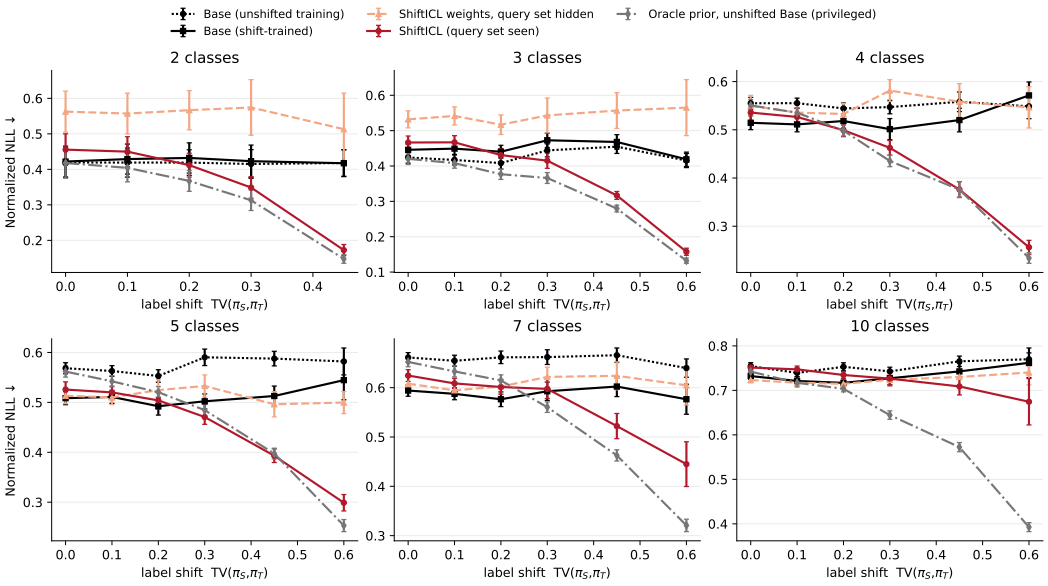



Figure 5: Normalized NLL under controlled label shift, by number of classes. Each panel shows the mean over 20 synthetic tasks with 256 context and 256 target rows, against the total-variation distance between the source and target priors.