Adapting TabPFN for Zero-Inflated Metagenomics Data

Giulia Perciballi*
UMMISCO
IRD, Sorbonne Université
Bondy, France
giulia.perciballi@ird.fr

Federica Granese*
UMMISCO
IRD, Sorbonne Université
Bondy, France
federica.granese@ird.fr

Ahmad Fall
UMMISCO
IRD, Sorbonne Université
Bondy, France
ahmad.fall@ird.fr

Farida Zehraoui
IBISC
Université d'Evry, Paris-Saclay
Evry, France
farida.zehraoui@ibisc.univ-evry.fr

Edi Prifti
UMMISCO
IRD, Sorbonne Université, Bondy, France
INSERM, NutriOmique,
AP-HP Hôpital Pitié-Salpêtrière Paris, France
edi.prifti@ird.fr

Jean-Daniel Zucker
UMMISCO
IRD, Sorbonne Université, Bondy, France
INSERM, NutriOmique,
AP-HP Hôpital Pitié-Salpêtrière Paris, France
jean-daniel.zucker@ird.fr

Abstract

This paper introduces a novel prior assumption for TabPFN—a meta-learning method designed to approximate Bayesian inference on synthetic datasets generated from a predefined prior—aimed at better accommodating the unique zero-inflated distributions characteristic of metagenomic data. We modify the model's prior assumptions without changing its architecture by generating synthetic training data replicating the sparsity and variability inherent in these datasets. Preliminary results from metagenomic classification tasks show significant improvements in predictive performance, exceeding that of the original TabPFN and competing with state-of-the-art methods. This work emphasizes the necessity of tailoring PFN priors to align with the specific statistical properties of biomedical data, thereby enhancing their effectiveness in precision medicine.

1 Introduction

Recent advancements in deep learning for tabular data have led to the development of highly effective methods [1, 2, 3, 4, 5, 6], particularly those leveraging Prior-Data Fitted Networks (PFNs) [7]. PFNs operate within the realm of meta-learning, aiming to create self-adaptive algorithms that dynamically refine their learning bias based on accumulated meta-knowledge [8]. The goal is to generalize across different datasets, enabling models trained on a set of datasets to perform well on unseen datasets. Notably, the authors in [7] utilize PFNs to approximate Bayesian models for supervised learning tasks. TabPFN [9] is a remarkable implementation of this approach, efficiently using a

*Equal contributions

Table Representation Learning Workshop at NeurIPS 2024

Transformer-based architecture to process tabular data. By adapting to a broad prior, TabPFN aims to deliver strong out-of-the-box performance across various classification tasks without requiring task-specific retraining.

This paper assesses PFNs in contexts that deviate from standard statistical assumptions, particularly in metagenomics, where zero-inflated data is prevalent [10]. Metagenomic data derived from sequencing genetic material in environmental or clinical samples often exhibit sparsity and a high proportion of zero counts due to the low abundance or absence of specific microbial taxa. This sparsity leads to statistical distributions that standard PFN training may inadequately capture. Indeed, zero-inflated distributions present significant challenges for machine learning models not explicitly designed to handle them. These models may misinterpret excess zeros as noise or outliers, resulting in biased parameter estimates and sub-optimal predictive performance. While traditional approaches involve specialized statistical models, such as zero-inflated Poisson or negative binomial distributions [11, 12, 13], integrating these considerations into deep learning frameworks remains an ongoing challenge.

To bridge this gap, we propose an adaptation of TabPFN that involves modifying its prior generation procedure to explicitly account for the zero-inflated and compositional nature of metagenomic data. Compositional data are complex to model as they indicate relative values, as in microbiome data. When a species' abundance increases, the other species' abundance will decrease, affecting the distribution of all features throughout the samples. By adjusting the synthetic data generation process during the offline training phase, we can create training samples that reflect the true statistical properties of the target datasets. This approach does not require changes to the underlying architecture of TabPFN, preserving its efficiency and general applicability while enhancing its performance on specialized tasks. Our contributions are two-fold:

- We propose modelling our metagenomics prior to using zero-inflated species abundance distributions (SADs), which more accurately capture the sparsity and variability in microbial presence and abundance typical of metagenomic datasets. For each sample, we assume that each feature represents the relative abundance of a corresponding species. The probability of species presence is modelled using a Bernoulli random variable with probability p. Conditional on species presence, we model the abundance using a Log-normal or Log-uniform distribution in this preliminary study.
- We conduct experiments to evaluate the performance of the proposed PFN-tailored models on metagenomic classification datasets [14] and the OpenML-CC18 benchmark [15]. The results suggest that our adapted PFN outperforms the original TabPFN in this specific context and achieves results that are competitive with classical machine learning methods.

2 Zero-inflated Species Abundance Distributions Priors

We adopt the same theoretical setting as described in [7]. Let q_θ be a parameterized model that accepts a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and a query $\mathbf{x}$ as input, predicting a distribution over possible values of y for the query $\mathbf{x}$. Models like TabPFN are trained on dynamically generated synthetic datasets to minimize the Prior-Data Negative Log-Likelihood. In our approach, we aim to ensure that the features of the samples generated in the synthetic datasets follow a specific type of distribution. In this preliminary work, we focus on modelling the features of our dataset while assigning corresponding labels using a Bayesian Neural Network (BNN) [16], following the methodology outlined in [7, 9].

Specifically, the species abundance distribution (SAD) characterizes the distribution of abundances of all species within an ecological community. The observation that most species are relatively rare, with only a few being common, is often described as one of the few ecological laws [11]. Moreover, zero-inflated distributions have been proposed as appropriate models for describing the spatial distribution of rare species due to their ability to account for excess absences in the data [12]. Specifically, a zero-inflated distribution can be viewed as a two-part model in which (1) the probability of species presence and (2) the abundance, given presence, are modelled from the same data.

Formally, let $\mathbf{X} = (X^{(1)}, \ldots, X^{(k)})$ be the random variable (r.v.) for which we have realization $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^k$. In our setting, we assume each one of the species $X^{(j)}$, $j \in \{1, \ldots, k\}$, can be described as the following mixture model

$$X^{(j)} = \Delta Z_2^{(j)} + (1 - \Delta^{(j)}) Z_1^{(j)} \tag{1}$$

where $\Delta^{(j)} \sim Ber(p)$, $Z_1^{(j)} \sim P_{Z_1^{(j)}}$ and $Z_2^{(j)} \sim P_{Z_2^{(j)}}$. Specifically, we let $P_{Z_2^{(j)}} \equiv \delta_{\{0\}}$ (i.e., $Z_2^{(j)}$ is almost surely zero) and we assume $Z_1^{(j)}$ discrete. Additionally, as we deal with abundance data, we assume $\Pr[Z_1^{(j)} \geq 0] = 1$. Given a sample, we assume that each feature represents the relative abundance of the corresponding species, and we model the probability of species presence using a Bernoulli random variable with probability p. Since we are dealing with a zero-inflated distribution, we set the probability p to be reasonably low. Several distributions, given presence, are available to model abundances, such as the Log-normal, Negative Binomial, Poisson, and Log-series distributions [11, 12]. In this preliminary version of our study, we use the most commonly applied model for describing the SAD: the log-normal distribution [13]. Moreover, we will consider log-uniform distribution as well. We address the compositionality of the data as described in Section 3.

3 Experimental setting and results

Datasets and data pre-processing. We evaluated the models using publicly curated metagenomic datasets from ExperimentHub [14], which includes six cohorts with data at six taxonomic levels. We focused on the species-level taxa (microbial species relative abundance) across five cohorts: Cirrhosis, Inflammatory Bowel Disease, Obesity, and Type 2 Diabetes (Chinese patients - T2D, European women - WT2D). We pre-processed the data as in [17] (see Table 2 and Appendix A.2).

Prior generation algorithm. In Algorithm 1 (see Appendix A.1), we show the prior feature generation procedure when considering zero-inflated uniform and zero-inflated log-normal distributions. Given n samples and k features, the aim is to generate n samples whose features are sampled from different priors. For each feature j, the algorithm samples two probabilities $(p_{\text{Ber}}^{(j)}, p_{\text{Log}}^{(j)})$ from different Gaussian distributions, both truncated to the interval $[0, 1]$. The value $p_{\text{Log}}^{(j)}$ is then used to construct a Bernoulli random variable that determines whether a feature is sampled from a uniform or a log-normal distribution (Algorithm 1, lines 3-5). In our experiments, if the uniform distribution is chosen, its parameters a and b are set to 0 and 1, respectively. To derive the mean $\mu^{(j)}$ and standard deviation $\sigma^{(j)}$ for the log-normal distribution Log-Norm$(\mu^{(j)}, \sigma^{(j)})$, $\mu^{(j)}$ is sampled from a normal distribution. In contrast, the standard deviation is calculated by multiplying $\mu^{(j)}$ with a value sampled from a different normal distribution $\mu^{(j)'}$ and taking the absolute value (Algorithm 1, lines 9-12). As a last step, for each sample i, zero-inflation is applied using the previously sampled $p_{\text{Ber}}^{(j)}$ and constructing a Bernoulli random variable to model feature presence. The resulting synthetic dataset has dimensions (n, k). In our experiments, for the zero-inflated distributions, $\mu_{p_{\text{Ber}}}$ is set to 0.8 and $\sigma_{p_{\text{Ber}}}$ is fixed at 0.1. The parameters $\mu_{p_{\text{Log}}}$ and $\sigma_{p_{\text{Log}}}$ for the log-normal distribution were set to 0.7 and 0.1, respectively. These empirical parameter values were derived based on a study on a separate set of the ExperimentHub dataset, which was not involved in testing the model [14]. We follow the same procedure as in [7, 9] for the BNN priors to generate the labels to associate with the synthetic samples. For further details, we refer the reader to the mentioned papers.

PFN-tailored models investigated. We trained six different PFN models. The term `LogN-U` refers to models trained on priors generated from Log-Normal and Log-Uniform distributions without zero-inflation (i.e., $p = 0$ in Eq. (1)). Conversely, `Z-LogN-U` refers to models trained on priors incorporating zero-inflation (i.e., $p > 0$ in Eq. (1)). To account for the compositionality of metagenomic data, we applied sample-wise normalization in this preliminary study, indicated by ($\Rightarrow$) in the results. Additionally, we considered feature-wise scaling ($\Downarrow$) to maintain a consistent data range at the model input, addressing the variability in feature ranges. The same training procedure was used for all models, as detailed in Appendix A.3.

3.1 Discussion of the results

TabPFN and the PFN-tailored models on metagenomics data. We analyze the performance of the original TabPFN model from [9] and our PFN-tailored models when applied to metagenomics data. The original TabPFN model presents a similar behaviour overall datasets as the classical machine learning techniques with better performance for the WT2D dataset, with an improvement of 8.2 percentage points in terms of accuracy and 16.8 in terms of AUROC (cf. Table 3).

Table 1: We performed 10-fold cross-validation on datasets from ExperimentHub [14]. (⇒) denotes the sample-wise normalization, (⇓) the feature-wise scaling.

Methods	Cirrhosis1 (181×100)		Cirrhosis2 (56×100)		IBD (396×100)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	83.4 ± 8.6	**93.1** ± 5.6	85.0 ± 17.4	93.3 ± 11.3	77.5 ± 7.1	**87.2** ± 5.3
`LogN-U`	83.4 ± 10.6	91.8 ± 6.8	85.0 ± 20.3	92.2 ± 14.1	79.8 ± 7.1	86.3 ± 5.0
`LogN-U` (⇒)	**84.5** ± 9.6	93.0 ± 5.5	83.0 ± 19.7	93.3 ± 10.2	77.3 ± 7.3	86.7 ± 6.5
`LogN-U` (⇓)	80.1 ± 10.6	91.0 ± 6.6	86.3 ± 16.4	**94.4** ± 11.4	77.0 ± 6.2	86.1 ± 4.8
`Z-LogN-U`	82.3 ± 8.9	92.2 ± 6.1	**86.7** ± 18.0	**94.4** ± 11.4	**80.3** ± 7.6	86.7 ± 5.2
`Z-LogN-U` (⇒)	82.9 ± 8.3	92.8 ± 5.7	84.7 ± 17.5	92.2 ± 12.2	**80.3** ± 6.3	86.7 ± 6.3
`Z-LogN-U` (⇓)	83.9 ± 10.1	92.6 ± 6.9	85.0 ± 17.4	93.3 ± 11.3	78.5 ± 5.5	87.1 ± 4.8
Methods	Obesity (263×100)		T2D (344×100)		WT2D (96×100)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	63.5 ± 4.7	60.0 ± 13.8	59.0 ± 4.0	61.6 ± 7.4	65.2 ± 16.6	69.6 ± 17.1
`LogN-U`	**65.5** ± 7.9	**61.9** ± 13.8	**59.9** ± 7.4	**63.3** ± 7.5	64.1 ± 19.8	74.4 ± 18.7
`LogN-U` (⇒)	62.4 ± 9.1	58.6 ± 13.6	56.7 ± 6.7	60.0 ± 8.1	63.3 ± 15.9	67.6 ± 19.3
`LogN-U` (⇓)	63.5 ± 6.9	60.1 ± 13.9	54.7 ± 6.4	59.6 ± 7.4	66.2 ± 13.4	73.7 ± 16.7
`Z-LogN-U`	63.1 ± 4.3	60.4 ± 13.9	57.0 ± 4.1	62.1 ± 8.9	65.1 ± 15.9	72.9 ± 16.9
`Z-LogN-U` (⇒)	65.0 ± 4.0	61.3 ± 15.0	51.5 ± 8.5	57.3 ± 11.3	63.4 ± 11.4	64.7 ± 18.1
`Z-LogN-U` (⇓)	65.1 ± 6.1	61.0 ± 15.0	56.1 ± 5.7	61.6 ± 8.9	**71.4** ± 14.0	**76.7** ± 17.4

In Table 1, we compare PFN-tailored and the original TabPFN models. Overall, the accuracy of our models exceeds that of the original TabPFN across all datasets. Similarly, our models generally achieve higher AUROC scores, except for the IBD and Cirrhosis1 datasets, with a difference of only 0.1 percentage points. This indicates improved performance when handling metagenomic data. Our analysis shows that using different priors enhances the model's ability to learn the variable behaviours characteristic of metagenomic data. Specifically, we find that introducing zero inflation in the IBD datasets improves model performance in terms of accuracy. The most notable performance gains compared to the original TabPFN occur on more challenging datasets. For example, with the `Z-LogN-U` model, we observe improvements of 2 percentage points in accuracy and one percentage point in AUROC on the Cirrhosis2 dataset, along with a 3-point accuracy increase for the IBD dataset, highlighting the importance of zero-inflation in certain contexts. Conversely, scaling contributes to a nearly 7-point increase in accuracy on the WT2D dataset. These results demonstrate that the combination of zero-inflation and scaling/normalization markedly enhances TabPFN's performance, paving the way for further improvements. Moreover, as shown in Table 3, our PFN-tailored models generalize better than classical machine learning models, outperforming RF, the best between the baselines methods, of 14.4 percentage points in terms of accuracy and 23.9 in terms of AUROC. Finally, we acknowledge the high standard deviation observed in the results, indicating potential variability in model performance across different folds. However, we note that similar behaviour is observed for the classical machine learning methods in Table 3, suggesting that small variations in the training set could significantly impact the final performance of the models. In Table 5, we provided the results by considering the Leave-One-Out Cross-Validation approach.

PFN-tailored models on OpenML-CC18. We analyzed the performance of our PFN-tailored models on datasets that may not conform to the SAD distribution. Specifically, we examined datasets from OpenML-CC18 [15], also evaluated in the original TabPFN paper [9]. Due to space constraints, we have relegated the complete set of results to Appendix B.2. As expected, our PFN-tailored models performed worse than the original TabPFN on datasets lacking relative abundance data. However, we noted some exceptions, particularly with ID-37, the *diabetes* dataset, which has the highest number of zero values among the datasets examined, with 7 out of 8 features with a maximum of 374 zero values across 768 entries per feature. Notably, the `Z-LogN-U` model improved by 1.3 percentage points over the original model. We observed a similar pattern in other datasets containing zero-valued features (cf. ID-29, ID-1063, ID-1464, and ID-1510 in Table 4). In contrast, we highlight a significant decline in performance for the retrained models on ID-54, ID-188, and ID-6332. The features in these datasets are specific to their respective studies (e.g., measures related to cylinders), which contradicts the typology of priors we considered during the meta-learning phase. Notably, in the case of ID-6332, we observed a performance decline of up to 17.8 percentage points in accuracy and 21.1 in AUROC.

The drop in the performance of PFN-tailored models on datasets outside our initial hypothesis is expected, as we aim to create context-specific models rather than general-purpose models like the original TabPFN. However, the models' strong performance in cases of pronounced zero inflation suggests they effectively capture the high prevalence of zeros in the feature distributions.

4 Conclusion

This paper explores microbial species' relative abundance data and proposes PFN-like models as a promising solution. By modifying the TabPFN [9, 7] model's prior assumptions—without altering its architecture—we generate synthetic training data that reflects the sparsity and variability typical of these datasets. The implications of our findings extend beyond metagenomics, suggesting that PFNs can be tailored for various specialized domains by properly adjusting their priors (as done, for example, in [18]). This adaptability positions PFNs as powerful tools in fields where data exhibit non-standard distributions, particularly where traditional deep learning models may struggle without significant retraining or architectural changes.

The framework proposed should be expanded to explicitly address data compositionality, especially relevant in metagenomics and other areas where features represent components of a whole. Identifying suitable parameters for zero-inflated distributions could yield PFN-tailored models that consistently outperform both the original TabPFN and classical methods across diverse metagenomic datasets.

This study serves as a preliminary comparison of our tailored PFNs against the original TabPFN model. We also plan to provide a comprehensive benchmark of zero-inflated and compositional datasets, enhancing the dataset suite used in the TabPFN family. This will validate the efficacy of our adapted model and offer valuable resources for the research community to develop and test new methods suited to complex data distributions.

5 Acknowledgements

The ANR supported this work (Agence Nationale de la Recherche) under grant number ANR-21-CE45-0030, as part of the DeepIntegromics project. This work was granted access to the HPC resources of IDRIS under the allocations 2023-AD011014580 and 2024-AD011014580R1 made by GENCI. The ANR-20-CE17-0022 DeepECG4U funding from the French National Research Agency supported the work of Federica Granese.

References

[1] W. Wydmański, O. Bulenok, M. Śmieja, Hypertab: Hypernetwork approach for deep learning on small tabular datasets, in: 2023 IEEE 10th International Conference on Data Science and Advanced Analytics (DSAA), IEEE, 2023, pp. 1–9.

[2] P. Chen, S. Sarkar, L. Lausen, B. Srinivasan, S. Zha, R. Huang, G. Karypis, Hytrel: Hypergraph-enhanced tabular data representation learning, Advances in Neural Information Processing Systems 36 (2024).

[3] D. Bonet, D. M. Montserrat, X. Giró-i Nieto, A. G. Ioannidis, Hyperfast: Instant classification for tabular data, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 11114–11123.

[4] D. Rundel, J. Kobialka, C. von Crailsheim, M. Feurer, T. Nagler, D. Rügamer, Interpretable machine learning for tabpfn, arXiv preprint arXiv:2403.10923 (2024).

[5] B. Feuer, R. T. Schirrmeister, V. Cherepanova, C. Hegde, F. Hutter, M. Goldblum, N. Cohen, C. White, Tunetables: Context optimization for scalable prior-data fitted networks, arXiv preprint arXiv:2402.11137 (2024).

[6] A. Müller, C. Curino, R. Ramakrishnan, Mothernet: A foundational hypernetwork for tabular classification, arXiv preprint arXiv:2312.08598 (2023).

[7] S. Müller, N. Hollmann, S. P. Arango, J. Grabocka, F. Hutter, Transformers can do bayesian inference, arXiv preprint arXiv:2112.10510 (2021).

[8] R. Vilalta, Y. Drissi, A perspective view and survey of meta-learning, Artificial intelligence review 18 (2002) 77–95.

[9] N. Hollmann, S. Müller, K. Eggensperger, F. Hutter, Tabpfn: A transformer that solves small tabular classification problems in a second, arXiv preprint arXiv:2207.01848 (2022).

[10] G. Roy, E. Prifti, E. Belda, J.-D. Zucker, Deep learning methods in metagenomics: a review, Microbial Genomics 10 (4) (2024) 001231.

[11] T. J. Matthews, R. J. Whittaker, On the species abundance distribution in applied ecology and biodiversity management, Journal of Applied Ecology 52 (2) (2015) 443–454.

[12] S. J. Wenger, M. C. Freeman, Estimating species occurrence, abundance, and detection probability using zero-inflated distributions, Ecology 89 (10) (2008) 2953–2959.

[13] F. W. Preston, The commonness, and rarity, of species, Ecology 29 (3) (1948) 254–283.

[14] E. Pasolli, L. Schiffer, P. Manghi, A. Renson, V. Obenchain, D. T. Truong, F. Beghini, F. Malik, M. Ramos, J. B. Dowd, et al., Accessible, curated metagenomic data through experimenthub, Nature methods 14 (11) (2017) 1023–1024.

[15] B. Bischl, G. Casalicchio, M. Feurer, F. Hutter, M. Lang, R. G. Mantovani, J. N. van Rijn, J. Vanschoren, Openml benchmarking suites, arXiv:1708.03731v2 [stat.ML] (2019).

[16] V. Mullachery, A. Khera, A. Husain, Bayesian neural networks, arXiv preprint arXiv:1801.07710 (2018).

[17] E. Prifti, Y. Chevaleyre, B. Hanczar, E. Belda, A. Danchin, K. Clément, J.-D. Zucker, Interpretable and accurate prediction models for metagenomics data, GigaScience 9 (3) (2020) giaa010.

[18] J. Ubbens, I. Stavness, A. G. Sharpe, Gpfn: Prior-data fitted networks for genomic prediction, bioRxiv (2023) 2023–09.

[19] J. Davis, M. Goadrich, The relationship between precision-recall and roc curves, in: Proceedings of the 23rd international conference on Machine learning, 2006, pp. 233–240.

A Additional Material to Section 3

A.1 Prior generation algorithm

In Algorithm 1 we denote by n the number of samples to generate; by k the dimension of each sample meaning the number of features; $\mu_{p_{\text{Ber}}}$ is set to 0.8 and $\sigma_{p_{\text{Ber}}}$ is fixed at 0.1, to constrain the generation process to create a high number of features with zero values. The parameters $\mu_{p_{\text{Log}}}$ and $\sigma_{p_{\text{Log}}}$ for the log-normal distribution were set to 0.7 and 0.1, respectively. These empirical parameter values were derived based on a study on a separate set of the ExperimentHub dataset, which was not involved in testing the model [14]. For the uniform distribution, we consider $a = 0$ and $b = 1$.

Algorithm 1: Zero-inflated SAD prior feature generation procedure

Input: k is the number of features in the synthetic dataset; n is the number of samples; $\mu_{p_{\text{Ber}}}$ and $\sigma_{p_{\text{Ber}}}$ are the parameters for sampling from the Gaussian distribution related to the probability p_{Ber}; $\mu_{p_{\text{Log}}}$ and $\sigma_{p_{\text{Log}}}$ are the parameters for sampling from the Gaussian distribution related to the probability p_{Log}; and a and b are the parameters for the uniform distribution.

Output: A synthetic matrix of n samples and k features $\mathcal{S} = (\mathbf{x}_i)_{i=1}^{n}$, with $\mathbf{x} \in \mathbb{R}^k$.

$\mathcal{S} \leftarrow \text{zeros}((n, k))$
for $j \leftarrow 1$ **to** k **do**
  Sample $p_{\text{Ber}}^{(j)} \sim \mathcal{N}(\mu_{p_{\text{Ber}}}, \sigma_{p_{\text{Ber}}})$ truncated in $[0, 1]$
  Sample $p_{\text{Log}} \sim \mathcal{N}(\mu_{p_{\text{Log}}}, \sigma_{p_{\text{Log}}})$ truncated in $[0, 1]$
  Sample $\delta_{\text{Log}}^{(j)} \sim Ber(p_{\text{Log}}^{(j)})$
  if $\delta_{Log}^{(j)} == 0$ **then**
    $P \leftarrow \text{Uniform}(a, b)$
  else
    Sample $\mu^{(j)} \sim \mathcal{N}(0, 1)$
    Sample $\mu^{(j)'} \sim \mathcal{N}(0, 1)$
    $\sigma^{(j)} \leftarrow |\mu^{(j)'} \cdot \mu^{(j)}|$
    $P \leftarrow \text{Log-norm}(\mu^{(j)}, \sigma^{(j)})$
  for $i \leftarrow 1$ **to** n **do**
    Sample $\delta_{\text{Ber}_i}^{(j)} \sim Ber(p_{\text{Ber}}^{(j)})$
    if $\delta_{Ber_i}^{(j)} == 0$ **then**
      Sample $x_i^{(j)} \sim P$
    else
      $x_i^{(j)} \leftarrow 0$
    $\mathcal{S}[i, j] \leftarrow x_i^{(j)}$
return $\mathcal{S}$

A.2 Features before and after filtering

The filtering process consisted of two steps: the first based on a **presence/absence** and **abundance** filtering by class, and the second on the ANOVA F-statistic to select the first 100 most important features. The whole preprocessing process was performed on the entire dataset, equally for all the methods, before implementing k-fold CV. This choice has been made because of the nature of the metagenomic datasets, for which the presence/absence of a feature could significantly affect the model performances in a fold.

In particular:

- **Presence/absence filtering** involved counting the non-zero occurrences (c) for each feature and retaining those where $c >$ total_samples $\cdot$ 0.2.

- **Abundance-based filtering** retained features with a maximum relative abundance exceeding a threshold of 0.01, set close to the detection limit.

Features that did not meet these criteria were removed. During prior fitting, if the synthetic input data is normalized by sample or scaled by feature, we apply the same procedure when evaluating real data. The table below shows the dimensions of the datasets before and after filtering.

Table 2: Summary of the dataset dimensions before and after filtering (samples $\times$ features)

Dataset	Before Filtering	After Filtering
Cirrhosis1 (Cir1)	181×1045	181×336
Cirrhosis2 (Cir2)	56×1045	56×291
Type 2 Diabetes (Chinese patients - T2D)	344×1045	344×288
Type 2 Diabetes (European women - WT2D)	96×1045	96×255
Inflammatory Bowel Disease (IBD)	396×1045	396×305
Obesity	263×1045	263×247

A.3 TabPFN [9] re-training procedure

We consider the same TabPFN architecture, built on a Transformer with an initial embedding layer of size 512 and 12 encoder blocks, each containing four attention layers, residual connections, and normalization layers. The architecture ends with a linear decoder layer. The attention module employs a self-attention mask, allowing training examples to attend to themselves while enabling validation examples to attend only to training examples. The number of input features is sampled from 0 to 100. Similarly, the number of output classes is randomly selected between 0 and a maximum during training. The final model contains 25.82M parameters. The training consists of 200 epochs and 200 steps to align with the total number of synthetic datasets generated and used from the original paper and uses an adaptive learning rate. Within each step, 128 datasets with a fixed size of 1024 samples are generated, resulting in 128×200 datasets per epoch ($\sim$ 5 million datasets). We split each dataset into training and validation sets randomly.

We conducted each experiment on a machine equipped with an Intel® Xeon® Gold 6226 CPU, a clock frequency of 2.70 GHz, and a Tesla V100-SXM2-32GB GPU. The training lasted about 15 hours using 8 GPUs, with each epoch taking approximately 450 seconds.

Evaluation metrics. A 10-fold cross-validation (CV) was performed to evaluate the performance of the models and ensure their generalizability. The metrics used for evaluation were the **Accuracy** score and the Area Under the Receiver Operating Characteristic curve score **(AUROC)** [19]. Both metrics are reported in the Table 3.

B Additional Results

B.1 Comparison with classical machine learning techniques

To gain insights into the inherent complexities of each dataset in the Experimental Hub [14], we evaluated them using classical machine learning methods, including Random Forest (RF), Support Vector Classifier (SVC), and LightGBM (lGBM). No hyperparameter tuning was performed for baselines and the results are presented in Table 3. As expected, the datasets Cirrhosis1, Cirrhosis2, and IBD were identified as the least complex, with RF achieving particularly high accuracy and AUROC scores. In contrast, the datasets Obesity, T2D, and WT2D proved to be the most challenging, as the baseline methods performed similarly to a random classifier. Notably, in these cases, our context-specific TabPFN demonstrated the greatest improvements.

B.2 SAD priors over OpenML-CC18

In this setting, we restricted our analysis to datasets with fewer than 1000 samples and 100 features, filtering out non-numerical data. The results are presented in Table 4. Each column indicates the identity number of the corresponding dataset.

Table 3: We performed 10-fold cross-validation on various datasets from ExperimentHub [14] and pre-processed as in [17]. The indicated shape reflects the data after filtering. ($\Rightarrow$) denotes the sample-wise normalization, ($\Downarrow$) the feature-wise scaling. The overall best result is **bold**, while the best among the PFN models is underlined.

Methods	Cirrhosis1 (181×100)		Cirrhosis2 (56×100)		IBD (396×100)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
RF	**86.2** ± 7.5	**94.1** ± 5.3	89.7 ± 11.3	**97.8** ± 4.4	77.7 ± 8.0	**88.0** ± 5.4
SVC	78.9 ± 8.6	86.4 ± 8.7	79.7 ± 22.1	86.7 ± 19.1	73.0 ± 6.2	81.1 ± 7.0
lGBM	86.2 ± 8.7	93.6 ± 5.0	**90.0** ± 11.1	92.2 ± 10.0	78.0 ± 7.0	87.1 ± 5.4
TabPFN [9]	83.4 ± 8.6	93.1 ± 5.6	85.0 ± 17.4	93.3 ± 11.3	77.5 ± 7.1	87.2 ± 5.3
LogN-U	83.4 ± 10.6	91.8 ± 6.8	85.0 ± 20.3	92.2 ± 14.1	79.8 ± 7.1	86.3 ± 5.0
LogN-U($\Rightarrow$)	84.5 ± 9.6	93.0 ± 5.5	83.0 ± 19.7	93.3 ± 10.2	77.3 ± 7.3	86.7 ± 6.5
LogN-U($\Downarrow$)	80.1 ± 10.6	91.0 ± 6.6	86.3 ± 16.4	94.4 ± 11.4	77.0 ± 6.2	86.1 ± 4.8
Z-LogN-U	82.3 ± 8.9	92.2 ± 6.1	86.7 ± 18.0	94.4 ± 11.4	**80.3** ± 7.6	86.7 ± 5.2
Z-LogN-U ($\Rightarrow$)	82.9 ± 8.3	92.8 ± 5.7	84.7 ± 17.5	92.2 ± 12.2	**80.3** ± 6.3	86.7 ± 6.3
Z-LogN-U ($\Downarrow$)	83.9 ± 10.1	92.6 ± 6.9	85.0 ± 17.4	93.3 ± 11.3	78.5 ± 5.5	87.1 ± 4.8
Methods	Obesity (263×100)		T2D (344×100)		WT2D (96×100)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
RF	61.3 ± 4.0	57.1 ± 13.0	57.9 ± 6.2	61.1 ± 8.9	57.0 ± 12.8	52.8 ± 18.8
SVC	62.4 ± 2.2	54.4 ± 11.9	51.8 ± 6.1	40.3 ± 5.5	52.9 ± 9.4	43.3 ± 17.4
lGBM	58.2 ± 6.6	55.4 ± 8.3	56.2 ± 8.5	60.1 ± 11.2	47.8 ± 13.1	47.2 ± 16.7
TabPFN [9]	63.5 ± 4.7	60.0 ± 13.8	59.0 ± 4.0	61.6 ± 7.4	65.2 ± 16.6	69.6 ± 17.1
LogN-U	**65.5** ± 7.9	**61.9** ± 13.8	**59.9** ± 7.4	**63.3** ± 7.5	64.1 ± 19.8	74.4 ± 18.7
LogN-U($\Rightarrow$)	62.4 ± 9.1	58.6 ± 13.6	56.7 ± 6.7	60.0 ± 8.1	63.3 ± 15.9	67.6 ± 19.3
LogN-U($\Downarrow$)	63.5 ± 6.9	60.1 ± 13.9	54.7 ± 6.4	59.6 ± 7.4	66.2 ± 13.4	73.7 ± 16.7
Z-LogN-U	63.1 ± 4.3	60.4 ± 13.9	57.0 ± 4.1	62.1 ± 8.9	65.1 ± 15.9	72.9 ± 16.9
Z-LogN-U ($\Rightarrow$)	65.0 ± 4.0	61.3 ± 15.0	51.5 ± 8.5	57.3 ± 11.3	63.4 ± 11.4	64.7 ± 18.1
Z-LogN-U ($\Downarrow$)	65.1 ± 6.1	61.0 ± 15.0	56.1 ± 5.7	61.6 ± 8.9	**71.4** ± 14.0	**76.7** ± 17.4

Below is a brief description of the datasets[2].

- **ID** (**11**): *balance-scale* dataset. This dataset models psychological experiments. Each example is classified as the balance scale tipping to the right, left, or staying balanced. (No features with zero values).
- **ID** (**15**): *breast-w* dataset. Features are computed from digitized images of fine needle aspirates (FNA) of breast masses, describing cell nuclei characteristics. The target feature indicates whether the prognosis is malignant or benign. (No features with zero values).
- **ID** (**29**): *credit-approval* dataset. This dataset comprises credit card applications, with all attribute names and values anonymized. It includes a mix of continuous and nominal features with varying numbers of values. (Feature 'A3' contains 19 zeros, 'A8' has 70, 'A11' has 395, 'A14' has 132, and 'A15' has 295 zeros, out of 690 entries per feature).
- **ID** (**37**): *diabetes* dataset. The diagnostic variable is binary, indicating if a patient has diabetes based on WHO criteria (plasma glucose level $\geq$ 200 mg/dl). Features include patient data like age, body mass index, and number of pregnancies. (Feature 'preg' has 111 zeros, 'plas' has 5, 'pres' has 35, 'skin' has 227, 'insu' has 374, and 'mass' has 11 zeros, out of 768 entries per feature).
- **ID** (**54**): *vehicle* dataset. This dataset classifies silhouettes of four vehicle types using features extracted by the HIPS system. Features include scale-independent measures like variance, skewness, kurtosis, hollows, circularity, rectangularity, and compactness. (No features with zero values).
- **ID** (**188**): *eucalyptus* dataset. The objective is to determine the best seed lots for soil conservation in dry hill countries by measuring height, diameter, survival, and other factors. (No features with zero values).

[2]From `https://openml.org/`.

ID (**1063**): *kc2* dataset. A NASA defect dataset from science software data processing. Features are derived from McCabe and Halstead extractors, which characterize software code quality. (Feature ‘v‘ has 16 zeros, ‘l‘, ‘d‘, ‘i‘, and ‘e‘ each have 23 zeros, out of 522 entries per feature).

ID (**1464**): *blood-transfusion-service-center* dataset. This classification dataset contains data from a blood transfusion centre in Taiwan. Features include months since the last donation, total donations, and total blood donated. (Feature ‘V1‘ has five zeros out of 748 entries per feature).

ID (**1480**): *ilpd* dataset. This dataset contains records of liver patients and non-liver patients from Andhra Pradesh, India. The class label divides patients into liver and non-liver groups. Features include patient information like age and gender. (No features with zero values).

ID (**1510**): *wdbc* dataset. Description as in *breast-w* dataset. (Features ‘V7‘ and ‘V8‘ each have 13 zeros out of 569 entries per feature).

ID (**6332**): *cylinder-bands* dataset. This dataset addresses process delays in rotogravure printing caused by cylinder banding. Features include measures related to cylinders. (Feature ‘varnish_pct‘ has 213 zeros, ‘press_speed‘ has 1 zero, out of 540 entries per feature).

ID (**40994**): *climate-model-simulation-crashes* dataset. This dataset records simulation crashes during climate model uncertainty quantification (UQ) ensembles. The goal is to predict simulation outcomes (success or failure) and analyze the causes of crashes using classification and feature selection. (No features with zero values).

Table 4: We performed 10-fold cross-validation on various datasets from OpenML-CC18 [15]. We considered datasets with fewer than 1000 samples, 100 features, and 10 classes. Columns containing non-numerical data were filtered out for datasets marked with a $\star$. The indicated shape reflects the data after filtering. ($\Rightarrow$) denotes the sample-wise normalization, ($\Downarrow$) the feature-wise scaling.

Methods	ID-11 (625×4)		ID-15 (699×9)		ID-29* (690×6)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	**99.2** ± 1.1	**100.0** ± 0.1	97.1 ± 0.9	**99.5** ± 0.3	77.1 ± 5.8	**84.3** ± 5.7
LogN-U	91.5 ± 1.4	96.6 ± 1.8	95.7 ± 1.7	98.8 ± 0.9	77.4 ± 5.5	84.1 ± 5.9
LogN-U($\Rightarrow$)	93.3 ± 2.3	97.7 ± 1.5	89.6 ± 2.5	96.1 ± 2.4	75.9 ± 4.6	82.1 ± 5.8
LogN-U($\Downarrow$)	90.9 ± 1.9	97.1 ± 1.3	95.7 ± 3.1	98.8 ± 0.9	**77.7** ± 4.9	83.4 ± 6.7
Z-LogN-U	90.9 ± 1.2	97.1 ± 1.1	**97.3** ± 1.4	98.9 ± 0.9	77.5 ± 6.0	84.2 ± 5.5
Z-LogN-U ($\Rightarrow$)	92.2 ± 1.8	98.9 ± 0.9	91.0 ± 3.1	96.1 ± 2.6	75.4 ± 5.6	81.7 ± 5.8
Z-LogN-U ($\Downarrow$)	91.2 ± 1.8	97.4 ± 1.0	96.4 ± 2.3	98.7 ± 0.8	77.1 ± 4.9	83.3 ± 6.6
Methods	ID-37 (768×8)		ID-54 (846×18)		ID-188* (736×14)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	76.2 ± 3.1	**83.8** ± 4.5	**84.5** ± 3.8	**96.9** ± 1.0	**69.7** ± 4.7	**92.7** ± 1.3
LogN-U	77.0 ± 2.5	83.4 ± 3.5	73.3 ± 4.2	93.0 ± 1.1	63.6 ± 4.6	89.2 ± 2.3
LogN-U($\Rightarrow$)	66.4 ± 1.9	72.0 ± 4.7	71.4 ± 5.4	92.5 ± 1.1	56.1 ± 4.0	84.5 ± 2.2
LogN-U($\Downarrow$)	76.2 ± 2.5	83.2 ± 3.9	74.7 ± 3.3	93.0 ± 1.2	41.0 ± 6.7	68.0 ± 4.7
Z-LogN-U	**77.5** ± 2.9	83.5 ± 3.6	72.7 ± 3.2	92.3 ± 1.0	62.5 ± 4.4	88.7 ± 2.5
Z-LogN-U ($\Rightarrow$)	70.4 ± 3.4	73.0 ± 4.9	70.9 ± 4.7	91.9 ± 1.1	57.1 ± 4.5	84.4 ± 2.5
Z-LogN-U ($\Downarrow$)	76.4 ± 2.8	83.1 ± 3.7	74.1 ± 4.6	92.6 ± 1.2	40.5 ± 5.6	67.7 ± 5.2
Methods	ID-1063 (522×21)		ID-1464 (748×4)		ID-1480* (583×9)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	**85.1** ± 3.7	84.3 ± 9.6	79.8 ± 2.6	**76.2** ± 5.2	**72.5** ± 4.5	**75.4** ± 7.1
LogN-U	83.9 ± 4.7	83.2 ± 9.5	**80.1** ± 3.8	**76.2** ± 6.2	71.7 ± 2.6	75.3 ± 6.6
LogN-U($\Rightarrow$)	82.2 ± 3.4	82.3 ± 8.4	75.9 ± 1.1	74.1 ± 6.1	71.4 ± 0.7	72.4 ± 6.8
LogN-U($\Downarrow$)	82.6 ± 4.1	82.2 ± 9.1	79.8 ± 3.7	75.6 ± 6.0	71.4 ± 1.9	74.6 ± 7.1
Z-LogN-U	79.3 ± 1.1	84.2 ± 9.4	77.1 ± 2.5	75.6 ± 5.1	71.2 ± 1.3	74.0 ± 4.9
Z-LogN-U ($\Rightarrow$)	79.5 ± 0.8	81.9 ± 8.3	76.2 ± 0.4	74.3 ± 7.2	70.3 ± 3.3	73.6 ± 6.3
Z-LogN-U ($\Downarrow$)	80.5 ± 1.9	**84.9** ± 7.9	76.2 ± 0.4	75.0 ± 5.7	71.4 ± 0.7	74.1 ± 6.1
Methods	ID-1510 (569×30)		ID-6332* (540×18)		ID-40994 (540×18)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	**98.1** ± 1.7	**99.6** ± 0.6	**74.8** ± 4.3	**80.6** ± 2.0	**96.3** ± 2.2	**95.5** ± 4.9
LogN-U	96.5 ± 1.9	99.4 ± 0.6	67.8 ± 3.5	73.5 ± 4.1	93.1 ± 1.7	94.7 ± 5.9
LogN-U($\Rightarrow$)	96.3 ± 3.1	99.4 ± 0.9	62.2 ± 6.2	68.2 ± 6.3	93.3 ± 1.9	94.6 ± 4.4
LogN-U($\Downarrow$)	96.3 ± 3.1	99.5 ± 0.6	58.9 ± 6.6	59.2 ± 10.6	94.8 ± 2.5	94.3 ± 6.1
Z-LogN-U	97.5 ± 2.0	**99.6** ± 0.7	68.5 ± 4.3	75.2 ± 2.5	94.3 ± 1.5	95.1 ± 6.0
Z-LogN-U ($\Rightarrow$)	95.6 ± 3.3	99.4 ± 0.8	60.4 ± 3.3	69.8 ± 7.8	92.4 ± 1.5	94.0 ± 4.8
Z-LogN-U ($\Downarrow$)	97.5 ± 2.5	**99.6** ± 0.7	57.0 ± 2.0	53.5 ± 10.2	95.0 ± 1.9	95.0 ± 6.0

Table 5: We performed leave-one-out on datasets from ExperimentHub [14]. ($\Rightarrow$) denotes the sample-wise normalization, ($\Downarrow$) the feature-wise scaling.

Methods	Cirrhosis1 (181×100)		Cirrhosis2 (56×100)		IBD (396×100)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	82.3	**92.7**	80.4	**94.8**	78.3	**87.4**
LogN-U	82.3	91.1	**85.7**	94.5	79.2	86.4
LogN-U ($\Rightarrow$)	83.4	92.6	83.9	94.1	76.0	86.7
LogN-U ($\Downarrow$)	81.8	90.9	82.1	94.2	76.8	85.9
Z-LogN-U	82.9	91.3	**85.7**	94.3	80.1	86.8
Z-LogN-U ($\Rightarrow$)	**84.5**	92.5	83.9	94.1	**80.6**	86.8
Z-LogN-U ($\Downarrow$)	83.4	91.9	**85.7**	94.5	78.8	86.6
Methods	Obesity (263×100)		T2D (344×100)		WT2D (96×100)	
	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC
TabPFN [9]	63.5	56.6	57.6	60.1	68.8	68.3
LogN-U	66.2	**60.9**	**58.4**	**61.6**	67.7	74.7
LogN-U ($\Rightarrow$)	64.3	56.2	56.1	59.0	63.5	66.7
LogN-U ($\Downarrow$)	**66.9**	60.7	54.7	**61.6**	65.6	72.8
Z-LogN-U	63.5	59.1	55.5	60.7	67.7	74.6
Z-LogN-U ($\Rightarrow$)	63.5	58.9	51.5	56.8	64.6	66.1
Z-LogN-U ($\Downarrow$)	63.9	58.8	56.4	59.6	**74**	**75.3**