Tree-based variational inference for Poisson log-normal models

Alexandre Chaussard[⋆], Anna Bonnet[⋆], Élisabeth Gassiat[†], and Sylvain Le Corff[⋆]

[⋆]LPSM, Sorbonne Université.
[†]LMO, Université Paris-Saclay.

Abstract

When studying ecosystems, hierarchical trees are often used to organize entities based on proximity criteria, such as the taxonomy in microbiology, social classes in geography, or product types in retail businesses, offering valuable insights into entity relationships. Despite their significance, current count-data models do not leverage this structured information. In particular, the widely used Poisson log-normal (PLN) model, known for its ability to model interactions between entities from count data, lacks the possibility to incorporate such hierarchical tree structures, limiting its applicability in domains characterized by such complexities. To address this matter, we introduce the PLN-Tree model as an extension of the PLN model, specifically designed for modeling hierarchical count data. By integrating structured variational inference techniques, we propose an adapted training procedure and establish identifiability results, enhancisng both theoretical foundations and practical interpretability. Additionally, we extend our framework to classification tasks as a preprocessing pipeline, showcasing its versatility. Experimental evaluations on synthetic datasets as well as real-world microbiome data (Pasolli et al., 2016) demonstrate the superior performance of the PLN-Tree model in capturing hierarchical dependencies and providing valuable insights into complex data structures, showing the practical interest of knowledge graphs like the taxonomy in ecosystems modeling. Our implementation is freely available at `https://github.com/AlexandreChaussard/PLN-Tree`.

1 Introduction

Count data appear in various domains such as ecology, metagenomics, retail, actuarial sciences, and social sciences. One significant interest in analyzing count data lies in understanding the relationships between entities within a specific environment, which can be framed as a network inference problem. Canonical methods for this involve undirected graphical models, which represent conditional dependencies among entities in an ecosystem, providing interpretable insights into community structures (Lauritzen, 1996; Harris, 2016). For continuous data, Gaussian graphical models (GGM) are widely used across multiple fields, including genomics to explore gene expressions and identify therapeutically relevant genes (Altenbuchinger et al., 2020), and to uncover functional pathways related to diseases (Yu et al., 2015). However, the Gaussian assumption is not suitable for discrete count data, and the commonly used log-transforms are being sidelined due to their lack of statistical groundings compared to modeling approaches O'Hara and Kotze (2010). Numerous statistical models have thus been developed to analyze count data, such as those discussed by

Hilbe (2014); Inouye et al. (2017). Among the graphical models for count data, the Poisson Log-Normal (PLN) model, originally proposed by Aitchison and Ho (1989), has become a standard, particularly after its rediscovery by Chiquet et al. (2021) which led to significant theoretical and methodological developments in interaction network inference for count data.

In practical applications, count data often exhibit hierarchical structures where observations are organized in a tree graph reflecting compositional relationships between entities at different levels of the hierarchy, like the taxonomy in ecology, the social classes in geography, or product types in marketing. In cases where no natural hierarchical structure is established in the domain, or when alternative clustering insights are desired, practitioners employ tree-inference approaches (Côme et al., 2021; Blei et al., 2003; Teh et al., 2004) to organize and describe entities in a comprehensible graph that incorporates domain-specific knowledge. In various applications, hierarchical data structures have been considered to enhance statistical models, resulting in improved performances in most cases (Crawford and Greene, 2020; Oliver et al., 2023). However, adhering strictly to predefined hierarchical structures can sometimes hinder model performance, as shown by Bichat et al. (2020) in the context of controlling the false discovery rate for the detection of differentially abundant microbial bacteria. This suggests the need for flexible modeling approaches that can exploit underlying tree graphs without being overly dependent on their structure. Yet, despite the potential interest of such hierarchical structures for multivariate counts modeling, existing models like PLN do not explicitly account for them, limiting their applicability in scenarios where hierarchical dependencies play a crucial role. To address this limitation, we introduce the PLN-Tree model, an extension of the PLN framework tailored to handle hierarchical count data represented by tree graphs. The PLN-Tree model leverages a top-down hidden Markov tree structure to capture hierarchical dependencies among counts, enabling more accurate and interpretable modeling of count data in hierarchical settings. While the observed counts are controlled by the underlying hierarchical structure in the PLN-Tree framework, the model maintains flexibility through a latent Markov chain to parameterize the counts which is not confined to the tree structure. Like its PLN parent, learning the PLN-Tree model through maximum-likelihood estimation is not tractable, requiring variational inference techniques (Blei et al., 2017). Hence, leveraging the true form of the posterior distribution, we propose a structured variational inference method based on backward Markov chains (Campbell et al., 2021). To ensure model scalability, we opt for a deep learning architectures by parameterizing the distributions with neural networks, allowing for efficient inference of the variational approximation using amortized backward inference (Chagneux et al., 2024). Furthermore, we introduce a residual variant of the amortized backward neural network architecture, which demonstrates superior performance in our experiments.

To ensure the interpretability of the latent variables in practical applications, we investigate the identifiability of the proposed model. Previous works on structured models, such as Gassiat et al. (2020); Hälvä et al. (2021), have demonstrated the ability to uniquely identify latent data models in the presence of Markov dependency structures. Thus, we establish the identifiability of a class of identifiability within the PLN-Tree structured framework, ensuring its applicability in demanding contexts where accurate and interpretable modeling of count data is crucial.

This paper is organized as follows. Section 2 provides background on the PLN framework and structured variational inference techniques motivating our model. Section 3 introduces the proposed PLN-Tree models and variational training procedures. It also displays identifiability results for tree-based PLN models. Finally, Section 4 provides synthetic and real-life applications, comparing the proposed backward variational approximation with the mean-field variant and the original PLN. We namely demonstrate the practical utility of PLN-Tree models through a data augmentation benchmark on real microbiome data from Pasolli et al. (2016), highlighting its effectiveness in capturing hierarchical dependencies, proving the inherent interest of the taxonomy in microbiome modeling. Extending beyond its generative features, we also illustrate the potential of considering PLN-Tree models as preprocessing pipelines for a one-vs-all disease classification task. Our implementation and experiments are freely available on our GitHub[1].

2 Background

2.1 Poisson log-normal models

The Poisson-Log Normal model, introduced by Aitchison and Ho (1989) and thoroughly extended by Chiquet et al. (2021), is a standard network inference model that has become popular due to its ability to handle over-dispersed count data and capture complex dependencies among variables. In its simplest form, for a sample i, the PLN approach models the interactions through a Gaussian latent variable $\mathbf{Z}_i \in \mathbb{R}^d$, with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and precision matrix $\boldsymbol{\Omega} \in \mathbb{R}^{d\times d}$. The observed counts $\mathbf{X}_i \in \mathbb{R}^d$ are modeled by a Poisson distribution such that $(\mathbf{Z}_i, \mathbf{X}_i)_{1\leqslant i\leqslant n}$ are independent and, for $1 \leqslant i \leqslant n$, conditionally on $\mathbf{Z}_i$ and X_{ik}, $1 \leqslant k \neq j \leqslant d$, X_{ij} depends on Z_{ij} only:

$$\begin{aligned} \text{latent space} \quad & \mathbf{Z}_i \sim \mathcal{N}\left(\boldsymbol{\mu}, \boldsymbol{\Omega}^{-1}\right) \ , \\ \text{observed space} \quad & \mathrm{X}_{ij} \mid \mathrm{Z}_{ij} \quad \text{indep.} \quad \mathbf{X}_i \mid \mathbf{Z}_i \sim \mathcal{P}(\exp(\mathbf{Z}_i)) \ . \end{aligned}$$

In the PLN model, the precision matrix $\boldsymbol{\Omega}$ yields the interaction network, as entailed by the faithful correlation property provided in Chiquet et al. (2021). On the other hand, the mean parameter $\boldsymbol{\mu}$ enables to model the fixed effects in the environment, such as the natural disproportion of species in an ecosystem. Individual-related environmental effects can also be accounted for in $\boldsymbol{\mu}$ by making it a function of covariates, or by adding sampling effort information through an offset, which can have a significant impact on the faithfulness of the reconstructed network, as shown numerically in Chiquet et al. (2019).

Performing maximum likelihood estimation in such latent data models is challenging as the conditional distribution of the latent variables given the observations is not tractable. Variational estimation (Blei et al., 2017) is an appealing alternative to computationally intensive Monte Carlo methods by approximating the true posterior using a family of variational distributions, yielding the Evidence Lower Bound (ELBO) as a suboptimal optimization objective (Kingma et al., 2019). Consequently, Chiquet et al. (2021) proposed an inference method for the PLN models based on variational inference called VEM, which consists in maximizing the ELBO in an alternate optimization resembling the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), except that the true posterior is replaced by its variational counterpart. In Chiquet et al. (2019), the variational approximation corresponds to the Gaussian mean-field approximation, where each sample is parameterized by a unique mean and diagonal covariance matrix, unlike usual neural network parameterizations (Kingma et al., 2019). This specific form enables fast inference, as it yields exact maximization steps of the true parameters given the variational parameters, making the inference process highly stable, efficient, and computationally expedient. However, it affects the model scalability to larger datasets as the number of parameters increases linearly with the number of samples.

In Chiquet et al. (2019), the network inference model also comes with a sparsity-informed penalty inspired by the graphical LASSO (Friedman et al., 2008), which introduces a hyperparameter that controls the sparsity of the reconstructed network, which is highly relevant for interpretability. Yet, tuning the penalty is a complex task, as thoroughly explored in Banerjee et al. (2008); Chiquet et al. (2019), and thus will not be

[1] `https://github.com/AlexandreChaussard/PLN-Tree`

studied in our framework. Additionally, various PLN variants have been proposed in Chiquet et al. (2021), such as PLN-PCA, PLN mixtures, and PLN-LDA. Although these variants can be naturally extended to our PLN-Tree framework, we opt not to explore them in this paper.

2.2 Variational inference for structured data

As underscored in the previous section, addressing the parameter inference problem for PLN models requires applying variational inference techniques, which requires to choose a variational family.

In scenarios devoid of specific structural constraints like PLN, the Gaussian mean-field approximation emerges as the prevalent choice for variational families. This approach entails modeling each latent coordinate with independent Gaussian densities, offering the advantage of explicit ELBO computation when the latent prior is Gaussian. The mean-field approximation has demonstrated efficacy across various applications, such as the Poisson Log-Normal network inference model (Chiquet et al., 2019) and in Variational Auto-Encoders (VAE) (Kingma et al., 2019). However, its inherent lack of expressivity and dependency modeling has encouraged the development of alternative variational families, including Gaussian mixture models with VAMPrior (Tomczak and Welling, 2018) and normalizing flows within the latent space to enhance posterior expressiveness (Kobyzev et al., 2020).

In this context, we set the focus to another class of variational approximations that explicitly incorporate data structures. These structured variational approximations can be formulated based on prior assumptions, as seen in approaches like NVAE (Vahdat and Kautz, 2020), or by deriving insights from the true posterior distribution, like auto-regressive models Marino et al. (2018) or hidden Markov models Campbell et al. (2021). While prior-based assumptions are pertinent to methodological advancements, structuring the variational approximation based on the true posterior aligns more closely with statistical principles while encouraging model interpretability (Arrieta et al., 2020). Notably, when the latent process follows a hidden Markov model, an enhanced variational approximation beyond the mean-field approach can be derived, as demonstrated by Johnson et al. (2016), further illustrated and extended in Lin et al. (2018); Hälvä et al. (2021); Schneider et al. (2023). Our work is closely related to advancements in this area, particularly in the context of hidden Markov models, where recent studies like Campbell et al. (2021); Chagneux et al. (2024) have highlighted the utility of backward variational inference, showcasing both empirical improvements and theoretical guarantees. Leveraging amortized inference techniques using recurrent networks, Chagneux et al. (2024) suggests a computationally efficient implementation of a variational approximation that partially captures the backward structure, thus enhancing experimental results over mean-field alternatives. Moreover, the theoretical underpinnings laid out in Chagneux et al. (2024); Gassiat and Le Corff (2024) and Campbell et al. (2021) regarding backward variational inference in Markov chains offer compelling motivations for its application in our specific context.

3 Tree-based variational inference

3.1 PLN-Tree model and parameters inference

Notations Let $\mathcal{T}$ be a finite rooted tree with L layers, where each layer $\ell \leqslant L$ comprises K_ℓ nodes. A branch contains at least one node in each layer, so that every branch has a depth equal to L. At layer $\ell \leqslant L$,

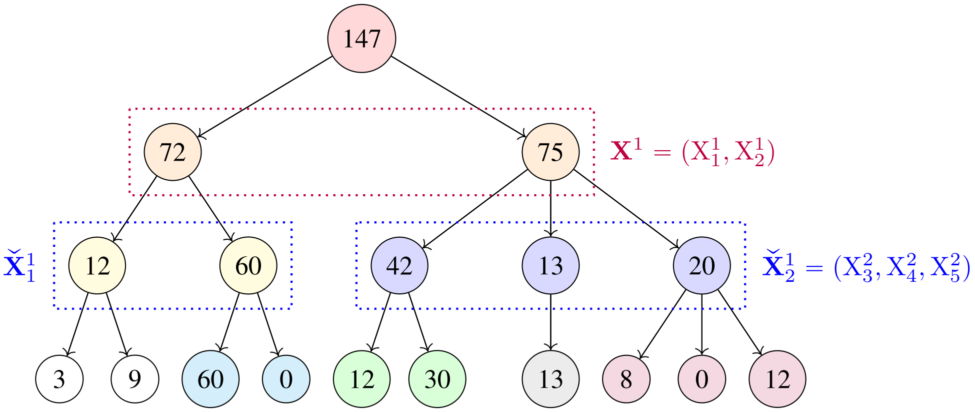

147
72
75
$\mathbf{X}^1 = (X^1_1, X^1_2)$
$\check{\mathbf{X}}^1_1$
12
60
42
13
20
$\check{\mathbf{X}}^1_2 = (X^2_3, X^2_4, X^2_5)$
3
9
60
0
12
30
13
8
0
12

Figure 1: Example of a hierarchical count data with $L = 4$. Nodes of the same color are independent of the other nodes conditionally to their parent node and their respective latent variables.

the random variable associated with node $k \leqslant K_\ell$ is denoted by V^ℓ_k. For layer $\ell \leqslant L-1$ and node $k \leqslant K_\ell$, the vector of children of the random variable V^ℓ_k is indexed by $\mathcal{C}^k_\ell$ and represented as $\check{V}^\ell_k = (V^{\ell+1}_j)_{j \in \mathcal{C}^k_\ell}$. We generally denote the hierarchical counts by $\mathbf{X}$ and the associated latent variables by $\mathbf{Z}$. A graphical representation is provided in Figure 1.

If the distribution of a random variable V has a density parameterized by $\boldsymbol{\theta}$ with respect to a reference measure, it is denoted by $p^{V}_{\boldsymbol{\theta},}$. In cases of clarity, we may express the density as $p_{\boldsymbol{\theta}}(V)$. If $\boldsymbol{\theta}$ is a vector, its k-th coordinate is denoted by $\boldsymbol{\theta}_k$, while for a diagonal matrix $\boldsymbol{\theta}$, the k-th diagonal term is denoted as $\boldsymbol{\theta}_k$. For a function $f_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta}$ and taking values in $\mathbb{R}^d$, the k-th coordinate of any of its outputs is denoted by $f_{\boldsymbol{\theta},k}(.)$. The sequence of random variables $(V^1, \ldots, V^L)$ is represented as $V^{1:L}$. For $V \in \mathbb{R}^d$, the exponential of V is defined as $\exp(V) = (\exp(V_j))_{1 \leqslant j \leqslant d}$, and the multivariate Poisson distribution with parameters $V \in \mathbb{R}^d_{>0}$ is denoted by $\mathcal{P}(V) = \otimes^d_{j=1} \mathcal{P}(V_j)$. We denote by $\mathcal{S}^d$ the simplex of dimension d, then if $V \in \mathcal{S}^d$, we denote the multinomial distribution with total count n and probabilities V by $\mathcal{M}(n, V)$.

Tree compositionality constraint Hierarchical count data can be constructed from a repeated clustering of entities in the community, up to having one group left, from bottom to top. Then, placing the observed counts at the lowest level of the tree and summing them with their respective siblings in an iterative process, we obtain a hierarchical count data sample. Therefore, hierarchical count data are subject to a tree compositionality constraint

$$\forall \ell < L, \forall k \leqslant K_\ell, \quad X^\ell_k = \sum_{j \in \mathcal{C}^k_\ell} X^{\ell+1}_j \ , \tag{1}$$

which needs to be accounted for in the modelization, thereby preventing an independent modeling of the layers. Furthermore, this constraint motivates a top-down propagation dynamic of the counts in the observed space, as a bottom-up would rely solely on the final layer to determine the entire hierarchical count data, thus failing to incorporate the tree structure in the modelization.

Model The PLN framework models tabular count data, which only applies to one layer of the tree at a time. Therefore, learning one PLN model at each layer does not satisfy the tree compositionality constraint since it models independent layers. Consequently, we propose a new model tailored to hierarchical structures named PLN-Tree.

H1
- The $(\mathbf{Z}_i, \mathbf{X}_i)_{1\leqslant i\leqslant n}$ are independent, and for $1 \leqslant \ell \leqslant L{-}1$, conditionally on $\{(\mathbf{Z}_i^u, \mathbf{X}_i^v)\}_{\substack{1\leqslant u\leqslant L\\ 1\leqslant v\neq \ell\leqslant L}}$, the random variables $(\check{\mathbf{X}}_{ik}^{\ell})_{1\leqslant k\leqslant K_\ell}$ are independent and the conditional law of $\check{\mathbf{X}}_{ik}^{\ell}$ depends only on $\check{\mathbf{Z}}_{ik}^{\ell}$ and X_{ik}^{ℓ}.
- The latent process $(\mathbf{Z}^\ell)_{1\leqslant \ell\leqslant L}$ is a Markov chain with initial distribution $\mathbf{Z}^1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and such that for all $1 \leqslant \ell \leqslant L-1$, the conditional distribution of $\mathbf{Z}^{\ell+1}$ given $\mathbf{Z}^\ell$ is Gaussian with mean $\boldsymbol{\mu}_{\boldsymbol{\theta}_{\ell+1}}(\mathbf{Z}^\ell)$ and variance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{\ell+1}}(\mathbf{Z}^\ell)$. Formally, the latent process writes

$$\begin{aligned} &\mathbf{Z}^1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\ , \\ \forall \ell < L, \quad &\mathbf{Z}^{\ell+1} \mid \mathbf{Z}^\ell \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\theta}_{\ell+1}}(\mathbf{Z}^\ell), \boldsymbol{\Sigma}_{\boldsymbol{\theta}_{\ell+1}}(\mathbf{Z}^\ell)\right)\ . \end{aligned}$$

- Conditionally on $\mathbf{Z}^1$, $\mathbf{X}^1 \sim \mathcal{P}(\mathrm{e}^{\mathbf{Z}^1})$ and for all $1 \leqslant \ell \leqslant L-1$, $1 \leqslant k \leqslant K_\ell$, conditionally on X_k^ℓ and $\check{\mathbf{Z}}_k^\ell$, $\check{\mathbf{X}}_k^\ell$ has a multinomial distribution with parameters $\sigma(\check{\mathbf{Z}}_k^\ell)$ and X_k^ℓ, where $\sigma(.)$ denotes the softmax function. Formally, the observed counts process writes

$$\begin{aligned} &\mathbf{X}^1 \mid \mathbf{Z}^1 \sim \mathcal{P}(\mathrm{e}^{\mathbf{Z}^1})\ , \\ \forall \ell < L, k \leqslant K_\ell, \quad &\check{\mathbf{X}}_k^\ell \mid \mathrm{X}_k^\ell, \check{\mathbf{Z}}_k^\ell \sim \mathcal{M}\left(\mathrm{X}_k^\ell, \sigma(\check{\mathbf{Z}}_k^\ell)\right)\ . \end{aligned}$$

The latent dynamic incorporates the tree structure through its Markov chain property while remaining flexible enough to model the interactions between all the nodes of a given layer, not just the siblings. Conversely, the observed counts are constrained to satisfy the tree compositionality constraint (1). In particular, the multinomial conditional distribution of the observations $\check{\mathbf{X}}_k^\ell$ for $1 \leqslant \ell \leqslant L-1$ is the conditional distribution of independent Poisson random variables with parameters $\exp(\check{\mathbf{Z}}_k^\ell)$ conditioned on the event $\{\sum_{j\in\mathcal{C}_k^\ell} \mathrm{X}_j^{\ell+1} = \mathrm{X}_k^\ell\}$.

Variational inference Under H1, the posterior distribution is a backward Markov chain. Since we approximate this quantity using a variational approximation, we suggest variational families that account for the backward structure of the true conditional distribution of the latent variables given the observations.

H2 The variational density is given by a backward Gaussian Markov Chain:

$$q_{\boldsymbol{\varphi},1:L}(\mathbf{Z}|\mathbf{X}) = q_{\boldsymbol{\varphi},L}(\mathbf{Z}^L|\mathbf{X}^{1:L}) \prod_{\ell=1}^{L-1} q_{\boldsymbol{\varphi},\ell|\ell+1}(\mathbf{Z}^\ell|\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})\ ,$$

where $q_{\boldsymbol{\varphi},L}(\cdot|\mathbf{X}^{1:L})$ is the Gaussian density with mean $\mathbf{m}_{\boldsymbol{\varphi}^L}(\mathbf{X}^{1:L})$ and variance $\mathbf{S}_{\boldsymbol{\varphi}^L}(\mathbf{X}^{1:L})$ and $q_{\boldsymbol{\varphi},\ell|\ell+1}(\cdot|\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})$ is the Gaussian density with mean $\mathbf{m}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})$ and variance $\mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})$.

The ELBO of the backward variational model can then be computed as proposed below.

Proposition 3.1. *Assume that H1 and H2 hold. Then, the ELBO of the PLN-Tree model writes*

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \sum_{\ell=1}^{L} \frac{1}{2} \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}} \left[\log |\boldsymbol{\Omega}_{\boldsymbol{\theta}^\ell}(\mathbf{Z}^{\ell-1})| - \operatorname{tr}(\widehat{\boldsymbol{\Sigma}}_\ell \boldsymbol{\Omega}_{\theta^\ell}(\mathbf{Z}^{\ell-1})) + \log |\mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})| \right] \\
+ \sum_{k=1}^{K_\ell} \left(\mathrm{X}_k^\ell \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}} \left[\mathbf{m}_{\boldsymbol{\varphi}^\ell,k}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell}) \right] - \mathbb{1}_{\ell=1} \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}} \left[\mathrm{M}^k_{\ell|\ell+1}(\mathbf{Z}^{\ell+1}) \right] \right) \\
- \mathbb{1}_{\ell>1} \sum_{k=1}^{K_{\ell-1}} \mathrm{X}_k^{\ell-1} \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}} \left[\log \sum_{j \in \mathcal{C}_k^{\ell-1}} \mathrm{e}^{Z_j^\ell} \right] - \mathbb{1}_{\ell=L} \sum_{k=1}^{K_\ell} \log \mathrm{X}_k^\ell ! - \frac{1}{2} K_\ell \ ,
\end{aligned}
$$

such that $\boldsymbol{\Omega}_{\boldsymbol{\theta}^1}(\mathbf{Z}^0) = \boldsymbol{\Omega}_1$, $\boldsymbol{\mu}_{\boldsymbol{\theta}^1}(\mathbf{Z}^0) = \boldsymbol{\mu}_1$, $\mathbf{S}_{\boldsymbol{\varphi}^L}(\mathbf{Z}^{L+1}, \mathbf{X}^{1:L}) = \mathbf{S}_{\boldsymbol{\varphi}^L}(\mathbf{X}^{1:L})$, $\mathbf{m}_{\boldsymbol{\varphi}^L}(\mathbf{Z}^{L+1}, \mathbf{X}^{1:L}) = \mathbf{m}_{\boldsymbol{\varphi}^L}(\mathbf{X}^{1:L})$*, and for all* $1 \leqslant \ell \leqslant L-1, 1 \leqslant k \leqslant K_\ell$ *,*

$$
\begin{aligned}
\widehat{\boldsymbol{\Sigma}}_\ell = \left(\boldsymbol{\mu}_{\boldsymbol{\theta}_\ell}(\mathbf{Z}^{\ell-1}) - \mathbf{m}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell}) \right) \left(\boldsymbol{\mu}_{\boldsymbol{\theta}_\ell}(\mathbf{Z}^{\ell-1}) - \mathbf{m}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell}) \right)^\top \\
+ \mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell}) \ ,
\end{aligned} \tag{2}
$$

$$
\mathrm{M}^k_{\ell|\ell+1}(\mathbf{Z}^{\ell+1}) = \exp \left(\frac{\mathbf{S}_{\boldsymbol{\varphi}^\ell,k}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})}{2} + \mathbf{m}_{\boldsymbol{\varphi}^\ell,k}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell}) \right) \ .
$$

Proof. The proof is postponed to Appendix B.1. □

Numerically, handling the inputs of the neural networks parameterizing the variational distributions is a challenging task due to the increasing dimension of the chains $(\mathbf{X}^{1:\ell})_{1 \leqslant \ell \leqslant L}$, and the value it takes relatively to the latent variables. To address this scalability issue, Chagneux et al. (2024) suggests performing amortized inference by encoding the chain of counts using a recurrent neural network. It enables us to control the number of parameters while neutralizing the increasing dimension of the input. Moreover, considering the current observation's pivotal influence on the latent variable distribution at layer ℓ, we introduce a residual connection yielding $\mathbf{X}^\ell$ as input of the current variational parameters. Combined with the amortized setting, this approach yields the residual amortized backward architecture illustrated in figure 2. Problem-specific networks must then be tuned, as thoroughly explored in our experiments in Section 4. While we focus on the residual amortized backward for its superior empirical performances in our experiment, other noteworthy methods could be employed for the variational parameters in certain cases, like the regular amortized backward, or a weak amortized variant taking only the current observation as input and the next latent.

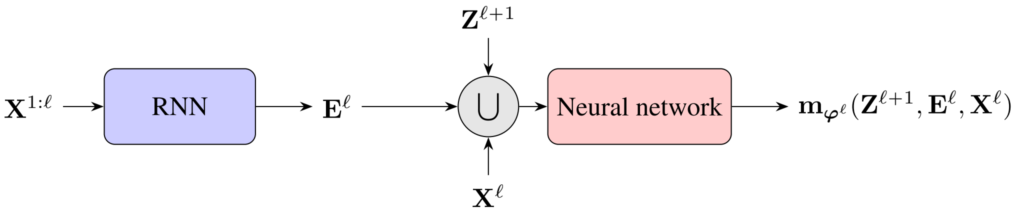


Figure 2: Residual amortized backward architecture for the variational mean at layer $\ell \leqslant L$.

Offset modeling Collecting count data within multiple ecosystems usually comes with a variable sampling effort in practice. This offset in the average total count often originates from the counting protocols in each environment or the difficulty of exploring an environment. In genomics for instance, the total count relates to the sequencing depth of the genome, which correlates with the counts of rarer species, introducing a bias in the data with higher total count (Lee et al., 2014; Xu et al., 2017). Consequently, the total count conveys information about the data capture protocol rather than the environments studied, and therefore it should not become a decisive feature in that context. Procedures like resampling, also called rarefaction, allow to overcome most of the offset effects while preprocessing the data (Weinroth et al., 2022; Schloss, 2024), at the cost of partial data loss. When using the raw count data, it is then recommended to model the offset in order to avoid spurious correlations (Chiquet et al., 2019). In the PLN model from Chiquet et al. (2021), the offset is modeled through a plug-in estimator from the abundances that shifts the latent variables mean by $\mathrm{O}_i = \log \sum_{k=1}^{K} \mathrm{X}_{ik}$. For the PLN-Tree framework, we propose to model the offset using a latent variable O satisfying H3.

H3

- The $(\mathrm{O}_i, \mathbf{Z}_i, \mathbf{X}_i)_{1 \leqslant i \leqslant n}$ are i.i.d., and for $\ell \leqslant L-1$, conditionally on $\{(\mathrm{O}, \mathbf{Z}^u, \mathbf{X}^v)\}_{\substack{1 \leqslant u \leqslant L \\ 1 \leqslant v \neq \ell \leqslant L}}$, the random variables $(\check{\mathbf{X}}_k^\ell)_{1 \leqslant k \leqslant K_\ell}$ are independent and the conditional law of $\check{\mathbf{X}}_k^\ell$ depends only on $\check{\mathbf{Z}}_k^\ell$ and and X_k^ℓ.
- The distribution of the offset O is a Gaussian mixture, and conditionally on $\mathbf{X}$, the offset O and the latent variables $\mathbf{Z}$ are independent.
- The latent process $(\mathbf{Z}^\ell)_{1 \leqslant \ell \leqslant L}$ is a Markov chain with initial distribution $\mathbf{Z}^1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and such that for all $1 \leqslant \ell \leqslant L-1$, the conditional distribution of $\mathbf{Z}^{\ell+1}$ given $\mathbf{Z}^\ell$ is Gaussian with mean $\boldsymbol{\mu}_{\boldsymbol{\theta}_{\ell+1}}(\mathbf{Z}^\ell)$ and variance $\boldsymbol{\Sigma}_{\boldsymbol{\theta}_{\ell+1}}(\mathbf{Z}^\ell)$.
- Conditionally on $\mathbf{Z}^1$, $\mathbf{X}^1$ has a Poisson distribution with parameter $\exp(\mathbf{Z}^1 + \mathrm{O})$ and for all $1 \leqslant \ell \leqslant L-1$, $1 \leqslant k \leqslant K_\ell$, conditionally on X_k^ℓ and $\check{\mathbf{Z}}_k^\ell$, $\check{\mathbf{X}}_k^\ell$ has a multinomial distribution with parameters $\sigma(\check{\mathbf{Z}}_k^\ell)$ and X_k^ℓ.

Since the softmax is invariant by constant translation, adding the offset in the lower layers of the observed dynamics has no impact on the modelization. By inheritance of the PLN-Tree model, the true posterior is still intractable. It is then approximated using a variational approximation that satisfies the following assumptions.

H4

- Inheriting the property of the true posterior, under the variational approximation, O and $\mathbf{Z}$ are independent conditionally to $\mathbf{X}$.
- The variational approximation $q_{\boldsymbol{\varphi}}^{\mathrm{O}}(\mathrm{O}|\mathbf{X})$ is a Gaussian with mean $\mathrm{m}_o(\mathbf{X})$ and variance $\mathrm{s}_o^2(\mathbf{X})$.
- The latent posterior $q_{\boldsymbol{\varphi},1:L}^{\mathbf{Z}}(\mathbf{Z}|\mathbf{X})$ is a backward Markov chain as defined in H2.

The ELBO of the offset-modeled PLN-Tree is derived directly from the PLN-Tree model.

Lemma 3.2. *Assume that H3 and H4 hold. Denote by $\mathcal{L}_{|\mathrm{O}}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ the ELBO of the generative model from proposition 3.1 with shifted latent means $\boldsymbol{\mu} + \mathrm{O}$ and $\mathbf{m}_{\boldsymbol{\varphi}}(.) + \mathrm{O}$, then the ELBO of the offset-modeled PLN-Tree is given by*

$$\mathcal{L}_{\mathrm{offset}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathcal{L}_{|\mathrm{O}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) + 2\mathbb{E}_{q_{\boldsymbol{\varphi}}^{\mathrm{O}}}\left[\log p_{\boldsymbol{\theta}}(\mathrm{O})\right] + \frac{1}{2} \log s_o^2(\mathbf{X}) + \frac{1 + \log 2\pi}{2} \,.$$

Proof. The proof is postponed to Appendix B.2. □

Learning the PLN-Tree model can be accelerated by exploiting the variational EM algorithm from Chiquet et al. (2021) applied at the first layer, which holds an explicit optimum in $\boldsymbol{\theta}$ when $\boldsymbol{\varphi}$ is known, so that at iteration $h + 1$

$$\boldsymbol{\mu}_1^{(h+1)} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{m}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{X}_i^{1:L}) \; ,$$

$$\boldsymbol{\Omega}_1^{(h+1)} = \left(\frac{1}{n}\sum_{i=1}^{n} \left(\boldsymbol{\mu}_1^{(h+1)} - \mathbf{m}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{X}_i^{1:L})\right)\left(\boldsymbol{\mu}_1^{(h+1)} - \mathbf{m}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{X}_i^{1:L})\right)^{\top} + \mathbf{S}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{X}_i^{1:L})\right)^{-1} .$$

3.2 Identifiability of Poisson-Log Normal models

In a nutshell, identifiability ensures we can uniquely determine a model given the data, and thus infer the law of the latent variables solely from the law of the observations. In real-world applications, it was shown that the lack of identifiability can severely undermine performances (D'Amour et al., 2022), and precludes the interpretability of the inferred networks. Fortunately, in many applications such as in Hälvä et al. (2021); Gassiat et al. (2020), the dependency structure of the data can disentangle parameters using inductive biases. This section presents two identifiability results related to the PLN model and the PLN-Tree extension.

Lemma 3.3 shows the identifiability of the PLN models and the identifiability of the first layer of the PLN-Tree model, which is illustrated in Section 4.1.

Lemma 3.3. *Let $\mathbf{Z} = (\mathrm{Z}^\ell)_{1\leqslant\ell\leqslant L}$ be a random variable supported on $(\mathbb{R}_+^*)^L$. Consider the observations $\mathbf{X} = (\mathrm{X}^\ell)_{1\leqslant\ell\leqslant L}$ such that for all $1 \leqslant \ell \leqslant L$, the conditional distribution of X^ℓ given Z is $\mathrm{X}^\ell \sim \mathcal{P}(\mathrm{Z}^\ell)$. Then, the law of $\mathbf{Z}$ is identifiable from the law of $\mathbf{X}$.*

Proof. Proof is postponed to Appendix C.2.1 □

PLN-Tree identifiability The previous result does not cover the whole scope of the PLN-Tree framework as it models independent layers conditionally to their respective latent variables. Instead, Theorem 3.4 establishes the identifiability of the PLN-Tree model up to a softmax transform, which is illustrated in Section 4.1.

Theorem 3.4. *Let $\mathcal{T}$ a given tree, $\mathbf{Z} = (\mathrm{Z}^1, \mathbf{Z}^2, \mathbf{Z}^3)$ be random variables such that $\mathrm{Z}^1 > 0$, $\mathbf{Z}^2 \in \mathcal{S}^{K_2}$, for all $k \leqslant K_2, \check{\mathbf{Z}}_k^2 \in \mathcal{S}^{\#\mathcal{C}_k^2}$. Suppose the observations $\mathbf{X} = (\mathrm{X}^1, \mathbf{X}^2, \mathbf{X}^3)$ are such that:*

- *conditionally on Z^1, X^1 has a Poisson distribution with parameter Z^1;*
- *conditionally on $(\mathrm{X}^1, \mathbf{Z}^2)$, $\mathbf{X}^2 \sim \mathcal{M}\left(\mathrm{X}^1, \mathbf{Z}^2\right)$;*
- *conditionally on $(\mathbf{X}^2, \mathbf{Z}^3)$, for all $1 \leqslant k \leqslant K_2$, $\check{\mathbf{X}}_k^2 \sim \mathcal{M}\left(\mathrm{X}_k^2, \check{\mathbf{Z}}_k^2\right)$, and $\check{\mathbf{X}}_k^2$ is independent of $(\check{\mathbf{X}}_j^2)_{j\neq k}$.*

Then, the law of $(\mathrm{Z}^1, \mathbf{Z}^2, \mathbf{Z}^3)$ is identifiable from the law of $(\mathrm{X}^1, \mathbf{X}^2, \mathbf{X}^3)$.

Proof. Proof is postponed to Appendix C.2.5. □

However, since the softmax function is constant along diagonals, obtaining the identifiability of $(\mathbf{Z}^1, \dots, \mathbf{Z}^L)$ is not a given if we do not set a constraint on the parameters space. Combining the previous result with Lemma C.4 shows we can identify the law of the latent variables up to a linear projection. Assuming the distribution of the latent variables is Gaussian, a direct application of the previous result yields the identifiability of every parent-children distribution of the PLN-Tree framework providing the parameters belong to a defined projection space.

Corollary 3.5. *Let $(\mathbf{Z}^1, \mathbf{Z}^2)$ and $(\tilde{\mathbf{Z}}^1, \tilde{\mathbf{Z}}^2)$ in $\mathbb{R}^m \times \mathbb{R}^d$ be such that conditionally on $\mathbf{Z}^1$ (resp. $\tilde{\mathbf{Z}}^1$), $\mathbf{Z}^2$ is Gaussian with mean $\boldsymbol{\mu}(\mathbf{Z}^1)$ (resp. $\tilde{\boldsymbol{\mu}}(\tilde{\mathbf{Z}}^1)$) and covariance $\boldsymbol{\Sigma}(\mathbf{Z}^1)$ (resp. $\tilde{\boldsymbol{\Sigma}}(\tilde{\mathbf{Z}}^1)$). Define $\mathbf{P} = \mathbf{I}_d - \frac{1}{d}\mathbf{1}_{d\times d}$ the projector on $\mathrm{Vect}\,(\mathbf{1}_d)^\perp$. Assume $(\mathbf{Z}^1, \sigma(\mathbf{Z}^2))$ has the same law as $(\tilde{\mathbf{Z}}^1, \sigma(\tilde{\mathbf{Z}}^2))$, then*

$$\mathbf{P}\boldsymbol{\mu}(\mathbf{z}) = \mathbf{P}\tilde{\boldsymbol{\mu}}(\mathbf{z}) \quad \text{and} \quad \mathbf{P}\boldsymbol{\Sigma}(\mathbf{z})\mathbf{P} = \mathbf{P}\tilde{\boldsymbol{\Sigma}}(\mathbf{z})\mathbf{P}\ , \quad \mathbb{P}_{\mathbf{Z}^1} - a.s.\ ,$$

where $\mathbb{P}_{\mathbf{Z}^1}$ is the law of $\mathbf{Z}^1$.

Proof. Proof is postponed to Appendix C.2.4. □

For all $\ell \geqslant 2$, denoting by $\mathbf{P}^\ell = \mathrm{diag}(\{\mathbf{P}^\ell_k\}_{1\leqslant k\leqslant K_{\ell-1}})$ with

$$\mathbf{P}^\ell_k = \mathbf{I}_{\#\mathcal{C}^{\ell-1}_k} - \frac{1}{\#\mathcal{C}^{\ell-1}_k}\mathbf{1}_{\#\mathcal{C}^{\ell-1}_k \times \#\mathcal{C}^{\ell-1}_k}\ ,$$

we obtain from Theorem C.2.5 and Corollary 3.5 that all PLN-Tree model parameterized by the latent variables $(\mathbf{Z}^1, \mathbf{P}^2\mathbf{Z}^2, \dots, \mathbf{P}^L\mathbf{Z}^L)$ are identifiable. This result is also illustrated in the experiments of Section 4.1.

4 Experiments

The goal of this section is to show the practical interest of considering the underlying tree graph structure behind hierarchical count data over unstructured approaches. In a first place, we consider two generative benchmarks on artificial datasets. The first synthetic dataset is generated along a PLN-Tree model and showcases the identifiability of the model, as well as the variational approximation performances and its limits in an ideal inference framework. Then, we generate hierarchical count data using a Markovian Dirichlet procedure as an extension of the simulation protocol proposed in Chiquet et al. (2019). This second experiment enables us to benchmark PLN-Tree against PLN in a fair yet controlled setup. Finally, we assess the model performance in comparison with PLN on real-life metagenomics data from microbiome samples of several disease-affected patients (Pasolli et al., 2016) from a data augmentation perspective, as well as a preprocessing for classification tasks.

Benchmarked models To assess the performance of the PLN-Tree model as a generative model, we would like to benchmark it against the PLN approach on data augmentation problems. However, the PLN framework is restricted to tabular data, allowing the modeling of only one layer of hierarchical count data at a time, thus yielding inconsistent samples regarding the hierarchical compositional constraint (1). Thankfully, by leveraging this constraint, PLN models can generate valid hierarchical count data by sampling solely the final layer of the tree. This generative procedure, denoted as PLN (fill), involves sampling the abundances of the last layer and then exploiting the compositional constraint to derive the values of the parent nodes, allowing us to obtain hierarchical count data that satisfies (1). For comprehensiveness, we still compute the regular PLN model at each layer of the hierarchical tree as a reference baseline, but we separate it from the other models for its consistency issues. In our experiments, the PLN baselines are computed using the *pyPLNmodels*[2] Python implementation from Chiquet et al. (2021). Finally, in this benchmark, we compare the efficiency of the proposed backward approximation H2 against the regular Gaussian mean-field (Blei et al., 2017), denoted as PLN-Tree (MF). Thus, the PLN-Tree tag is retained for backward variational approximation modeling.

Metrics for model evaluation In the context of variational deep generative models, comparing the quality of estimated parameters is often impractical due to variations in model architectures, which adds up to identifiability concerns in neural networks. Instead, we assess the generative performance of trained models by their ability to faithfully replicate the dataset distribution. To achieve this in our context, we use alpha diversity metrics, commonly employed in ecosystem studies (see Appendix A), which provide insights into species richness and evenness, thereby characterizing the diversity within an ecosystem (Gotelli and Colwell, 2001). Among these metrics, the Shannon entropy (A.1) and the Simpson index (A.2) indices are widely employed . The Shannon index quantifies the uncertainty in predicting the entities in the ecosystem (degree of surprise), while the Simpson index represents the probability that two entities chosen at random represent the same entity. Both estimators are qualified as robust, but they quantify complementary aspects of the ecosystems (Nagendra, 2002). Our objective is to ensure that the generated data closely approximates the alpha diversity distribution of the original dataset, as measured by the Wasserstein distance. Other distances or divergences are considered in the appendix for each experiment, such as the Kullback-Leibler divergence, Kolmogorov-Smirnov statistic, and total variation distance. Additionally, to compare the distribution of the generated data with the original data, we evaluate the empirical Wasserstein distance between generated samples and the initial dataset in normalized forms at each layer using the `emd2` function from *POT* (Flamary et al., 2021). Finally, we employ correlation measures between the original data and their reconstructions to assess the quality of the variational approximations at the reconstruction task.

Selection of the variational architectures To provide a comprehensive and equitable evaluation of the PLN-Tree variants, we determine efficient architectures for the variational approximations tailored to each experimental scenario. To that end, we propose several network architectures and assess their generative capabilities, leveraging the above evaluation metrics. The model demonstrating superior overall performance is identified by averaging its rank across all computed metrics. The considered architectures and numerical considerations are detailed in Appendix D. Since the models are trained using variational approximations, convergence may result in different model parameters depending on the initialization. Specifically, the analysis of training variability in Appendix D.1.2 reveals that the mean-field approximation is less stable compared to the proposed residual backward approach, but this does not affect the performance ranking of the two methods. Consequently, training is conducted once for each model, and performance variability is assessed based on the generations.

[2] `https://github.com/PLN-team/pyPLNmodels`

4.1 Synthetic data

4.1.1 PLN-Tree retrieval

To evaluate the efficiency of the proposed backward variational approximation H2 and demonstrate the identifiability results discussed in Section 3.2, we conduct an initial study on data generated from a PLN-Tree model. We begin by defining a tree $\mathcal{T}$ (see Fig. 7), a reference PLN-Tree model with parameters $\boldsymbol{\theta}^*$, and a synthetic dataset $(\mathbf{X}, \mathbf{Z})$ generated using the dynamic specified in H1 with $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ (see Fig. 8), consisting of $n = 2000$ samples. In our experiments, we ensure that the latent dynamic is parameterized by identifiable parameters as detailed in Section 3.2. Upon selecting candidate architectures (see Appendix D.1), we conduct the training procedure for each model until convergence. Then, we generate data by sampling $M = 25$ times 2000 samples from the trained models and aggregate the results to address sampling variability.

Model performance analysis We start our evaluation by analyzing the performance on the synthetic dataset using alpha diversity metrics, summarized in Table 1 using Wasserstein distance (see other statistics in Table 13). As anticipated, the PLN-Tree models exhibit superior performance compared to the PLN variants, with the backward variational approximation outperforming the mean-field variant despite being in an amortized setting. Upon delving into the layers of the tree, we observe a gradual decrease in performance across all criteria in the PLN models, attributable to the Markov tree propagation of the counts, a factor not accounted for by the PLN approach. Additionally, Table 2 demonstrates that PLN-Tree-based approaches consistently approximate the distribution of the counts at each depth of the tree, contrasting with the PLN approaches, which exhibits a progressive performance decline as we descend the tree matching with the alpha diversities observations. Looking at the encoders performance in Table 3, it appears the backward approximation conserves more information than the mean-field approach in an ideal PLN-Tree framework on unseen samples, illustrating the upside of considering the backward Markov structure of the true posterior for model inference.

Alpha diversity	**PLN-Tree**	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
Wasserstein Distance ($\times 10^2$)				
Shannon $\ell = 1$	**1.57** (0.50)	11.23 (0.73)	14.64 (1.15)	<u>1.40</u> (0.61)
Shannon $\ell = 2$	**3.67** (1.33)	5.14 (1.20)	32.04 (1.62)	26.28 (1.31)
Shannon $\ell = 3$	**5.82** (1.51)	7.86 (1.47)	35.03 (1.68)	34.03 (1.68)
Simpson $\ell = 1$	**0.62** (0.21)	2.69 (0.27)	4.91 (0.41)	<u>0.53</u> (0.24)
Simpson $\ell = 2$	**0.71** (0.24)	1.40 (0.31)	7.35 (0.41)	6.05 (0.36)
Simpson $\ell = 3$	**0.85** (0.24)	1.55 (0.34)	7.21 (0.41)	7.21 (0.41)

Table 1: Wasserstein distance between alpha diversities distributions from synthetic data sampled under the original PLN-Tree model and simulated data under each model trained, averaged over the samplings, with standard deviation. Since PLN does not verify the tree compositionality constraint, it is placed aside as a reference. The best-performing model in each row is indicated in bold, and PLN is underlined when it outperforms others.

PLN-Tree identifiability We conduct Principal Component Analysis (PCA) (Hotelling, 1933) on the true latent variables and the latent variables of the trained models at each layer, as depicted in Figure 3. When the inferred counts closely approximate the true counts at a given layer, we observe congruence in the distributions of latent variables across layers, as evident for $\ell = 1$ and $\ell = 2$ in Figure 3, illustrating our identifiability results of Section 3.2. However, for $\ell = 3$, the model fails to capture sufficient information from the true count distribution, resulting in disparate latent distributions. This discrepancy may be attributed to limitations in the proposed variational inference framework.

4.1.2 Artificial data from Markovian Dirichlet

In order to provide fair comparisons of the performances of each model in a controlled setup, we simulate hierarchical count data from a process unrelated to PLN framework, extended from the synthetic experiments protocol of Chiquet et al. (2019). First, we define a hierarchical tree $\mathcal{T}$ that fixes the dataset structure. Then, the steps of the generative process are defined as follows.

- **Base network generation.** Sample an adjacency matrix $\boldsymbol{G} \in \mathcal{M}_{K_1 \times K_1}$ using a random graph model like Erdos-Rényi (no particular structure), preferential attachment (scale-free property) or affiliation models (community structure). Choose $u, v > 0$ to control the partial correlation and conditioning of the network at the first layer, and deduce a precision matrix $\boldsymbol{\Omega} = v\boldsymbol{G} + \mathrm{diag}(|\min(\mathrm{eig}(v\boldsymbol{G}))| + u)$. In our experiments, $v = 0.3$ and $u = 0.1$.
- **First counts generations.** Draw counts $\mathbf{a} \in \mathbb{N}^{K_1}$ such that $\log(\mathbf{a}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Omega})$. Compute a probability vector $\boldsymbol{\pi} = \sigma(\mathbf{a})$ and draw a sampling effort $\mathrm{N} = \exp(\mathrm{O})$ from a negative binomial distribution. We obtain the counts of the first layer using a multinomial distribution $\mathbf{X}^1 \sim \mathcal{M}(\mathrm{N}, \boldsymbol{\pi})$.
- **Counts propagation.** For each $k \leqslant K_1$, compute $\boldsymbol{\alpha}_k^1(\mathbf{X}^1) \in \mathbb{R}_{>0}^{\#\mathcal{C}_k^1}$, where $\boldsymbol{\alpha}_k^1(.)$ is an arbitrary function, like a neural network with softplus output in our experiments. Sample weights $\omega_k^1 \in \mathcal{S}^{\#\mathcal{C}_k^1}$ from a Dirichlet of parameters $\boldsymbol{\alpha}_k^1(\mathbf{X}^1)$. Draw the counts of the children of the node k using a multinomial with total count X_k^1 and probabilities ω_k^1. Repeat that procedure for the next layers using the counts of the previous layer.

In Figure 9, we present the chosen tree graph for our experiments. To derive the covariance matrix of the

	PLN-Tree	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
	Wasserstein Distance $(\times 10^2)$			
$\ell = 1$	**5.20** (0.62)	8.61 (0.11)	10.70 (0.34)	<u>4.71</u> (0.15)
$\ell = 2$	**13.01** (0.14)	16.37 (0.29)	17.59 (0.28)	16.42 (0.27)
$\ell = 3$	**14.08** (0.13)	18.13 (0.32)	20.04 (0.03)	20.04 (0.03)

Table 2: Empirical Wasserstein distance between normalized synthetic data sampled under the original PLN-Tree model and normalized simulated data under each modeled trained, for each layer, averaged over the trainings, with standard deviation. Since PLN does not verify the tree compositionality constraint, it is placed aside as a reference. The best-performing model in each row is indicated in bold, and PLN is underlined when it outperforms others.

	PLN-Tree	PLN-Tree (MF)
$\ell = 1$	**0.999** (0.002)	0.901 (0.209)
$\ell = 2$	**0.993** (0.050)	0.910 (0.137)
$\ell = 3$	**0.996** (0.020)	0.990 (0.028)

Table 3: Correlation between reconstructed counts and the test dataset (1000 samples) from the original PLN-Tree model, averaged over the samples, with standard deviation.

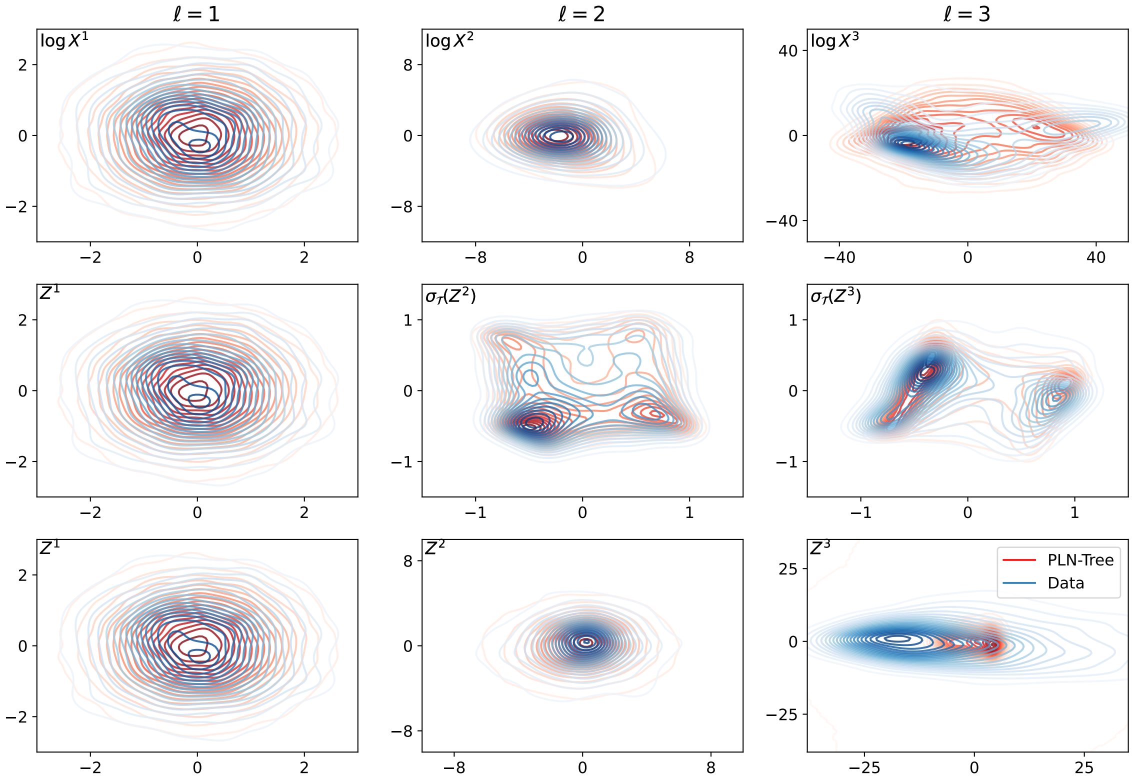


Figure 3: PCA at each depth of the tree, using the training data and the generated data from the PLN-Tree model. The first row corresponds to the projection of the layer count data in log scale, the second row corresponds to the projection of the latent variables as the multinomial parameters (softmax per group of children, denoted by $\sigma_{\mathcal{T}}$), the third row corresponds to the projection of the raw latent variables.

Alpha diversity	PLN-Tree	PLN-Tree (MF)	PLN (fill)	PLN
		Wasserstein Distance $(\times 10^2)$		
Shannon $\ell = 1$	**17.70** (0.47)	21.42 (0.59)	72.27 (1.70)	17.53 (0.59)
Shannon $\ell = 2$	**22.23** (0.94)	29.10 (1.06)	111.53 (1.81)	92.89 (1.92)
Shannon $\ell = 3$	**24.32** (0.83)	37.72 (1.14)	142.28 (1.99)	142.28 (1.99)
Simpson $\ell = 1$	**5.69** (0.16)	5.84 (0.16)	21.74 (0.60)	5.65 (0.22)
Simpson $\ell = 2$	**5.21** (0.17)	5.90 (0.19)	26.70 (0.59)	20.76 (0.64)
Simpson $\ell = 3$	**3.91** (0.11)	5.16 (0.16)	28.55 (0.59)	28.55 (0.59)

Table 4: Wasserstein distance on the distribution of alpha diversities at each layer computed between synthetic data sampled under the Markov Dirichlet model and simulated data under each modeled trained, averaged over the trainings, with standard deviation. Since PLN does not verify the tree compositionality constraint, it is placed aside as a reference. The best-performing model in each row is indicated in bold, and PLN is underlined when it outperforms others.

	PLN-Tree	PLN-Tree (MF)	PLN (fill)	PLN
		Wasserstein distance $(\times 10^2)$		
$\ell = 1$	**11.51** (0.25)	12.47 (0.30)	25.50 (0.59)	10.93 (0.19)
$\ell = 2$	**19.68** (0.25)	22.02 (0.36)	43.26 (0.61)	45.56 (0.90)
$\ell = 3$	**24.33** (0.24)	27.15 (0.30)	51.84 (0.57)	51.84 (0.57)

Table 5: Empirical Wasserstein distance between normalized synthetic data sampled under the Markov Dirichlet model and normalized simulated data under each modeled trained, for each layer, averaged over the trainings, with standard deviation. Since PLN does not verify the tree compositionality constraint, it is placed aside as a reference. The best-performing model in each row is indicated in bold, and PLN is underlined when it outperforms others.

first layer, we generate a random adjacency matrix using the Erdos-Rényi graph model. In our architecture, for all layers ℓ up to L and nodes k up to K_ℓ, α_k^ℓ is structured as a one-layer network with softplus output and a random weight matrix. We set the sampling effort to $N = 20000$, and we sample $n = 2000$ hierarchical count data points, constituting our synthetic dataset. Following the selection of candidate architectures (detailed in Appendix D.2), we conduct a single training procedure for each model. Subsequently, we sample data from the trained models $M = 10$ times and aggregate the results to address sampling variability.

We provide a summary of the model performances in Table 4, Table 5, (see Table 15 for other distances), and Table 6. Notably, the PLN-Tree models exhibit superior performance compared to the PLN approach, which does not account for the underlying Markovian tree structure of the data. Similar to our previous synthetic experiment, we observe that as we traverse deeper into the tree structure, the performance of PLN models deteriorates significantly. When looking at the alpha diversities in Table 15, the backward variational approach demonstrates superior performance compared to the mean-field approach, which is supported by its higher efficiency at the reconstruction task on unseen samples summarized in Table 6. This experiment then demonstrates the practical interest of considering the structure of the true posterior when doing variational inference to learn the model.

	PLN-Tree	**PLN-Tree (MF)**
$\ell = 1$	**0.995** (0.062)	0.967 (0.103)
$\ell = 2$	**0.989** (0.065)	0.967 (0.078)
$\ell = 3$	**0.987** (0.075)	0.973 (0.087)

Table 6: Correlation between reconstructed abundances and the test dataset from the Markov Dirichlet model (1000 samples), averaged over the samples, with standard deviation.

Label	**Nb of training samples**	**Nb of test samples**	**Total**
IBD (Crohn)	20	5	25
Colorectal Cancer	38	10	48
Leaness	71	18	89
Liver Cirrhosis	94	24	118
IBD (UC)	118	30	148
Obesity	131	33	164
Type 2 Diabetes	178	45	223
Total	650	165	815

Table 7: Metagenomics dataset considered in our experiments, extracted from Pasolli et al. (2016). The samples are drawn randomly for each label to satisfy these counts.

4.2 Metagenomics dataset: application to the gut microbiome

Description of the dataset and preprocessing We assess the efficacy of the PLN-Tree model using a metagenomics dataset introduced in Pasolli et al. (2016). This dataset comprises microbial compositions from both control individuals and patients with various diseases, totaling 3610 samples. Our analysis focuses exclusively on the gut microbial compositions of disease-associated patients, as recapitulated in Table 7. Each sample is characterized by hierarchical proportion data, with the base tree representing the taxonomy of Archaea, Eukaryota, and Bacteria. Sequencing was conducted using MetaPhlAn2, optimized for bacterial sequencing (Truong et al., 2015), thus restricting our study to bacteria. Besides, we limit our analysis to the layers of the taxonomy exploration comprised between the second and fifth layers, which respectively correspond to the "class" and the "family". To convert the proportions of taxa within each patient's gut into count data, we sample counts from a multinomial distribution with a total count of $\exp(12)$ and gut sample compositions as probabilities, as generally done in microbiome rarefaction procedures to standardize count data (Schloss, 2024). Additionally, we implement prevalence filtering using a threshold of $1 \times \mathrm{e}^{-12}$ to filter very rare Operational Taxonomic Units (OTUs).

4.2.1 Data augmentation with PLN-based methods

In Appendix D.3, we provide a summary of the tested and selected architectures for the PLN-Tree models. Each compared model is trained once, while sampling is repeated $M = 25$ with 2000 samples to account for sampling variability in the model evaluation.

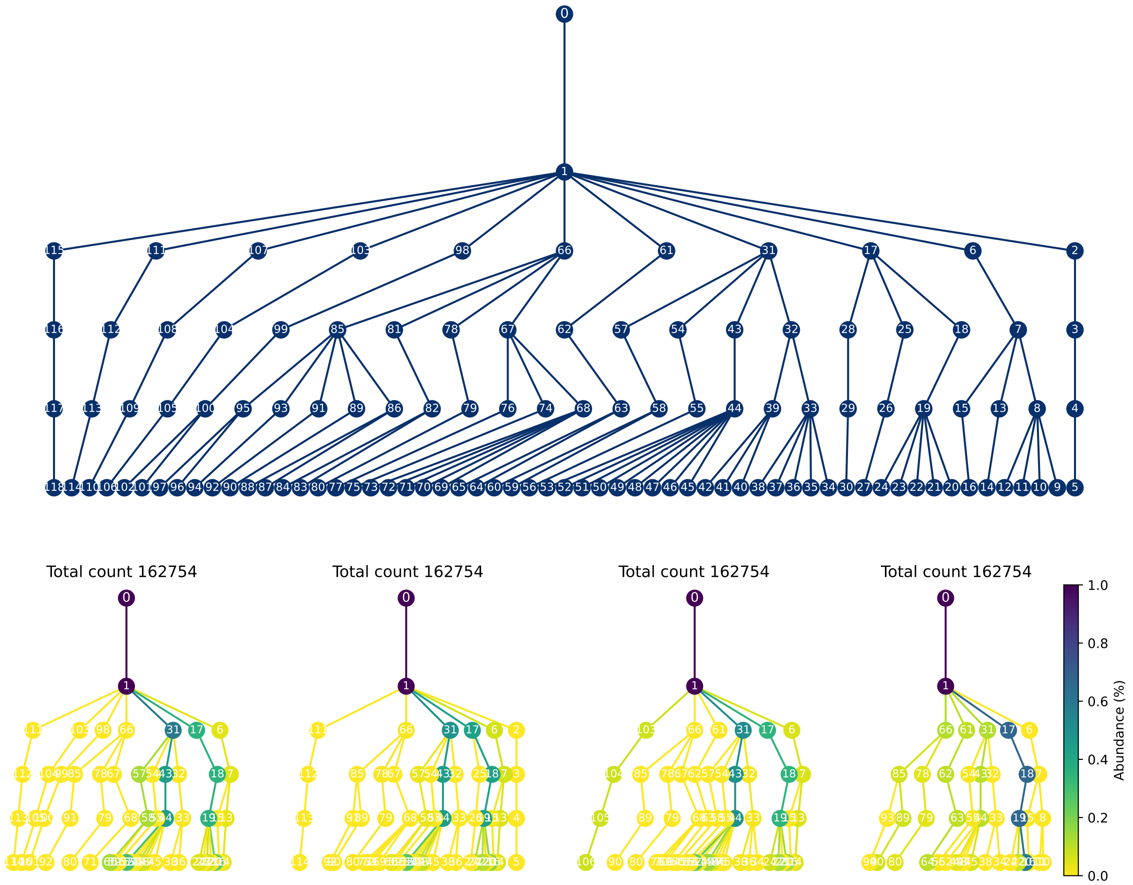


Figure 4: Graph of the taxonomy considered in the metagenomics experiments (top), and four samples from the dataset (bottom).

	PLN-Tree	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
		Wasserstein distance $(\times 10^2)$		
$\ell = 1$	5.89 (0.29)	**4.67** (0.25)	15.57 (0.59)	8.38 (0.36)
$\ell = 2$	8.83 (0.28)	**7.55** (0.14)	20.65 (0.71)	15.49 (0.63)
$\ell = 3$	9.27 (0.27)	**7.76** (0.12)	20.86 (0.70)	18.34 (0.83)
$\ell = 4$	17.00 (0.22)	**15.59** (0.13)	29.29 (0.72)	29.29 (0.72)

Table 8: Empirical Wasserstein distance between normalized metagenomics data and normalized simulated data under each modeled trained, for each layer, averaged over the trainings, with standard deviation. Since PLN does not verify the tree compositionality constraint, it is placed aside as a reference. The best-performing model in each row is indicated in bold.

Alpha diversity	**PLN-Tree**	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
		Wasserstein distance $(\times 10^2)$		
Shannon $\ell = 1$	**1.73** (0.44)	3.00 (0.44)	16.49 (1.14)	4.00 (0.50)
Shannon $\ell = 2$	**2.22** (0.73)	5.70 (0.97)	23.21 (1.64)	8.37 (1.21)
Shannon $\ell = 3$	**2.29** (0.63)	6.58 (1.02)	23.96 (1.67)	9.28 (1.43)
Shannon $\ell = 4$	**2.08** (0.62)	20.39 (1.08)	55.32 (2.38)	55.32 (2.38)
Simpson $\ell = 1$	0.84 (0.14)	**0.71** (0.12)	7.18 (0.48)	1.71 (0.23)
Simpson $\ell = 2$	0.92 (0.24)	**0.73** (0.19)	7.49 (0.57)	2.55 (0.33)
Simpson $\ell = 3$	0.91 (0.23)	**0.72** (0.19)	7.46 (0.57)	2.88 (0.29)
Simpson $\ell = 4$	**0.53** (0.13)	2.41 (0.21)	12.91 (0.67)	12.91 (0.67)

Table 9: Wasserstein distance on alpha diversities distributions computed between metagenomics data and simulated data under each modeled trained, averaged over the trainings, with standard deviation. Since PLN does not verify the tree compositionality constraint, it is placed aside as a reference. The best-performing model in each row is indicated in bold.

Exploiting the taxonomy improves the performances We provide a summary of the model performances in Tables 8 and 9, while Figure 6 illustrates the variability of the generations for each model. Notably, the tree-based models exhibit superior performance compared to the PLN approach, which does not account for the taxonomy. Specifically, as we delve deeper into the tree structure, the performance of PLN models declines, while the PLN-Tree models maintain consistency with depth. These findings suggest that taxonomy provides pertinent insights into the distribution of bacteria and their interactions within the host ecosystem, bearing significant biological implications. However, as shown in Figure 6, PLN-Tree approaches struggle with modeling zero-valued abundances (see Bacteria 2, 61, 107 for instance), particularly when using the mean-field approximation. This issue, which accumulates across layers due to the top-down dynamic of the model, could be addressed using zero-inflation techniques, similar to the approach taken for PLN in Batardière et al. (2024).

Variational approximation performances Analysis of Table 17 underscores the consistently superior performance of the residual amortized backward approximation compared to the mean-field approach. This

	PLN-Tree	**PLN-Tree (MF)**
$\ell = 1$	**0.971** (0.113)	0.850 (0.184)
$\ell = 2$	**0.971** (0.084)	0.843 (0.185)
$\ell = 3$	**0.826** (0.243)	0.804 (0.258)
$\ell = 4$	**0.917** (0.165)	0.736 (0.212)

Table 10: Correlation between reconstructed abundances and the test samples from the metagenomics dataset (see Table 7), averaged over the samples, with standard deviation.

observation is further supported by the reconstruction task results summarized in Table 10, where structured variational inference exhibits a distinct advantage over the conventional mean-field method in this practical context. Even when the mean-field approximation outperforms the backward approach, as evidenced by the sample distributions in Table 8, the backward approach remains competitive, indicating its overall effectiveness as the preferred variational approximation method on the metagenomics dataset.

4.2.2 Data preprocessing using PLN-Tree for classification tasks

The metagenomics dataset from Pasolli et al. (2016) involves a one-vs-all disease classification problem using microbiome proportion data, which are highly sparse and compositional, presenting challenges for direct use in machine learning algorithms (Rodriguez, 2022). To mitigate these constraints, several preprocessing techniques have been proposed, including the additive, centered, and isometric log-ratio transforms, which are commonly used for standard preprocessing (Greenacre, 2021) even though they struggle in highly sparse context and lack theoretical groundings O'Hara and Kotze (2010). More recently, Chiquet et al. (2018) introduced the use of the PLN model to perform PCA in the latent space, demonstrating that latent variables can facilitate machine learning tasks. Therefore, PLN-based approaches can serve as preprocessing pipelines by encoding observations into a latent space, using these latent variables as input data for machine learning models instead of the raw observations. Given the significant improvements in data generation when accounting for underlying hierarchical structures, we aim to investigate whether exploiting the taxonomy through PLN-Tree can also yield meaningful features for solving classification problems. To benchmark the effect of PLN-based preprocessing on the metagenomics dataset, we focus on the T2D-vs-all classification problem, as well as the IBD-vs-all scenario in Appendix D.3.2. The dataset description is provided in Table 7, the considered taxonomic levels remain the same as in the previous experiment.

Building tabular latent variables with PLN-Tree In the PLN-Tree framework, the latent variable of the first layer models the total count, while the next variables account for how the counts progressively distribute over the layers in the observed space. To design a tabular latent input accounting for both the counts and how they propagate, we suggest a feature-engineered latent feature called Latent Tree Counts (LTC). Define $\mathbf{V}^1 = \exp(\mathbf{Z}^1)$ and for all $\ell < L, k \leqslant K_\ell$,

$$\check{\mathbf{V}}_k^\ell = \sigma(\check{\mathbf{Z}}_k^\ell) \times \mathbf{V}_k^\ell \ .$$

Benchmark procedure We seek to compare the influence of the preprocessing techniques using the conventional PLN model and the PLN-Tree log-LTC transform, against the raw normalized data employed in the study Pasolli et al. (2016). To that end, we train the PLN-based models on the entire dataset using the previously selected architectures, thereby obtaining an encoder for each model, which enables the mapping of raw counts to latent features of interest. In the case of the PLN-Tree models, we also apply the log-LTC transform to the latent features. Then, we select several tabular classifiers with fixed architectures (see Appendix D.3.2) and proceed to a 50 stratified K-Fold cross-validation for each model, which allows to account for the training variability on the performances, using 80% of the most precise taxa-abundance data to train the models (family level). In this experiment, we assume the availability of the full dataset, using all available data to train the encoders for preprocessing. In practical applications, preprocessing models are typically trained on an existing dataset and then applied to new data, raising questions about the generalization capabilities of the encoders. We partially explored this generalization in a prior correlation analysis for PLN-Tree variants (see Table 10), demonstrating the superiority of the residual backward approximation. However, regular PLN models do not support encoding samples outside of the training dataset, as one pair of variational parameters is learned per sample (see Chiquet et al. (2019)). Given the small sample sizes of the test datasets and to prevent model biases, we train each compared model on the entire dataset. This approach mitigates the advantages of the residual backward PLN-Tree method over the mean-field variant, and its scalability in this context compared to the regular PLN model.

T2D-vs-all experiment We consider the classification task of patients with type 2 diabetes against patients with other diseases. In Table 11, we present the performance of various classifiers using the raw data, as well as data preprocessed with PLN latents or the log-LTC transform from PLN-Tree models, employing either the residual backward amortized variational approximation or the mean-field approximation. Overall, our results indicate that preprocessing with PLN-based models generally enhances performances, except for random forests in this application. The regular PLN and PLN-Tree preprocessing using the backward approximation exhibit similar performance, as their metrics mostly overlap within the variance region. We also observe similar performances between the backward PLN-Tree and its mean-field counterpart, with a slight advantage for the backward approach, indicating that both methods enable an efficient scalable preprocessing of microbiome data. The IBD-vs-all experiment conducted in Appendix D.3.2 highlights similar results. Overall, these results indicate a slight advantage in considering the underlying taxonomy to provide relevant latent variables for disease classification. Further exploration of deep recursive architectures appears promising for enhancing these results by leveraging PLN-Tree latent variables given their Markov nature.

	Raw data	**log-LTC (PLN-Tree)**	**log-LTC (MF)**	**PLN**
Logistic Regression				
Balanced Accuracy	0.632 (0.042)	**0.737** (0.038)	0.730 (0.037)	0.716 (0.037)
Precision	0.701 (0.032)	**0.781** (0.029)	0.775 (0.028)	0.765 (0.028)
Recall	0.645 (0.039)	**0.748** (0.029)	0.742 (0.026)	0.727 (0.032)
F1 score	0.661 (0.036)	**0.757** (0.028)	0.751 (0.026)	0.738 (0.030)
ROC AUC	0.677 (0.045)	**0.795** (0.035)	0.781 (0.035)	0.791 (0.037)
ROC Precision-Recall	0.438 (0.061)	0.570 (0.063)	0.543 (0.055)	**0.601** (0.069)
Linear SVM				
Balanced Accuracy	0.586 (0.042)	**0.738** (0.034)	0.732 (0.036)	0.714 (0.033)
Precision	0.673 (0.035)	**0.782** (0.026)	0.777 (0.028)	0.763 (0.025)
Recall	0.584 (0.061)	**0.748** (0.029)	0.737 (0.030)	0.720 (0.031)
F1 score	0.598 (0.062)	**0.757** (0.027)	0.748 (0.029)	0.732 (0.029)
ROC AUC	0.545 (0.127)	**0.797** (0.034)	0.782 (0.034)	0.791 (0.036)
ROC Precision-Recall	0.336 (0.085)	0.581 (0.070)	0.534 (0.058)	**0.597** (0.067)
Neural Network				
Balanced Accuracy	0.704 (0.036)	**0.751** (0.030)	0.701 (0.042)	0.727 (0.034)
Precision	0.773 (0.026)	**0.812** (0.025)	0.771 (0.03)	0.795 (0.026)
Recall	0.777 (0.028)	**0.816** (0.024)	0.769 (0.032)	0.802 (0.025)
F1 score	0.772 (0.027)	**0.812** (0.024)	0.765 (0.03)	0.795 (0.026)
ROC AUC	0.782 (0.036)	0.842 (0.029)	0.787 (0.04)	**0.854** (0.024)
ROC Precision-Recall	0.620 (0.062)	**0.694** (0.061)	0.622 (0.071)	0.692 (0.051)
Random Forest				
Balanced Accuracy	**0.673** (0.032)	0.639 (0.030)	0.672 (0.031)	0.610 (0.027)
Precision	**0.827** (0.026)	0.798 (0.030)	0.787 (0.028)	0.790 (0.033)
Recall	**0.811** (0.019)	0.789 (0.018)	0.794 (0.021)	0.776 (0.016)
F1 score	**0.781** (0.026)	0.751 (0.025)	0.772 (0.025)	0.727 (0.024)
ROC AUC	**0.903** (0.022)	0.861 (0.026)	0.829 (0.028)	0.887 (0.026)
ROC Precision-Recall	**0.790** (0.052)	0.727 (0.057)	0.660 (0.061)	0.736 (0.069)

Table 11: Classification T2D-vs-all performances for several classifiers on the metagenomics dataset using different preprocessing strategies, averaged over training, with standard deviation. We perform 50 stratified K-folds using 80% of the dataset, using only the "family" level of the taxonomy.

5 Discussion

In this paper, we introduced the PLN-Tree model as an extension of the Poisson log-normal framework, designed to accommodate hierarchical count data. To learn the parameters of the PLN-Tree model, we proposed a structured variational inference approximation to effectively learn model parameters by exploiting the true form of the posterior distribution using deep learning parameterizations, showing highly competitive performances against the regular mean-field approximation. Additionally, we established the identifiability properties of the PLN-Tree model, providing insights into its theoretical foundations and validating its practical reliability.

To assess the performance of the PLN-Tree model, we conducted comprehensive experiments on both synthetic and real-world datasets, benchmarking it against the established PLN framework on data augmentation and classification tasks. By using the underlying tree structure, our results underscored the efficacy and consistency of the PLN-Tree model in capturing the diversity of the data at all depths, contrary to the regular PLN approach. This highlights the relevance of hierarchical structures organizing entities, such as the taxonomy, in modeling complex biological systems like the microbiome. Furthermore, we illustrated the potential of PLN-Tree models as a preprocessing pipeline to facilitate machine learning tasks on microbiome data, showing the versatility of the model. Overall, our contribution offers valuable insights into the practical utility of considering knowledge graphs in modeling approaches, particularly in domains characterized by intricate data structures such as ecology or microbiology.

However, the PLN-Tree model has certain limitations. While it precisely models proportion-based alpha diversities, it does not account for sparse structures effectively due to its propagation dynamics. Inspired by the ZI-PLN model Batardière et al. (2024), a zero-inflated PLN-Tree variant could address this limitation and represent a promising direction for future research. Additionally, the proposed PLN-Tree model does not include covariates for simplicity. However, adding covariates into the mean through a linear regression model is a natural extension from the original PLN model (Chiquet et al., 2021). The modular nature of the PLN-Tree model also allows for the injection of covariates at each layer to model their impact on the latent dynamics. Exploring the effect of covariates on preprocessing using PLN-based models in practical applications is an interesting research direction that could enhance classification results.

5.1 Acknowledgments

We would like to gratefully thank Harry Sokol, co-director of Alexandre Chaussard's PhD program and direct medical advisor for this work. We also acknowledge the Institute of Computing and Data Sciences (ISCD) from Sorbonne Université for funding the PhD thesis of Alexandre Chaussard.

References

Aitchison, J. and Ho, C. (1989). The multivariate poisson-log normal distribution. *Biometrika*, 76(4):643–653.

Altenbuchinger, M., Weihs, A., Quackenbush, J., Grabe, H. J., and Zacharias, H. U. (2020). Gaussian and mixed graphical models as (multi-) omics data analysis tools. *Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms*, 1863(6):194418.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516.

Batardière, B., Chiquet, J., Gindraud, F., and Mariadassou, M. (2024). Zero-inflation in the multivariate poisson lognormal family. *arXiv preprint arXiv:2405.14711*.

Bichat, A., Plassais, J., Ambroise, C., and Mariadassou, M. (2020). Incorporating phylogenetic information in microbiome differential abundance studies has no effect on detection power and fdr control. *Frontiers in microbiology*, 11:489364.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Campbell, A., Shi, Y., Rainforth, T., and Doucet, A. (2021). Online variational filtering and parameter learning. *Advances in Neural Information Processing Systems*, 34:18633–18645.

Chagneux, M., Gassiat, É., Gloaguen, P., and Le Corff, S. (2024). Additive smoothing error in backward variational inference for general state-space models. *Journal of Machine Learning Research*, 25(28):1–33.

Chiquet, J., Mariadassou, M., and Robin, S. (2018). Variational inference for probabilistic poisson pca. *Annals of Applied Statistics*.

Chiquet, J., Mariadassou, M., and Robin, S. (2021). The poisson-lognormal model as a versatile framework for the joint analysis of species abundances. *Frontiers in Ecology and Evolution*, 9:588292.

Chiquet, J., Robin, S., and Mariadassou, M. (2019). Variational inference for sparse network reconstruction from count data. In *International Conference on Machine Learning*, pages 1162–1171. PMLR.

Côme, E., Jouvin, N., Latouche, P., and Bouveyron, C. (2021). Hierarchical clustering with discrete latent variable models and the integrated classification likelihood. *Advances in Data Analysis and Classification*, 15(4):957–986.

Crawford, J. and Greene, C. S. (2020). Incorporating biological structure into machine learning models in biomedicine. *Current opinion in biotechnology*, 63:126–134.

D'Amour, A., Heller, K., Moldovan, D., Adlam, B., Alipanahi, B., Beutel, A., Chen, C., Deaton, J., Eisenstein, J., Hoffman, M. D., et al. (2022). Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., et al. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Gassiat, É. and Le Corff, S. (2024). Variational excess risk bound for general state space models. *Transactions on Machine Learning Research*.

Gassiat, E., Le Corff, S., and Lehéricy, L. (2020). Identifiability and consistent estimation of nonparametric translation hidden markov models with general state space. *Journal of Machine Learning Research*, 21(115):1–40.

Gotelli, N. J. and Colwell, R. K. (2001). Quantifying biodiversity: procedures and pitfalls in the measurement and comparison of species richness. *Ecology letters*, 4(4):379–391.

Greenacre, M. (2021). Compositional data analysis. *Annual Review of Statistics and its Application*, 8:271–299.

Hälvä, H., Le Corff, S., Lehéricy, L., So, J., Zhu, Y., Gassiat, E., and Hyvarinen, A. (2021). Disentangling identifiable features from noisy data with structured nonlinear ica. *Advances in Neural Information Processing Systems*, 34:1624–1633.

Harris, D. J. (2016). Inferring species interactions from co-occurrence data with markov networks. *Ecology*, 97(12):3308–3314.

Hilbe, J. M. (2014). *Modeling count data*. Cambridge University Press.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Inouye, D. I., Yang, E., Allen, G. I., and Ravikumar, P. (2017). A review of multivariate distributions for count data derived from the poisson distribution. *Wiley Interdisciplinary Reviews: Computational Statistics*, 9(3):e1398.

Johnson, M. J., Duvenaud, D. K., Wiltschko, A., Adams, R. P., and Datta, S. R. (2016). Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29.

Jost, L. (2006). Entropy and diversity. *Oikos*, 113(2):363–375.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020). Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979.

Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.

Lee, S., Abecasis, G. R., Boehnke, M., and Lin, X. (2014). Rare-variant association analysis: study designs and statistical tests. *The American Journal of Human Genetics*, 95(1):5–23.

Lin, W., Khan, M. E., and Hubacher, N. (2018). Variational message passing with structured inference networks. In *International Conference on Learning Representations*.

Marino, J., Cvitkovic, M., and Yue, Y. (2018). A general method for amortizing variational filtering. *Advances in neural information processing systems*, 31.

Nagendra, H. (2002). Opposite trends in response for the shannon and simpson indices of landscape diversity. *Applied geography*, 22(2):175–186.

O'Hara, R. and Kotze, J. (2010). Do not log-transform count data. *Nature Precedings*, pages 1–1.

Oliver, A., Kay, M., and Lemay, D. G. (2023). Taxahfe: a machine learning approach to collapse microbiome datasets using taxonomic structure. *Bioinformatics Advances*, 3(1):vbad165.

Pasolli, E., Truong, D. T., Malik, F., Waldron, L., and Segata, N. (2016). Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. *PLoS computational biology*, 12(7):e1004977.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Rodriguez, E. G. (2022). *Advances in Machine Learning for Compositional Data*. Columbia University.

Schloss, P. D. (2024). Rarefaction is currently the best approach to control for uneven sequencing effort in amplicon sequence analyses. *Msphere*, pages e00354–23.

Schneider, S., Lee, J. H., and Mathis, M. W. (2023). Learnable latent embeddings for joint behavioural and neural analysis. *Nature*, 617(7960):360–368.

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2004). Sharing clusters among related groups: Hierarchical dirichlet processes. *Advances in neural information processing systems*, 17.

Thukral, A. K. (2017). A review on measurement of alpha diversity in biology. *Agricultural Research Journal*, 54(1).

Tomczak, J. and Welling, M. (2018). Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pages 1214–1223. PMLR.

Truong, D. T., Franzosa, E. A., Tickle, T. L., Scholz, M., Weingart, G., Pasolli, E., Tett, A., Huttenhower, C., and Segata, N. (2015). Metaphlan2 for enhanced metagenomic taxonomic profiling. *Nature methods*, 12(10):902–903.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679.

Weinroth, M. D., Belk, A. D., Dean, C., Noyes, N., Dittoe, D. K., Rothrock Jr, M. J., Ricke, S. C., Myer, P. R., Henniger, M. T., Ramírez, G. A., et al. (2022). Considerations and best practices in animal science 16s ribosomal rna gene sequencing microbiome studies. *Journal of animal science*, 100(2):skab346.

Xu, C., Wu, K., Zhang, J.-G., Shen, H., and Deng, H.-W. (2017). Low-, high-coverage, and two-stage dna sequencing in the design of the genetic association study. *Genetic epidemiology*, 41(3):187–197.

Yu, X., Zeng, T., Wang, X., Li, G., and Chen, L. (2015). Unravelling personalized dysfunctional gene network of complex diseases based on differential network model. *Journal of translational medicine*, 13:1–13.

A Alpha diversity metrics

Alpha diversities are a set of metrics used in ecology and biology to quantify the variety and distribution of species within a particular ecosystem (Gotelli and Colwell, 2001; Thukral, 2017). These measures consider the diversity within a single sample (a given ecosystem) without considering interactions with other samples. There exist numerous indices to compute alpha diversity, which evaluate species richness and/or evenness. Species richness refers to the total number of different species present in the sample, while evenness measures how evenly the entities are distributed among the species. High alpha diversity often indicates a healthy ecosystem with a wide variety of species, while low alpha diversity suggests a less diverse or possibly disturbed ecosystem.

A.1 Shannon entropy

Originally introduced for information theory, the Shannon entropy is a widely used alpha diversity metric in ecology to measure species diversity within a given community (Thukral, 2017). It considers both species richness and evenness by considering the relative abundance of each species. The Shannon entropy calculates the uncertainty or randomness in species composition, reflecting the information content of the community. Higher values of Shannon entropy indicate greater diversity, where species are more evenly distributed, while lower values suggest lower diversity or dominance by a few species. Denoting by p_s the empirical proportion of the species s in the ecosystem, the Shannon entropy is computed as

$$H = -\sum_{s=1}^{S} p_s \log p_s \ .$$

The interpretation of the Shannon entropy as an alpha diversity is described for instance in Jost (2006).

A.2 Simpson index

The Simpson alpha diversity metric assesses species diversity within a specific habitat (Thukral, 2017). It focuses on the probability that two individuals randomly selected from the community belong to different species. Letting p_s the empirical proportion of species s in the ecosystem, the Simpson index is computed as

$$S = \sum_{s=1}^{S} p_s^2 \ .$$

This metric emphasizes the importance of species evenness in a community, giving more weight to rare species. The interpretation of the Simpson index as an alpha diversity is given by its reciprocal as the Inverse Simpson index (Jost, 2006).

B ELBO derivation for PLN-Tree

B.1 Proof of Proposition 3.1

The prior distribution of $\mathbf{Z}$ is denoted by $p_{\boldsymbol{\theta},1:L}(\mathbf{Z}) = p_{\boldsymbol{\theta},1}(\mathbf{Z}^1)p_{\boldsymbol{\theta},\ell|\ell-1}(\mathbf{Z}^{\ell+1}|\mathbf{Z}^{\ell})$. By definition of the ELBO,

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\varphi}) = \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},1:L}(\mathbf{X}|\mathbf{Z})\right] - \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},1:L}\|p_{\boldsymbol{\theta},1:L}\right] \ .$$

Using the Markov tree structure of the observed counts yields

$$\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},1:L}(\mathbf{X}|\mathbf{Z})\right] = \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},1}(\mathbf{X}^1|\mathbf{Z}^1)\right] + \sum_{\ell=1}^{L-1}\sum_{k=1}^{K_\ell}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},\ell+1|\ell}(\check{\mathbf{X}}_k^{\ell}|\check{\mathbf{Z}}_k^{\ell},\mathrm{X}_k^{\ell})\right] \ .$$

The first layer is modeled by a Poisson lognormal distribution, thus it can be expressed as

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},1}(\mathbf{X}^1|\mathbf{Z}^1)\right] = &\sum_{k=1}^{K_1}\mathrm{X}_k^1\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathbf{m}_{\boldsymbol{\varphi}^1,k}(\mathbf{Z}^2,\mathbf{X}^1)\right] \\ &- \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\exp\left(\frac{\mathbf{S}_{\boldsymbol{\varphi}^1,k}(\mathbf{Z}^2,\mathbf{X}^1)}{2} + \mathbf{m}_{\boldsymbol{\varphi}^{\ell},k}(\mathbf{Z}^2,\mathbf{X}^1))\right)\right] - \log(\mathrm{X}_k^1!) \ .\end{aligned}$$

The propagation of the counts along the tree conditionally to the respective latent variables and the parent counts is given by a multinomial distribution, which enables to explicit the second term as

$$\begin{aligned}\sum_{\ell=1}^{L-1}\sum_{k=1}^{K_\ell}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},\ell+1|\ell}(\check{\mathbf{X}}_k^{\ell}|\check{\mathbf{Z}}_k^{\ell},\mathrm{X}_k^{\ell})\right] = &\sum_{k=1}^{K_1}\log(\mathrm{X}_k^1!) - \sum_{k=1}^{K_L}\log(\mathrm{X}_k^L!) \\ &+ \sum_{\ell=1}^{L-1}\sum_{k=1}^{K_\ell}\left\{\sum_{j\in\mathcal{C}_k^{\ell}}\mathrm{X}_j^{\ell+1}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathrm{Z}_j^{\ell+1}\right] - \mathrm{X}_k^{\ell}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log\left(\sum_{j\in\mathcal{C}_k^{\ell}}\mathrm{e}^{\mathrm{Z}_j^{\ell+1}}\right)\right]\right\} \ .\end{aligned}$$

Combining those results and applying the tower property to the latent expectations yields

$$\begin{aligned}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta},1:L}(\mathbf{X}|\mathbf{Z})\right] = \sum_{\ell=1}^{L}\sum_{k=1}^{K_\ell}\Bigg(&\mathrm{X}_k^{\ell}\left(\mathbb{1}_{\ell<L}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathbf{m}_{\boldsymbol{\varphi}^{\ell},k}(\mathbf{Z}^{\ell+1},\mathbf{X}^{1:\ell})\right] + \mathbb{1}_{\ell=L}\mathbf{m}_{\boldsymbol{\varphi}^{L},k}(\mathbf{X}^{1:L})\right) \\ &- \mathbb{1}_{\ell=1}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\exp\left(\frac{\mathbf{S}_{\boldsymbol{\varphi}^{\ell},k}(\mathbf{Z}^{\ell+1},\mathbf{X}^{1:\ell})}{2} + \mathbf{m}_{\boldsymbol{\varphi}^{\ell},k}(\mathbf{Z}^{\ell+1},\mathbf{X}^{1:\ell})\right)\right]\Bigg) \\ &- \mathbb{1}_{\ell>1}\sum_{k=1}^{K_{\ell-1}}\mathrm{X}_k^{\ell-1}\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log\left(\sum_{j\in\mathcal{C}_k^{\ell-1}}\mathrm{e}^{\mathrm{Z}_j^{\ell}}\right)\right] \\ &- \mathbb{1}_{\ell=L}\sum_{k=1}^{K_\ell}\log(\mathrm{X}_k^{\ell}!) + \frac{1}{2}K_\ell \ .\end{aligned}$$

The divergence term can be expressed as

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},1:L} \| p_{\boldsymbol{\theta},1:L}\right] \\
&\quad= \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log\left(\frac{q_{\boldsymbol{\varphi},1|2}(\mathbf{Z}^1 \mid \mathbf{Z}^2, \mathbf{X}^{1:2})}{p_{\boldsymbol{\theta},1}(\mathbf{Z}^1)} \prod_{\ell=2}^{L-1} \frac{q_{\boldsymbol{\varphi},\ell|\ell+1}(\mathbf{Z}^\ell \mid \mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})}{p_{\boldsymbol{\theta},\ell|\ell-1}(\mathbf{Z}^\ell \mid \mathbf{Z}^{\ell-1})} \frac{q_{\boldsymbol{\varphi},L}(\mathbf{Z}^L \mid \mathbf{X}^{1:L})}{p_{\boldsymbol{\theta},L|L-1}(\mathbf{Z}^L \mid \mathbf{Z}^{L-1})}\right)\right] \\
&\quad= \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},1|2} \| p_{\boldsymbol{\theta},1}\right]\right] + \sum_{\ell=2}^{L-1} \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},\ell|\ell+1} \| p_{\boldsymbol{\theta},\ell|\ell-1}\right]\right] \\
&\qquad\qquad + \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},L} \| p_{\boldsymbol{\theta},L|L-1}\right]\right].
\end{aligned}$$

For $1 < \ell < L$, the Kullback-Leibler divergence writes

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},\ell|\ell+1} \| p_{\boldsymbol{\theta},\ell|\ell-1}\right] &= -\frac{1}{2}\left[\log|\boldsymbol{\Omega}_{\theta^\ell}(\mathbf{Z}^{\ell-1})| + \log|\mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})| + K_\ell\right] \\
&\qquad + \frac{1}{2}\mathrm{tr}\left(\widehat{\boldsymbol{\Sigma}}_\ell \boldsymbol{\Omega}_{\boldsymbol{\theta}^\ell}(\mathbf{Z}^{\ell-1})\right) ,
\end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}}_\ell$ is defined in (2). Following the same steps for the other terms yields

$$\begin{aligned}
&\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},1:L} \| p_{\boldsymbol{\theta},1:L}\right] \\
&\quad= -\frac{1}{2}\sum_{\ell=1}^{L} \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log|\boldsymbol{\Omega}_{\boldsymbol{\theta}^\ell}(\mathbf{Z}^{\ell-1})| + \log|\mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})| - \mathrm{tr}\left(\widehat{\boldsymbol{\Sigma}}_\ell \boldsymbol{\Omega}_{\boldsymbol{\theta}^\ell}(\mathbf{Z}^{\ell-1})\right)\right] + K_\ell ,
\end{aligned}$$

which concludes the proof.

B.2 Proof of Lemma 3.2

By definition,

$$\mathcal{L}_{\text{offset}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\log p_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{Z}, \mathrm{O})\right] - \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi},1:L} \| p_{\boldsymbol{\theta},}^{(\mathrm{O},\mathbf{Z})}\right] .$$

Conditioning $(\mathbf{X}, \mathbf{Z})$ by O yields

$$\mathcal{L}_{\text{offset}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathcal{L}_{|\mathrm{O}}(\boldsymbol{\theta}, \boldsymbol{\varphi}) + \mathbb{E}_{q_{\boldsymbol{\varphi}}^{\mathrm{O}}}\left[\log p_{\boldsymbol{\theta}}(\mathrm{O})\right] + \mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi}}^{\mathrm{O}} \| p_{\boldsymbol{\theta},}^{\mathrm{O}}\right] .$$

Using the KL divergence definition

$$\mathrm{D}_{\mathrm{KL}}\left[q_{\boldsymbol{\varphi}}^{\mathrm{O}} \| p_{\boldsymbol{\theta},}^{\mathrm{O}}\right] = -\mathrm{H}_{q_{\boldsymbol{\varphi}}^{\mathrm{O}}} - \mathbb{E}_{q_{\boldsymbol{\varphi}}^{\mathrm{O}}}\left[\log p_{\boldsymbol{\theta}}(\mathrm{O})\right] ,$$

since $\mathrm{H}q_{\boldsymbol{\varphi}}^{\mathrm{O}}$ is Gaussian, its entropy is given by $\frac{1}{2}\log(2\pi \mathrm{e} s_o^2(\mathbf{X}))$, which concludes the proof.

C Identifiability results

C.1 PLN identifiability

Lemma C.1. *Let* Z *and* $\tilde{\mathrm{Z}}$ *be supported on* $\mathbb{R}_+^*$*, and* $\mathrm{X} \sim \mathcal{P}(\mathrm{Z})$ *and* $\tilde{\mathrm{X}} \sim \mathcal{P}(\mathrm{Z})$*. Then, if* X *and* $\tilde{\mathrm{X}}$ *have the same distribution,* Z *and* $\tilde{\mathrm{Z}}$ *have the same distribution.*

Proof. Let h be a measurable function, then we have

$$\mathbb{E}\left[h(\mathrm{X})\right] = \mathbb{E}\left[\mathbb{E}\left[h(\mathrm{X}) \mid \mathrm{Z}\right]\right] = \mathbb{E}\left[\sum_{x \in \mathbb{N}} \mathrm{e}^{-\mathrm{Z}} \frac{\mathrm{Z}^x}{x!} h(x)\right] .$$

For all $t \in \mathbb{R}$, taking $h(x) = t^x$ yields

$$\mathbb{E}\left[h(\mathrm{X})\right] = \mathbb{E}\left[\mathrm{e}^{-\mathrm{Z}} \sum_{x \in \mathbb{N}} \frac{(\mathrm{Z}t)^x}{x!}\right] = \mathbb{E}\left[\mathrm{e}^{(t-1)\mathrm{Z}}\right] = \mathrm{M}_{\mathrm{Z}}(t-1) .$$

Since X and $\tilde{\mathrm{X}}$ have the same law, then we have for all $u \leqslant 0, \mathrm{M}_{\mathrm{Z}}(u) = \mathrm{M}_{\tilde{\mathrm{Z}}}(u)$. Write $\mathrm{Y} = \exp(-\mathrm{Z})$ and $\tilde{\mathrm{Y}} = \exp(-\tilde{\mathrm{Z}})$. The random variables $\tilde{\mathrm{Y}}$ and Y are compactly supported so by the Stone-Weierstrass theorem their distribution is characterized by their moments $(\mathbb{E}\left[\mathrm{Y}^k\right])_{k \geqslant 0}$ and $(\mathbb{E}[\tilde{\mathrm{Y}}^k])_{k \geqslant 0}$. Therefore $\tilde{\mathrm{Y}}$ and Y have the same law, which concludes the proof. □

Lemma C.2. *Let* Z *and* $\tilde{\mathrm{Z}}$ *be two real random variables, and* $\mathrm{X} \sim \mathcal{P}(\mathrm{e}^{\mathrm{Z}})$ *and* $\tilde{\mathrm{X}} \sim \mathcal{P}(\mathrm{e}^{\tilde{\mathrm{Z}}})$. *Then, if* X *and* $\tilde{\mathrm{X}}$ *have the same distribution,* Z *and* $\tilde{\mathrm{Z}}$ *have the same distribution.*

Proof. By Lemma C.1, e^{Z} and $\mathrm{e}^{\tilde{\mathrm{Z}}}$ have the same distribution which is enough to conclude the proof. □

C.2 PLN-Tree identifiability

C.2.1 Proof of Lemma 3.3

Let $h(\mathrm{X}^1, \ldots, \mathrm{X}^L) = \prod_{\ell=1}^{L} h_\ell(\mathrm{X}^\ell)$ where $\{h_\ell\}_{1 \leqslant \ell \leqslant L}$ are measurable functions. Then,

$$\begin{aligned}\mathbb{E}\left[h(\mathrm{X}^1, \ldots, \mathrm{X}^L)\right] = \mathbb{E}\left[\mathbb{E}\left[h(\mathrm{X}^1, \ldots, \mathrm{X}^L) \mid \mathbf{Z}\right]\right] &= \mathbb{E}\left[\prod_{\ell=1}^{L} \mathbb{E}\left[h_\ell(\mathrm{X}^\ell) \mid \mathrm{Z}^\ell\right]\right] \\ &= \mathbb{E}\left[\prod_{\ell=1}^{L} \sum_{x \in \mathbb{N}} \mathrm{e}^{-\mathrm{Z}^\ell} \frac{(\mathrm{Z}^\ell)^x}{x!} h_\ell(x)\right] .\end{aligned}$$

Choosing $h_\ell(x) = t_\ell^x$, yields

$$\mathbb{E}\left[h(\mathrm{X}^1, \ldots, \mathrm{X}^L)\right] = \mathbb{E}\left[\prod_{\ell=1}^{L} \mathrm{e}^{(t_\ell - 1)\mathrm{Z}^\ell}\right] .$$

By setting $\mathbf{u} = \{t_\ell - 1\}_{1 \leqslant \ell \leqslant L}$, we obtain

$$\mathbb{E}\left[h(\mathrm{X}^1, \ldots, \mathrm{X}^L)\right] = \mathbb{E}\left[\mathrm{e}^{\mathbf{u}^\top \mathbf{Z}}\right] = \mathrm{M}_{\mathbf{Z}}(\mathbf{u}) .$$

The proof is concluded by the same arguments as in Lemma C.2.

C.2.2 Identifiability of parent-children distributions at the first layer

Lemma C.3. *Let $\mathbf{Z} = (\mathrm{Z}^1, \mathbf{Z}^2)$ be random variables such that $\mathrm{Z}^1 > 0$, $\mathbf{Z}^2 \in \mathcal{S}^K$, where $\mathcal{S}^K$ denotes the simplex in $\mathbb{R}^K$. Assume that the observations $\mathbf{X} = (\mathrm{X}^1, \mathbf{X}^2)$ are such that conditionally on Z^1, $\mathrm{X}^1 \sim \mathcal{P}(\mathrm{Z}^1)$ and conditionally on $(\mathrm{X}^1, \mathbf{Z}^2)$, $\mathbf{X}^2 \sim \mathcal{M}\left(\mathrm{X}^1, \mathbf{Z}^2\right)$. Then, the law of $(\mathrm{Z}^1, \mathbf{Z}^2)$ is identifiable from the law of $(\mathrm{X}^1, \mathbf{X}^2)$.*

Proof. Let h be a measurable function. For all $x_1 \geqslant 1$, let $x^1\mathcal{S}^K = \{(x^2_1, \dots, x^2_K) \in \mathbb{R}^K \mid \sum_{k=1}^K x^2_k = x^1\}$, then

$$\begin{aligned}\mathbb{E}\left[h(\mathrm{X}^1, \mathbf{X}^2)\right] &= \mathbb{E}\left[\mathbb{E}\left[h(\mathrm{X}^1, \mathbf{X}^2) \mid \mathbf{Z}\right]\right] \\ &= \mathbb{E}\left[\sum_{x^1 \in \mathbb{N}} \sum_{\mathbf{x}^2 \in x^1\mathcal{S}^K} \mathrm{e}^{-\mathrm{Z}^1} \prod_{k=1}^K \frac{(\mathrm{Z}^1\mathrm{Z}^2_k)^{x^2_k}}{x^2_k!} h(x^1, \mathbf{x}^2)\right] .\end{aligned}$$

Using that $\mathbf{Z}^2$ lies in the simplex yields

$$\mathbb{E}\left[h(\mathrm{X}^1, \mathbf{X}^2)\right] = \mathbb{E}\left[\sum_{\mathbf{x}^2 \in \mathbb{N}^K} \prod_{k=1}^K \mathrm{e}^{-\mathrm{Z}^1\mathrm{Z}^2_k} \frac{(\mathrm{Z}^1\mathrm{Z}^2_k)^{x^2_k}}{x^2_k!} h\left(\sum_k^K x^2_k, \mathbf{x}^2\right)\right] .$$

Therefore, $(\mathbf{X}^2_1, \dots, \mathbf{X}^2_K)$ are conditionally independent with Poisson distribution with parameters $(\mathrm{Z}^1\mathrm{Z}^2_k)_{1 \leqslant k \leqslant K}$. Hence, by Lemma 3.3, the law of $(\mathrm{Z}^1\mathrm{Z}^2_1, \dots, \mathrm{Z}^1\mathrm{Z}^2_K)$ is identifiable. Since $\mathbf{Z}^2$ lies in the simplex, conditionally on $\mathbf{U} = \mathrm{Z}^1\mathbf{Z}^2$, Z^1 has a Dirac distribution with mass at $\sum_{k=1}^K \mathbf{U}_k$. Then, as the law of Z^1 is identifiable from the law of $\mathbf{X}^1$ by Lemma C.2, the law of $(\mathrm{Z}^1, \mathbf{Z}^2)$ is identifiable from the law of $(\mathrm{Z}^1, \mathrm{Z}^1\mathbf{Z}^2)$, which concludes the proof. □

C.2.3 Identifiability through softmax transform

Lemma C.4. *Let $\mathbf{Z}$, $\tilde{\mathbf{Z}}$ be two random variables in $\mathbb{R}^d$. Define $\mathbf{P} = \mathbf{I}_d - \mathbf{1}_{d\times d}/d$ the projector on $\mathrm{Vect}\,(\mathbf{1}_d)^\perp$. Then, if $\sigma(\mathbf{Z})$ and $\sigma(\tilde{\mathbf{Z}})$ have the same distribution, $\mathbf{P}\mathbf{Z}$ and $\mathbf{P}\tilde{\mathbf{Z}}$ have the same distribution and conversely.*

Proof. We start with the direct sense of the equivalence. Let $B \in \mathcal{B}(\mathcal{S}^d)$, since $\sigma(\cdot)$ is surjective on $\mathcal{S}^d$ there exists $C \in \mathbb{R}^d$ such that $\sigma(C) = B$. Then, assuming $\sigma(\mathbf{Z})$ has the same law as $\sigma(\tilde{\mathbf{Z}})$,

$$\mathbb{P}(\sigma(\mathbf{Z}) \in B) = \mathbb{P}(\sigma(\tilde{\mathbf{Z}}) \in B) ,$$

so that

$$\mathbb{P}(\sigma(\mathbf{Z}) \in \sigma(C)) = \mathbb{P}(\sigma(\tilde{\mathbf{Z}}) \in \sigma(C)) .$$

On the event $\{\sigma(\mathbf{Z}) \in \sigma(C)\}$, there exists $\boldsymbol{c} \in C$ such that $\sigma(\mathbf{Z}) = \sigma(\boldsymbol{c})$, which yields

$$\mathbf{Z} = \boldsymbol{c} + K(\boldsymbol{c}, \mathbf{Z})\mathbf{1}_d ,$$

with $K(\boldsymbol{c}, \mathbf{Z}) = \log(\sum_{k=1}^d \mathrm{e}^{\mathbf{Z}_k} / \sum_{k=1}^d \mathrm{e}^{\boldsymbol{c}_k})$. Since $\mathbf{P}$ is the projector on $\mathrm{Vect}\,(\mathbf{1}_d)^\perp$, we have $\mathbf{P}\mathbf{1}_d = 0$, which yields $\mathbf{P}\mathbf{Z} = \mathbf{P}\boldsymbol{c} \in \mathbf{P}C$, the projection of C on $\mathrm{Vect}\,(\mathbf{1}_d)^\perp$ and therefore $\{\sigma(\mathbf{Z}) \in \sigma(C)\} \subset \{\mathbf{P}\mathbf{Z} \in \mathbf{P}C\}$. We obtain similarly $\{\mathbf{P}\mathbf{Z} \in \mathbf{P}C\} \subset \{\sigma(\mathbf{Z}) \in \sigma(C)\}$ so that

$$\mathbb{P}(\mathbf{P}\mathbf{Z} \in \mathbf{P}C) = \mathbb{P}(\sigma(\mathbf{Z}) \in \sigma(C)) = \mathbb{P}(\sigma(\tilde{\mathbf{Z}}) \in \sigma(C)) = \mathbb{P}(\mathbf{P}\tilde{\mathbf{Z}} \in \mathbf{P}C) ,$$

which concludes the direct sense of the equivalence. The converse statement is obtained similarly. □

C.2.4 Proof Corollary 3.5

Since conditionally to $\mathbf{Z}^1$ (resp. $\tilde{\mathbf{Z}}^1$), $\mathbf{Z}^2$ (resp. $\tilde{\mathbf{Z}}^2$) is Gaussian, observing that $\mathbf{P} = \mathbf{P}^\top$, the law of $\mathbf{P}\mathbf{Z}^2$ (resp. $\mathbf{P}\tilde{\mathbf{Z}}^2$) is given by $\mathcal{N}(\mathbf{P}\boldsymbol{\mu}(\mathbf{Z}^1), \mathbf{P}\boldsymbol{\Sigma}(\mathbf{Z}^1)\mathbf{P})$ (resp. $\mathcal{N}(\mathbf{P}\tilde{\boldsymbol{\mu}}(\tilde{\mathbf{Z}}^1), \mathbf{P}\tilde{\boldsymbol{\Sigma}}(\tilde{\mathbf{Z}}^1)\mathbf{P})$), which concludes the proof.

C.2.5 Proof of Theorem 3.4

By Lemma 3.3 and Lemma C.3 we obtain the identifiability of the Poisson layer and the identifiability of all parent-children distributions between the Poisson layer and the second Multinomial one. By conditional independence of the group of children conditionally to their respective latent variables and their parents, we only have to show the identifiability of any parent-children distributions for $\ell \geqslant 2$. To represent the tree compositionality constraint, we denote the events $\{\mathrm{X}_k^2 = \sum_{j \in \mathcal{C}_k^2} \mathrm{X}_j^3\}_{k \leqslant K_2}$ by $\{\mathbf{X}^2 = \widehat{\mathbf{X}}^3\}$. The joint distribution then writes

$$
\begin{aligned}
p(\mathbf{X}^2, \mathbf{X}^3 | \mathrm{X}^1, \mathbf{Z}^2, \mathbf{Z}^3) &= \mathbb{1}_{\mathbf{X}^2 = \hat{\mathbf{X}}^3} p(\mathbf{X}^2 | \mathbf{Z}^2, \mathrm{X}^1) \prod_{k=1}^{K_2} p(\check{\mathbf{X}}_k^2 | \check{\mathbf{Z}}_k^2, \mathrm{X}_k^2) \\
&= \mathbb{1}_{\mathbf{X}^2 = \hat{\mathbf{X}}^3} \frac{\mathrm{X}^1!}{\prod_{k=1}^{K_2} \prod_{j \in \mathcal{C}_k^2} \mathrm{X}_j^3!} \left[\prod_{k=1}^{K_2} (\mathrm{Z}_k^2)^{\mathrm{X}_k^2} \right] \cdot \left[\prod_{k=1}^{K_2} \prod_{j \in \mathcal{C}_k^2} (\mathrm{Z}_j^3)^{\mathrm{X}_j^3} \right] .
\end{aligned}
$$

Using that $\{\mathcal{C}_k^2\}_{k \leqslant K_2}$ is a partition of $\{1, \dots, K_3\}$ yields

$$
p(\mathbf{X}^2, \mathbf{X}^3 \mid \mathrm{X}^1, \mathbf{Z}^2, \mathbf{Z}^3) = \mathbb{1}_{\mathbf{X}^2 = \hat{\mathbf{X}}^3} \frac{\mathrm{X}^1!}{\prod_{k=1}^{K_3} \mathrm{X}_k^3!} \left[\prod_{k=1}^{K_2} (\mathrm{Z}_k^2)^{\mathrm{X}_k^2} \right] \cdot \left[\prod_{k=1}^{K_3} (\mathrm{Z}_k^3)^{\mathrm{X}_k^3} \right] .
$$

For all $k \leqslant K_2$, the compositionality constraint yields

$$
\prod_{k=1}^{K_2} (\mathrm{Z}_k^2)^{\mathrm{X}_k^2} = \prod_{k=1}^{K_2} \prod_{j \in \mathcal{C}_k^2} (\mathrm{Z}_k^2)^{\mathrm{X}_j^3} = \prod_{k=1}^{K_3} (\widehat{\mathrm{Z}}_k^3)^{\mathrm{X}_k^3} ,
$$

and therefore

$$
p(\mathbf{X}^2, \mathbf{X}^3 \mid \mathrm{X}^1, \mathbf{Z}^2, \mathbf{Z}^3) = \mathbb{1}_{\mathbf{X}^2 = \hat{\mathbf{X}}^3} \frac{\mathrm{X}^1!}{\prod_{k=1}^{K_3} \mathrm{X}_k^3} \prod_{k=1}^{K_3} (\mathrm{Z}_k^3 \widehat{\mathrm{Z}}_k^3)^{\mathrm{X}_k^3} ,
$$

yielding that the conditional distribution of $\mathbf{X}^3$ is multinomial. Hence, by Lemma C.2.2 the law of $(\mathrm{Z}^1, (\mathrm{Z}_k^3 \widehat{\mathrm{Z}}_k^3)_{k \leqslant K_3})$ is identifiable, or equivalently $(\mathrm{Z}^1, (\mathrm{Z}_k^2 \check{\mathbf{Z}}_k^2)_{k \leqslant K_2})$ is identifiable. Since for all $1 \leqslant k \leqslant K_2$, $\check{\mathbf{Z}}_k^2$ lies in the simplex, using the same argument as for the proof of Lemma C.2.2 enables us to identify the law of $(\mathrm{Z}^1, (\mathrm{Z}_k^2, \check{\mathbf{Z}}_k^2)_{k \leqslant K_2})$, which concludes the proof.

D Experimental setup

Latent prior architectures The latent prior is a Markov chain made of Gaussian transition kernels parameterized by neural networks, such that at layer $1 < \ell \leqslant L$, the mean $\boldsymbol{\mu}_{\boldsymbol{\theta}^\ell}(.) \in \mathbb{R}^{K_\ell}$ and precision matrix

$\boldsymbol{\Omega}_{\boldsymbol{\theta}^\ell}(.) \in \mathbb{R}^{K_\ell \times K_\ell}$ use $\mathbf{Z}^{\ell-1} \in \mathbb{R}^{K_{\ell-1}}$ as input. In our experiments, the mean and covariance share the same network architecture but consist of two separate networks. At layer $1 < \ell \leqslant L$, we fix the number of neurons in all hidden layers to $K_{\ell-1}$, and only adjust the number of hidden layers. Finally, we initialize the parameters of the first layers based on PLN initialization such that for all $k \leqslant K_1$,

$$\boldsymbol{\mu}_{1,k} = \frac{1}{n} \sum_{i=1}^{n} \log \mathrm{X}_{ik}^{1}$$

and

$$\boldsymbol{\Sigma}_1 = \frac{1}{n-1} (\log \mathbf{X}^1 - \mathbf{1}_{n \times K_1} \boldsymbol{\mu}_1)^\top (\log \mathbf{X}^1 - \mathbf{1}_{n \times K_1} \boldsymbol{\mu}_1) \ ,$$

the other parameters are initialized at random.

Mean-field architectures In the mean-field approximation, the parametrization of the Gaussian kernels at layer $\ell \leqslant L$ is made of two neural networks with inputs $\mathbf{X}$. In our experiments, the input of the networks at layer ℓ is limited to $\mathbf{X}^\ell$ (see Blei et al. (2017)). At layer ℓ, the mean $\boldsymbol{m}_{\boldsymbol{\varphi}^\ell}(\mathbf{X}^\ell)$ and the diagonal covariance matrix $\mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{X}^\ell)$ have the same network architecture but consists of two different fully connected neural networks with output of dimension K_ℓ. In our experiments, the architecture of the networks is solely parameterized by the number of hidden layers, while the number of neurons at each hidden layer is fixed to K_ℓ at depth ℓ of the tree.

Backward Markov architectures The backward variational approximation is a backward Markov chain with Gaussian transition kernels, such that at layer L the mean and diagonal covariance matrix use $\mathbf{X}^{1:L}$ as inputs, and for layer $\ell < L$, the mean and diagonal covariance matrix use $(\mathbf{X}^{1:\ell}, \mathbf{Z}^{\ell+1})$ as inputs (see H2). Due to the computational burdens of the chain $\mathbf{X}^{1:\ell}$, Chagneux et al. (2024) suggest performing amortized inference by encoding the chain using a recurrent neural network architecture into $\mathbf{E}^{1:\ell}$. Consequently, the backward architecture consists of an embedding block common to all layers, and for each layer $1 \leqslant \ell < L$ a fully connected network for each parameter of the Gaussian taking as input $\mathbf{E}^{1:\ell}$ and $\mathbf{Z}^{\ell+1}$. In our experiments, we define the embedder using a GRU or LSTM from the PyTorch library (Paszke et al., 2019), and we design the fully connected network at each layers by their number of hidden layers solely, fixing the intermediate hidden neurons to the input size.

Model optimization and numerical considerations The computation of the ELBO presents several numerical challenges that arise due to the need for exponentiation of parameters and inversion of the precision matrix. To mitigate issues related to numerical overflow, we impose constraints on the variational parameters. Specifically, we restrict the means to the interval $[-100, 25]$ and the variance terms to $[10^{-8}, 10]$. Additionally, to ensure the invertibility of the considered matrices, we introduce a bias of $\lambda = 10^{-4}$ to the diagonal. Subsequently, we opt to employ the Adam optimizer (Kingma and Ba, 2014) for training our neural networks with learning rate 10^{-3} using PyTorch implementation (Paszke et al., 2019). This choice is motivated by its demonstrated stability and efficacy, surpassing alternative optimization techniques in our experiments.

D.1 PLN-Tree generated data experiments

D.1.1 Model selection experiments

In this experiment, the latent prior optimal architecture is already known from the original model. Consequently, we only optimize the hyperparameters of the variational approximation. The training dataset consists of 2000 samples from a PLN-Tree model. For each model, we sample 3000 samples 5 times and select the model with the best overall performances regarding the alpha diversity criteria.

Mean-field architectures We try 3 architectures of mean-field variational approximations, where the amount of hidden layers in the variation approximation spans in $\{1, 2, 3\}$. The results indicate the optimal architecture is given for 1 hidden layer.

Backward Markov architectures The tested architectures are summarized in Table 12. The performances of each architecture orientate the choice of the optimal architecture towards the Model 4.

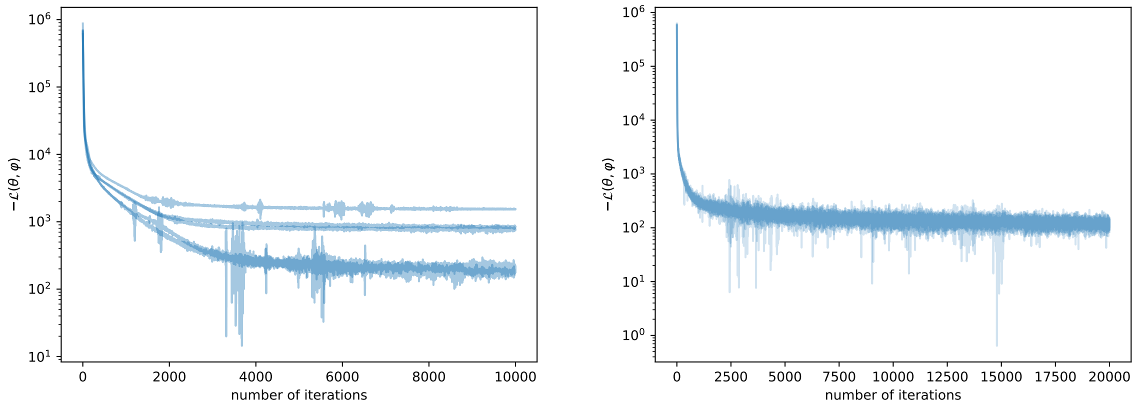
Parameter	**Model 1**	**Model 2**	**Model 3**	**Model 4**
Embedder type	GRU	GRU	GRU	GRU
Hidden layers size	32	32	32	32
Number of hidden layers	2	2	3	3
Embedding size	64	64	64	120
Number of layers (Gaussian parameters)	1	2	1	2

Table 12: Tested backward variational architectures in the PLN-Tree synthetic data experiments.

D.1.2 Performance benchmark

For each selected model, we perform multiple training runs and present the resulting objective values in Figure 5. We observe that the mean-field approximation does not converge to the same value of the ELBO value across different runs (see Figure 5a), indicating variability in performance. Conversely, our method consistently converges to the same ELBO values (see Figure 5b), demonstrating stable performance and consistently outperforming the mean-field approach. Thus, in all our experiments, we do not explore the training variability of the mean-field model, and only account for the sampling variability.

For the performance benchmark of the selected models, we sample 2000 samples 25 times for each model and show the average result with standard deviation between brackets.



(a) Mean-field approximation

(b) Residual amortized backward approximation

Figure 5: ELBO convergence over iterations for PLN-Tree models on the PLN-Tree generated dataset, repeated 5 times, performed for mean-field and residual amortized backward variational approximations. Negative values are eluded in log scale.

Alpha diversity	**PLN-Tree**	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
Wasserstein Distance $(\times 10^2)$				
Shannon $\ell = 1$	**1.57** (0.50)	11.23 (0.73)	14.64 (1.15)	1.40 (0.61)
Shannon $\ell = 2$	**3.67** (1.33)	5.14 (1.20)	32.04 (1.62)	26.28 (1.31)
Shannon $\ell = 3$	**5.82** (1.51)	7.86 (1.47)	35.03 (1.68)	34.03 (1.68)
Simpson $\ell = 1$	**0.62** (0.21)	2.69 (0.27)	4.91 (0.41)	0.53 (0.24)
Simpson $\ell = 2$	**0.71** (0.24)	1.40 (0.31)	7.35 (0.41)	6.05 (0.36)
Simpson $\ell = 3$	**0.85** (0.24)	1.55 (0.34)	7.21 (0.41)	7.21 (0.41)
Kolmogorov Smirnov $(\times 10^{-2})$				
Shannon $\ell = 1$	**2.60** (0.70)	14.69 (1.06)	11.0 (0.99)	2.59 (0.78)
Shannon $\ell = 2$	4.63 (1.29)	**4.42** (1.00)	20.68 (0.87)	18.32 (0.94)
Shannon $\ell = 3$	**5.34** (1.08)	5.54 (0.99)	20.2 (1.15)	20.2 (1.15)
Simpson $\ell = 1$	**2.65** (0.69)	11.14 (0.93)	9.87 (0.78)	2.67 (0.86)
Simpson $\ell = 2$	4.37 (1.24)	**4.18** (0.89)	19.14 (0.97)	16.96 (0.94)
Simpson $\ell = 3$	4.99 (0.92)	**4.58** (0.79)	18.92 (0.98)	18.92 (0.98)
Total variation $(\times 10^{-2})$				
Shannon $\ell = 1$	**1.14** (0.29)	5.67 (0.41)	4.41 (0.39)	1.16 (0.30)
Shannon $\ell = 2$	**1.21** (0.32)	1.24 (0.21)	5.34 (0.24)	4.76 (0.25)
Shannon $\ell = 3$	**1.19** (0.22)	1.31 (0.17)	4.32 (0.23)	4.32 (0.23)
Simpson $\ell = 1$	**3.13** (0.71)	10.84 (0.94)	11.38 (0.91)	3.31 (0.87)
Simpson $\ell = 2$	**4.29** (0.94)	4.66 (0.97)	19.53 (0.91)	17.17 (0.97)
Simpson $\ell = 3$	4.92 (0.73)	**4.48** (0.74)	18.63 (0.94)	18.63 (0.94)
Kullback-Leibler Divergence $(\times 10^{-2})$				
Shannon $\ell = 1$	**0.24** (0.11)	4.83 (0.50)	14.17 (1.22)	10.62 (0.82)
Shannon $\ell = 2$	**0.57** (0.26)	0.80 (0.23)	14.39 (1.15)	14.39 (1.15)
Shannon $\ell = 3$	**1.07** (0.36)	1.63 (0.45)	4.56 (0.50)	0.23 (0.13)
Simpson $\ell = 1$	**0.23** (0.11)	2.40 (0.31)	4.56 (0.50)	0.23 (0.13)
Simpson $\ell = 2$	**0.47** (0.19)	0.80 (0.28)	10.64 (0.91)	7.99 (0.74)
Simpson $\ell = 3$	**0.68** (0.20)	0.84 (0.26)	10.57 (0.87)	10.57 (0.87)

Table 13: Distribution metrics on alpha diversities computed between synthetic data sampled under the original PLN-Tree model and simulated data under each modeled trained, averaged over the trainings, with standard deviation.

D.2 Synthetic data with Markov Dirichlet experiments

D.2.1 Model selection experiments

Dataset description and selection procedure The training dataset consists of 2000 samples from a Markov Dirichlet model. For each model, when compared to this dataset, we sample 3000 samples 5 times and select the model with the best overall performances regarding the alpha diversity criteria.

Mean-field architectures In this experiment, the number of hidden layers in the latent priors spans in $\{1, 2, 3\}$, while the number of hidden layers in the mean-field approximations spans in $\{1, 2\}$. Trying all combinations, we obtain the best-performing architecture in our experiment has 2 hidden layers in the latent prior, and 1 hidden layer in the variational approximation parameters.

Backward Markov architectures For the backward architectures, the number of layers tested in the latent priors spans in $\{1, 2\}$. The various tested architectures for the embedders are summarized in Table 14. The architecture of the best-performing model is yielded for 1 layers in the latent prior with the embedding architecture E8.

Name	Embedding size	Hidden layers	Nb neurons
E1	16	2	32
E2	32	2	32
E3	32	3	32
E4	32	2	64
E5	32	3	64
E6	60	2	64
E7	60	3	64
E8	60	3	120

Table 14: Tested backward variational architectures in the Embedder in the Markov Dirichlet synthetic experiments. All embedders are GRU, stacked with a 2 layers neural network to model the parameters.

D.2.2 Performance benchmark

For the performance benchmark of the selected models, we sample 2000 samples 25 times for each model and show the average result with standard deviation between brackets.

Alpha diversity	**PLN-Tree**	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
Wasserstein Distance ($\times 10^2$)				
Shannon $\ell = 1$	**17.70** (0.47)	21.42 (0.59)	72.27 (1.70)	17.53 (0.59)
Shannon $\ell = 2$	**22.23** (0.94)	29.10 (1.06)	111.53 (1.81)	92.89 (1.92)
Shannon $\ell = 3$	**24.32** (0.83)	37.72 (1.14)	142.28 (1.99)	142.28 (1.99)
Simpson $\ell = 1$	**5.69** (0.16)	5.84 (0.16)	21.74 (0.60)	5.65 (0.22)
Simpson $\ell = 2$	**5.21** (0.17)	5.90 (0.19)	26.70 (0.59)	20.76 (0.64)
Simpson $\ell = 3$	**3.91** (0.11)	5.16 (0.16)	28.55 (0.59)	28.55 (0.59)
Kolmogorov Smirnov ($\times 10^2$)				
Shannon $\ell = 1$	**16.81** (0.93)	24.28 (0.9)	45.12 (1.02)	16.17 (0.91)
Shannon $\ell = 2$	**19.29** (1.06)	25.94 (0.97)	58.83 (0.88)	54.14 (0.55)
Shannon $\ell = 3$	**20.80** (0.75)	30.50 (0.98)	66.62 (0.71)	66.62 (0.71)
Simpson $\ell = 1$	**13.95** (0.94)	20.93 (0.9)	39.77 (1.10)	13.41 (0.89)
Simpson $\ell = 2$	**18.35** (1.03)	23.47 (0.97)	55.42 (0.87)	49.37 (0.64)
Simpson $\ell = 3$	**22.00** (0.87)	30.43 (0.82)	62.10 (0.71)	62.10 (0.71)
Total variation ($\times 10^2$)				
Shannon $\ell = 1$	**7.75** (0.29)	9.78 (0.35)	14.87 (0.33)	7.55 (0.33)
Shannon $\ell = 2$	**6.67** (0.30)	8.08 (0.28)	15.59 (0.21)	14.23 (0.17)
Shannon $\ell = 3$	**5.60** (0.16)	7.47 (0.24)	14.93 (0.14)	14.93 (0.14)
Simpson $\ell = 1$	**19.33** (0.68)	23.72 (1.02)	38.32 (1.02)	18.92 (0.84)
Simpson $\ell = 2$	**20.11** (0.89)	24.60 (1.01)	50.64 (0.79)	43.88 (0.73)
Simpson $\ell = 3$	**19.21** (0.64)	26.42 (0.96)	56.78 (0.66)	56.78 (0.66)
Kullback-Leilbler divergence ($\times 10^2$)				
Shannon $\ell = 1$	**20.72** (2.42)	23.72 (1.70)	60.51 (3.14)	19.21 (3.23)
Shannon $\ell = 2$	**28.77** (5.75)	33.04 (3.33)	153.73 (8.38)	120.30 (5.20)
Shannon $\ell = 3$	**25.02** (4.03)	40.96 (3.73)	226.13 (11.96)	226.13 (11.96)
Simpson $\ell = 1$	**15.21** (1.71)	15.75 (1.38)	39.04 (2.18)	14.15 (2.50)
Simpson $\ell = 2$	**26.26** (8.32)	26.84 (5.95)	81.47 (3.49)	57.97 (2.07)
Simpson $\ell = 3$	**21.68** (7.61)	29.71 (7.99)	106.28 (3.40)	106.28 (3.40)

Table 15: Distribution metrics on alpha diversities computed between synthetic data sampled under the Markov Dirichlet model and simulated data under each modeled trained, averaged over the trainings, with standard deviation.

D.3 Metagenomics dataset experiments

D.3.1 Model selection experiments

Selection procedure For each model, when compared to the metagenomics dataset, we sample 3000 samples 5 times and select the model with the best overall performances regarding the alpha diversity criteria.

Mean-field architectures We try all combinations of the number of hidden layers for the latent prior and the variational approximation taking values in $\{1, 2, 3\}$. The best-performing architecture in our experiment has 1 hidden layers in the latent prior, and 2 hidden layers in the variational approximation parameters.

Backward Markov architectures We decide on a grid of embedders summarized in Table 16, which we combine with latent prior architecture with a number of hidden layers in $\{1, 2, 3\}$. In our experiment, the best architecture is yielded by the embedding architecture E4.

Name	**Embedding size**	**Hidden layers**	**Nb neurons**	**Parameters layers**
E1	16	2	32	2
E2	32	2	32	2
E3	32	3	32	2
E4	32	2	64	2
E5	32	3	64	2
E6	32	3	64	3
E7	60	2	64	2
E8	60	3	64	2
E9	60	3	64	3
E10	60	3	120	2
E11	60	3	120	3

Table 16: Tested backward variational architectures in the Embedder in the metagenomics experiments. All embedders are GRU.

Alpha diversity	**PLN-Tree**	**PLN-Tree (MF)**	**PLN (fill)**	**PLN**
Wasserstein distance $(\times 10^2)$				
Shannon $\ell = 1$	**1.73** (0.44)	3.00 (0.44)	16.49 (1.14)	4.00 (0.50)
Shannon $\ell = 2$	**2.22** (0.73)	5.70 (0.97)	23.21 (1.64)	8.37 (1.21)
Shannon $\ell = 3$	**2.29** (0.63)	6.58 (1.02)	23.96 (1.67)	9.28 (1.43)
Shannon $\ell = 4$	**2.08** (0.62)	20.39 (1.08)	55.32 (2.38)	55.32 (2.38)
Simpson $\ell = 1$	0.84 (0.14)	**0.71** (0.12)	7.18 (0.48)	1.71 (0.23)
Simpson $\ell = 2$	0.92 (0.24)	**0.73** (0.19)	7.49 (0.57)	2.55 (0.33)
Simpson $\ell = 3$	0.91 (0.23)	**0.72** (0.19)	7.46 (0.57)	2.88 (0.29)
Simpson $\ell = 4$	**0.53** (0.13)	2.41 (0.21)	12.91 (0.67)	12.91 (0.67)
Kolmogorov Smirnov $(\times 10^2)$				
Shannon $\ell = 1$	**4.71** (1.44)	8.4 (1.35)	23.17 (1.26)	7.75 (1.07)
Shannon $\ell = 2$	**3.42** (0.99)	10.3 (1.25)	22.14 (1.58)	9.35 (1.14)
Shannon $\ell = 3$	**3.48** (0.68)	10.66 (1.32)	22.07 (1.47)	10.30 (1.00)
Shannon $\ell = 4$	**3.64** (1.06)	22.66 (1.3)	36.65 (1.50)	36.65 (1.50)
Simpson $\ell = 1$	4.8 (0.93)	**4.17** (0.58)	21.25 (1.47)	7.19 (1.20)
Simpson $\ell = 2$	**4.46** (1.06)	5.6 (1.46)	19.64 (1.45)	8.15 (1.11)
Simpson $\ell = 3$	**4.17** (1.03)	5.7 (1.53)	19.53 (1.42)	8.79 (0.90)
Simpson $\ell = 4$	**4.09** (1.06)	12.26 (1.46)	32.07 (1.69)	32.07 (1.69)
Total variation $(\times 10^2)$				
Shannon $\ell = 1$	**2.34** (0.63)	4.51 (0.75)	10.00 (0.63)	3.17 (0.38)
Shannon $\ell = 2$	**1.42** (0.34)	3.87 (0.52)	7.47 (0.63)	3.00 (0.31)
Shannon $\ell = 3$	**1.36** (0.24)	3.81 (0.48)	7.19 (0.59)	3.18 (0.36)
Shannon $\ell = 4$	**0.82** (0.27)	6.3 (0.42)	8.94 (0.38)	8.94 (0.38)
Simpson $\ell = 1$	**6.71** (1.18)	8.17 (1.8)	27.59 (1.80)	9.09 (1.07)
Simpson $\ell = 2$	**5.51** (0.89)	7.41 (1.61)	22.23 (1.60)	10.43 (1.32)
Simpson $\ell = 3$	**5.63** (0.93)	7.35 (1.64)	21.83 (1.56)	12.21 (1.16)
Simpson $\ell = 4$	**3.75** (1.15)	14.88 (1.72)	34.56 (1.58)	34.56 (1.58)
Kullback-Leilbler divergence $(\times 10^2)$				
Shannon $\ell = 1$	**0.88** (0.32)	2.32 (0.82)	15.98 (1.33)	2.14 (0.42)
Shannon $\ell = 2$	**0.86** (0.32)	2.68 (0.71)	15.33 (1.59)	4.08 (0.55)
Shannon $\ell = 3$	**0.71** (0.29)	2.87 (0.75)	15.30 (1.57)	4.64 (0.56)
Shannon $\ell = 4$	**0.57** (0.22)	18.02 (4.71)	35.01 (2.64)	35.01 (2.64)
Simpson $\ell = 1$	**1.04** (0.33)	1.54 (0.68)	15.57 (1.40)	2.26 (0.40)
Simpson $\ell = 2$	**0.96** (0.28)	1.14 (0.50)	13.61 (1.42)	3.70 (0.59)
Simpson $\ell = 3$	**0.97** (0.32)	1.11 (0.51)	13.58 (1.41)	4.56 (0.58)
Simpson $\ell = 4$	**0.81** (0.30)	9.70 (3.48)	29.66 (2.38)	29.66 (2.38)

Table 17: Distribution metrics on alpha diversities computed between metagenomics data and simulated data under each modeled trained, averaged over the trainings, with standard deviation.

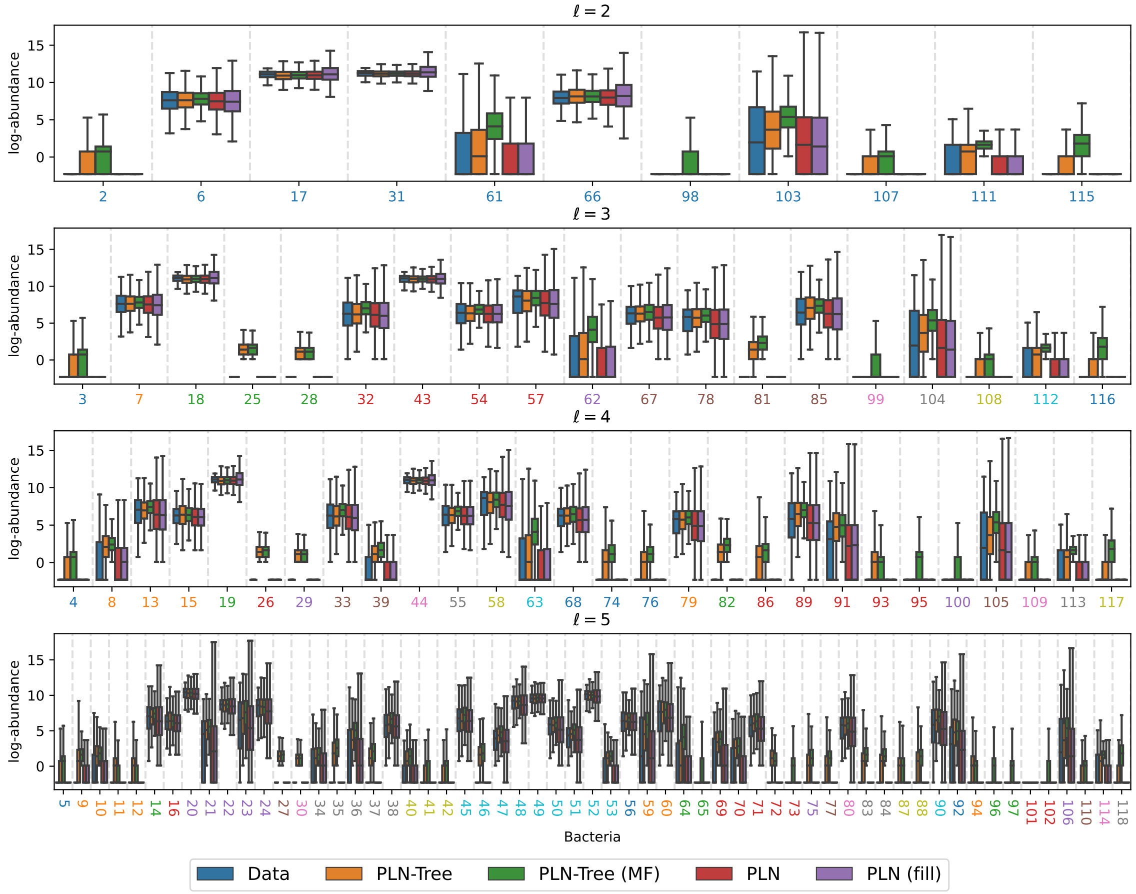


Figure 6: Boxplot of log abundances of the metagenomics dataset and generated data from several PLN-based models learned on this dataset, with 20000 points per model. Zero abundances are artificially shifted to 10^{-1} to represent them in log scale. The bacteria are denoted by a unique integer on the x-axis, with colors indicating the brotherhoods in the taxonomic tree at a given depth.

D.3.2 Classification using PLN-based preprocessing

For the classification benchmark on the metagenomics dataset, we consider four different types of inputs for various classifiers: the raw data, the PLN latent variables, the backward PLN-Tree latent variables, and the corresponding mean-field variant. Using the same taxa-abundance data as in the previous experiment, we adopt the PLN-Tree architectures selected from our prior model selection on the metagenomics dataset. We then proceed to the training of each model on the entire dataset, then proceed to encode the taxa-abundance data into respective latent variables. We then select various classifiers (see Table 18) for which the unspecified hyperparameters are selected from default Scikit-Learn proposals (Pedregosa et al., 2011). In this experiment, we only consider the deepest layer of the input data

Model	**Parameters**
Logistic Regression	class weight: balanced
SVC	probability: true, kernel: linear, C: 0.1, class weight: balanced
MLP	hidden layers sizes: 256, 256, 124
Random Forests	number of estimators: 100, class weight: balanced

Table 18: Considered classifiers in the metagenomics preprocessing experiment, with hyperparameters based on Scikit-Learn implementation.

To further illustrate the impact of the preprocessing on the classifiers' performances, we study the IBD-vs-all problem in addition to the T2D-vs-all presented in the article. The results are presented in Table 19, demonstrating similar interpretations to what is observed in the T2D-vs-all problem.

	Raw data	**log-LTC (PLN-Tree)**	**log-LTC (MF)**	**PLN**
Logistic Regression				
Balanced Accuracy	0.673 (0.043)	**0.767** (0.039)	0.713 (0.046)	0.746 (0.036)
Precision	0.770 (0.027)	**0.827** (0.024)	0.795 (0.029)	0.815 (0.022)
Recall	0.680 (0.03)	**0.777** (0.026)	0.716 (0.038)	0.765 (0.029)
F1 score	0.705 (0.027)	**0.791** (0.023)	0.738 (0.034)	0.780 (0.026)
ROC AUC	0.735 (0.042)	**0.835** (0.037)	0.786 (0.037)	0.814 (0.031)
ROC Precision-Recall	0.410 (0.062)	**0.587** (0.073)	0.539 (0.076)	0.586 (0.064)
Linear SVM				
Balanced Accuracy	0.573 (0.055)	**0.760** (0.043)	0.705 (0.044)	0.743 (0.033)
Precision	0.765 (0.113)	**0.823** (0.026)	0.791 (0.028)	0.813 (0.02)
Recall	0.390 (0.132)	**0.767** (0.031)	0.701 (0.038)	0.764 (0.028)
F1 score	0.361 (0.195)	**0.783** (0.028)	0.725 (0.033)	0.779 (0.025)
ROC AUC	0.416 (0.186)	**0.834** (0.037)	0.783 (0.036)	0.821 (0.03)
ROC Precision-Recall	0.215 (0.103)	0.589 (0.080)	0.534 (0.069)	**0.610** (0.065)
Neural Network				
Balanced Accuracy	0.726 (0.044)	**0.748** (0.042)	0.704 (0.051)	0.739 (0.045)
Precision	0.826 (0.026)	**0.838** (0.026)	0.812 (0.028)	0.837 (0.026)
Recall	0.830 (0.024)	0.841 (0.025)	0.813 (0.034)	**0.844** (0.023)
F1 score	0.825 (0.023)	**0.838** (0.025)	0.807 (0.031)	**0.838** (0.025)
ROC AUC	0.839 (0.034)	**0.865** (0.030)	0.813 (0.037)	0.847 (0.035)
ROC Precision-Recall	0.607 (0.062)	**0.667** (0.070)	0.581 (0.066)	0.648 (0.073)
Random Forest				
Balanced Accuracy	**0.647** (0.035)	0.602 (0.033)	0.609 (0.034)	0.606 (0.031)
Precision	**0.857** (0.021)	0.817 (0.03)	0.802 (0.035)	0.838 (0.025)
Recall	**0.844** (0.016)	0.82 (0.015)	0.817 (0.017)	0.826 (0.014)
F1 score	**0.810** (0.024)	0.776 (0.024)	0.779 (0.025)	0.781 (0.024)
ROC AUC	**0.917** (0.021)	0.867 (0.028)	0.815 (0.037)	0.888 (0.022)
ROC Precision-Recall	**0.794** (0.050)	0.655 (0.071)	0.579 (0.078)	0.730 (0.052)

Table 19: Classification IBD-vs-all performances for several classifiers on the metagenomics dataset using different preprocessing strategies, averaged over training, with standard deviation.

E Additional experiments visualisations

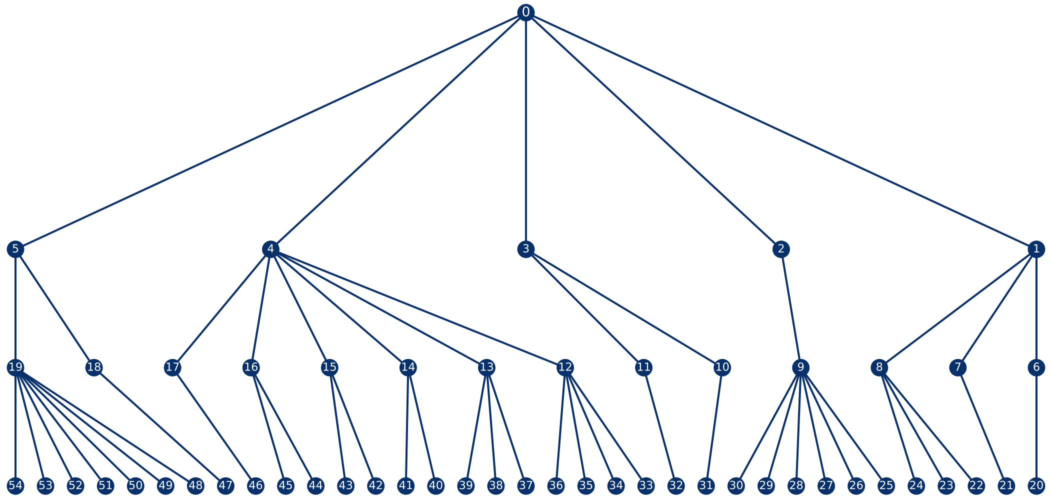


Figure 7: Graph of the tree considered in the PLN-Tree synthetic experiments.

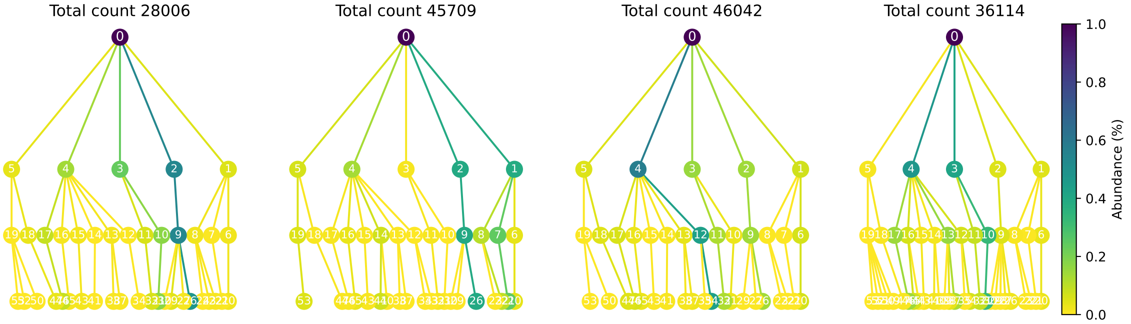


Figure 8: Synthetic hierarchical samples from the artificial dataset $(\mathbf{X}, \mathbf{Z})$.

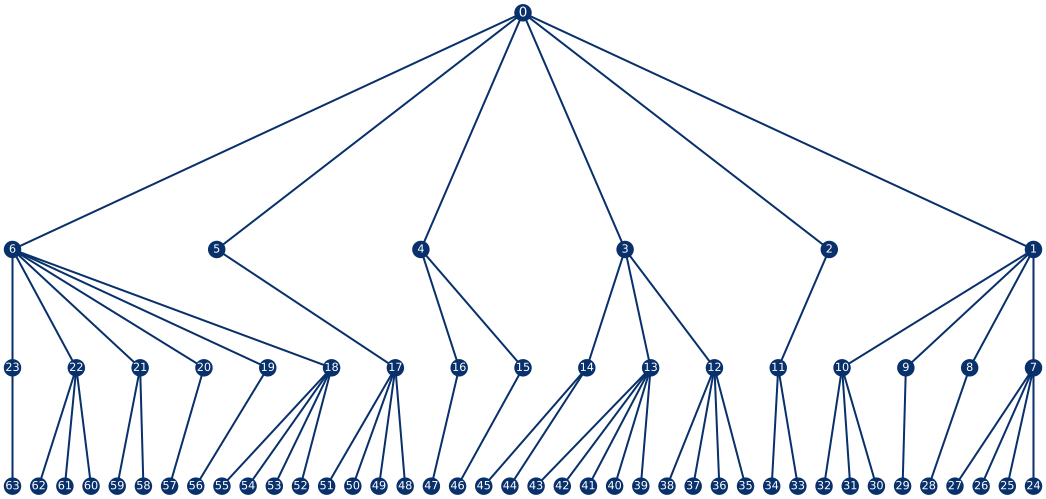


Figure 9: Graph of the tree considered in the Markov Dirichlet synthetic experiments.