Tree-based variational inference for Poisson log-normal models

Alexandre Chaussard, Anna Bonnet, Elisabeth Gassiat, Sylvain Le Corff

Tree-based variational inference for Poisson log-normal models

Alexandre Chaussard[1*], Anna Bonnet[1], Elisabeth Gassiat[2], Sylvain Le Corff[1]

[1*]LPSM, Sorbonne Université.
[2]LMO, Université Paris-Saclay.

*Corresponding author(s). E-mail(s): alexandre.chaussard@sorbonne-universite.fr;
Contributing authors: anna.bonnet@sorbonne-universite.fr;
elisabeth.gassiat@universite-paris-saclay.fr; sylvain.le_corff@sorbonne-universite.fr;

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, enhancing both theoretical foundations and practical interpretability. Additionally, we extend our framework to classification tasks as a preprocessing pipeline for compositional data, showcasing its versatility. Experimental evaluations on synthetic datasets as well as real-world microbiome data 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.

Keywords: Hierarchical count data, Poisson lognormal, Structured variational inference, Deep generative models, Identifiability, Microbiome

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 standard, particularly after the work of Chiquet et al. (2021) which led to significant theoretical and methodological developments for count data interaction-based modeling.

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 often 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 via maximum likelihood estimation is intractable, but this challenge can be circumvented using 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 modeling flexibility and scalability, we opt for 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 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. Furthermore, by leveraging the identifiable features of the model, we suggest an interpretable transform of the counts towards the latent space showing competitive performances when used as a preprocessing tool and illustrating the versatility of the PLN-Tree framework.

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 other state-of-art interaction-based count data models like SPiEC-Easi (Kurtz et al., 2015) and PLN. We namely demonstrate the practical utility of

PLN-Tree models through a generative 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 in Section 4.2.2. Our implementation and experiments are freely available on our GitHub[1].

2 Background

2.1 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 $\ell \leqslant L$, the random variable associated with node $k \leqslant K_\ell$ is denoted by V_k^ℓ. For layer $\ell \leqslant L-1$ and node $k \leqslant K_\ell$, the vector of children of the random variable V_k^ℓ is indexed by $\mathcal{C}_k^\ell$ and represented as $\check{\mathrm{V}}_k^\ell = (\mathrm{V}_j^{\ell+1})_{j \in \mathcal{C}_k^\ell}$. 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 $p_{\boldsymbol{\theta}}^{\mathrm{V}}$. In cases of clarity, we may express the density as $p_{\boldsymbol{\theta}}(\mathrm{V})$. 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 $\mathbb{R}^d$, $d > 0$, the k-th coordinate of any of its outputs is denoted by $f_{\boldsymbol{\theta},k}(.)$. The sequence of random variables $(\mathrm{V}^1, \ldots, \mathrm{V}^L)$ is represented as $\mathrm{V}^{1:L}$. For $\mathbf{V} \in \mathbb{R}^d$, the exponential of $\mathbf{V}$ is defined as $\exp(\mathbf{V}) = (\exp(\mathrm{V}_j))_{1 \leqslant j \leqslant d}$, and the multivariate Poisson distribution with parameters $\mathbf{V} \in \mathbb{R}^d_{>0}$ is denoted by $\mathcal{P}(\mathbf{V}) = \otimes_{j=1}^d \mathcal{P}(\mathrm{V}_j)$. We denote by $\mathcal{S}^d$ the simplex of dimension d, then if $\mathbf{V} \in \mathcal{S}^d$, we denote the multinomial distribution with total count n and probabilities $\mathbf{V}$ by $\mathcal{M}(n, \mathbf{V})$. Finally,

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

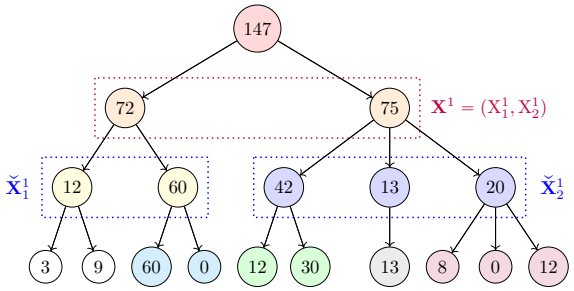


Fig. 1: Example of a hierarchical count data with $L = 4$. Nodes of the same color are independent of the other nodes conditionally to their parent node and their respective latent variables.

for $\mathbf{V} \in \mathbb{R}^d$ we denote its projection on the simplex through the softmax transform by $\sigma(\mathbf{V}) = (\mathrm{e}^{\mathrm{V}_i} / \sum_{j=1}^d \mathrm{e}^{\mathrm{V}_j})_{1 \leqslant i \leqslant d}$.

2.2 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:

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

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

2.3 Variational inference for structured data

As underscored in the previous section, addressing the parameter inference problem for PLN models can be achieved by leveraging variational inference techniques, which requires choosing a variational family.

In scenarios devoid of specific structural constraints, 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, they suggest 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

Tree compositionality constraint

Hierarchical count data are generated through the repeated aggregation of counts from the deepest-level entities in the hierarchy, moving from the bottom to the top layer of the tree. Formally, this process involves placing the observed counts at the deepest level of the tree, then summing these counts with their respective siblings to compute the counts at their parent nodes, continuing this process recursively up to the root layer. This construction induces the following tree compositionality constraint

$$\forall \ell < L, \forall k \leqslant K_\ell, \quad \mathrm{X}_k^\ell = \sum_{j \in \mathcal{C}_\ell^k} \mathrm{X}_j^{\ell+1} \,, \qquad (1)$$

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 approach would rely solely on the final layer to determine the entire hierarchical count data, thus failing to incorporate the tree structure in the modelization.

PLN-Tree 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 (1) since it models independent layers. Consequently, we propose a new model tailored to hierarchical structures named PLN-Tree.

- 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 up to $\ell < L$ writes

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

- 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 $\sigma(.)$ is the softmax transform. Formally, the observed counts process up to $\ell < L$ writes

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

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 $1 \leqslant \ell < L$ is the conditional distribution of independent Poisson random variables with parameters $\exp(\check{\mathbf{Z}}_k^\ell)$ conditioned on the event $\{\sum_{j \in \mathcal{C}_k^\ell} \mathrm{X}_j^{\ell+1} = \mathrm{X}_k^\ell\}$.

Interaction networks modeling

The latent Gaussian process in PLN-Tree captures interactions between entities at each layer through the covariance matrices of the Markov chain. In the simplest case, a diagonal covariance matrix at a given layer suggests no interaction between entities at that layer, conditionally on the previous latent variables. Notably, if all covariance matrices are diagonal except for the final one, then the hierarchical structure is essentially useless for modeling interactions within the ecosystem. In contrast, block-diagonal covariance matrices indicate the presence of clusters at each layer conditionally on the preceding latent variables. For instance, if the covariance matrix at a specific layer is block-diagonal with blocks corresponding to clades in the hierarchy, this implies that the hierarchy reflects an interaction-based clustering at that level. Further exploration is possible by analyzing the structure of the precision matrices (inverse of covariance), which can also provide insights into interaction patterns. Thankfully, by parameterizing PLN-Tree with neural networks, this framework allows for a wide range of architectural choices such as low-rank, block-diagonal, or sparse covariance structures, enabling the model to explore various interaction scenarios within the data.

Variational inference

Under the PLN-Tree model, 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. The variational density is given by a backward Gaussian Markov Chain:

$$q_{\boldsymbol{\varphi},1:L}(\mathbf{Z}|\mathbf{X}) = q_{\boldsymbol{\varphi},L}(\mathbf{Z}^L|\mathbf{X}^{1:L}) \times \prod_{\ell=1}^{L-1} q_{\boldsymbol{\varphi},\ell|\ell+1}(\mathbf{Z}^\ell|\mathbf{Z}^{\ell+1},\mathbf{X}^{1:\ell}) \,, \tag{2}$$

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 $\mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1},\mathbf{X}^{1:\ell})$.

Using the backward variational approximation (2), we can compute the surrogate objective given by the ELBO of the PLN-Tree model, for which the complete derivation is provided in Appendix 3. Interestingly, the PLN-Tree ELBO shares similarities with a per-layer PLN ELBO, where the latent variables $(\mathbf{Z}^\ell)_{1\leqslant\ell\leqslant L}$ would be treated as independent across layers. However, PLN-Tree relaxes this independence assumption, incorporating Markov dependencies between layers. These dependencies are reflected in the ELBO, which is expressed only up to an expectation rather than in closed form. Additionally, the propagation of multinomial distributions across children groups introduces distinctive terms between the root layer ($\ell = 1$) and deeper layers, setting PLN-Tree apart from traditional PLN models. As a result, the PLN-Tree optimization objective exhibits a greater complexity than a layer-wise PLN.

Residual amortized architecture

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.

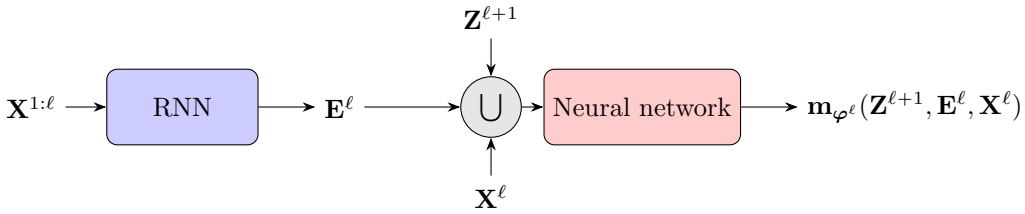


Fig. 2: Residual amortized backward architecture for the variational mean at layer $\ell \leqslant L$. The amortizing Recurrent Neural Network is denoted by RNN, while the symbol "$\cup$" indicates a concatenation of entries. The variable $\mathbf{E}^{\ell}$ is the last output of the recurrent network after inputting the sequence $\mathbf{X}^{1:\ell}$.

Partial closed-form optimization

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 $\boldsymbol{\theta}^1$ when $\boldsymbol{\varphi}^1$ is known, so that at iteration $h+1$,

$$\begin{aligned}
\boldsymbol{\mu}_1^{(h+1)} &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\varphi}}}\left[\mathbf{m}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{Z}^2, \mathbf{X}_i^1)\right], \\
\boldsymbol{\Sigma}_1^{(h+1)} &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\varphi}}}\Big[\left(\boldsymbol{\mu}_1^{(h+1)} - \mathbf{m}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{Z}^2, \mathbf{X}_i^1)\right) \\
&\quad \times \left(\boldsymbol{\mu}_1^{(h+1)} - \mathbf{m}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{Z}^2, \mathbf{X}_i^1)\right)^\top + \mathbf{S}_{\boldsymbol{\varphi}_1^{(h)}}(\mathbf{X}_i^{1:L})\Big].
\end{aligned} \tag{3}$$

The availability of these closed-form expressions is essential for practical model training, as they significantly accelerate the optimization of the ELBO and enable the learning of deeper layers in the model. Without these closed-form solutions, the learning process becomes prohibitively slow.

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). As a result, the offset often reflect sampling protocols rather than the ecological properties of the environments being studied, making them unreliable as direct features.

To mitigate these effects, preprocessing techniques such as resampling (rarefaction) can be applied to reduce the influence of variable sampling efforts, albeit with some loss of data (Weinroth et al., 2022; Schloss, 2024). An alternative approach is to model the offset directly within the statistical framework to avoid introducing spurious correlations (Chiquet et al., 2019). In the PLN models Chiquet et al. (2021), the offset is handled via a plug-in estimator that shifts the latent variable means based on the log of the total count in each sample. Extending this idea, we propose modeling the offset as a latent variable following a Gaussian Mixture in the PLN-Tree framework. This formulation captures variability in sampling efforts both across different groups of samples and within groups, resolving the need for domain-specific assumptions. The flexibility of this approach comes with the introduction of an hyperparameter (the number of mixture components), which allows users to tailor the model to different offset scenarios but increases the complexity of parameter estimation during training. Interestingly, 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, restricting its usage to the root layer. Details on the suggested variational approximation and the associated ELBO for PLN-Tree models with offset modeling can be found in Appendix B.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 1 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 1. *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 1 establishes the identifiability of the PLN-Tree model up to a softmax transform, which is illustrated in Section 4.1.

Theorem 1. *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 $(\mathbf{Z}^1, \ldots, \mathbf{Z}^L)$ is not a given if we do not set a constraint on the parameters space. Combining the previous result with Lemma 8 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 2. *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*

$$\mathbf{P}\boldsymbol{\mu}(\mathbf{z}) = \mathbf{P}\tilde{\boldsymbol{\mu}}(\mathbf{z}) \quad \text{and} \quad \mathbf{P}\boldsymbol{\Sigma}(\mathbf{z})\mathbf{P} = \mathbf{P}\tilde{\boldsymbol{\Sigma}}(\mathbf{z})\mathbf{P}\ ,$$

$\mathbb{P}_{\mathbf{Z}^1} - a.s.$, *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 $\mathbf{P}^\ell = \mathrm{diag}(\{\mathbf{P}_k^\ell\}_{1\leqslant k\leqslant K_{\ell-1}})$ with

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

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

Using identifiable features as counts preprocessing

Using latent variables as inputs for machine learning tasks is a standard practice that can significantly improve performance. In the case of PLN-Tree, Theorem 1 suggests that the identifiable latent variables $(\mathbf{P}^\ell\mathbf{Z}^\ell)_{2\leqslant\ell\leqslant L}$ may provide meaningful representations. This encoding process moves the data from a constrained and discrete space to a real-valued hyperplane, making the latent features potentially more effective for tasks such as classification, PCA, or regression. However, it is difficult to directly associate a latent variable with a specific entity in the tree, rendering comparisons with the regular PLN impractical.

Based on this remark, we introduce a latent feature, referred to as the latent proportions (LP), which maps hierarchical count data to their latent

representation $\mathbf{V}$ such that:

$$\begin{aligned} \mathbf{V}^1 &= \sigma(\mathbf{Z}^1)\ , \\ \forall \ell < L, k \leqslant K_\ell, \quad \check{\mathbf{V}}_k^\ell &= \sigma(\check{\mathbf{Z}}_k^\ell) \times \mathrm{V}_k^\ell\ . \end{aligned} \tag{4}$$

Since the latent proportions are compositional in nature, they can be further transformed using standard log transforms commonly employed in compositional data analysis (Ibrahimi et al., 2023), such as the centered log-ratio (CLR) transform. By combining the LP with the CLR transform (LP-CLR), we can map the observed counts from their constrained compositional space into an unconstrained latent space, which can improve the performance of machine learning models. It can also serve as a foundation for estimating covariance matrices at different layers and for conducting network inference. Similarly, PLN features can benefit from the LP-CLR transform which sums up to projecting the latents on $\mathrm{Vect}\,(\mathbf{1})^\perp$ (Proj-PLN).

The proposed LP-CLR transform of PLN-Tree's features is benchmarked against Proj-PLN features and the CLR transform of the true proportions in Section 4.2.2.

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 the 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 non-hierarchical competitors in a fair setup. Finally, we assess the model performance in comparison with PLN and SPiEC-Easi (Kurtz et al., 2015) on real-life metagenomics data from microbiome samples of several disease-affected patients (Pasolli et al., 2016) from generative 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 benchmark it against other interaction-based count data models. However, state-of-art models like PLN Chiquet et al. (2021), SparCC Friedman and Alm (2012) or SPiEC-Easi Kurtz et al. (2015) are restricted to tabular data, allowing the modeling of only one layer of hierarchical count data at a time. Thankfully, by leveraging the hierarchical compositional constraint (1), tabular count data models can generate valid hierarchical count data by modeling only the last layer of the tree, which is usually the one at stake for practitioners. This generative procedure involves sampling the abundances of the last layer under a given model and then exploiting the compositional constraint to derive the values of the parent nodes, allowing us to obtain hierarchical count data that satisfies (1).

In our experiments, PLN baselines are computed using the *pyPLNmodels*[2] Python implementation from Chiquet et al. (2018). Conversely, SparCC and SPiEC-Easi were implemented within our package as generative models, as both methods usually only estimate the covariance and precision matrices of the log-centered ratio (CLR) transformation of compositional data. After estimating the mean of the normalized and CLR-transformed count data, we sample from the inferred Gaussian distribution and invert the CLR transformation using the softmax function, obtaining proportion data that can be used to generate count data via a multinomial distribution. Additionally, since our model does not involve sparsity, we set the sparsity parameter of the estimated matrices to 0 in both SparCC and SPiEC-Easi, making both models equivalent. Consequently, we only compare PLN-Tree to PLN and SPiEC-Easi. Finally, in this benchmark, we compare the efficiency of the proposed backward approximation (2) against the regular Gaussian mean-field (Blei et al., 2017), denoted as PLN-Tree (MF). The PLN-Tree tag is retained for the residual backward variational approximation modeling.

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

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 replicate the distribution of the original dataset faithfully. To achieve this in our context, we use alpha diversity and beta diversity metrics that are commonly employed in ecosystem studies, as well as empirical Wasserstein on normalized counts (proportion hierarchical data) and correlation measures.

Alpha diversity metrics provide insights into species richness, evenness, thereby partially characterizing the diversity within an ecosystem (see Appendix A.1 and Gotelli and Colwell (2001)). Among these, the Shannon entropy and the Simpson index are widely employed. The Shannon index quantifies the uncertainty in predicting the entities in the ecosystem, while the Simpson index represents the probability that two entities chosen at random represent the same entity. Both estimators are qualified as robust and quantify complementary aspects of the ecosystems (Nagendra, 2002). Our first 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.

While alpha diversity metrics evaluate the intrinsic statistics of one ecosystem, beta diversity metrics enable the quantitative comparison of the composition of two ecosystems (see Appendix A.2). These metrics are often referred to as dissimilarity measures, taking values between 0 and 1 to indicate the degree of dissimilarity between pairs of samples. Among the beta diversity metrics, the UniFrac (Lozupone and Knight, 2005) and Jaccard diversities can account for the hierarchical nature of the data, while the Bray-Curtis dissimilarity Beals (1984), commonly applied in microbiological studies (Kleine Bardenhorst et al., 2021), operates at a single level of the hierarchy. To ensure that the benchmark remains independent of the underlying tree structure, we restrict our assessment to the Bray-Curtis dissimilarity to evaluate the quality of the generations at each layer of the tree. To compare the beta diversity, we draw $n = 100$ samples from the true dataset and from the trained model, and compute the dissimilarity between each pair of samples. Repeating that sampling process $m = 50$ times, we obtain m symmetric dissimilarity matrices of shape $n \times n$. For each matrix, we perform PERMANOVA (Anderson, 2014) and PERMDISP (Anderson, 2006) to test respectively whether the centroids and the dispersions of the two groups are the same. Both tests are performed m times on 1000 permutations, providing finally m associated p-values for each test, the distribution of which will assess the dissimilarity between original and generated data. PERMANOVA and PERMDISP tests are detailed in Appendix A.2.1 and implemented in the `scikit-bio`[3] package.

Finally, 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 (proportion hierarchical data) at each layer using the `emd2` function from *POT* (Flamary et al., 2021). Additionally, we employ correlation measures between the original data and their reconstructions to assess the quality of the variational approximations at the reconstruction task. Computational efficiency between implementations is discussed in Appendix D.

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

[3] https://github.com/scikit-bio/scikit-bio

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.

4.1 Synthetic data

4.1.1 PLN-Tree retrieval

To evaluate the efficiency of the proposed backward variational approximation (2) 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 Figure E3), a reference PLN-Tree model with parameters $\boldsymbol{\theta}^*$, and a synthetic dataset $(\mathbf{X}, \mathbf{Z})$ generated using the PLN-Tree dynamic specified in Section 3 with $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ (see Figure E4), 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. The considered tree of Figure E3 has a small depth and not too many species for computational speed reasons, but it is sufficient to explore scenarios of interest in this benchmark.

PLN-Tree successfully outperforms others under its model

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 distances in Table D2 in Appendix D.1.2). As anticipated, the PLN-Tree models exhibit superior performance compared to the other method, 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 and SPiEC-Easi models, attributable to the Markov tree propagation of the counts, a factor not accounted for by these approaches.

Analyzing beta diversity through the PERMANOVA and PERMDISP tests (see Figure 3) reveals that, at the deepest layer ($\ell = L$), the centroids and dispersions of PLN and SPiEC-Easi significantly deviate from the original data. Specifically, the rejection rates at 5% significance level are 82% and 96% for PLN, and 100% for both tests applied to SPiEC-Easi. In contrast, the PLN-Tree model with backward approximation exhibits rejection rates of only 8% for PERMANOVA and 6% for PERMDISP, suggesting that this model better preserves the beta diversity patterns of the original data compared to the competing methods. Interestingly, the mean-field approximation of PLN-Tree displays a considerably higher rejection rate of around 90% for both tests. At upper layers ($\ell < L$), the backward PLN-Tree model continues to be accepted, demonstrating its robustness across the hierarchy. In comparison, the acceptance rate of PLN improves from 18% at $\ell = L$ to 80% at $\ell = 1$, while SPiEC-Easi remains consistently rejected across all layers at the 95% confidence level. These results highlight the consistency and improved performance of our method in modeling hierarchical beta diversity and the specific interest of the backward approximation over the mean-field approach.

Additionally, Table 2 demonstrates that PLN-Tree-based approaches consistently approximate the distribution of the proportions of the entities at each depth of the tree, contrasting with the other approaches, which exhibits a noticeable 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.

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 4. 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

Alpha diversity	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
		Wasserstein Distance ($\times 10^2$)		
Shannon $\ell = 1$	**1.57** (0.50)	11.23 (0.73)	14.64 (1.15)	46.72 (1.63)
Shannon $\ell = 2$	**3.67** (1.33)	5.14 (1.20)	32.04 (1.62)	89.62 (2.31)
Shannon $\ell = 3$	**5.82** (1.51)	7.86 (1.47)	35.03 (1.68)	98.49 (2.31)
Simpson $\ell = 1$	**0.62** (0.21)	2.69 (0.27)	4.91 (0.41)	15.91 (0.65)
Simpson $\ell = 2$	**0.71** (0.24)	1.40 (0.31)	7.35 (0.41)	22.13 (0.72)
Simpson $\ell = 3$	**0.85** (0.24)	1.55 (0.34)	7.21 (0.41)	22.05 (0.70)

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.

	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
		Wasserstein Distance ($\times 10^2$)		
$\ell = 1$	**5.20** (0.62)	8.61 (0.11)	10.70 (0.34)	24.21 (0.73)
$\ell = 2$	**13.01** (0.14)	16.37 (0.29)	17.59 (0.28)	31.35 (0.67)
$\ell = 3$	**14.08** (0.13)	18.13 (0.32)	20.04 (0.03)	37.36 (0.87)

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.

	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.

for $\ell = 1$ and $\ell = 2$ in Figure 4, 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.

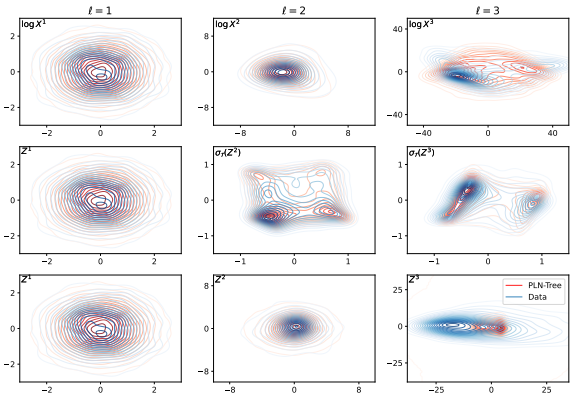
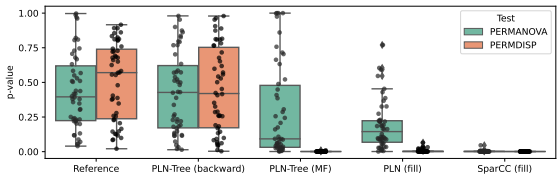
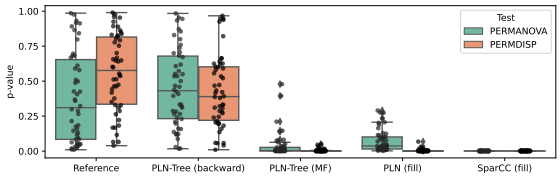


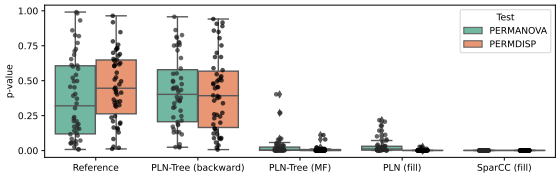
Fig. 4: PCA at each depth of the tree, using the training data and the generated data from the PLN-Tree model. The first row corresponds to the projection of the layer count data in log scale, the second row corresponds to the projection of the latent variables as the multinomial parameters (softmax per group of children, denoted by $\sigma_{\mathcal{T}}$), the third row corresponds to the projection of the raw latent variables.



(a) $\ell = 1$



(b) $\ell = 2$



(c) $\ell = 3$

Fig. 3: p-values for PERMANOVA and PERMDISP tests applied on Bray Curtis dissimilarities (layer-wise) computed between 100 generated data with each model and 100 sampled PLN-Tree generated data from the training dataset, repeated 50 times. Reference model corresponds to generated data from the original model to assess the bootstrap variability.

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 $\boldsymbol{\Omega} = v\boldsymbol{G} + \text{diag}(|\min(\text{eig}(v\boldsymbol{G}))| + u)$. 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.

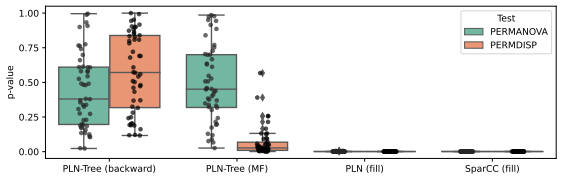
We provide the chosen tree graph for our experiments in Appendix E5. To derive the covariance matrix of the first layer, we generate a random adjacency matrix using the Erdos-Rényi graph model. In our architecture, for all layers ℓ up to L and nodes k up to K_ℓ, α_k^ℓ is structured as a one-layer network 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.

PLN-Tree outperforms others in hierarchical scenarios

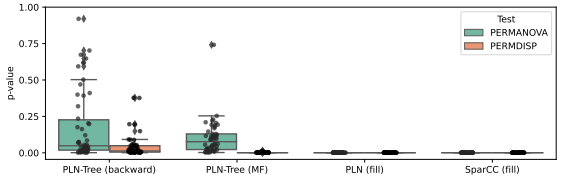
We provide a summary of the model performances in Table 4, Table 5, (see Table D4 for other distances), and Table 6. Notably, the PLN-Tree models exhibit superior performance compared to the PLN and SPiEC-Easi approaches, which do not account for the underlying Markovian tree structure of the data. Similar to our previous synthetic experiment, we observe that as we delve deeper into the tree structure, the performance of PLN and SPiEC-Easi deteriorates significantly. When looking at the alpha diversities in Table D4,

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. The results of the beta diversity tests, presented in Figure 5, reveal a 100% rejection rate for not-tree-based methods at the 5% significance level for both PERMANOVA and PERMDISP tests, confirming their inability to capture beta diversity patterns in this hierarchical context. Among PLN-Tree methods, the backward approximation shows a notably lower rejection rate (4% to 58% for PERMANOVA) compared to the mean-field approach (2% to 84% for PERMANOVA), highlighting the residual backward approximation superiority over the mean-field in learning PLN-Tree models. However, PERMDISP tests at the deepest layer ($\ell = L$) reveal a 100% rejection rate for all models, indicating that even PLN-Tree methods still struggle to fully capture beta diversity patterns at the deepest levels in this particular hierarchical dynamic defined by the Markov Dirichlet framework.

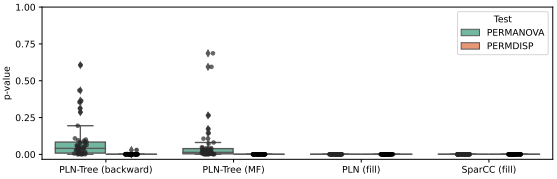
Thus, this experiment demonstrates the inability for not-tree-based method to capture count data distributions in hierarchical context, as well as the interest of considering the backward structure of the true posterior when doing variational inference to learn PLN-Tree. However, progress is still to be made for PLN-Tree methods to fully capture counts distributions in generalized hierarchical context.



(a) $\ell = 1$



(b) $\ell = 2$



(c) $\ell = 3$

Fig. 5: p-values for PERMANOVA and PERMDISP tests applied on Bray Curtis dissimilarities (layer-wise) computed between 100 generated data with each model and 100 sampled Markov Dirichlet generated data from the training dataset, repeated 50 times.

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, for computational speed reasons, we limit our analysis to the layers of the taxonomy comprised between the second and fifth layers, which respectively correspond to the "class" and the "family", as these levels yield sufficient performance disparities between the considered models of this benchmark. 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).

Alpha diversity	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
		Wasserstein Distance ($\times 10^2$)		
Shannon $\ell = 1$	**17.70** (0.47)	21.42 (0.59)	72.27 (1.70)	125.10 (1.25)
Shannon $\ell = 2$	**22.23** (0.94)	29.10 (1.06)	111.53 (1.81)	177.18 (1.50)
Shannon $\ell = 3$	**24.32** (0.83)	37.72 (1.14)	142.28 (1.99)	224.07 (1.62)
Simpson $\ell = 1$	**5.69** (0.16)	5.84 (0.16)	21.74 (0.60)	39.01 (0.46)
Simpson $\ell = 2$	**5.21** (0.17)	5.90 (0.19)	26.70 (0.59)	46.26 (0.54)
Simpson $\ell = 3$	**3.91** (0.11)	5.16 (0.16)	28.55 (0.59)	50.12 (0.54)

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.

	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
		Wasserstein distance ($\times 10^2$)		
$\ell = 1$	**11.51** (0.25)	12.47 (0.30)	25.50 (0.59)	41.84 (0.50)
$\ell = 2$	**19.68** (0.25)	22.02 (0.36)	43.26 (0.61)	59.09 (0.55)
$\ell = 3$	**24.33** (0.24)	27.15 (0.30)	51.84 (0.57)	68.21 (0.52)

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.

	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.

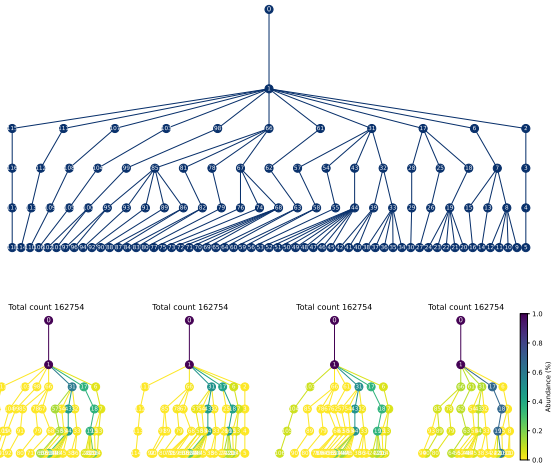


Fig. 6: Graph of the taxonomy considered in the metagenomics experiments (top), and four samples from the dataset (bottom).

4.2.1 Generating microbiome compositions with PLN-Tree

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

Exploiting the taxonomy improves the performances

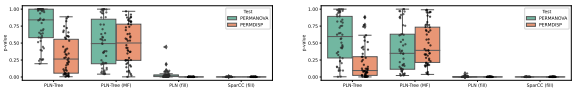
We provide a summary of the model performances in Tables 8 and 9, while Figure D2 illustrates the variability of the generations for each model. Notably, the tree-based models exhibit superior performance for alpha diversity and distribution

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.

of proportions modeling compared to the state-of-art approaches, which do 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. At the deepest layer ($\ell = L$) in Figure 7, rejection rates obtained from the PERMANOVA test at the 5% significance level show that the PLN-Tree model with backward approximation is rejected in only 36% of the tests, compared to 48% with the mean-field approximation. In contrast, the benchmark models are rejected in all cases. Similarly, for the PERMDISP test, PLN-Tree with backward approximation is rejected in only 8% of tests, compared to 46% with the mean-field approach, while the other methods are consistently rejected. These findings suggest that the PLN-Tree models provide a significantly better approximation of the original beta diversity than PLN and SPiEC-Easi. For layers above the deepest ($\ell < L$), the acceptance rate for PLN-Tree residual backward model continues to rise over 80% on average, whereas the benchmark models remain largely rejected for both tests, showing only marginal improvements. This highlights the robustness and consistency of the PLN-Tree model across different layers of the taxonomy.

These findings suggest that the taxonomy provides pertinent insights into the distribution of bacteria and their interactions within the host's ecosystem, bearing significant biological implications. However, as shown in Appendix D2, 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



(a) $\ell = 2$ (b) $\ell = 3$

(c) $\ell = 4$ (d) $\ell = 5$

Fig. 7: p-values for PERMANOVA and PERMDISP tests applied on Bray Curtis dissimilarities (layer-wise) computed between 100 generated data with each model and 100 sampled microbiome data from the metagenomics dataset, repeated 50 times.

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

The analysis of the alpha diversity (see Appendix D6) and the beta diversity underscores the consistently superior performance of the residual amortized backward approximation compared to the mean-field approach. This 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,

	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
		Wasserstein distance ($\times 10^2$)		
$\ell = 1$	5.89 (0.29)	**4.67** (0.25)	15.57 (0.59)	36.16 (1.02)
$\ell = 2$	8.83 (0.28)	**7.55** (0.14)	20.65 (0.71)	42.52 (1.17)
$\ell = 3$	9.27 (0.27)	**7.76** (0.12)	20.86 (0.70)	42.75 (1.16)
$\ell = 4$	17.00 (0.22)	**15.59** (0.13)	29.29 (0.72)	56.19 (0.88)

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.

Alpha diversity	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
		Wasserstein distance ($\times 10^2$)		
Shannon $\ell = 1$	**1.73** (0.44)	3.00 (0.44)	16.49 (1.14)	43.12 (1.57)
Shannon $\ell = 2$	**2.22** (0.73)	5.70 (0.97)	23.21 (1.64)	57.73 (2.02)
Shannon $\ell = 3$	**2.29** (0.63)	6.58 (1.02)	23.96 (1.67)	59.16 (2.00)
Shannon $\ell = 4$	**2.08** (0.62)	20.39 (1.08)	55.32 (2.38)	127.11 (3.03)
Simpson $\ell = 1$	0.84 (0.14)	**0.71** (0.12)	7.18 (0.48)	17.99 (0.71)
Simpson $\ell = 2$	0.92 (0.24)	**0.73** (0.19)	7.49 (0.57)	19.59 (0.81)
Simpson $\ell = 3$	0.91 (0.23)	**0.72** (0.19)	7.46 (0.57)	19.50 (0.80)
Simpson $\ell = 4$	**0.53** (0.13)	2.41 (0.21)	12.91 (0.67)	31.62 (0.99)

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.

	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.

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 the identifiable latent variables as input data for machine learning models instead of the raw observations (see Section 3.2). 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. In particular, we benchmark the proposed latent proportions combined with the CLR (LP-CLR) transform (4) as a preprocessing pipeline using PLN-Tree identifiable latent variables learned 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.

Benchmark procedure

We seek to compare the influence of the preprocessing techniques using the conventional PLN latent features, the CLR transform used in SPiEC-Easi, and the PLN-Tree LP-CLR transform (4), 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 LP-CLR transform to the latent features, while PLN features are projected on $\text{Vect}(\mathbf{1})^{\perp}$ (see Corollary 2) and are thus denoted Proj-PLN. 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 projected PLN latents, CLR transform, or the LP-CLR transform (4) from PLN-Tree models, employing either the residual backward amortized variational approximation or the mean-field approximation. Overall, our results indicate that all the proposed preprocessing procedures enhance performances, except for random forests. It has already been observed in previous works that random forests do not benefit from existing compositional preprocessing with microbiome data (Yerke et al., 2024). We also observe similar performances between the backward PLN-Tree and its mean-field counterpart, 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 demonstrate that PLN-based features can improve classification performances. In particular, using the latent features rather than the true proportions enhances the preprocessing quality of the CLR transform, significantly outperforming the results obtained with the true proportions. Further improvements could potentially be attained by using PLN-Tree's identifiable features rather than their LP-CLR transform within specific deep architectures. Exploring such preprocessing methods is out of the scope of this paper.

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 the

	Proportions	LP-CLR	LP-CLR (MF)	Proj-PLN	CLR
Logistic Regression					
Balanced Accuracy	0.632 (0.042)	0.739 (0.035)	0.731 (0.035)	**0.748** (0.034)	0.729 (0.041)
Precision	0.701 (0.032)	0.783 (0.027)	0.776 (0.027)	**0.789** (0.026)	0.775 (0.031)
Recall	0.645 (0.039)	0.749 (0.028)	0.742 (0.026)	**0.753** (0.030)	0.738 (0.033)
F1 score	0.661 (0.036)	0.758 (0.026)	0.752 (0.025)	**0.763** (0.028)	0.748 (0.031)
ROC AUC	0.677 (0.045)	0.795 (0.034)	0.778 (0.034)	**0.813** (0.035)	0.804 (0.035)
ROC PR	0.438 (0.061)	0.568 (0.065)	0.529 (0.057)	**0.635** (0.063)	0.600 (0.063)
Linear SVM					
Balanced Accuracy	0.586 (0.042)	**0.742** (0.034)	0.728 (0.038)	0.737 (0.037)	0.730 (0.035)
Precision	0.673 (0.035)	**0.784** (0.026)	0.774 (0.029)	0.781 (0.029)	0.776 (0.027)
Recall	0.584 (0.061)	**0.749** (0.028)	0.734 (0.032)	0.746 (0.030)	0.735 (0.034)
F1 score	0.598 (0.062)	**0.758** (0.026)	0.745 (0.031)	0.756 (0.028)	0.746 (0.031)
ROC AUC	0.545 (0.127)	0.798 (0.034)	0.778 (0.034)	**0.810** (0.037)	0.798 (0.036)
ROC PR	0.336 (0.085)	0.588 (0.068)	0.519 (0.058)	**0.630** (0.068)	0.587 (0.066)
Neural Network					
Balanced Accuracy	0.704 (0.036)	**0.745** (0.041)	0.694 (0.040)	**0.745** (0.031)	0.740 (0.035)
Precision	0.773 (0.026)	0.803 (0.032)	0.767 (0.028)	**0.810** (0.023)	0.804 (0.027)
Recall	0.777 (0.028)	0.806 (0.030)	0.767 (0.031)	**0.816** (0.021)	0.809 (0.026)
F1 score	0.772 (0.027)	0.803 (0.031)	0.761 (0.028)	**0.811** (0.023)	0.804 (0.026)
ROC AUC	0.782 (0.036)	0.841 (0.034)	0.775 (0.042)	**0.873** (0.024)	0.859 (0.03)
ROC PR	0.620 (0.062)	0.688 (0.064)	0.611 (0.065)	**0.723** (0.048)	0.717 (0.06)
Random Forest					
Balanced Accuracy	0.673 (0.032)	0.645 (0.028)	**0.676** (0.030)	0.592 (0.027)	0.629 (0.031)
Precision	**0.827** (0.026)	0.803 (0.026)	0.786 (0.031)	0.776 (0.039)	0.791 (0.031)
Recall	**0.811** (0.019)	0.793 (0.016)	0.794 (0.023)	0.766 (0.017)	0.784 (0.018)
F1 score	**0.781** (0.026)	0.757 (0.023)	0.773 (0.025)	0.710 (0.025)	0.743 (0.027)
ROC AUC	**0.903** (0.022)	0.855 (0.029)	0.831 (0.032)	0.868 (0.028)	0.864 (0.03)
ROC PR	**0.790** (0.052)	0.708 (0.057)	0.675 (0.060)	0.690 (0.061)	0.705 (0.061)

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.

model's 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 established interaction-based count data models on generative 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 and SPiEC-Easi approaches. 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 with compositional data using identifiable latent features, 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. Investigating the effect of covariates on the latent variables generated by PLN-Tree is a compelling direction for practical applications. In addition, exploring deep architectures that fully leverage the Markov structure of the identifiable latent variables presents another promising lead for improving the classification performance of PLN-Tree-based preprocessing. Finally, the identification of meaningful interaction networks from the PLN-Tree framework remains an open question. The latent dynamics acting like a fragmentation process, the latent variables are not directly tied to the observed counts but rather to their propagation through the tree, making it challenging to directly associate the latent variables with a given entity in the tree. Investigating covariance properties, in the wake of the faithful correlations in the PLN model Chiquet et al. (2021), could offer deeper insights into the relationship between the inferred covariance structure and the observed counts.

Acknowledgements. 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.

Declarations

Author contributions. All authors conceived the ideas, contributed to the design methodology and investigated the formal analysis. Alexandre Chaussard developed the code, analyzed the data and wrote the first draft of the manuscript. Reviewing and editing has been performed by all authors.

Data availability. The microbiome dataset from Pasolli et al. (2016) is available online on open access. The generated datasets are reproducible from the provided GitHub of PLN-Tree.

References

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.*: Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information fusion **58**, 82–115 (2020)

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

Anderson, M.J.: Distance-based tests for homogeneity of multivariate dispersions. Biometrics **62**(1), 245–253 (2006)

Anderson, M.J.: Permutational multivariate analysis of variance (permanova). Wiley statsref: statistics reference online, 1–15 (2014)

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

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

Beals, E.W.: Bray-curtis ordination: an effective strategy for analysis of multivariate ecological

data. In: Advances in Ecological Research vol. 14, pp. 1–55. Elsevier, Amsterdam (1984)

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

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

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

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

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

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

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

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

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

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

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

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

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

Friedman, J., Alm, E.J.: Inferring correlation networks from genomic survey data. PLOS Computational Biology (2012)

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

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

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

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

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

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

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

Hilbe, J.M.: Modeling Count Data. Cambridge University Press, Cambdrige (2014)

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

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

Ibrahimi, E., Lopes, M.B., Dhamo, X., Simeon, A., Shigdel, R., Hron, K., Stres, B., D'Elia, D., Berland, M., Marcos-Zambrano, L.J.: Overview of data preprocessing for machine learning applications in human microbiome research. Frontiers in Microbiology **14**, 1250909 (2023)

Inouye, D.I., Yang, E., Allen, G.I., Ravikumar, P.: A review of multivariate distributions for count data derived from the poisson distribution. Wiley Interdisciplinary Reviews: Computational Statistics **9**(3), 1398 (2017)

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

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

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

Kleine Bardenhorst, S., Berger, T., Klawonn, F., Vital, M., Karch, A., Rübsamen, N.: Data analysis strategies for microbiome studies in human populations—a systematic review of current practice. Msystems **6**(1), 10–1128 (2021)

Kurtz, Z.D., Müller, C.L., Miraldi, E.R., Littman, D.R., Blaser, M.J., Bonneau, R.A.: Sparse and compositionally robust inference of microbial ecological networks. PLoS computational biology **11**(5), 1004226 (2015)

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

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

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

Lauritzen, S.L.: Graphical Models vol. 17. Clarendon Press, Oxford (1996)

Lozupone, C., Knight, R.: Unifrac: a new phylogenetic method for comparing microbial communities. Applied and environmental microbiology **71**(12), 8228–8235 (2005)

Lin, W., Khan, M.E., Hubacher, N.: Variational message passing with structured inference networks. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=HyH9lbZAW

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

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

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

Oliver, A., Kay, M., Lemay, D.G.: Taxahfe: a machine learning approach to collapse microbiome datasets using taxonomic structure. Bioinformatics Advances **3**(1), 165 (2023)

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

Pasolli, E., Truong, D.T., Malik, F., Waldron, L., Segata, N.: Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. PLoS computational biology **12**(7), 1004977 (2016)

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

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

Schloss, P.D.: Rarefaction is currently the best approach to control for uneven sequencing effort in amplicon sequence analyses. Msphere, 00354–23 (2024)

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

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

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

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

Tomczak, J., Welling, M.: Vae with a vampprior. In: International Conference on Artificial Intelligence and Statistics, pp. 1214–1223 (2018). PMLR

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

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.*: Considerations and best practices in animal science 16s ribosomal rna gene sequencing microbiome studies. Journal of animal science **100**(2), 346 (2022)

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

Yerke, A., Fry Brumit, D., Fodor, A.A.: Proportion-based normalizations outperform compositional data transformations in machine learning applications. Microbiome **12**(1), 45 (2024)

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

Appendix A Diversity metrics

A.1 Alpha diversity

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.

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).

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).

A.2 Beta diversity

Beta diversity measures the variation in species composition between different communities, providing insight into how ecosystems differ from one another, and are thus often referred to as dissimilarity metrics. Unlike alpha diversity, which quantifies species richness and evenness within a single community (sample), beta diversity assesses differences in species composition across multiple ecosystems (pairwise dissimilarity). This measure is crucial in ecology studies, where understanding community structure, biogeography, or the effects of environmental changes is essential. Common beta diversity metrics include Bray-Curtis dissimilarity Beals (1984), which evaluates compositional differences based on species abundances, UniFrac (both unweighted and weighted) Lozupone and Knight (2005), which incorporates phylogenetic distances between communities, and the Jaccard index, which compares species presence and absence. These metrics enable biologist to unveil patterns in communities, going further in the environment's characteristics than agglomerated statistics like alpha diversity.

Bray Curtis dissimilarity

The Bray-Curtis beta diversity is used to quantify the compositional dissimilarity between two communities based on species abundances. It ranges from 0 (completely identical) to 1 (completely dissimilar). The metric emphasizes species abundances, making it sensitive to both shared species and their relative quantities, and is widely used in ecological and microbiome studies for comparing community compositions. Given two samples i, j, let C_{ij} the amount of entities shared in both samples, S_i the total count in site i and S_j the total count in site j, then the Bray Curtis dissimilarity between i and j is given by

$$\mathrm{BC}_{ij} = 1 - \frac{2C_{ij}}{S_i + S_j} \ .$$

A.2.1 Comparing Beta diversities

Computing the pairwise beta diversity between two ecosystems results in a matrix which captures the dissimilarity in species composition between the two ecosystems. To quantify and assess the overall similarity between these ecosystems, this matrix can be further used in statistical analyses such as PERMANOVA and PERMDISP, thus providing a statistical framework for comparing ecosystem differences based on beta diversity metrics.

PERMANOVA

Permutational Multivariate Analysis of Variance (PERMANOVA) (Anderson, 2014) is a non-parametric multivariate statistical test based on permutations. In our context, it used to compare beta diversity between two ecosystems by testing the null hypothesis that the centroids and dispersions of these groups are the same, as defined in the measured space given by the dissimilarity matrix. A rejection of the null hypothesis indicates that there are significant differences between groups in terms of their centroids, their dispersion, or both.

PERMDISP

Permutational Analysis of Multivariate Dispersions (PERMDISP) (Anderson, 2006) is a non-parametric multivariate test that assesses the homogeneity of group dispersions. It tests whether the spread of beta diversity within each ecosystem differs significantly, regardless of group centroids, according to the dissimilarity matrix provided by the beta diversity. The test is commonly used in conjunction with PERMANOVA to distinguish whether differences between groups arise from variability in dispersion rather than differences in central tendency. A rejection of the null hypothesis in PERMDISP suggests that the groups exhibit different degrees of variability, making it particularly valuable for interpreting beta diversity in ecological studies.

Appendix B ELBO derivation for PLN-Tree

Proposition 3. *Consider the PLN-Tree model of Section 3. Then, when using the backward variational approximation (2), the ELBO of the PLN-Tree model writes*

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

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* $1 \leqslant \ell \leqslant L-1, 1 \leqslant k \leqslant K_\ell$,

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}_\ell = &\left(\boldsymbol{\mu}_{\boldsymbol{\theta}_\ell}(\mathbf{Z}^{\ell-1}) - \mathbf{m}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})\right)\left(\boldsymbol{\mu}_{\boldsymbol{\theta}_\ell}(\mathbf{Z}^{\ell-1}) - \mathbf{m}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})\right)^\top \\ &+ \mathbf{S}_{\boldsymbol{\varphi}^\ell}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell}) ,\end{aligned} \tag{B1}$$

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

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

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

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

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

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

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

The tower property yields $\mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}[\mathrm{Z}^\ell_j] = \mathbb{E}_{q_{\boldsymbol{\varphi},1:L}}\left[\mathbf{m}_{\boldsymbol{\varphi}^\ell,j}(\mathbf{Z}^{\ell+1}, \mathbf{X}^{1:\ell})\right]$, thus combining the previous results provides the expected conditional log-likelihood as

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

$$- \mathbb{1}_{\ell=L} \sum_{k=1}^{K_\ell} \log(\mathrm{X}_k^\ell!) + \frac{1}{2} K_\ell \ .$$

The divergence term can be expressed as

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

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

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

where $\widehat{\boldsymbol{\Sigma}}_\ell$ is defined in (B1). Following the same steps for the other terms yields

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

which concludes the proof. □

B.1 PLN-tree ELBO with offset modeling

For a sample i, let $\mathrm{O}_i \in \mathbb{R}$ its offset, following H1: (**H1**)

- 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 $(\breve{\mathbf{X}}_k^\ell)_{1 \leqslant k \leqslant K_\ell}$ are independent and the conditional law of $\breve{\mathbf{X}}_k^\ell$ depends only on $\breve{\mathbf{Z}}_k^\ell$ 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 $\breve{\mathbf{Z}}_k^\ell$, $\breve{\mathbf{X}}_k^\ell$ has a multinomial distribution with parameters $\sigma(\breve{\mathbf{Z}}_k^\ell)$ and X_k^ℓ.

We define the following variational approximation to compute the unknown posterior: (**H2**)

- 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 $q^{\mathbf{Z}}_{\boldsymbol{\varphi},1:L}(\mathbf{Z}|\mathbf{X})$ is a backward Markov chain as defined in (2).

Lemma 4. *Assume that H1 and H2 hold. Denote by $\mathcal{L}_{|\mathrm{O}}(\boldsymbol{\theta},\boldsymbol{\varphi})$ the ELBO of the generative model from proposition 3 with shifted latent means $\boldsymbol{\mu}_1 + \mathrm{O}$ and $\mathbf{m}_{\boldsymbol{\varphi}^1}(.) + \mathrm{O}$, then the ELBO of the offset-modeled PLN-Tree is given by*

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

Proof. By definition,

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

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

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

Using the KL divergence definition

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

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

Appendix C Identifiability results

C.1 PLN identifiability

Lemma 5. *Let Z and $\tilde{\mathrm{Z}}$ be supported on $\mathbb{R}_+^*$, and $\mathrm{X} \sim \mathcal{P}(\mathrm{Z})$ and $\tilde{\mathrm{X}} \sim \mathcal{P}(\tilde{\mathrm{Z}})$. 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 6. *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 5, 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 1

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

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

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

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

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

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

The proof is concluded by the same arguments as in Lemma 6.

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

Lemma 7. *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 $x^1 \mathcal{S}^K = \{(x_1^2, \dots, x_K^2) \in \mathbb{R}^K \mid \sum_{k=1}^K x_k^2 = x^1\}$, then

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

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

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

Therefore, $(\mathbf{X}_1^2, \dots, \mathbf{X}_K^2)$ are conditionally independent with Poisson distribution with parameters $(\mathrm{Z}^1 \mathrm{Z}_k^2)_{1 \leqslant k \leqslant K}$. Hence, by Lemma 1, the law of $(\mathrm{Z}^1 \mathrm{Z}_1^2, \dots, \mathrm{Z}^1 \mathrm{Z}_K^2)$ 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 6, 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 8. *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, $\mathbf{PZ}$ 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 $\mathbf{PZ} = \mathbf{P}\boldsymbol{c} \in \mathbf{P}C$, the projection of C on $\mathrm{Vect}\,(\mathbf{1}_d)^\perp$ and therefore $\{\sigma(\mathbf{Z}) \in \sigma(C)\} \subset \{\mathbf{PZ} \in \mathbf{P}C\}$. We obtain similarly $\{\mathbf{PZ} \in \mathbf{P}C\} \subset \{\sigma(\mathbf{Z}) \in \sigma(C)\}$ so that

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

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

C.2.4 Proof Corollary 2

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 $\mathbf{PZ}^2$ (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 1

By Lemma 1 and Lemma 7 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 $\{\mathbf{X}^2 = \hat{\mathbf{X}}^3\}$. The joint distribution then writes

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

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

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

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.

Appendix D Experimental setup

Latent prior architectures

The latent prior is a Markov chain with 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 precision of the latent dynamic are both composed of two modules. The first module consists of a fully connected neural network, such that at layer $1 < \ell \leqslant L$ of the tree, we fix the number of neurons in the hidden layers to $K_{\ell-1}$ for the mean, and $K_{\ell-1}(K_\ell+1)/2$ for the precision, and only tune the number of hidden layers. Then, for the mean, we add a module to compute the projector associated with the layer ℓ of the tree to ensure the identifiability of the mean parameter (see Section 3.2). Similarly, for the precision matrix, we attach a module that turns the output of the first module into a lower triangular matrix $\mathbf{L}_{\boldsymbol{\theta}^\ell}(.)$ with positive diagonal terms using softplus, thus obtaining the Cholesky decomposition of a positive definite matrix. To prevent computational issues, we add a perturbation term of amplitude $\lambda = 10^{-4}$, ensuring the numerical invertibility of the covariance matrix which is given by $\boldsymbol{\Sigma}_{\boldsymbol{\theta}^\ell}(.) = \mathbf{L}_{\boldsymbol{\theta}^\ell}(.)\mathbf{L}_{\boldsymbol{\theta}^\ell}(.)^\top + \lambda \mathbf{I}_{K_\ell}$. We then apply the projector given in Section 3.2 to ensure the identifiability of the model, and proceed to taking its inverse to obtain the precision matrix.

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 (2)). 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.

Computational efficiency

PLN-Tree training is more computationally intensive than classical PLN alternatives, particularly as the depth of the taxonomy increases, the dimensionality of the layers grows (excluding only-child nodes that do not require parameterization), and the dataset size expands. In our experiments, conducted on a CPU with i5−1335U×12 configuration, training a single-layer PLN model using the `pyPLNmodels` package leads to an average iteration time of 0.01s. In contrast, training the entire PLN-Tree hierarchy has an average iteration time of 0.36s (batch size set to 512). This indicates that while PLN-Tree convergence is achieved, the hierarchical nature of the model and its neural network parameterization significantly slow down the process compared to PLN. It should be noted that both `pyPLNmodels` and our PLN-Tree implementation support GPU acceleration through `CUDA`, though we did not benchmark GPU performance for this study.

Despite the slower training times observed in CPU-based experiments, the PLN-Tree model is inherently scalable to larger datasets contrary to PLN. The critical difference lies in the parameterization of the variational distributions. Indeed, traditional PLN models use a per-individual parameterization of the variational parameters, allowing for fast optimization through closed-form updates at each iteration (Chiquet et al., 2021). However, this approach scales linearly with the number of data points, which can become a bottleneck for very large datasets. In contrast, the PLN-Tree model employs a backward variational approximation that cannot be parameterized per individual due to the structured dependencies imposed by the backward Markov chain. While this makes optimization more challenging and slower at each iteration, the number of parameters in PLN-Tree remains fixed, regardless of the size of the dataset. This property is crucial for scalability, as it allows PLN-Tree to handle large datasets efficiently.

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.

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 D1: Tested backward variational architectures in the PLN-Tree synthetic data experiments.

Mean-field architectures

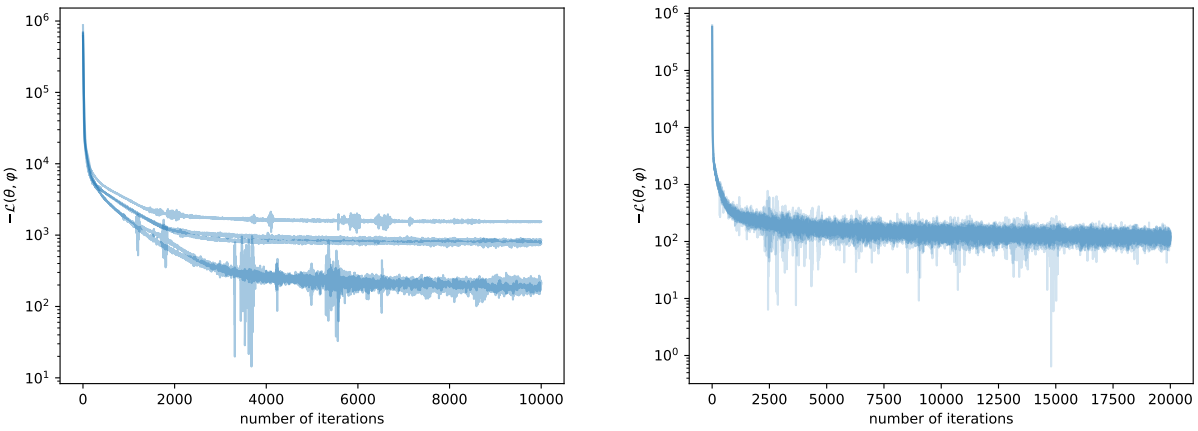
We try 3 architectures of mean-field variational approximations, where the amount of hidden layers in the variation approximation spans in $\{1, 2, 3\}$. The results indicate the optimal architecture is given for 1 hidden layer.

Backward Markov architectures

The tested architectures are summarized in Table D1. The performances of each architecture orientate the choice of the optimal architecture towards the Model 4.

D.1.2 Performance benchmark

For each selected model, we perform multiple training runs and present the resulting objective values in Figure D1. We observe that the mean-field approximation does not converge to the same value of the ELBO value across different runs (see Figure D1a), indicating variability in performance. Conversely, our method consistently converges to the same ELBO values (see Figure D1b), 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.



(a) Mean-field approximation

(b) Residual amortized backward approximation

Fig. D1: 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	SPiEC-Easi
Wasserstein Distance ($\times 10^2$)				
Shannon $\ell = 1$	**1.57** (0.50)	11.23 (0.73)	14.64 (1.15)	46.72 (1.63)
Shannon $\ell = 2$	**3.67** (1.33)	5.14 (1.20)	32.04 (1.62)	89.62 (2.31)
Shannon $\ell = 3$	**5.82** (1.51)	7.86 (1.47)	35.03 (1.68)	98.49 (2.31)
Simpson $\ell = 1$	**0.62** (0.21)	2.69 (0.27)	4.91 (0.41)	15.91 (0.65)
Simpson $\ell = 2$	**0.71** (0.24)	1.40 (0.31)	7.35 (0.41)	22.13 (0.72)
Simpson $\ell = 3$	**0.85** (0.24)	1.55 (0.34)	7.21 (0.41)	22.05 (0.70)
Kolmogorov Smirnov ($\times 10^{-2}$)				
Shannon $\ell = 1$	**2.60** (0.70)	14.69 (1.06)	11.0 (0.99)	32.91 (1.22)
Shannon $\ell = 2$	4.63 (1.29)	**4.42** (1.00)	20.68 (0.87)	47.25 (1.10)
Shannon $\ell = 3$	**5.34** (1.08)	5.54 (0.99)	20.2 (1.15)	45.84 (1.11)
Simpson $\ell = 1$	**2.65** (0.69)	11.14 (0.93)	9.87 (0.78)	28.59 (1.21)
Simpson $\ell = 2$	4.37 (1.24)	**4.18** (0.89)	19.14 (0.97)	42.17 (1.08)
Simpson $\ell = 3$	4.99 (0.92)	**4.58** (0.79)	18.92 (0.98)	42.29 (1.20)
Total variation ($\times 10^{-2}$)				
Shannon $\ell = 1$	**1.14** (0.29)	5.67 (0.41)	4.41 (0.39)	13.12 (0.49)
Shannon $\ell = 2$	**1.21** (0.32)	1.24 (0.21)	5.34 (0.24)	12.27 (0.29)
Shannon $\ell = 3$	**1.19** (0.22)	1.31 (0.17)	4.32 (0.23)	9.97 (0.22)
Simpson $\ell = 1$	**3.13** (0.71)	10.84 (0.94)	11.38 (0.91)	31.82 (1.42)
Simpson $\ell = 2$	**4.29** (0.94)	4.66 (0.97)	19.53 (0.91)	43.60 (1.05)
Simpson $\ell = 3$	4.92 (0.73)	**4.48** (0.74)	18.63 (0.94)	42.23 (1.03)
Kullback-Leibler Divergence ($\times 10^{-2}$)				
Shannon $\ell = 1$	**0.24** (0.11)	4.83 (0.50)	14.17 (1.22)	30.09 (1.81)
Shannon $\ell = 2$	**0.57** (0.26)	0.80 (0.23)	14.39 (1.15)	71.78 (3.24)
Shannon $\ell = 3$	**1.07** (0.36)	1.63 (0.45)	4.56 (0.50)	87.93 (4.22)
Simpson $\ell = 1$	**0.23** (0.11)	2.40 (0.31)	4.56 (0.50)	23.70 (1.65)
Simpson $\ell = 2$	**0.47** (0.19)	0.80 (0.28)	10.64 (0.91)	51.78 (2.25)
Simpson $\ell = 3$	**0.68** (0.20)	0.84 (0.26)	10.57 (0.87)	52.62 (2.13)

Table D2: 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.

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.

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.

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 D3: 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.

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 D3. The architecture of the best-performing model is yielded for 1 layers in the latent prior with the embedding architecture E8.

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.

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 D5, 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.

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 CLR transformed, the projected PLN latent variables on Vect $(\mathbf{1})^{\perp}$ (see Corollary 2), the backward PLN-Tree LP-CLR 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

Alpha diversity	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
Wasserstein Distance ($\times 10^2$)				
Shannon $\ell = 1$	**17.70** (0.47)	21.42 (0.59)	72.27 (1.70)	125.10 (1.25)
Shannon $\ell = 2$	**22.23** (0.94)	29.10 (1.06)	111.53 (1.81)	177.18 (1.50)
Shannon $\ell = 3$	**24.32** (0.83)	37.72 (1.14)	142.28 (1.99)	224.07 (1.62)
Simpson $\ell = 1$	**5.69** (0.16)	5.84 (0.16)	21.74 (0.60)	39.01 (0.46)
Simpson $\ell = 2$	**5.21** (0.17)	5.90 (0.19)	26.70 (0.59)	46.26 (0.54)
Simpson $\ell = 3$	**3.91** (0.11)	5.16 (0.16)	28.55 (0.59)	50.12 (0.54)
Kolmogorov Smirnov ($\times 10^2$)				
Shannon $\ell = 1$	**16.81** (0.93)	24.28 (0.9)	45.12 (1.02)	66.09 (0.69)
Shannon $\ell = 2$	**19.29** (1.06)	25.94 (0.97)	58.83 (0.88)	76.04 (0.56)
Shannon $\ell = 3$	**20.80** (0.75)	30.50 (0.98)	66.62 (0.71)	83.14 (0.31)
Simpson $\ell = 1$	**13.95** (0.94)	20.93 (0.9)	39.77 (1.10)	61.65 (0.73)
Simpson $\ell = 2$	**18.35** (1.03)	23.47 (0.97)	55.42 (0.87)	70.24 (0.69)
Simpson $\ell = 3$	**22.00** (0.87)	30.43 (0.82)	62.10 (0.71)	77.63 (0.32)
Total variation ($\times 10^2$)				
Shannon $\ell = 1$	**7.75** (0.29)	9.78 (0.35)	14.87 (0.33)	21.50 (0.29)
Shannon $\ell = 2$	**6.67** (0.30)	8.08 (0.28)	15.59 (0.21)	19.54 (0.18)
Shannon $\ell = 3$	**5.60** (0.16)	7.47 (0.24)	14.93 (0.14)	18.23 (0.08)
Simpson $\ell = 1$	**19.33** (0.68)	23.72 (1.02)	38.32 (1.02)	58.01 (0.86)
Simpson $\ell = 2$	**20.11** (0.89)	24.60 (1.01)	50.64 (0.79)	64.60 (0.75)
Simpson $\ell = 3$	**19.21** (0.64)	26.42 (0.96)	56.78 (0.66)	71.10 (0.47)
Kullback-Leilbler divergence ($\times 10^2$)				
Shannon $\ell = 1$	**20.72** (2.42)	23.72 (1.70)	60.51 (3.14)	1.8236 (7.55)
Shannon $\ell = 2$	**28.77** (5.75)	33.04 (3.33)	153.73 (8.38)	423.20 (57.72)
Shannon $\ell = 3$	**25.02** (4.03)	40.96 (3.73)	226.13 (11.96)	784.49 (160.02)
Simpson $\ell = 1$	**15.21** (1.71)	15.75 (1.38)	39.04 (2.18)	119.83 (4.30)
Simpson $\ell = 2$	**26.26** (8.32)	26.84 (5.95)	81.47 (3.49)	198.69 (6.90)
Simpson $\ell = 3$	**21.68** (7.61)	29.71 (7.99)	106.28 (3.40)	265.42 (7.48)

Table D4: 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.

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 D5: Tested backward variational architectures in the Embedder in the metagenomics experiments. All embedders are GRU.

Alpha diversity	PLN-Tree	PLN-Tree (MF)	PLN	SPiEC-Easi
Wasserstein distance ($\times 10^2$)				
Shannon $\ell = 1$	**1.73** (0.44)	3.00 (0.44)	16.49 (1.14)	43.12 (1.57)
Shannon $\ell = 2$	**2.22** (0.73)	5.70 (0.97)	23.21 (1.64)	57.73 (2.02)
Shannon $\ell = 3$	**2.29** (0.63)	6.58 (1.02)	23.96 (1.67)	59.16 (2.00)
Shannon $\ell = 4$	**2.08** (0.62)	20.39 (1.08)	55.32 (2.38)	127.11 (3.03)
Simpson $\ell = 1$	0.84 (0.14)	**0.71** (0.12)	7.18 (0.48)	17.99 (0.71)
Simpson $\ell = 2$	0.92 (0.24)	**0.73** (0.19)	7.49 (0.57)	19.59 (0.81)
Simpson $\ell = 3$	0.91 (0.23)	**0.72** (0.19)	7.46 (0.57)	19.50 (0.80)
Simpson $\ell = 4$	**0.53** (0.13)	2.41 (0.21)	12.91 (0.67)	31.62 (0.99)
Kolmogorov Smirnov ($\times 10^2$)				
Shannon $\ell = 1$	**4.71** (1.44)	8.4 (1.35)	23.17 (1.26)	45.26 (1.53)
Shannon $\ell = 2$	**3.42** (0.99)	10.3 (1.25)	22.14 (1.58)	45.65 (1.63)
Shannon $\ell = 3$	**3.48** (0.68)	10.66 (1.32)	22.07 (1.47)	45.57 (1.58)
Shannon $\ell = 4$	**3.64** (1.06)	22.66 (1.3)	36.65 (1.50)	65.15 (1.32)
Simpson $\ell = 1$	4.8 (0.93)	**4.17** (0.58)	21.25 (1.47)	43.30 (1.86)
Simpson $\ell = 2$	**4.46** (1.06)	5.6 (1.46)	19.64 (1.45)	41.76 (1.94)
Simpson $\ell = 3$	**4.17** (1.03)	5.7 (1.53)	19.53 (1.42)	41.53 (1.91)
Simpson $\ell = 4$	**4.09** (1.06)	12.26 (1.46)	32.07 (1.69)	58.34 (1.49)
Total variation ($\times 10^2$)				
Shannon $\ell = 1$	**2.34** (0.63)	4.51 (0.75)	10.00 (0.63)	19.54 (0.75)
Shannon $\ell = 2$	**1.42** (0.34)	3.87 (0.52)	7.47 (0.63)	15.38 (0.63)
Shannon $\ell = 3$	**1.36** (0.24)	3.81 (0.48)	7.19 (0.59)	14.88 (0.62)
Shannon $\ell = 4$	**0.82** (0.27)	6.3 (0.42)	8.94 (0.38)	15.69 (0.30)
Simpson $\ell = 1$	**6.71** (1.18)	8.17 (1.8)	27.59 (1.80)	51.63 (2.13)
Simpson $\ell = 2$	**5.51** (0.89)	7.41 (1.61)	22.23 (1.60)	45.53 (2.12)
Simpson $\ell = 3$	**5.63** (0.93)	7.35 (1.64)	21.83 (1.56)	44.71 (2.05)
Simpson $\ell = 4$	**3.75** (1.15)	14.88 (1.72)	34.56 (1.58)	59.51 (1.28)
Kullback-Leilbler divergence ($\times 10^2$)				
Shannon $\ell = 1$	**0.88** (0.32)	2.32 (0.82)	15.98 (1.33)	48.32 (2.94)
Shannon $\ell = 2$	**0.86** (0.32)	2.68 (0.71)	15.33 (1.59)	48.99 (3.24)
Shannon $\ell = 3$	**0.71** (0.29)	2.87 (0.75)	15.30 (1.57)	49.10 (3.13)
Shannon $\ell = 4$	**0.57** (0.22)	18.02 (4.71)	35.01 (2.64)	116.63 (3.97)
Simpson $\ell = 1$	**1.04** (0.33)	1.54 (0.68)	15.57 (1.40)	45.21 (3.10)
Simpson $\ell = 2$	**0.96** (0.28)	1.14 (0.50)	13.61 (1.42)	42.20 (3.24)
Simpson $\ell = 3$	**0.97** (0.32)	1.11 (0.51)	13.58 (1.41)	41.81 (3.18)
Simpson $\ell = 4$	**0.81** (0.30)	9.70 (3.48)	29.66 (2.38)	90.79 (3.80)

Table D6: Distribution metrics on alpha diversities computed between metagenomics data and simulated data under each modeled trained, averaged over the trainings, with standard deviation.

into respective latent variables. We then select various classifiers (see Table D7) 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.

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 D8, demonstrating similar interpretations to what is observed in the T2D-vs-all problem.

Appendix E Additional experiments visualisations

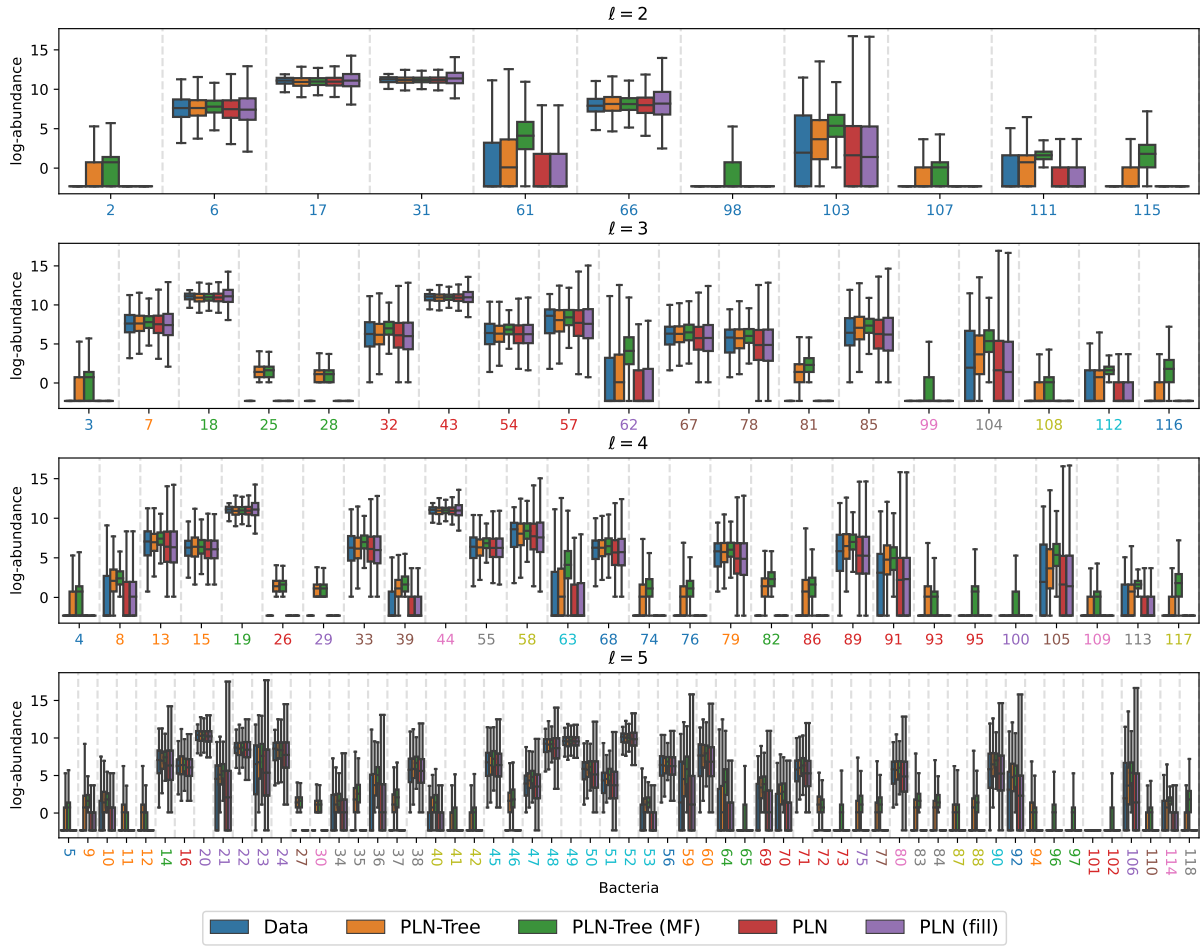


Fig. D2: Boxplot of log abundances of the metagenomics dataset and generated data from several PLN-based models learned on this dataset, with 20000 points per model. Zero abundances are artificially shifted to 10^{-1} to represent them in log scale. The bacteria are denoted by a unique integer on the x-axis, with colors indicating the brotherhoods in the taxonomic tree at a given depth.

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 D7: Considered classifiers in the metagenomics preprocessing experiment, with hyperparameters based on Scikit-Learn implementation.

	Proportions	LP-CLR	LP-CLR (MF)	Proj-PLN	CLR
Logistic Regression					
Balanced Accuracy	0.673 (0.043)	**0.765** (0.046)	0.722 (0.045)	0.763 (0.038)	0.75 (0.035)
Precision	0.770 (0.027)	**0.827** (0.029)	0.801 (0.028)	0.825 (0.024)	0.818 (0.021)
Recall	0.680 (0.030)	0.771 (0.034)	0.723 (0.036)	**0.782** (0.031)	0.764 (0.029)
F1 score	0.705 (0.027)	0.787 (0.030)	0.744 (0.032)	**0.795** (0.028)	0.779 (0.026)
ROC AUC	0.735 (0.042)	0.830 (0.037)	0.793 (0.036)	**0.836** (0.035)	0.83 (0.026)
ROC PR	0.410 (0.062)	0.571 (0.067)	0.541 (0.074)	**0.615** (0.072)	0.569 (0.068)
Linear SVM					
Balanced Accuracy	0.573 (0.055)	**0.756** (0.046)	0.719 (0.04)	0.754 (0.030)	0.752 (0.029)
Precision	0.765 (0.113)	**0.821** (0.028)	0.799 (0.025)	0.820 (0.018)	0.819 (0.018)
Recall	0.390 (0.132)	0.762 (0.034)	0.711 (0.032)	**0.774** (0.026)	0.762 (0.025)
F1 score	0.361 (0.195)	0.779 (0.031)	0.734 (0.028)	**0.788** (0.023)	0.778 (0.022)
ROC AUC	0.416 (0.186)	0.829 (0.036)	0.789 (0.036)	**0.834** (0.032)	0.825 (0.029)
ROC PR	0.215 (0.103)	0.577 (0.076)	0.533 (0.069)	**0.626** (0.065)	0.574 (0.067)
Neural Network					
Balanced Accuracy	0.726 (0.044)	0.741 (0.043)	0.712 (0.043)	**0.749** (0.046)	0.743 (0.039)
Precision	0.826 (0.026)	0.837 (0.023)	0.819 (0.025)	**0.847** (0.027)	0.84 (0.025)
Recall	0.830 (0.024)	0.842 (0.022)	0.821 (0.029)	**0.853** (0.024)	0.844 (0.025)
F1 score	0.825 (0.023)	0.837 (0.024)	0.815 (0.026)	**0.847** (0.026)	0.84 (0.025)
ROC AUC	0.839 (0.034)	0.860 (0.031)	0.816 (0.037)	**0.862** (0.036)	0.858 (0.034)
ROC PR	0.607 (0.062)	0.656 (0.061)	0.578 (0.073)	**0.695** (0.072)	0.666 (0.066)
Random Forest					
Balanced Accuracy	**0.647** (0.035)	0.604 (0.029)	0.610 (0.036)	0.606 (0.037)	0.635 (0.036)
Precision	**0.857** (0.021)	0.814 (0.028)	0.806 (0.038)	0.839 (0.031)	0.845 (0.025)
Recall	**0.844** (0.016)	0.819 (0.014)	0.819 (0.019)	0.827 (0.017)	0.837 (0.017)
F1 score	**0.810** (0.024)	0.777 (0.021)	0.780 (0.027)	0.781 (0.028)	0.801 (0.025)
ROC AUC	**0.917** (0.021)	0.865 (0.028)	0.817 (0.036)	0.875 (0.025)	0.898 (0.021)
ROC PR	**0.794** (0.050)	0.653 (0.067)	0.586 (0.077)	0.709 (0.062)	0.743 (0.055)

Table D8: Classification IBD-vs-all performances for several classifiers on the metagenomics dataset using different preprocessing strategies, averaged over training, with standard deviation.

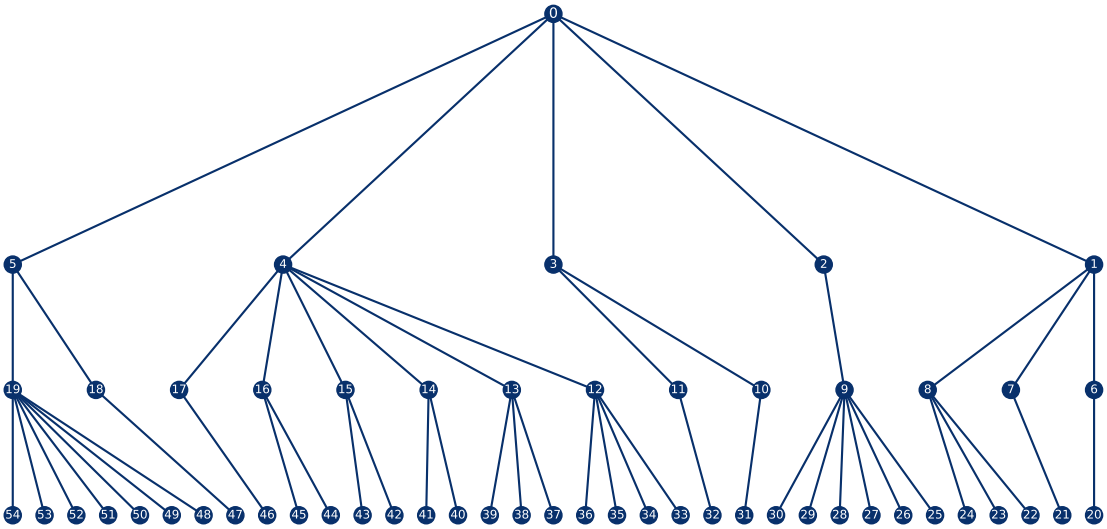


Fig. E3: Graph of the tree considered in the PLN-Tree synthetic experiments.

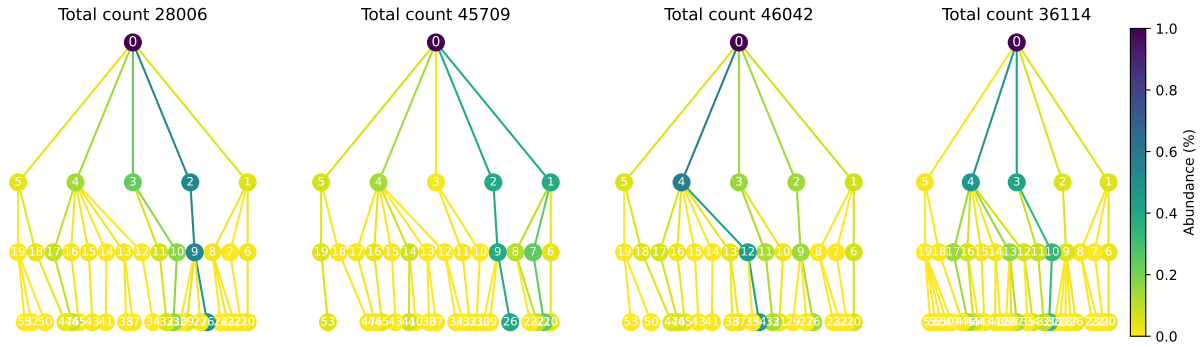


Fig. E4: Synthetic hierarchical samples from the artificial dataset $(\mathbf{X}, \mathbf{Z})$.

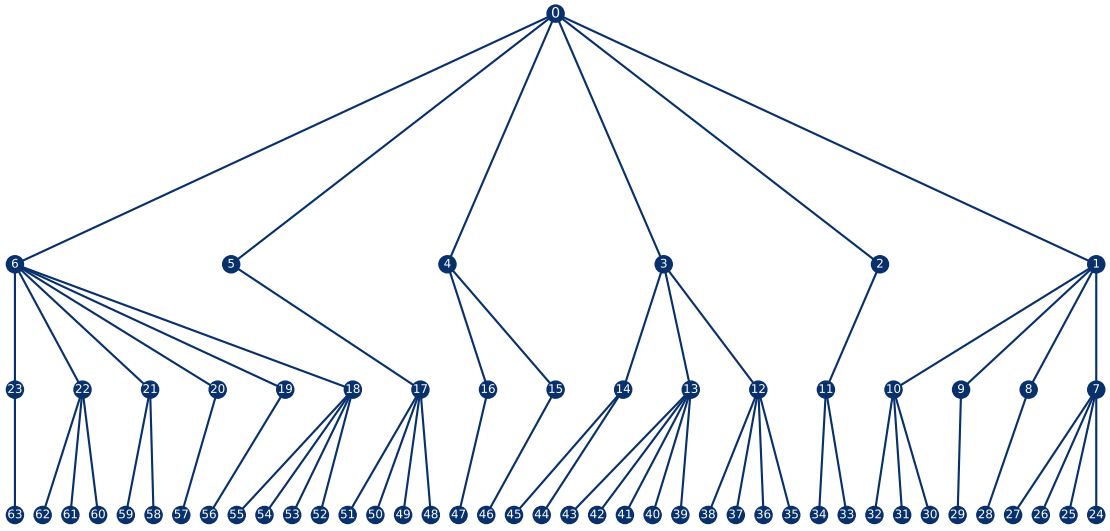


Fig. E5: Graph of the tree considered in the Markov Dirichlet synthetic experiments.