UNCERTAINTY QUANTIFICATION FOR PRIOR-FITTED NETWORKS USING MARTINGALE POSTERIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior-fitted networks (PFNs) have emerged as promising foundation models for prediction from tabular data sets, achieving state-of-the-art performance on small to moderate data sizes without tuning. While PFNs are motivated by Bayesian ideas, they do not provide any uncertainty quantification for predictive means, quantiles, or similar quantities. We propose a principled and efficient method to construct Bayesian posteriors for such estimates based on Martingale Posteriors. Several simulated and real-world data examples are used to showcase the resulting uncertainty quantification of our method in inference applications.

1 INTRODUCTION

Prior-fitted networks (PFNs) are foundation models (Müller et al., 2022; Hollmann et al., 2022) that allow for in-context learning, i.e., the ability to learn at inference time without any parameter updates (Garg et al., 2022). TabPFN, a transformer pre-trained on synthetic data for in-context learning on tabular data sets, has recently attracted a lot of interest. TabPFN (Hollmann et al., 2022; 2025) and extensions such as TuneTables (Feuer et al., 2024) and LocalPFN (Thomas et al., 2024) have been shown to achieve state-of-the-art performance on tabular benchmarks by pre-training on purely synthetic data. Since PFNs and extensions learn in-context, there is no need for further model (fine-)tuning on the inference task.

Recent extensions of PFNs allow their applicability to large data sets (Feuer et al., 2024), the use of PFN "priors" for latent variable models (Reuter et al., 2025), and simultaneously minimizing bias and variance to improve their performance (Liu & Ye, 2025). PFNs are also related to simulation-based inference and amortized inference but have slightly different goals and do not amortize across a single but multiple data sets (Reuter et al., 2025). While introduced as a Bayesian method and approximation to the posterior predictive, PFNs can also be interpreted as pre-tuned untrained predictors (Nagler, 2023). This also relates to the question of what uncertainty PFN models can provide.

PFNs approximate the posterior predictive distribution for the label given some feature values. Despite the name, this only yields point estimates of the most relevant predictive quantities, such as the conditional mean or quantiles. Due to the complex nature of PFNs, it is difficult to assess the uncertainty of these point estimates.

We propose a principled and efficient method to construct Bayesian posteriors for such estimates using the idea of *Martingale Posteriors (MPs)*. In particular:

1. We introduce a new extension of the MP framework of Fong et al. (2023) for inference of predictive quantities conditional on a specific feature value x.

2. We propose an efficient, nonparametric resampling scheme yielding an approximate posterior for the point estimates derived from a PFN.

3. We illustrate the new method in several simulated and real-world data applications.

Our proposal should be understood as a proof-of-concept, leaving some practical considerations to future work; see our discussion in Section 5.

2 BACKGROUND

We consider a tabular prediction task with labels $y \in \mathbb{R}$ and features $x \in \mathbb{R}^d$ drawn from a joint distribution P. A typical problem in such tasks is to estimate predictive quantities such as conditional means $\mathbb{E}[y|x]$, conditional probabilities $P(y|x)$, or conditional quantiles $P^{-1}(\alpha|x)$. Because the true distribution P is unknown and only a finite amount of data $\mathcal{D}_n = (y_i, x_i)_{i=1}^n$ is available, estimates of such quantities bear some uncertainty. Our goal is to quantify this uncertainty.

2.1 PRIOR-FITTED NETWORKS

Prior-fitted networks are foundation models trained to approximate the posterior predictive density

$$\mathrm{PPD}(y|x) = p(y|x, \mathcal{D}_n),$$

which quantifies the likelihood of observing label y given that the feature is x and $\mathcal{D}_n$ has been observed. The PPD is a Bayesian concept and implicitly involves a prior over the distributions P that could have generated the data. To approximate the PPD with a PFN, a deep neural network—typically a transformer—is pre-trained on simulated data sets with diverse characteristics. After pre-training, the network weights are fixed, and the approximate PPD for a new training set can be computed through a single forward pass without additional training or tuning.

2.2 BAYESIAN INFERENCE

In classical Bayesian inference, the set of possible distributions $P = P_\theta$ is indexed by some parameter θ. A *prior* distribution $\pi(\theta)$ is elicited to quantify our beliefs about the likelihood of the possible values of θ before seeing any data. After observing $\mathcal{D}_n$, this belief is updated to a *posterior* $\pi(\theta|\mathcal{D}_n)$ of the parameter θ given the data. For predictive inference, the PPD can be computed as

$$\mathrm{PPD}(y|x) = \int p_\theta(y|x)\pi(\theta|\mathcal{D}_n)\, d\theta.$$

The posterior $\pi(\theta|\mathcal{D}_n)$ also quantifies uncertainty for other interest quantities. For example, the posterior distribution for the conditional mean $\mu(x) = \int p_\theta(y|x)dy$ is given by

$$\Pi(\mu(x) \in A) = \int \mathbb{1}\left\{\int p_\theta(y|x)dy \in A\right\} \pi(\theta|\mathcal{D}_n)\ d\theta.$$

PFNs neither provide an explicit model for p_θ nor an explicit prior $\pi(\theta)$, although both may be implicit in the PPD. The following shows how Bayesian posterior inference can be approached when only the PPD is available.

2.3 MARTINGALE POSTERIORS

Martingale posteriors were recently introduced by Fong et al. (2023) as a new method for Bayesian uncertainty quantification. Its core idea is to reverse the direction of the Bayesian inference. In classical Bayesian inference, the posterior is derived from a prior and likelihood, which then implicitly leads to the PPD. MP inference starts from the PPD and leaves the prior $\pi(\theta)$ implicit. An appropriate sampling scheme and Doob's theorem then allow us to derive posteriors for virtually all quantities of interest (e.g., the conditional mean $\mu(x)$).

To simplify our outline of the approach, consider the case where there are no features, and we are interested in unconditional inference. An extension to our predictive inference setting will be made explicit in Section 3.1. Suppose we have observed data $y_{1:n} = (y_1, \ldots, y_n)$.

The MP approach involves iteratively sampling

$$y_{n+1} \sim p(y|y_{1:n}), \quad y_{n+2} \sim p(y|y_{1:(n+1)}), \quad y_{n+3} \sim p(y|y_{1:(n+2)}), \quad \ldots,$$

N times, which yields a sample $y_{(n+1):(n+N)}$ drawn from the predictive joint distribution

$$p(y_{(n+1):(n+N)}|y_{1:n}) = \prod_{i=1}^{N} p(y_{n+i}|y_{1:(n+i-1)}).$$

Observe, however, that the samples are neither independent nor identically distributed. As a consequence, the long-run empirical distribution of the obtained sample,

$$F_\infty(y) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y_{n+i} \leq y),$$

is a random function and comes out differently whenever the sampling procedure is repeated. Denote by $\Pi(F_\infty|\mathcal{D}_n)$ the distribution of this function (which depends on the data $\mathcal{D}_n$ we start with). For any parameter $\theta = \theta(P)$ of interest, the martingale posterior is now given as

$$\Pi(\theta \in A|\mathcal{D}_n) = \int \mathbb{1}\{\theta(F_\infty)\} d\Pi(F_\infty|\mathcal{D}_n).$$

Furthermore, Doob's theorem (Doob, 1949) implies that $\Pi(\theta|\mathcal{D}_n)$ coincides with the classical Bayes posterior for the prior $\pi(\theta)$ implicit in the PPD (Fong et al., 2023).

3 Efficient martingale posteriors for prior-fitted networks

Martingale posteriors allow for Bayesian inference directly from the PPD. PFNs approximate the PPD, so using PFNs to construct a martingale posterior seems natural. However, there are two problems. First, modern PFNs are based on transformer architectures that require $\Omega(n^2)$ operations for a forward pass on a training set size of n. Iteratively computing $p(y|y_{1:(n+k)})$ for $k = 1, \ldots, N$ thus has complexity $\Omega(N^3)$, which is prohibitive. Second, Falck et al. (2024) found that modern transformer-based LLMs substantially deviate from the *martingale property*

$$\mathbb{E}[p(y|y_{1:(n+k)})|y_{1:n}] = p(y|y_{1:n}).$$

Without this property, the MP sampling procedure leads to meaningless results. Instead, we propose to use the PPD implied by the PFN only as a starting point for the sampling scheme. This PPD is then iteratively updated using the Gaussian copula approach of Fong et al. (2023), which ensures the martingale property.

3.1 Martingale posteriors for conditional inference

We extend the unconditional sampling scheme outlined in the previous section to the conditional inference setting. Fong et al. (2023) already proposed one such extension. Their scheme involves forward sampling of the features $x_{(n+1):(n+N)}$. The distribution of the features isn't of primary interest but significantly complicates the sampling procedure, which the authors resolved through heuristic simplifications. In contrast to Fong et al. (2023), we propose to sample only the labels $y_{(n+1):(n+N)}$ conditional on the event that $x_{n+k} = x$, for a fixed value of x and all $k = 1, \ldots, N$.

Specifically, our goal is to simulate data from the distribution

$$p(y_{(n+1):(n+N)}|x_{(n+1):(n+N)} = x, \mathcal{D}_n).$$

Set $x_{n+k} = x$ for all $k \geq 1$ and define

$$p_k(y) = p(y_{n+k+1}|y_{1:(n+k)}, x_{1:(n+k)})$$

and P_k as the corresponding CDF. Applying Bayes' rule recursively gives

$$p(y_{(n+1):(n+N)}|x_{(n+1):(n+N)} = x, \mathcal{D}_n) = \prod_{k=0}^{N-1} p_k(y_{k+1}),$$

which suggests that we can iteratively sample

$$y_{n+1} \sim P_0, \quad y_{n+2} \sim P_1, \quad y_{n+3} \sim P_2, \quad \ldots.$$

Denote the long-run empirical distribution of the obtained sample by

$$F_{\infty,x} = \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y_{n+i} \leq y),$$

which is again a random function, even in the limit. Repeating the iterative sampling procedure gives us its distribution $\Pi(F_{\infty,x}|\mathcal{D}_n)$. For any conditional parameter $\theta(x) = \theta(P(\cdot\,|x))$ of interest, the martingale posterior is now given as

$$\Pi(\theta(x) \in A|\mathcal{D}_n) = \int \mathbb{1}\{\theta(F_{\infty,x})\} d\Pi(F_{\infty,x}|\mathcal{D}_n).$$

Common examples of the parameter $\theta(x)$ are the conditional mean

$$\theta(x) = \int y\, dP(y|x)dy$$

or a conditional α-quantile

$$\theta(x) = P^{-1}(\alpha|x).$$

3.2 Efficient PPD updates based on the Gaussian copula

Observe that $p_0(y) = p(y|x, \mathcal{D}_n)$ is the PPD approximated by the PFN g_θ. However, the following update distributions $p_1, p_2, \dots$ are generally intractable. To alleviate this, Fong et al. (2023) proposed a computationally efficient, nonparametric method based on Dirichlet Process Mixture Models (DPMMs) and a copula decomposition of the conditional p_k. Specifically, we set

$$P_k(y) = (1 - \alpha_{n+k})P_{k-1}(y) + \alpha_{n+k} H_\rho(P_{k-1}(y), P_{k-1}(y_{n+k})), \tag{1}$$

where P_k is the CDF corresponding to p_k,

$$\alpha_i = \left(2 - \frac{1}{i}\right)\frac{1}{i+1}, \quad H_\rho(u,v) = \Phi\left(\frac{\Phi^{-1}(u) - \rho\Phi^{-1}(v)}{\sqrt{1-\rho^2}}\right),$$

and Φ is the standard normal cumulative distribution function. The method has a hyperparameter ρ, corresponding to a bandwidth that smoothes the updates. Fong et al. (2023) proposed to tune this by maximizing the likelihood of the updated densities over the observed data. To increase the alignment of the updates with the PFN baseline, we simulate a new sample from the PFN and optimize the bandwidth on the simulated data.

3.3 Theoretical properties

Despite the simplicity of the updates given in Equation (1), they provide several important theoretical guarantees. The following is a direct consequence of Theorem 3 by Fong et al. (2023).

Proposition 3.1. *It holds $(y_{N+1}, y_{N+2}, \dots) \to_d (z_1, z_2, \dots)$ as $N \to \infty$ where $(z_1, z_2, \dots)$ has an exchangeable distribution.*

By de Finetti's theorem (e.g., Schervish, 2012, Theorem 1.49) it then follows that there is a random variable Θ such that $(z_1, z_2, \dots)$ is conditionally *iid* given Θ. The distribution of Θ can be interpreted as an implicit prior. In our setting, this prior depends both on the initial PPD implied by the PFN and the Gaussian copula updates specified by Equation (1). In particular, the following result follows immediately from Proposition 3.1 above and Theorem 2.2 of Berti et al. (2004).

Proposition 3.2. *Suppose that P_0 is absolutely continuous with respect to the Lebesgue measure. Then there exists a random probability distribution $P_{\infty,x}$ such that $\lim_{N\to\infty} P_N(y) = P_{\infty,x}(y) = F_{\infty,x}(y)$ almost surely.*

The proposition implies that the (random) limit $P_{\infty,x}$ is well defined and that the iterative sampling scheme is a valid way to draw from its distribution. We can be more precise about how fast this limit is approached.

Proposition 3.3. *For any $N \geq 0$ and $\epsilon > 0$, there is a constant $C \in (0, \infty)$ such that*

$$\sup_y \limsup_{M\to\infty} \Pr\left(|P_M(y) - P_N(y)| \geq \epsilon\right) \leq 2\exp\left(-C\epsilon^2(n+N)\right).$$

The proof is given in Appendix A. The result quantifies how well the distribution P_N approximates $P_{\infty,x}$ after only N updates. In particular, the approximation error decays exponentially fast in N, so we can expect the sampling scheme to approximate $P_{\infty,x}$ well already after a moderate number of steps. Further, setting $N = 0$ corresponds to the case where $P_N(y)$ equals the initial PPD given only the observed data. In our setting, this is the output from the PFN. If this PPD converges to a fixed distribution as $n \to \infty$, the proposition implies that $P_{\infty,x}(y)$ must have the same deterministic limit. Hence, the martingale posterior contracts at roughly the same speed at which the PPD converges.

3.4 Practical implementation

In practice, we can only sample finite sequences and replace the MP by its finite approximation. The procedure is summarized in Algorithm 1.

Algorithm 1 Computation of Martingale Posterior

```
Input: Estimated $\widehat{\text{PPD}}(y|x)$ obtained from the PFN.
for $b = 1, \dots, B$ do
    Initialize $P_0^{(b)} \leftarrow \widehat{\text{PPD}}(y|x)$.
    for $k = 1, \dots, N$ do
        Sample $y_{n+k}^{(b)} \sim P_k^{(b)}$.
        Update $(P_k^{(b)}, y_{n+k}^{(b)}) \to P_{k+1}^{(b)}$ as in Equation (1).
        Compute
            $\widehat{P}_N^{(b)}(y) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\Big\{ y_{n+i}^{(b)} \leq y \Big\}.$
        Set $\theta^{(b)}(x) \leftarrow \theta\big(\widehat{P}_N^{(b)}\big)$.
    end for
end for
Compute the estimated Martingale Posterior:
    $\widehat{\Pi}\Big(\theta(x) \in A | \mathcal{D}_n\Big) = \frac{1}{B} \sum_{b=1}^{B} \mathbf{1}\Big\{ \theta^{(b)}(x) \in A \Big\}.$
```

4 Numerical experiments

To demonstrate the efficacy of our approach, we conduct various experiments. Here, we focus on conditional posterior and coverage properties, but we also provide an unconditional posterior estimation example and comparison with classical predictive resampling in the Appendix. In all cases, we use the aforementioned combination of DPMMs and copula decomposition. To estimate ρ, we randomly draw 1000 data points simulated by the PFN.

4.1 Conditional regression posterior

As discussed in the previous section, our approach enables posterior estimation given new features x. We demonstrate this using two different regression data sets, a diffusion process and the data from Izmailov et al. (2020a). The diffusion process data is challenging as it evolves from an unimodal to a trimodal heteroscedastic Gaussian distribution with linear trends and sinusoidal changes depending on x. The second data set is used to evaluate how out-of-distribution (OOD) values for x affect the estimated posterior. In both cases, we use $B = 200$ replications, $N = 10{,}000$, and data sizes of $n = 200$ and $n = 400$, respectively.

Results The resulting posteriors are depicted by their respective density values in Figure 1. Both results suggest that the processes themselves and their uncertainty are well captured. For the diffusion process, it becomes apparent from the testing data that our approach can provide very accurate values for the posterior density despite the relatively few training data points. The results for the data from

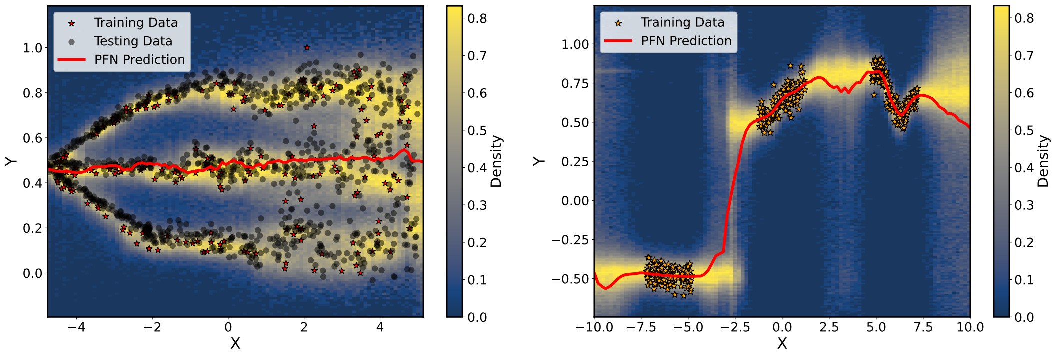

Figure 1: Visualization of PFN prediction and estimated Martingale Posterior (shaded area). Left: Diffusion process data. Right: OOD data as provided in Izmailov et al. (2020b).

Izmailov et al. (2020b), on the other hand, confirm that our approach is also able to detect the regions of OOD values and attribute higher probability to regions further away from the PFN prediction.

4.2 Conditional quantile coverages

While the previous example evaluates the obtained posterior as a whole, we now investigate its coverage properties for conditional statistics. For this, we simulate data using a heteroscedastic Gaussian funnel $\mathcal{N}(\sin(3x), x^2)$ for $x \in [0,1]$ for $n = 100$ data points. We then compute the posterior 90%-credible intervals (CIs) for the quantiles $P^{-1}(\alpha|x, \mathcal{D}_n)$, various α-levels, and three values of interest $x \in \{0.2, 0.5, 0.8\}$. For our routine, we use $B = 400$ and $N = 5000$. We repeat this process 20 times with a new random data set and check how often the true quantile is included in the computed CI in each repetition.

Table 1: Coverages (± two std. errors) of different quantiles (columns) at different x values (rows), highlighted where ranges include the nominal coverage.

$x\backslash\alpha$	**0.05**	**0.10**	**0.25**	**0.5**	**0.75**	**0.90**	**0.95**
$x = 0.2$	0.65 ± 0.21	0.60 ± 0.22	0.65 ± 0.21	0.75 ± 0.19	0.80 ± 0.18	0.85 ± 0.16	0.85 ± 0.16
$x = 0.5$	0.70 ± 0.21	0.70 ± 0.21	0.75 ± 0.19	0.85 ± 0.16	0.90 ± 0.13	0.75 ± 0.19	0.70 ± 0.21
$x = 0.8$	0.95 ± 0.10	0.85 ± 0.16	0.90 ± 0.13	1.00 ± 0.00	0.85 ± 0.16	0.70 ± 0.21	0.65 ± 0.21

Results The data is visualized in Figure 3 in Appendix B. The resulting coverages are shown in Table 1. Our method provides (close to) nominal coverage for many of the combinations of x and α but is less accurate for $x = 0.2$ and extremer quantiles in general. This can be explained by the fact that the data at $x = 0.2$ has almost no variation, while little data is available for extreme values, making these cases more challenging.

5 Discussion

This work proposes an efficient and principled Bayesian uncertainty quantification method for estimates derived from prior fitted networks. While our preliminary experimental results are promising, we aim to address several open problems in future work.

PPD updates using a Gaussian copula function enjoy nice computational and theoretical properties. Notably, the updates satisfy the martingale property, which is typically violated for transformer-based models. However, this introduces a slight inconsistency between the PPD computed from only observed data and subsequent PPDs involving both observed and simulated data. The computational and theoretical convenience is not exclusive to the Gaussian copula but is shared by many other copula models. To reduce the discrepancy between in- and out-of-sample PPD updates, one could search through a catalog of different copula models to see which fits the in-sample updates best.

REFERENCES

Bernard Bercu, Bernard Delyon, Emmanuel Rio, Bernard Bercu, Bernard Delyon, and Emmanuel Rio. *Concentration inequalities for martingales*. Springer, 2015.

Patrizia Berti, Luca Pratelli, and Pietro Rigo. Limit theorems for a class of identically distributed random variables. *The Annals of Probability*, 32(3):2029 – 2052, 2004.

Joseph L Doob. Application of the theory of martingales. *Le calcul des probabilites et ses applications*, pp. 23–27, 1949.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12784–12805. PMLR, 21–27 Jul 2024.

Benjamin Feuer, Robin Tibor Schirrmeister, Valeriia Cherepanova, Chinmay Hegde, Frank Hutter, Micah Goldblum, Niv Cohen, and Colin White. Tunetables: Context optimization for scalable prior-data fitted networks. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 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, 2023.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

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

David Huk, Yuanhe Zhang, Mark Steel, and Ritabrata Dutta. Quasi-bayes meets vines. *arXiv preprint arXiv:2406.12764*, 2024.

Pavel Izmailov, Wesley J Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace inference for bayesian deep learning. In *Uncertainty in Artificial Intelligence*, pp. 1169–1179. PMLR, 2020a.

Pavel Izmailov, Wesley J. Maddox, Polina Kirichenko, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Subspace Inference for Bayesian Deep Learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pp. 1169–1179, 2020b.

Si-Yang Liu and Han-Jia Ye. TabPFN Unleashed: A Scalable and Effective Solution to Tabular Classification Problems. *arXiv preprint arXiv:2502.02527*, 2025.

Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2022.

Thomas Nagler. Statistical foundations of prior-data fitted networks. In *International Conference on Machine Learning*, pp. 25660–25676. PMLR, 2023.

Arik Reuter, Tim G. J. Rudner, Vincent Fortuin, and David Rügamer. Can transformers learn full bayesian inference in context?, 2025.

Mark J Schervish. *Theory of statistics*. Springer Science & Business Media, 2012.

Valentin Thomas, Junwei Ma, Rasa Hosseinzadeh, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony Caterini. Retrieval & Fine-Tuning for In-Context Tabular Models. *arXiv preprint arXiv:2406.05207*, 2024.

A Proof of Proposition 3.3

We use arguments similar to Fong et al. (2023). It holds

$$\mathbb{E}[P_M(y)|y_{(n+1):(n+M-1)}] = (1-\alpha_{n+M})P_M(y) + \alpha_{n+M}\mathbb{E}_{y_{n+M}\sim P_{M-1}}[H_\rho(P_{M-1}(y), P_{M-1}(y_{n+M}))].$$

By the probability integral transform, it holds $P_M(y_{n+M+1}) \sim \text{Uniform}[0,1]$. Thus,

$$\mathbb{E}_{y_{n+M}\sim P_{M-1}}[H_\rho(P_{M-1}(y), P_M(y_{n+M}))] = \int H_\rho(P_{M-1}(y), u)du = P_{M-1}(y),$$

by the properties of the Gaussian copula. Hence,

$$\mathbb{E}[P_M(y)|y_{(n+1):(n+M-1)}] = P_{M-1}(y),$$

which implies that $P_M(y)$ is a martingale. Furthermore,

$$|P_M(y) - P_{M-1}(y)| \leq \alpha_{n+M} = O\left(\frac{1}{n+M}\right), \quad \text{for all } y \in \mathbb{R},$$

and

$$\sum_{i=N}^{\infty} \alpha_{n+i}^2 = O\left(\frac{1}{n+N}\right).$$

Now the Azuma-Hoeffding inequality (e.g., Bercu et al., 2015) yields the desired result. □

B Further numerical experiments and details

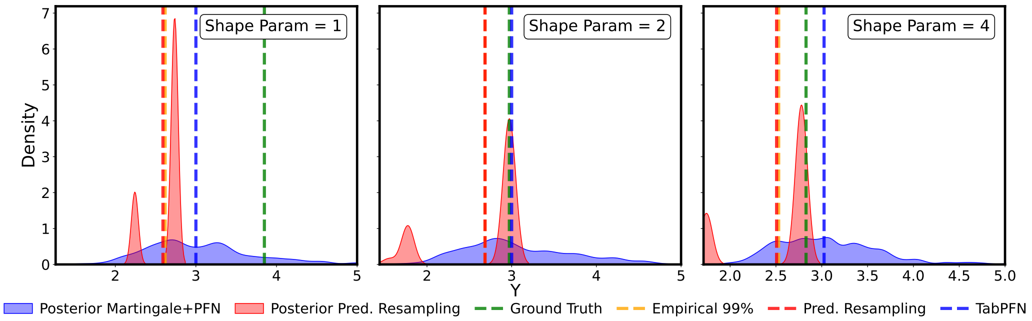

Figure 2: Comparison of the posterior from our method (blue) and the one obtained by predictive resampling (red) for different shape parameters γ (columns) of the Gamma distribution.

B.1 Unconditional quantile posterior

We here provide another experiment where we analyze the ability of our approach to estimate the posterior of an extreme quantile of a skewed distribution. For this, we simulate a Gamma distribution with different shape parameters $\gamma \in \{1, 2, 4\}$, inducing varying left-skewness. We then task PFN to estimate the 99%-quantile (the function T) and use our approach to compute the posterior uncertainty for PFN's estimate. In this experiment, we use $B = 1,000$ replications and $N = 10,000$. To make the task of quantile estimation more challenging, we use a relatively small data set size of $n = 25$. To evaluate the performance, we compare the distribution against the true value and a posterior estimate by the predictive resampling approach from Fong et al. (2023) that does not have access to the PFN.

Results An exemplary result for the posteriors for different shape values is visualized in Figure 2, showing the general trend of the results. The predictive resampling methods, which do not have access to the simulated data from TabPFN, usually result in a much more concentrated and bimodal posterior. This, however, comes at the cost of not always covering the true value. In contrast, the posterior for our method is much wider, thereby always covering the true value independent of the shape parameter.

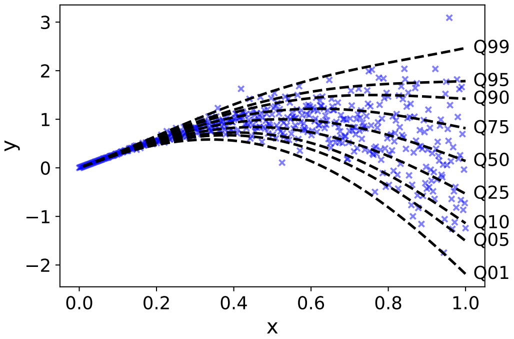

Figure 3: Depiction of the data used in Section 4.2.

B.2 Data generation for diffusion process

For the diffusion process data in Section 4.1, we use the following data-generating process:

1. Input generation: The input variable x is sampled uniformly from $[2.5, 12.5]$.
2. Piecewise functional behavior: The output y is determined by three different functions applied to x, separated by a data-dependent midpoint m, approximately at $\text{median}(x) - 2$. The three functional forms are:
   - $f_1(x)$: A linear and sinusoidal function switching at m.
   - $f_2(x)$: A mirrored version of $f_1(x)$.
   - $f_3(x)$: A piecewise function with a constant component and a high-frequency sine term.

   Each sample is randomly assigned one of these three functions.
3. Heteroscedastic noise: Additive noise is introduced with a scale that increases quadratically with x, making uncertainty larger for larger x.
4. Output transformation: The input is centered to be within $[-2.5, 2.5]$, and the output values are normalized based on their min-max range for potential use in quantile-based learning.

B.3 Data generation for quantile regression

For the data in Section 4.2, we sample values $x \sim U(0, 1)$ and generate the corresponding outcome via $y \sim \mathcal{N}(\sin(3x), x^2)$. A visualization of the data set is given in Figure 3.

B.4 Computational environment

All computations were performed on a user PC with Intel(R) Core(TM) i7-8665U CPU @ 1.90GHz, 8 cores, 16 GB RAM using Python 3.8, R 4.2.1, and TensorFlow 2.10.0. Run times of each experiment do not exceed 24 hours.

C Further discussion

Another open problem is joint posterior inference for a collection of parameters $\theta = \{\theta(x) \colon x \in \mathcal{X}\}$. Although our proposed inference procedure can be repeated for many values of x, the resulting posteriors are disconnected. For obtaining a full joint posterior $\Pi(\theta|\mathcal{D}_n)$, the distribution of the features $x_{1:n}$ can no longer be ignored. Fong et al. (2023) proposed a general joint update of the PPDs for all values of x simultaneously. However, this general update is intractable, and the heuristic simplifications proposed by Fong et al. (2023) are neither particularly simple nor theoretically justified. There are several potential ways forward. A simple practical solution would be to specify a joint distribution that combines the individual posteriors $P_{\infty,x_1}, \ldots, P_{\infty,x_K}$ in a plausible way; for example, using a multivariate Gaussian copula with covariance kernel depending on the distance between values $x_i \neq x_j$. We expect such a heuristic correction to work reasonably well in many applications. A more sophisticated alternative was recently proposed by Huk et al. (2024) and involves nonparametric estimation of the implicit dependence between PPDs by a nonparametric vine copula.