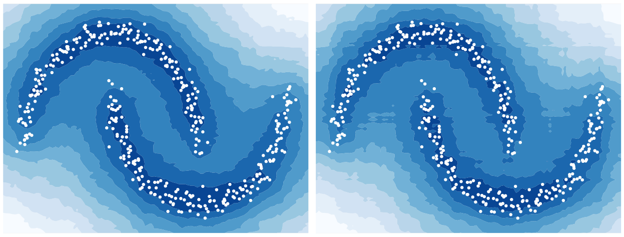
# On Priors in Bayesian Probabilistic Circuits and Multivariate Pólya Trees

**Martin Trapp**[1] **Arno Solin**[1]

[1]Department of Computer Science, Aalto University, Espoo, Finland

## Abstract

Bayesian formulations of probabilistic circuits (PCs) have gained increasing attention, *e.g.*, to regularize parameter or structure learning or perform model selection. However, prior specification, an essential part of the Bayesian workflow, is often not adequately addressed. In this work, we discuss priors in Bayesian PCs and show that certain constructions are related to Pólya tree processes in the limit of infinite depth. Furthermore, we show that Bayesian PCs can accurately represent mixtures of multivariate Pólya trees with only a fraction of the random variables required in the former. We verify our findings with simulations on synthetic data.



(a) Mixture of Multivariate Pólya trees (b) Bayesian Probabilistic Circuit

Figure 1: Illustration of a density function drawn from a finite mixture of multivariate Pólya trees (a) and a Bayesian probabilistic circuit (b) on a 2D toy data set. Both represent priors over a similar (in KL divergence) set of densities, but probabilistic circuits require only a fraction (less than 10%) of the random variables needed for the Pólya tree.

## 1 INTRODUCTION & RELATED WORK

Over the years, probabilistic circuits (PCs) (*e.g.*, Darwiche [2003], Poon and Domingos [2011], Peharz [2015], Trapp [2020], Choi et al. [2020], Vergari et al. [2021]) have gained increasing attention in the machine learning community as an effective approach for tractable and flexible probabilistic modelling. Consequently, a plethora of approaches for parameter and structure learning of PCs have been proposed, including various techniques that leverage Bayesian learning (*e.g.*, Zhao et al. [2016], Trapp et al. [2019], Vergari et al. [2019], Trapp [2020]). In comparison to frequentist approaches, Bayesian learning of PCs has shown to be more robust, allows learning despite missing data [Trapp et al., 2019], and enables interpretable inferences useful for data type discovery [Vergari et al., 2019].

Orthogonal to the research on PCs, Bayesian deep learning, which aims to leverage Bayesian learning for deep neural networks, has recently gained large attention in the machine learning community. Alongside advancements in approximated Bayesian inference for Bayesian neural networks (BNNs), investigating prior specification has become a major research direction (*e.g.*, Pearce et al. [2020], Fortuin et al. [2021], Meronen et al. [2021]). However, somewhat surprisingly, prior specification in Bayesian formulations of PCs is still in its infancy and lacks a thorough investigation.

In this work, we study prior selection in Bayesian formulations of PCs and show that specific constructions of PCs are related to tail-free processes, such as the Pólya tree process and the Dirichlet process, in the limit of infinite depth. In fact, finite Pólya trees can be seen as a special case of Bayesian PCs. Moreover, while the number of random variables (RVs) in finite multivariate Pólya trees (MPTs) scales exponentially in depth $J$ and input dimensionality $p$, Bayesian PCs obtain comparable results often with only a fraction (<10%) of the number of RVs, *cf.* Fig. 1.

Our contributions can be summarised as follow: (i) we show that certain Bayesian PCs are related to Pólya trees in the limit of infinite depth, (ii) we show that Bayesian PCs can represent mixtures of finite MPTs accurately and efficiently, (iii) and lastly, we verify our results through simulations on synthetic data sets.

*Accepted for the 5th Workshop on Tractable Probabilistic Modeling at UAI* (TPM 2022).

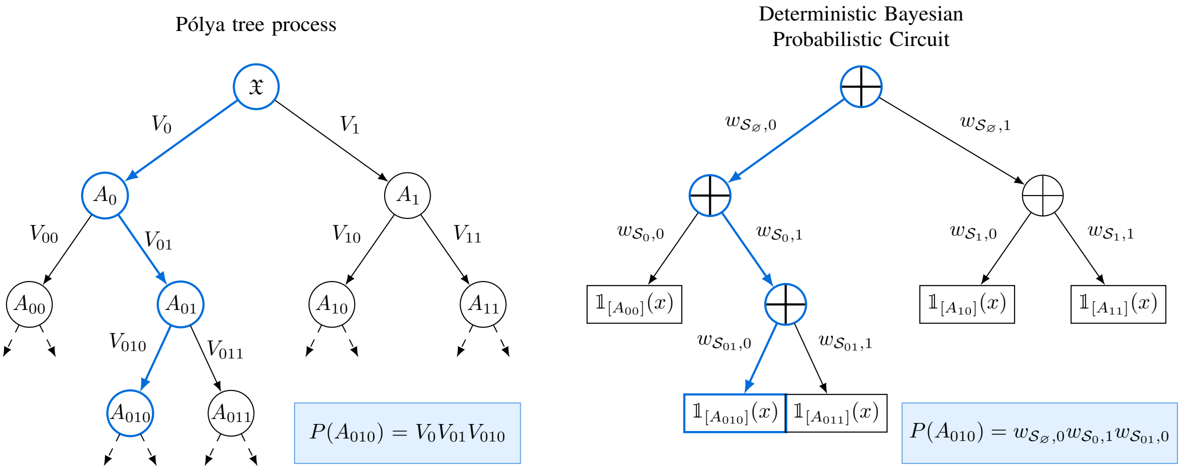

Pólya tree process
$\mathfrak{X}$
$V_0$
$V_1$
$A_0$
$A_1$
$V_{00}$
$V_{01}$
$V_{10}$
$V_{11}$
$A_{00}$
$A_{01}$
$A_{10}$
$A_{11}$
$V_{010}$
$V_{011}$
$A_{010}$
$A_{011}$
$P(A_{010}) = V_0 V_{01} V_{010}$
Deterministic Bayesian Probabilistic Circuit
$w_{\mathcal{S}_\varnothing,0}$
$w_{\mathcal{S}_\varnothing,1}$
$w_{\mathcal{S}_0,0}$
$w_{\mathcal{S}_0,1}$
$w_{\mathcal{S}_1,0}$
$w_{\mathcal{S}_1,1}$
$\mathbb{1}_{[A_{00}]}(x)$
$\mathbb{1}_{[A_{10}]}(x)$
$\mathbb{1}_{[A_{11}]}(x)$
$w_{\mathcal{S}_{01},0}$
$w_{\mathcal{S}_{01},1}$
$\mathbb{1}_{[A_{010}]}(x)$
$\mathbb{1}_{[A_{011}]}(x)$
$P(A_{010}) = w_{\mathcal{S}_\varnothing,0} w_{\mathcal{S}_0,1} w_{\mathcal{S}_{01},0}$

Figure 2: Illustration of a Pólya tree process and a deterministic Bayesian PC representing a truncation of the Pólya tree. Both recursively sub-divide the domain and associate each sub-part with a random probability (random weight). For example, computing the probability of $\boxed{P(A_{010})}$ in the PC is equivalent to the computation in a finite/truncated Pólya tree.

## 2 PRELIMINARIES

We briefly review background information on probabilistic circuits and Pólya trees and introduce the notation used in this work. See appendix for a notation summary.

### 2.1 PROBABILISTIC CIRCUITS

For consistency with recent works, we will briefly introduce probabilistic circuits (PCs) based on the formalism used in Trapp et al. [2019] and Choi et al. [2020]. A PC on a set of RVs $\boldsymbol{X} = \{X_d\}_{d=1}^{p}$, denoted as $\mathcal{C}(\boldsymbol{X})$, is defined as a tuple $(\mathcal{G}, \psi)$ consisting of a directed acyclic graph $\mathcal{G}$ and a scope function $\psi$. Nodes in $\mathcal{G}$ are either sum nodes $(\mathrm{S}(\boldsymbol{x}) = \sum_{\mathrm{N}\in\mathrm{ch}(\mathrm{S})} w_{\mathrm{S},\mathrm{N}}\,\mathrm{N}(\boldsymbol{x}))$, product nodes $(\mathrm{P}(\boldsymbol{x}) = \prod_{\mathrm{N}\in\mathrm{ch}(\mathrm{P})} \mathrm{N}(\boldsymbol{x}))$, or leaf nodes $(\mathrm{L}(\boldsymbol{x}) = p(\boldsymbol{x} \mid \theta_{\mathrm{L}}))$. We use N to denote a generic node and boldface to indicate sets of nodes. The scope function $\psi\colon \mathbf{N} \to \mathscr{P}(\boldsymbol{X})$ assigns each node in the graph $\mathcal{G}$ a scope, *i.e.*, a subset $\boldsymbol{Y} \subseteq \boldsymbol{X}$, where $\mathscr{P}(\boldsymbol{X})$ is the power set including $\emptyset$ and $\boldsymbol{X}$.

Depending on the structural properties of the circuit, specific probabilistic inference queries can be answered tractably. We will assume throughout the paper that the circuit is *smooth* and *decomposable*, see Choi et al. [2020] for details.

We will additionaly consider the property called *determinism*. A sum node is *deterministic* if, for any fully-instantiated input, the output of at most one of its children is nonzero. A PC is *deterministic* if all of its sum nodes are *deterministic*. This is a strong structural constraint, but it has been shown by prior work that ensembles thereof can obtain competitive results in density estimation tasks (*e.g.*, Peharz et al. [2014], Liang et al. [2017]) and can be effective surrogate models for Bayesian inference (*e.g.*, Shih and Ermon [2020]).

### 2.2 PÓLYA TREES

The Pólya tree process (*e.g.*, Mauldin et al. [1992], Lavine [1992], Hanson [2006]) has been widely used in the statistics literature and applied to various applications, including modeling of censored data (*e.g.*, Neath [2003]), modeling regression errors (*e.g.*, Hanson and Johnson [2002]), and survival analysis (*e.g.*, Muliere and Walker [1997], Zhao et al. [2009]). We will give a brief and pragmatic introduction to Pólya trees and refer to Ghosal and van der Vaart [2017] for a more rigourous and thorough discussion.

Let $(\Omega, \mathscr{A}, \mathbb{P})$ be some abstract probability space, $(\mathfrak{X}, \mathscr{X})$ some sample space of interest and let $\mathfrak{M}$ denote the set of probability measures on the sample space. We are interested in representing a random probability measure $\xi\colon \Omega \times \mathscr{X} \to [0,1]$ through a Pólya tree process. For this, let the sample space $\mathfrak{X}$ be recursively partitioned into measurable subsets, *i.e.*, $\Pi^1 = \{\mathfrak{X}\} = A_0 \cup A_1$, $\Pi^2 = \{A_0, A_1\} = \{A_{00} \cup A_{01}, A_{10} \cup A_{11}\}$, and $\Pi^J = \{A_{\varepsilon 0} \cup A_{\varepsilon 1} \mid \varepsilon \in \mathbb{B}^J\}$ where $A_{\varepsilon 0} \cap A_{\varepsilon 1} = \emptyset$ for every $\varepsilon$. Finally, let each part $A$ be associated with a positive *random* weight $V$ representing the following conditional probabilities:

$$V_{\varepsilon 0} = P(A_{\varepsilon 0} \mid A_\varepsilon) \quad \text{and} \quad V_{\varepsilon 1} = P(A_{\varepsilon 1} \mid A_\varepsilon), \qquad (1)$$

and let the partitioning be *tree additive*, *i.e.*, $P(A_\varepsilon) = P(A_{\varepsilon 0} \mid A_\varepsilon) + P(A_{\varepsilon 1} \mid A_\varepsilon)$. Therefore, we have:

$$P(A_{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_J}) = V_{\varepsilon_0} V_{\varepsilon_0 \varepsilon_1} \dots V_{\varepsilon_0 \dots \varepsilon_J}\,, \qquad (2)$$

with $\varepsilon_i \in \mathbb{B}$ for the joint probability.

**Definition 2.1 (Pólya tree process)** *A random measure is a Pólya tree process wrt a sequence of partitions $\Pi^j, j \in 1, \dots, J$ if $\{V_0\} \perp\!\!\!\perp \{V_{00}, V_{10}\} \perp\!\!\!\perp \dots \perp\!\!\!\perp \{V_{\varepsilon 0} \mid \varepsilon \in \mathbb{B}^J\}$*

*and each random weight is distributed according to $V_{\varepsilon 0} \sim$ Beta$(\alpha_{\varepsilon 0}, \alpha_{\varepsilon 1})$ and $V_{\varepsilon 1} = 1 - V_{\varepsilon 0}$. If $J = \infty$, we call the process an infinite Pólya tree process and finite otherwise.*

In the following we use $\mathrm{PT}^J(c, \rho, \Pi)$ to denote a finite Pólya tree process of depth $J$ with partitions $\Pi$ on $\mathfrak{X}$ parameterized with $c > 0$ and $\rho(\cdot) > 0$. Moreover, we will use a prior specification given as $V_{\varepsilon_j(k)0}, V_{\varepsilon_j(k)1} \sim \mathrm{Dir}(c\rho(j), c\rho(j))$ with $k \in 1, \dots, 2^{j-1}$ and $j \in 1, \dots, J$, where $\varepsilon_j(\cdot)$ is used to encode elements into a linear index at each level $j$. Note that depending on the choice of $\rho$ a Pólya tree process is a process over a.c. distributions with probability 1, *i.e.*, if $\rho(j) = j^2$, or a process over discrete distributions, *e.g.*, if $\rho(j) = 2\rho(j+1)$ and $\mathfrak{X} = \mathbb{R}$, as $J \to \infty$. Throughout this work we will assume that $\rho(j) = j^2$ and refer to Lavine [1992], Ghosal and van der Vaart [2017] for details on parameterisations of Pólya tree processes.

# 3 MAIN RESULTS

Given a sample space $(\mathfrak{X}, \mathscr{X})$, let $\mathcal{C} = (\mathcal{G}, \psi)$ be a finite binary tree of depth $J$ that recursively partitions $\mathfrak{X}$ into measurable subsets $A_\mathrm{N} \in \Pi$ defined by a computational graph and a scope function mapping to $\mathscr{X}$ (*cf.*, Trapp et al. [2020]). Further, let $A_\mathrm{N} = \mathrm{supp}(\mathrm{N})$ denote the support of nodes in the tree and let $w_{\mathrm{S}_{\varepsilon_j(k)0}}, w_{\mathrm{S}_{\varepsilon_j(k)1}} \sim \mathrm{Dir}(c\rho(j), c\rho(j))$ denote *random* weights associated to the sum nodes $\mathbf{S} = \{\mathrm{S}_{\varepsilon_j(k)} \mid k \in 1, \dots 2^{j-1}, j \in 1, \dots, J\}$ of the tree.

Then, probabilities wrt sum nodes $\mathrm{S}_{\varepsilon_j(k)}$ are computed as:

$$\begin{aligned} p_{\mathrm{S}_{\varepsilon_j(k)}}(A_{\mathrm{S}_{\varepsilon_j(k)}}) = {} & w_{\mathrm{S}_{\varepsilon_j(k)0}}\, p_{\mathrm{S}_{\varepsilon_j(k)0}}(A_{\mathrm{S}_{\varepsilon_j(k)0}}) \\ & + w_{\mathrm{S}_{\varepsilon_j(k)1}}\, p_{\mathrm{S}_{\varepsilon_j(k)1}}(A_{\mathrm{S}_{\varepsilon_j(k)1}}). \end{aligned} \tag{3}$$

Further, we assume indicator functions at the leaves, *i.e.*, $p_\mathrm{L}(A_\mathrm{L}) = \mathbb{1}_{[x \in A_\mathrm{L}]}$, which return one if the argument is true and zero otherwise. Moreover, as commonly assumed in Bayesian approaches to PCs (*e.g.*, Zhao et al. [2016], Trapp et al. [2019]), we will assume the random weight sets for sum nodes to be mutually independent across the tree.

From the description above, it becomes evident that specific constructions of Bayesian PCs are indeed finite Pólya trees.

**Corollary 1** *A finite Pólya tree process is a special case of deterministic Bayesian probabilistic circuit.*

Fig. 2 illustrates a Pólya tree process and a deterministic Bayesian PC respresenting a truncation of the process.

**Implications:** There are various direct implications arising from the correspondence between certain deterministic Bayesian PCs and Pólya tree processes.

1. Specific constructions of deterministic Bayesian PCs allow tractable posterior inference.
2. Specific constructions of deterministic Bayesian PCs correspond to Dirichlet processes in the limit.
3. Specific constructions of deterministic Bayesian PCs are distributions over a.c. distributions in the limit.

## 3.1 MULTIVARIATE PÓLYA TREES

Our definition of Pólya trees can easily be extended to the multivariate case. For this, let us assume that $\mathfrak{X} = \mathbb{R}^2$. Then $\Pi^1$ will divide the 2D plane into four sets, each of those sets will be further divided into four sets resulting in 16 sets at level $j = 2$. Generally, at a depth of $j$, we obtain a sub-division of $\mathbb{R}^p$ with $p > 0$ into $2^{pj}$ disjoint sub-sets.

This highlights a key issue when working with finite MPTs, the number of RVs scales exponentially in depth and dimensionality, *i.e.*, $\sum_{j=1}^{J} 2^{pj}$. Therefore, one is restricted to using MPTs only in low-dimensional settings. Alternatively, one may use the marginal model by marginalising out the random measure in higher-dimensional cases. However, doing so still requires storing counts for all of the $\sum_{j=1}^{J} 2^{pj}$ many RVs.

Fortunately, by leveraging a representation with a deterministic Bayesian PC that places a product node (*i.e.*, assumes independence between input dimensions) after each sum node, we can approximate a MPT with only $2^{pJ_p} + 2^{(p-1)J_{p-1}} + \dots + 2^{J_1} + \sum_{j=\tilde{J}}^{J} 2^j$ RVs where $\tilde{J} = 1 + \sum_{d=1}^{p} J_d$ and $\sum_{d=1}^{p} J_d \leq J$. Thus, a construction through deterministic Bayesian PCs can substantially reduce the computational and memory costs required for finite MPTs. However, in practice, such an approximation might be too rough, and we ought to introduce product nodes after several consecutive sum nodes instead. In Section 3.3 we discuss the effect of representing a mixture of MPTs with an increasing number of consecutive sum nodes (increasing fraction of RVs).

## 3.2 MIXTURES OF PÓLYA TREES

Mixtures of Pólya trees have been widely studied (*e.g.*, Hanson [2006], Paddock et al. [2003]), for ease of the discussion, we will focus on the construction proposed by Hanson [2006]. Let $g_\theta$ denote the density of an a.c. parametric distribution indexed by $\theta$ with CDF $G_\theta$. Further, let the partitions be given as $\Pi_\theta^j = \{B_\theta(e_j(k)) \colon k \in 1, \dots, 2^j\}$ where $B_\theta(e_j(k)) = (G_\theta^{-1}((k-1)2^{-j}), G_\theta^{-1}(k2^{-j}))$ and let $G \sim \mathrm{PT}(c, \rho, \Pi)$ follow $G_\theta$ on sets of $\Pi_\theta^j$. Then, we obtain a mixture of Pólya trees if $\theta$ is considered to be random. The above can be generalised to the multivariate case with some additional technical details.

As for finite Pólya trees, we can represent a mixture of finite Pólya trees (or a mixture of MPTs) through a Bayesian PC. In the case of a mixture, we obtain an ensemble of deterministic Bayesian PCs. Details can be found in Appendix A.

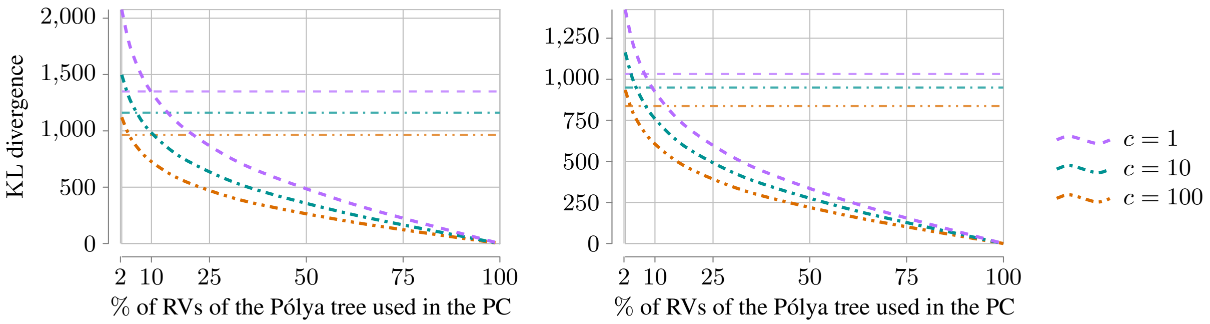


Figure 3: Comparison between the posterior expectation of a mixture of multivariate Pólya trees and a Bayesian probabilistic circuit based on Monte Carlo estimates of the KL divergence under different values of $c \in [1, 10, 100]$. **Left hand-side** shows results on the two moons data set, and **right hand-side** shows results on the two circles data set. Horizontal lines show Monte Carlo estimates for the entropy of the posterior expectation of the mixture of multivariate Pólya trees for comparison.

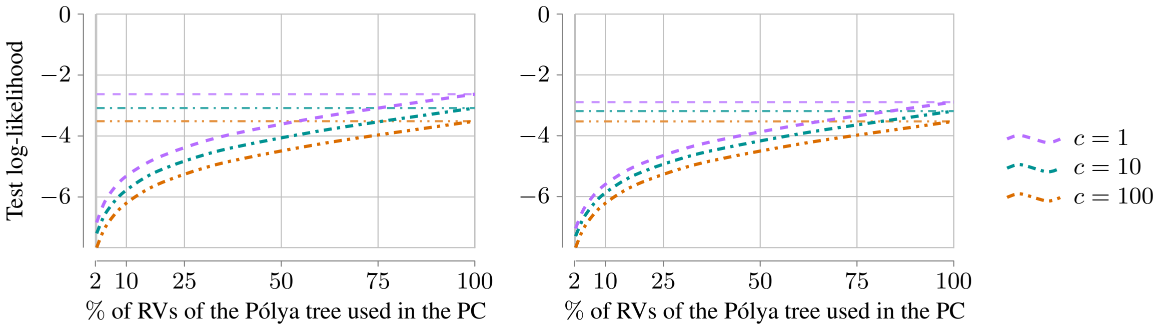


Figure 4: Average test set log-likelihoods for draws from a Bayesian probabilistic circuit under different values of $c \in [1, 10, 100]$. **Left hand-side** shows results on the two moons data set, and **right hand-side** shows results on the two circles data set. Horizontal lines show average results for draws from a mixture of multivariate Pólya trees for comparison.

As in the case of MPTs, we can obtain a more efficient representation through the use of Bayesian PCs.

### 3.3 SIMULATION RESULTS

We evaluated our approach by comparing draws from the posterior of a mixture of MPTs against draws from Bayesian PCs as well as their posterior expectations under varying concentration parameters $c$. For this purpose, we generated two synthetic data sets, the two moons and two circles data sets, each consisting of $n_{\text{train}} = 400$ training samples and $p = 2$ dimensions. In each experiment, we analysed the performance of Bayesian PCs as a surrogate model for a mixture of MPTs under different values of the control parameter $c \in [1, 10, 100]$. We refer to Appendix B for details.

Fig. 3 shows the Kullback–Leibler (KL) divergence from the posterior expectation of a mixture of MPTs to the posterior expectation of a Bayesian PC following the same partitioning applied by the MPTs. With an increasing number of consecutive sum nodes (represented by the $\%$ of RVs of the Pólya tree used for the circuit), the KL divergence quickly decays, indicating that a Bayesian PC can accurately represent the marginal model of a mixture of MPTs with a fraction of the computational and memory costs.

Fig. 4 shows the average test log-likelihood for draws from a mixture of MPTs and a Bayesian PC following the same partitioning. Bayesian PCs that use as little as $25\%$ of the RVs required for a mixture of MPTs provide a reasonable approximation, indicating that Bayesian PCs are a promising alternative to fully instantiated MPTs. Additionaly, Fig. 5 shows a comparison of the computational costs and approximation quality, and Fig. 6 qualitatively compares draws from both models, verifying our results.

## 4 CONCLUSION AND DISCUSSION

We have shown that certain constructions of deterministic Bayesian probabilistic circuits (PCs) correspond to finite Pólya trees and henceforth to infinite Pólya tree processes in the limit of infinite depth. Moreover, we have shown that Bayesian PCs can represent multivariate Pólya trees (MPTs) and mixtures thereof more efficiently. Our simulation results indicate that Bayesian PCs provide a computationally attractive alternative to fully instantiated MPTs and mixtures of MPTs. In future, we plan to further explore the aforementioned connection and investigate applications thereof.

## References

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. Technical report, UCLA, 2020.

Adnan Darwiche. A differential approach to inference in Bayesian networks. *Journal of the ACM (JACM)*, 50(3): 280–305, 2003.

Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv preprint*, 2021.

Subhashis Ghosal and Aad van der Vaart. *Fundamentals of Nonparametric Bayesian Inference*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2017.

Timothy Hanson and Wesley O Johnson. Modeling regression error with a mixture of Pólya trees. *Journal of the American Statistical Association*, 97(460):1020–1033, 2002.

Timothy E Hanson. Inference for mixtures of finite Pólya tree models. *Journal of the American Statistical Association*, 101(476):1548–1565, 2006.

Michael Lavine. Some aspects of Pólya tree distributions for statistical modelling. *The Annals of Statistics*, 20(3): 1222 – 1235, 1992.

Yitao Liang, Jessa Bekker, and Guy Van den Broeck. Learning the structure of probabilistic sentential decision diagrams. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

R Daniel Mauldin, William D Sudderth, and Stanley C Williams. Pólya trees and random distributions. *The Annals of Statistics*, pages 1203–1221, 1992.

Lassi Meronen, Martin Trapp, and Arno Solin. Periodic activation functions induce stationarity. *Advances in Neural Information Processing Systems*, 34:1673–1685, 2021.

Pietro Muliere and Stephen Walker. A Bayesian nonparametric approach to survival analysis using Pólya trees. *Scandinavian Journal of Statistics*, 24(3):331–340, 1997.

Andrew A Neath. Pólya tree distributions for statistical modeling of censored data. *Journal of Applied Mathematics and Decision Sciences*, 7(3):175–186, 2003.

Susan M Paddock, Fabrizio Ruggeri, Michael Lavine, and Mike West. Randomized Pólya tree models for nonparametric Bayesian inference. *Statistica Sinica*, pages 443–460, 2003.

Tim Pearce, Russell Tsuchida, Mohamed Zaki, Alexandra Brintrup, and Andy Neely. Expressive priors in Bayesian neural networks: Kernel combinations and periodic functions. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 134–144. PMLR, 2020.

Robert Peharz. *Foundations of Sum-Product Networks for Probabilistic Modeling*. PhD thesis, Graz University of Technology, 2015.

Robert Peharz, Robert Gens, and Pedro Domingos. Learning selective sum-product networks. In *ICML Workshop on LTPM*, 2014.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 689–690. IEEE, 2011.

Andy Shih and Stefano Ermon. Probabilistic circuits for variational inference in discrete graphical models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:4635–4646, 2020.

Martin Trapp. *Sum-Product Networks for Complex Modelling Scenarios*. PhD thesis, Graz University of Technology, 2020.

Martin Trapp, Robert Peharz, Hong Ge, Franz Pernkopf, and Zoubin Ghahramani. Bayesian learning of sum-product networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6347–6358, 2019.

Martin Trapp, Robert Peharz, Franz Pernkopf, and Carl Edward Rasmussen. Deep structured mixtures of Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108, pages 2251–2261. PMLR, 2020.

Antonio Vergari, Alejandro Molina, Robert Peharz, Zoubin Ghahramani, Kristian Kersting, and Isabel Valera. Automatic Bayesian density analysis. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 5207–5215, 2019.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. *Advances in Neural Information Processing Systems (NeurIPS)*, 34: 13189–13201, 2021.

Han Zhao, Tameem Adel, Geoff Gordon, and Brandon Amos. Collapsed variational inference for sum-product networks. In *International Conference on Machine Learning (ICML)*, pages 1310–1318. PMLR, 2016.

Luping Zhao, Timothy E Hanson, and Bradley P Carlin. Mixtures of Pólya trees for flexible spatial frailty survival modelling. *Biometrika*, 96(2):263–276, 2009.

# Supplementary Material: On Priors in Bayesian Probabilistic Circuits and Multivariate Pólya trees

This appendix contains further derivations in Appendix A and details on the experiments in Appendix B. Moreover, we will summarize the notation used in the paper.

**General Notation** Scalars are written lowercase (*e.g.*, $x, y$) vectors are written lowercase bold (*e.g.*, $\boldsymbol{x}, \boldsymbol{y}$) and matrices are written uppercase bold (*e.g.*, $\boldsymbol{X}, \boldsymbol{Y}$). Furthermore, the following is used for general mathematical objects.

| | |
|---|---|
| $n$ | Number of data points, *e.g.*, size of training set |
| $p$ | Number of dimensions / features |
| $J$ | Depth of a finite Pólya tree |
| $\boldsymbol{x}_i$ | $i^{\text{th}}$ observation |
| $X$ | random variable |
| $\theta$ | Parameter |
| $\mathscr{P}(\boldsymbol{X})$ | Power set including empty set |
| $\mathbb{1}_{[\cdot]}$ | Indicator function |
| $\mathbb{R}$ | Reals |
| $\mathbb{B}$ | Booleans |
| $(\mathfrak{X}, \mathscr{X})$ | sample space |
| $A$ | measurable subset of $\mathfrak{X}$ |
| $\Pi$ | A partition |
| $\mathrm{PT}(c, \rho, \Pi)$ | A Pólya tree |
| $c > 0$ | Control parameter |
| $\rho(\cdot)$ | Prior parameter |
| $G_\theta$ | CDF of a parametric distribution |
| $g_\theta$ | PDF of a parametric distribution |
| $G$ | A draw from a Pólya tree |

**Notation on Probabilistic Circuits** The following notation is used for objects related to probabilistic circuits.

| | |
|---|---|
| $\mathcal{G}$ | Graph, *i.e.*, computational graph |
| $\psi$ | Scope function |
| $\mathrm{S}, \mathrm{P}, \mathrm{L}$ | Sum, Product and Leaf node (respectively) |
| $\mathrm{N}$ | Generic node, *i.e.*, a sum, product of leaf node |
| $\mathbf{N}$ | Set of nodes |
| $w_{\mathrm{S},\mathrm{N}}$ | Edge weight from S to N |

## A MATHEMATICAL DETAILS

### A.1 MIXTURE OF MULTIVARIATE PÓLYA TREES

In this work, we consider a mixture of multivariate Pólya trees (MPT) given as:

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \mid G \sim G \tag{4}$$

$$G \mid \theta \sim \mathrm{PT}_p^J(c, \rho, \Pi_\theta) \tag{5}$$

$$\theta \sim p(\theta) \tag{6}$$

where $G$ follows $G_\theta$ on sets of $\Pi_\theta$ and MPT has the parametric location-scale distribution $G_\theta$ as centering distribution.

In particular, we will assume that $g_\theta(\cdot)$ is the PDF of a Normal distribution denoted as $\phi_\theta$. Further, we denote the Normal CDF parameterized by $\mu, \Sigma$ as $\Phi_{\mu,\Sigma}(\cdot)$ and, in the case of a standard Normal, we use $\phi_0$ and $\Phi_0$ respectively. Note that, we have that $\mathbb{E}[\boldsymbol{Y}] = \mu$, $\mathrm{Cov}[\boldsymbol{Y}] = \Sigma$, and $p(\boldsymbol{Y} \in A) = \int_A \phi_{\mu,\Sigma}(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x}$ where $A \subset \mathbb{R}^d$.

To compute the likelihood of a vector $\boldsymbol{x} \in \mathbb{R}^p$, one first needs to find the path (sets of sub-regions) $\boldsymbol{x}$ falls into. For this, we define the following function:

$$k_\theta(j, d, \boldsymbol{z}) = \min\{\lfloor 2^j \Phi_0(z_d) + 1, 2^j\}\rfloor \tag{7}$$

which selects the sub-region at each level $j \in 1, \ldots, J$ respective to each dimension $d \in 1, \ldots, p$. Given:

$$\boldsymbol{k}_\theta(j, \boldsymbol{z}) = \begin{bmatrix} k_\theta(j, 1, \boldsymbol{z}) \\ \vdots \\ k_\theta(j, d, \boldsymbol{z}) \end{bmatrix} \tag{8}$$

we can compute the probabilities:

$$p_{\mathcal{X}}(\boldsymbol{k}) = \prod_{j=1}^{J} X_{e_j(\lceil k_1 2^{j-J} \rceil \ldots \lceil k_d 2^{j-J} \rceil)} \tag{9}$$

which allows us to obtain the likelihood given as:

$$\begin{aligned} p(\boldsymbol{x} \mid \mathcal{X}, \mu, \Sigma) = 2^{Jd} p_{\mathcal{X}}(\boldsymbol{k}_\theta(J, \Sigma^{-1/2}(\boldsymbol{x} - \mu))) \\ \det(\Sigma)^{-1/2} \phi_0(\Sigma^{-1/2}(\boldsymbol{x} - \mu)). \end{aligned} \tag{10}$$

We first reparameterize $\boldsymbol{x}$ to evaluate the mixture of MPTs using a standard Normal as the centring distribution. In practice, all computations are performed in log-space to ensure numerical stability.

### A.1.1 Bayesian Probabilistic Circuit construction of a Mixture of Multivariate Pólya trees

To model a mixture of MPTs, we define $G$ recursively as follows:

$$
\begin{aligned}
G &= \tilde{G}_{1:p}(\mathcal{X}) && (11)\\
\tilde{G}_{d_1:d_2}(A) &= G_{d_1:d_2}(A) \\
&\quad + \sum_{A^i \in \Pi(A)} \tilde{G}_{d_1:s_i}(A^i_{d_1:s_i}) \\
&\qquad \times \tilde{G}_{s_i+1:d_2}(A^i_{s_i+1:d_2}) && (12)\\
G_{d_1:d_2}(A) \mid \theta &\sim \mathrm{PT}^{J_{d_2-d_1}}_{d_1:d_2}(c, \rho, G_\theta, A) && (13)
\end{aligned}
$$

where $\Pi(A)$ denotes a partition of $A$ into disjoint sub-sets, $s_i$ denotes a splitting variable associated to each to each $A^i$, and as before $\theta \sim p(\theta)$. Note that the splitting variable might be different for each product node in the model.

Note that now, the likelihood function is also recursively given. Let $J_d$ be the number of layers if the input domain has $d \leq p$ dimensions. Then we have that:

$$
\begin{aligned}
p(\boldsymbol{x} \mid \mathcal{X}, \mu, \Sigma) &= f_{1:p}(\mathcal{X}, \Sigma^{-1/2}(\boldsymbol{x} - \mu)) \\
&\quad \det(\Sigma)^{-1/2} \phi_0(\Sigma^{-1/2}(\boldsymbol{x} - \mu)), && (14)
\end{aligned}
$$

with

$$
\begin{aligned}
f_{d_1:d_2}(A, \boldsymbol{z}) &= 2^{J_{d_2-d_1}} p_A(\boldsymbol{k}_\theta(J_{d_2}, \boldsymbol{z})) \\
&\quad \sum_{A^i \in \Pi(A)} f(A^i_{d_1:s_i}, \boldsymbol{z}_{d_1:s_i}) \times f(A^i_{s_i+1:d_2}, \boldsymbol{z}_{s_i+1:d_2}). && (15)
\end{aligned}
$$

Similar to before, we reparameterize $\boldsymbol{x}$ to evaluate the Bayesian PC using a standard Normal as centring distribution and perform all computations in log-space.

# B EXPERIMENTS

To evaluate the proposed approach, we used two synthetic data sets, both of which are generated based on the implementation provided in `https://github.com/wildart/NMoons.jl`.

**Two Moons:** The two moons data set uses $n_{\text{train}} = 400$ training samples generated with noise $\varepsilon = 0.1$ and repulse $(-0.25, -0.25)$. The test set is base on $n_{\text{test}} = 200$ samples generated with increased noise of $\varepsilon = 0.3$ and the same repulse.

**Two Circles:** The two circles data set uses $n_{\text{train}} = 400$ training samples generated with noise $\varepsilon = 0.1$ and translation of $(-0.25, -0.25)$. The test set is base on $n_{\text{test}} = 200$ samples generated with increased noise of $\varepsilon = 0.3$ and the same translation.

## B.1 EXPERIMENTAL SETUP

All experiments are based on the generative models described in Appendix A.1 and use a Normal density as the base distribution.

For simplicity we did not sample $\boldsymbol{V}$ (or $\boldsymbol{w}$) and $\theta$ jointly, but instead used a fixed sample of $\theta_1, \dots, \theta_T$ generated by injecting random noise, *i.e.*,

$$
\begin{aligned}
\mu_t &= \bar{\mu} + \epsilon, \quad \epsilon \sim \mathrm{N}(0, 1) && (16)\\
\sigma_t &= 2\bar{\sigma}_t + \epsilon, \quad \epsilon \sim \mathrm{U}(0, 0.5) && (17)
\end{aligned}
$$

where $\bar{\mu}$ is the sample mean and $\bar{\sigma}^2$ the sample variance. Moreover, we assumed that the base distribution has diagonal covariance structure, simplifying the computations.

## B.2 QUANTITATIVE COMPARISION

To quantitatively compare finite mixtures of MPTs to Bayesian PCs, we assessed the Kullback–Leibler (KL) divergence from the posterior expectation of a mixture of MPTs to the posterior expectation of a Bayesian PC and compared the test log-likelihood for posterior draws of both models. The results are found in Fig. 3 and Fig. 4 respectively. Fig. 5 shows an additional visualisation comparing the computational costs and approximation quality.

To compute the KL divergences, we estimated the divergence using Monte Carlo (MC) integration using $100k$ samples. For comparison, we also estimated the entropy of the posterior expectation of a mixture of MPTs estimated based on the same $100k$ samples. We did not systematically assess the variance, but found that $100k$ samples are sufficient to obtain consistent results across multiple reruns. To estimate the test log-likelihoods, we used ten random draws from the posterior and averaged across the draws.

In all experiments, we assessed the performance for varying values of the control parameter $c$. Note that the parameter $c$ controls how much the Pólya tree (or the deterministic Bayesian PC) can deviate from the centring distribution (larger $c$ means less deviation is allowed). Larger values of $c$ require a larger number of observations to let the data overwhelm the effect of the centring distribution.

## B.3 QUALITATIVE COMPARISON

Fig. 6 shows 2D density function draws from a finite mixture of MPTs and draws from a Bayesian PC for different values of $J_2$ and different values of the control parameter $c$. In all cases, $J = 9 \approx \log_2(n_{\text{train}})$, $J_1 = J - J_2$ and $\rho(j) = j^2$. We can observe that draws from a Bayesian PC with small $J_2$ result in grid-like density functions with strong discontinuities and values of $J_2 = 4$ or greater provide visually similar density draws to those generated from a finite mixture of MPTs.

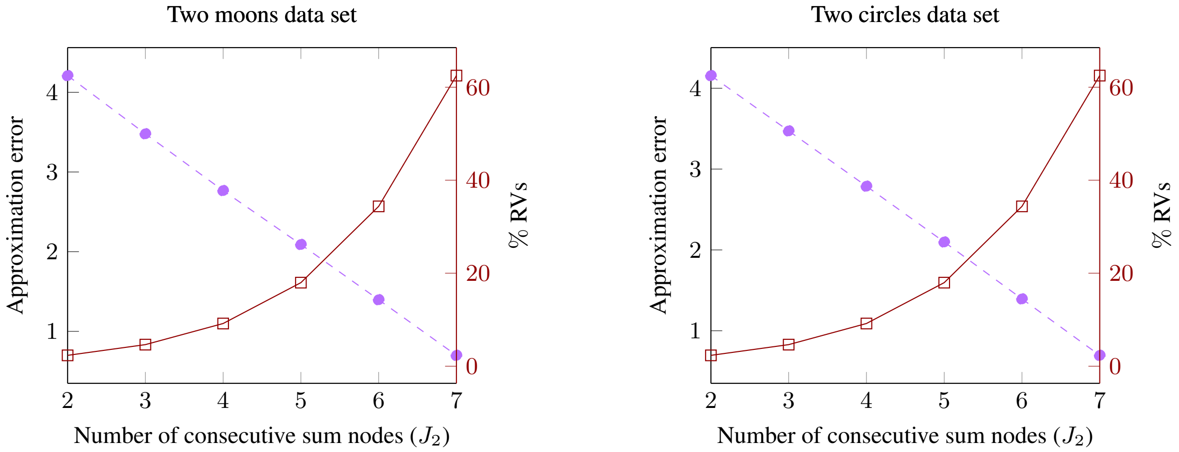


Figure 5: Average approximation error of the test set log-likelihoods for draws of a Bayesian probabilistic circuit compared to draws from a mixture of MPTs with $c = 1$. Red lines show the fraction of RVs used in the Bayesian probabilistic circuit, reflecting the computational and memory costs compared to a mixture of MPTs.

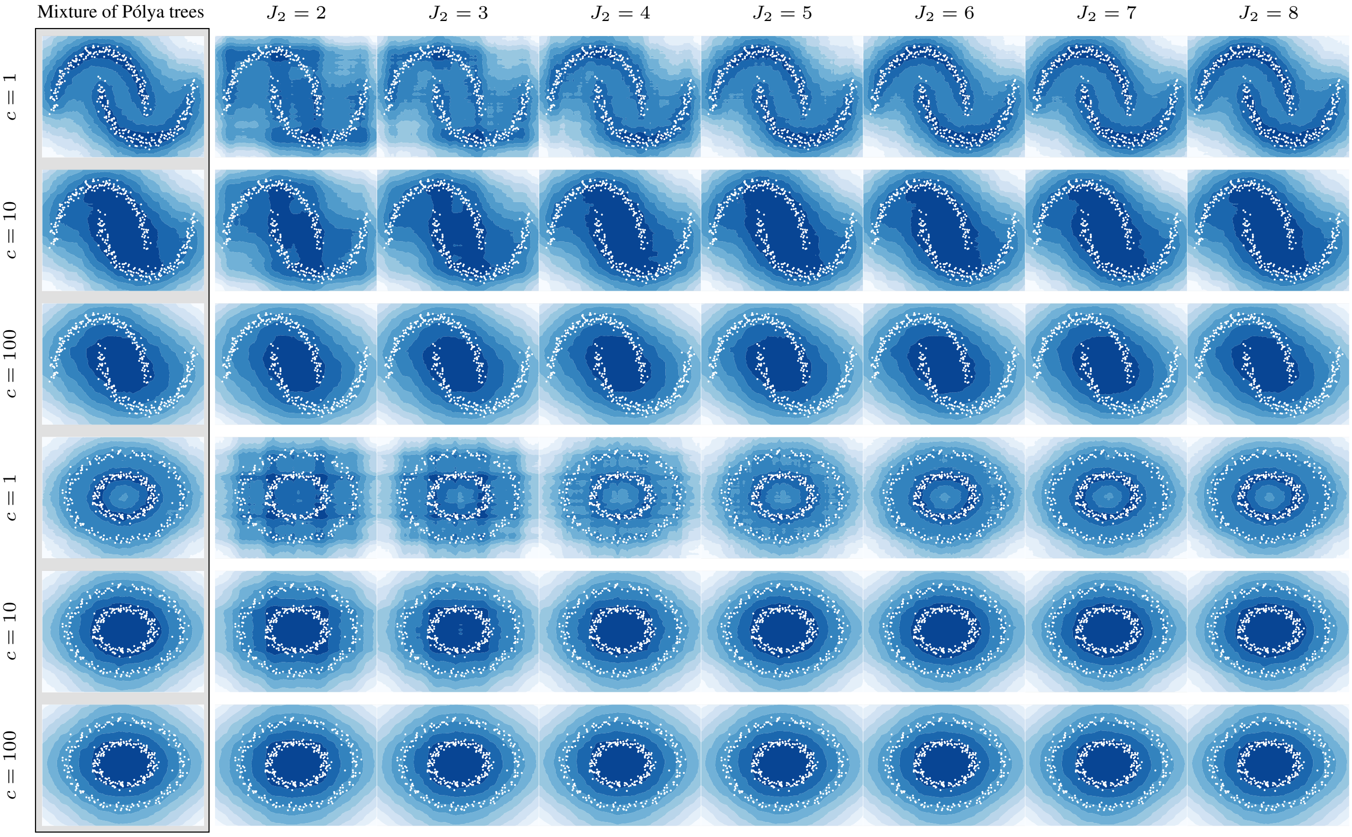


Figure 6: Comparison of 2D density function draws from a finite mixture of multivariate Pólya trees and draws from a Bayesian probabilistic circuit for different values of $J_2$ and different values of $c$. The top three rows show results on the two moons data set, and the bottom rows show results on the two circles data set. Brighter colours indicate a higher probability.