Mixtures of All Trees

Nikil Roashan Selvam
UCLA Computer Science
nikilrselvam@ucla.edu

Honghua Zhang
UCLA Computer Science
hzhang19@cs.ucla.edu

Guy Van den Broeck
UCLA Computer Science
guyvdb@cs.ucla.edu

Abstract

Tree-shaped graphical models are widely used for their tractability. However, they unfortunately lack expressive power as they require committing to a particular sparse dependency structure. We propose a novel class of generative models called mixtures of *all* trees: that is, a mixture over all possible (n^{n-2}) tree-shaped graphical models over n variables. We show that it is possible to parameterize this Mixture of All Trees (MoAT) model compactly (using a polynomial-size representation) in a way that allows for tractable likelihood computation and optimization via stochastic gradient descent. Furthermore, by leveraging the tractability of tree-shaped models, we devise fast-converging conditional sampling algorithms for approximate inference, even though our theoretical analysis suggests that exact computation of marginals in the MoAT model is NP-hard. Empirically, MoAT achieves state-of-the-art performance on density estimation benchmarks when compared against powerful probabilistic models including hidden Chow-Liu Trees.

1 INTRODUCTION

Probabilistic graphical models (PGMs) have been extensively studied due to their ability to exploit structure in complex high-dimensional distributions and yield compact representations. The underlying graph structure of these models typically dictates the trade-off between expressive power and tractable probabilistic inference. On one end of the spectrum lie tree-shaped graphical models including Chow-Liu trees (Chow and Liu, 1968), where the underlying graph is a spanning tree $T = (V, E)$ on n vertices. Tree distributions allow for efficient sampling and exact inference on a variety of queries such as computing marginals (Pearl, 1988; Darwiche, 2003) and are widely used in practice (Zhang and Poon, 2017). However, by committing to a single sparse dependency structure (by choice of spanning tree) their expressive power is limited. On the other end of the spectrum, we have densely connected graphical models such as Markov random fields (MRFs) (Koller and Friedman, 2009; Rabiner and Juang, 1986), Bayesian networks (Pearl, 1988), and factor graphs (Loeliger, 2004), which excel at modelling arbitrarily complex dependencies (Mansinghka et al., 2016), but do so at the cost of efficient computation of marginal probabilities. This spectrum and the underlying tradeoff extends beyond graphical models to generative models at large. For instance, deep generative models like variational autoencoders (VAEs) (Maaløe et al., 2019) are extremely expressive, but do not support tractable inference.

In this work, we propose a novel class of probabilistic models called Mixture of *All* Trees (MoAT): a mixture over all possible (n^{n-2}) tree-shaped MRFs over n variables; e.g., MoAT represents a mixture over 10^{196} components when modeling joint distributions on 100 variables. Despite the large number of mixture components, MoAT can be compactly represented by $O(n^2)$ parameters, which are shared across the tree components. The MoAT model strikes a new balance between expressive power and tractability: (i) it concurrently models all possible tree-shaped dependency structures, thereby greatly boosting expressive power; (ii) by leveraging the tractability of the spanning tree distributions and the tree-shaped MRFs, it can not only tractably compute *normalized likelihood* but also efficiently estimate *marginal probabilities* via sampling. In addition, as a fixed-structure model, MoAT circumvents the problem of structure learning, which plagues most probabilistic graphical models.

This paper is organized as follows. Section 2 defines the MoAT model and shows the tractability of exact (normalized) likelihood computation despite the presence of super-exponentially many mixture components. In Section 3, we discuss the MoAT model's parameterization and learning, and demonstrate state-of-the-art performance on density estimation for discrete tabular data. Next, in Section 4, we discuss the tractability of marginals and MAP inference in MoAT and prove hardness results. Finally, we view MoAT as a latent variable model and devise fast-converging impor-

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

tance sampling algorithms that let us leverage the extensive literature on inference in tree distributions.

2 MIXTURES OF ALL TREES

In this section, we propose mixture of all trees (MoAT) as a new class of probabilistic models. We first introduce tree-shaped Markov random fields (MRFs) and define the MoAT model as a mixture over all possible tree distributions weighted by the spanning tree distribution. Then, we demonstrate how to tractably compute normalized likelihood on the MoAT model.

2.1 Mixture of Tree-shaped Graphical Models

A *tree-shaped MRF* with underlying graph structure $G(V, E)$ represents a joint probability distribution $\Pr_G$ over n random variables $\mathsf{X} = X_1, \quad , X_n$ by specifying their univariate and pairwise marginal distributions. Specifically, assuming G is a tree with vertex set $V = \{1, \quad , n\}$, we associate with each edge $(u, v) \in E$ a pairwise marginal distribution $P_{uv}(X_u, X_v)$ and each vertex u a univariate marginal distribution $P_v(X_v)$. Assuming that P_{uv} and P_v are consistent, then the normalized joint distribution $\Pr_G$ is given by Meilă et al. (2000):

$$\Pr_G(\mathsf{x}) = \frac{\prod_{(u,v)\in E} P_{uv}(x_u, x_v)}{\prod_{u\in V} P_v(x_v)^{\deg v - 1}}, \qquad (1)$$

where $\mathsf{x} = (x_1, \quad , x_n)$ denotes assignment to X and $\deg v$ denotes the degree of v in G; see $\Pr_1(X_1, X_2, X_3)$ in Figure 1 as an example tree-shaped MRF.

Despite the tractability of tree-shaped MRFs, they suffer from the problem of limited expressive power. To improve the expressive power, prior works propose to learn mixtures of tree models (Anandkumar et al., 2012; Meilă et al., 2000), where they focus on simple mixtures of a few trees, and propose EM algorithms for parameter and structure learning. This idea, however, suffers from several limitations. Firstly, while it is known how to optimally pick a single tree distribution with respect to the training data via the Chow-Liu algorithm (Chow and Liu, 1968), no known closed form solution exists for picking the optimal set of tree distributions as mixture components from the super-exponentially many possible choices for spanning trees. Secondly, by having a small fixed number of (even possibly optimal) mixture components, the model forces us to commit to a few sparse dependency structures that might not be capable of capturing complex dependencies anyway.

Though mixture of trees model becomes more expressive as more tree structures are included, the number of parameters increases with the number of mixture components, which seem to suggest that a mixture over a large number of tree components is infeasible. Despite this, we propose the mixture of **all** trees model (MoAT), a polynomial-size representation for the mixture over all possible (super-exponentially many) tree-shaped MRFs.

Formally, we define:

$$\Pr_{\text{MoAT}}(\mathsf{x}) = \frac{1}{Z} \sum_{T\in \mathsf{ST}(K_n)} \left(\prod_{e\in T} w_e \right) \Pr_T(\mathsf{x}) \qquad (2)$$

where K_n denotes the complete graph on n vertices, $\mathsf{ST}(G)$ denotes the set of spanning trees of a connected graph G, and Z is the normalization constant. Each mixture component is a tree-shaped MRF $\Pr_T$ weighted by $\prod_{e\in T} w_e$, that is, *product of the edge weights of the tree*. Note that we define the weight of each tree to be proportional to its probability in the *spanning tree distribution* (Borcea et al., 2009), which is tractable, allowing for efficient likelihood computation on MoAT (Section 2.2.)

Though a MoAT model represents a mixture over super-exponentially many tree-shaped MRFs, the number of parameters in MoAT is polynomial-size due the the *parameter sharing* across its mixture components. Specifically, all tree-shaped MRFs $\Pr_T(\mathsf{x})$ share the same univariate and pair-wise marginals (i.e., $P_u(x_u)$ and $P_{uv}(x_u, x_v)$); in addition, each edge in the graph K_n is parameterized by a positive weight w_{uv}. To summarize, a MoAT model over n variables has $O(n^2)$ parameters.

Figure 1 shows an example MoAT model over 3 binary random variables X_1, X_2, X_3, for which there are 3 possible spanning trees. Note that each of the mixture components (tree distributions) share the same set of marginals, but encode different distributions by virtue of their different dependency structures.

For example, for the distribution represented in Figure 1,

$$\begin{aligned}
&\Pr_{\text{MoAT}}(X_1 = 1, X_2 = 0, X_3 = 1) \\
&= \frac{1}{Z} \sum_{T\in \mathsf{ST}(K_n)} \left(\prod_{e\in T} w_e \right) T(\mathsf{x}) \\
&= \frac{1}{2 \quad 3+3 \quad 6+6 \quad 2} \left[2 \quad 3 \quad \frac{0.5 \quad 0.3}{0.7} \right. \\
&\quad \left. + \quad 2 \quad 6 \quad \frac{0.5 \quad 0.2}{0.6} \quad + \quad 3 \quad 6 \quad \frac{0.2 \quad 0.3}{0.5} \right]
\end{aligned}$$

By Cayley's formula (Chaiken and Kleitman, 1978), the number of spanning trees increases super-exponentially with respect to the number of random variables, thus preventing us from evaluating them by enumeration.

2.2 Tractable Likelihood for MoAT

Despite a super-exponential number (n^{n-2}) of mixture components, we show that computing (normalized) likelihood on MoAT is tractable. Our approach primarily leverages the tractability of spanning tree distributions and their compact

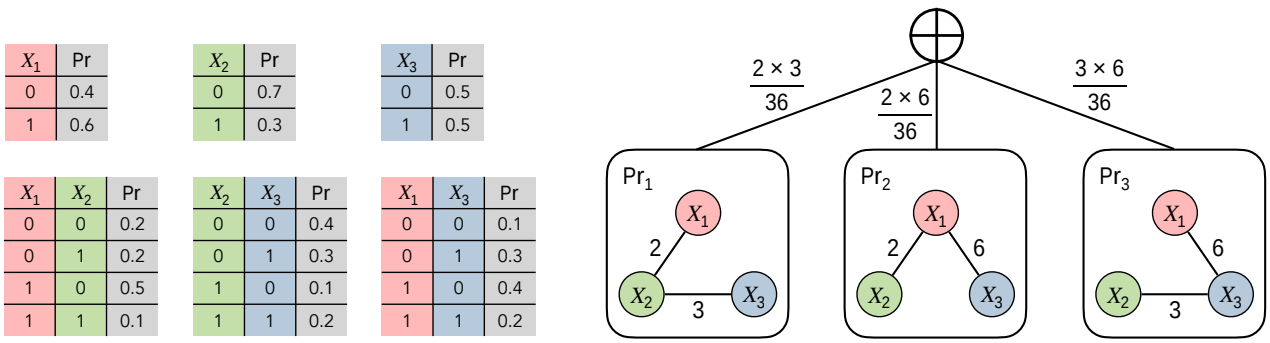


Figure 1: An example MoAT distribution over 3 binary random variables X_1, X_2, and X_3. The summation at the top denotes a mixture distribution where the mixture weights are given by the weights of the corresponding spanning tree (shown on the edges). The tables on the left shows the univariate and pairwise marginal distributions, which are shared across the mixture components (3 possible spanning trees).

representation as *probability generating polynomials*, which has been extensively studied in the context of machine learning (Li et al., 2016; Mariet et al., 2018; Robinson et al., 2019; Zhang et al., 2021).

Definition 1. Let $\Pr(\,)$ be a probability distribution over n binary random variables $\mathbf{X} = X_1, X_2, \ldots, X_n$, then the *probability generating polynomial* for $\Pr$ is defined as

$$\sum_{\mathbf{x} \in \{0,1\}^n} \Pr(\mathbf{X} = \mathbf{x}) \prod_{i \text{ s.t. } x_i = 1} z_i \quad ,$$

where each z_i is an indeterminate associated with X_i.

To define spanning tree distributions and present their representation as probability generating polynomials, we first introduce some notation. Let $G = (V, E)$ be a connected graph with vertex set $V = \{1, \ldots, n\}$ and edge set E. Associate to each edge $e \in E$ an indeterminate z_e and a weight $w_e \in \mathbb{R}_{\geq 0}$. If $e = \{i, j\}$, let A_e be the $n \times n$ matrix where $A_{ii} = A_{jj} = 1$, $A_{ij} = A_{ji} = -1$ and all other entries equal to 0. Then the *weighted Laplacian* of G is given by $L(G) = \sum_{e \in E} w_e z_e A_e$,

For instance, the weighted Laplacian for the example MoAT distribution in Figure 1 is

$$\begin{bmatrix} 2z_{ab} + 6z_{ac} & -2z_{ab} & -6z_{ac} \\ -2z_{ab} & 2z_{ab} + 3z_{bc} & -3z_{bc} \\ -6z_{ac} & -3z_{bc} & 3z_{bc} + 6z_{ac} \end{bmatrix}$$

Using $L(G)_{\setminus\{i\}}$ to denote the principal minor of $L(G)$ that is obtained by removing its i^{th} row and column, by the Matrix Tree Theorem (Chaiken and Kleitman, 1978), the probability generating polynomial for the spanning tree distribution is given by:

$$\det(L(G)_{\setminus\{i\}}) = \sum_{T \in \mathsf{ST}(G)} \prod_{e \in T} w_e z_e \tag{3}$$

Now we derive the formula for computing $\Pr_{\text{MoAT}}(\mathbf{x})$ efficiently. We first set $G = K_n$ and $z_e = \frac{P_{uv}}{P_u P_v}$ and define:

$$L^* := L(K_n)_{\setminus\{i\}} \;\; {}_{z_e = \frac{P_{uv}}{P_u P_v}};$$

and it follows from Equation 3 that

$$\det(L^*) = \sum_{T \in \mathsf{ST}(K_n)} \prod_{e \in T} w_e \prod_{(u,v) \in T} \frac{P_{uv}}{P_u P_v};$$

note that $\prod_{(u,v) \in T} P_u P_v = \prod_u P_u^{\deg(u)}$; hence,

$$\begin{aligned} \det(L^*) &= \prod_{v \in V} \frac{1}{P_v} \sum_{T \in \mathsf{ST}(K_n)} \prod_{e \in T} w_e \prod_{v \in V} P_v^{\deg v - 1} \\ &= \prod_{v \in V} \frac{Z}{P_v} \Pr_{\text{MoAT}}, \end{aligned}$$

where the second equality follows from the definition of MoAT (Equation 2). Finally, we multiply both sides by $\prod_{v \in V} P_v$ and divide by Z thus $\Pr_{\text{MoAT}}(\mathbf{x})$ can be evaluated as:

$$\Pr_{\text{MoAT}}(\mathbf{x}) = \frac{1}{Z} \prod_{v \in V} P_v(x_v) \det(L^*_{\mathbf{x}}).$$

Note that the normalization constant of the MoAT model $Z = \sum_{T \in \mathsf{ST}(K_n)} \prod_{e \in T} w_e$ can be evaluated efficiently as a determinant by replacing the indeterminate z_e with the constant 1. As the computational bottleneck is the determinant calculation, the time complexity is upper bounded as $O(n^\omega)$, where ω is the matrix multiplication exponent.

3 DENSITY ESTIMATION

In the previous section, we introduced the MoAT model and described how we can compute likelihood tractably. In this section, we describe how to parameterize the MoAT model

in a way that is amenable to learning and subsequently effective density estimation on real world datasets. There are few desirable properties we seek from this parameterization (of univariate and pairwise marginals in particular). Firstly, we need to parameterize the marginals in way that are *consistent* with each other. This is essential as it guarantees that all tree-shaped mixture components (Equation 1) in the MoAT model are normalized. Secondly, we want our parameterization to capture the entire space of *consistent* combinations of univariate and pairwise marginals. In particular, this also ensures that every tree distribution is representable by our parameterization.

3.1 MoAT Parameter Learning

For a MoAT model over n binary random variables $V = \{X_1, X_2, ..., X_n\}$, we propose the following parameterization (as illustrated in Figure 2):

- Edge weights: $w_e \in \mathbb{R}_{\geq 0}$ for $e \in \binom{V}{2}$.
- Univariate marginals: $p_v = P(X_v = 1) \in [0, 1] \; \forall v \in V$.
- Pairwise marginals: $p_{uv} = P(X_u = 1, X_v = 1) \in [\max(0, p_u + p_v - 1), \min(p_u, p_v)]$ for $\{u, v\} \in \binom{V}{2}$.

X_1	Pr
0	$1 - \alpha_1$
1	α_1

X_2	Pr
0	$1 - \alpha_2$
1	α_2

X_3	Pr
0	$1 - \alpha_3$
1	α_3

X_1	X_3	Pr
0	0	$1 - \alpha_1 - \alpha_3 - \beta_{13}$
0	1	$\alpha_3 - \beta_{13}$
1	0	$\alpha_1 - \beta_{13}$
1	1	β_{13}

X_2	X_3	Pr
0	0	$1 - \alpha_2 - \alpha_3 - \beta_{23}$
0	1	$\alpha_3 - \beta_{23}$
1	0	$\alpha_2 - \beta_{23}$
1	1	β_{23}

X_1	X_2	Pr
0	0	$1 - \alpha_1 - \alpha_2 - \beta_{12}$
0	1	$\alpha_2 - \beta_{12}$
1	0	$\alpha_1 - \beta_{12}$
1	1	β_{12}

Figure 2: Parameterization for multivariate and univariate marginals for the example distribution on three binary random variables. The α_is and β_{ij}s are the free parameters.

As mentioned in Section 2, to ensure that all the mixture components of MoAT are normalized, our parameterization for P_u and P_{uv} needs to be consistent; specifically, they need to satisfy the following constraints:

- $P(X_v = 0) + P(X_v = 1) = 1$ for all $v \in V$.
- $\sum_{a \in \{0,1\}} P(X_u = a, X_v = b) = P(X_v = b) \; \forall b \in \{0, 1\}, \forall \{u, v\} \in \binom{V}{2}$.
- $\sum_{(a,b) \in \{0,1\}^2} P(X_u = a, X_v = b) = 1 \; \forall \{u, v\} \in \binom{V}{2}$.

Lemma 1. For any distribution $\Pr(\,)$ over binary random variables $X_1, \ldots, X_n$, there exists a set of parameters (i.e., p_v and p_{uv}) in our hypothesis space such that $\Pr(X_u) = P_u(X_u)$ and $\Pr(X_u, X_v) = P_{uv}(X_u, X_v)$ for all $1 \leq u, v \leq n$; i.e., the univariate and pair-wise marginals of $\Pr$ are the same as P_u and and P_{uv}.

See appendix for proof. This lemma shows that the MoAT parameterization is not just valid, but also fully general in the sense that it covers all possible *consistent* combinations of univariate and pairwise marginals. Further, the MoAT parameterization naturally extends to categorical variables. For categorical random variables $V = \{X_1, X_2, ..., X_n\}$, let $\mathsf{val}(X_i) = \{1, 2, \ldots, k_i\}$. It is easy to see that the values $P(X_v = i)$ for $i \in \{1, 2, \ldots, k_v - 1\}$ uniquely determine the univariate marginals. Similarly, the values $P(X_u = i, X_v = j)$ for $(i, j) \in \{1, 2, \ldots, k_u - 1\} \times \{1, 2, \ldots, k_v - 1\}$ uniquely determine the pairwise marginals. This extension is provably valid, but not fully general. For MoAT over categorical variables, whether there exists a fully general parameterization (i.e., Lemma 1 holds) is unknown. See appendix for a detailed discussion.

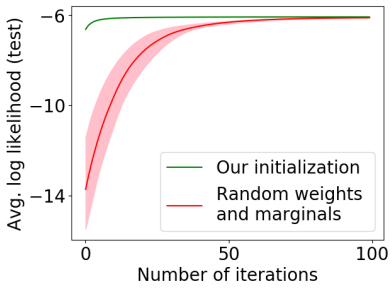
Parameter Learning For individual tree distributions, the optimal tree structure (as measured by KL divergence from training data) is the maximum weight spanning tree of the complete graph, where edge weights are given by mutual information between the corresponding pairs of variables (Chow and Liu, 1968). Following this intuition, we use mutual information to initialize w_e; besides, we also initialize the univariate and pairwise marginals of the MoAT model by estimating them from training data. Finally, given our parameter initialization, we train the MoAT model by performing maximum likelihood estimation (MLE) via stochastic gradient descent.

It is worth noting that *our parameter initialization is deterministic*. We perform ablation studies to check the effectiveness of our initialization. As shown in Figure 3, compared to random initialization, we observe that our special initialization always leads to better initial log likelihood, faster convergence and better final log likelihood.

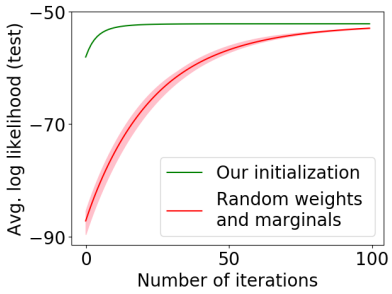
3.2 Density Estimation via MoAT

We evaluate MoAT on a suite of density estimation datasets called the Twenty Datasets (Van Haaren and Davis, 2012), which contains 20 real-world datasets covering a wide range of application domains including media, medicine, and retail. This benchmark has been extensively used to evaluate tractable probabilistic models. We compare MoAT against two baselines: (1) hidden Chow-Liu trees (HCLTs) (Liu and Van den Broeck, 2021), which are a class of probabilistic models that achieve state-of-the-art performance on the Twenty Datasets benchmark and (2) the mixture of trees model (MT) (Meilă et al., 2000).

Table 1 summarizes the experiment results. MoAT outperforms both HCLT and MT on 14 out of 20 datasets. In particular, the MoAT model beats baselines by large margins on all datasets with more than 180 random variables. It is also worth noting that despite having fewer parameters ($O(n^2)$) than MT ($O(k \cdot n^2)$, where k is the number of mix-



(a) NLTCS Dataset



(b) Jester Dataset

Figure 3: Average log-likelihood throughout training on datasets across various data dimensionalities: our initialization vs. random initialization (averaged over 5 runs).

Dataset	# vars	MoAT	HCLT	MT
nltcs	16	-6.07	**-5.99**	-6.01
msnbc	17	-6.43	**-6.05**	-6.07
kdd	65	**-2.13**	-2.18	**-2.13**
plants	69	-13.50	-14.26	**-12.95**
baudio	100	**-39.03**	-39.77	-40.08
jester	100	**-51.65**	-52.46	-53.08
bnetflix	100	**-55.52**	-56.27	-56.74
accidents	111	-31.59	**-26.74**	-29.63
tretail	135	**-10.81**	-10.84	-10.83
pumsb	163	-29.89	**-23.64**	-23.71
dna	180	-87.10	**-79.05**	-85.14
kosarek	190	**-10.57**	-10.66	-10.62
msweb	294	**-9.80**	-9.98	-9.85
book	500	**-33.46**	-33.83	-34.63
tmovie	500	**-49.37**	-50.81	-54.60
cwebkb	839	**-147.70**	-152.77	-156.86
cr52	889	**-84.78**	-86.26	-85.90
c20ng	910	**-149.44**	-153.4	-154.24
bbc	1058	**-243.82**	-251.04	-261.84
ad	1556	**-15.30**	-16.07	-16.02

Table 1: Comparison of average log likelihood of MoAT, HCLT, and MT across the Twenty Datasets benchmarks. Best results are presented in bold.

ture components in MT), MoAT almost always outperforms MT, with the exception of a few smaller datasets, where MoAT does not have enough parameters to fit the data well.

4 ON THE HARDNESS OF MARGINALS AND MAP INFERENCE

In this section, we prove the hardness of semiring queries (which is a generalization of marginals) and Maximum a posteriori (MAP) inference on the MoAT model.

4.1 On the Hardness of Computing Marginals

First, we define the notion of semiring queries.

Definition 2. Semiring Queries (SQ): Let $p(\mathbf{X})$ be a real-valued function over random variables $\mathbf{X}$. The class of semiring queries Q_{F} is the set of queries that compute values of the following form:

$$f(\mathbf{e}) = \bigoplus_{\mathbf{z}} p(\mathbf{z}, \mathbf{e})$$

where $e \sqsubseteq \mathrm{val}(\mathbf{E})$ is a partial configuration for any subset of random variables $\mathbf{E} \subseteq \mathbf{X}$, and $\mathbf{Z} = \mathbf{X} \setminus \mathbf{E}$ is the set of remaining random variables.

When the semiring sum/product operations correspond to the regular sum/product operations and the function p is a likelihood function, the semiring query $f(\mathbf{e})$ actually computes marginal probabilities.

In fact, if p is the likelihood function for the MoAT model, for an assignment $\mathbf{e}$ to $\mathbf{E} \subseteq \mathbf{X}$,

$$f(\mathbf{e}) = \frac{1}{Z} \left(\sum_{\mathbf{z}} \sum_{T \in \mathsf{ST}(K_n)} \prod_{v \in V} P_v(x_v) \right) \left(\prod_{(u,v) \in T} w_{(u,v)} \frac{P_{uv}(x_u, x_v)}{P_u(x_u) P_v(x_v)} \right),$$

where $Z = \sum_{T \in \mathsf{ST}(K_n)} \prod_{e \in T} w_e$ is the normalization constant and $\mathbf{z}$ enumerates over all instantiations of $\mathbf{Z} = \mathbf{X} \setminus \mathbf{E}$. Thus, in this case, $f(\mathbf{e})$ actually computes marginals in the MoAT model. However, the generality of the semiring queries allows for negative parameter values and hence negative "probabilities", which we leverage to prove hardness of semiring queries on the MoAT model.

Since most marginal computation algorithms on tractable probabilistic models (such as the jointree algorithm which relies on variable elimination (Zhang and Poole, 1996; Dechter, 1996) and circuit compilation based methods (Chavira and Darwiche, 2008; Darwiche, 2002)) are semiring generalizable (Wachter et al., 2007; Kimmig et al., 2017; Bacchus et al., 2009), the hardness of semiring queries on the MoAT model would strongly suggest the hardness of marginal computation. In other words, the hardness of semiring queries would rule out most marginal inference techniques in the literature as they perform purely algebraic computations on the parameter values without any restric-

tions/assumptions on the range of these values. We dedicate the rest of this subsection to establishing the same, while deferring most technical proof details to the appendix.

Theorem 1. Computation of semiring queries on the MoAT model is NP-hard.

Proof. To prove the hardness of semiring queries, we proceed by a reduction from the subset spanning tree problem (denoted SST), which we define below.

Lemma 2. Define SST as the following decision problem: given a connected graph $G = (V, E)$ and a subset $K \quad V$ of the vertices, decide if there exists a spanning tree of G whose leaves are exactly K. SST is NP-hard.

Consider an arbitrary connected graph $G = (V, E)$ on $jVj = n \quad 3$ vertices and a subset of vertices $K \quad V$. Set MoAT likelihood function parameters on n binary random variables $\mathsf{X} = \mathit{f}X_1, X_2, \ldots, X_n\mathit{g}$ (corresponding to the vertices of G) as follows:

- $0 < \epsilon < 1$
- $w_e = \begin{cases} 1, & e = \mathit{f}i, j\mathit{g} \ \mathit{2} \ E \\ 0, & \text{otherwise} \end{cases}$
- $\mathit{val}(X_i) = \mathit{f}0, 1\mathit{g}.$
- $P_v(0) = \epsilon, P_v(1) = \quad 1$ for all $v \ \mathit{2} \ V$.
- $P_{uv}(\alpha, \beta) = \begin{cases} \epsilon, & \alpha = \beta = 0 \\ 0, & \alpha = \beta = 1 \\ \epsilon, & \text{otherwise} \end{cases}$

One can intuitively interpret an assignment of 1 as corresponding to labelling a node as a leaf, and 0 as marking it as unknown. The univariate and pairwise marginals have been carefully chosen to ensure that any tree assigns higher probability to assignments where all the nodes assigned 1 are leaves in the tree and lower probabilities to assignments where one or more nodes that are assigned 1 are actually internal nodes. In fact, for any tree, there exists a likelihood separation of ϵ between assignments that agree on the leaves and those that do not. By assigning 1 to all the variables in $K \quad V$ and 0 to others, and by choosing a sufficiently small ϵ, we can now effectively use the MoAT likelihood as an indicator for the presence of an spanning tree whose leaves are a superset of K. More impressively, we can exactly count the number of spanning trees that satisfy the desired property, and we formalize the same in the following lemma.

Lemma 3. Let x be a complete assignment, and denote by $\mathsf{ONES}(\mathsf{x})$ the set of variables are are set to 1 in x. Denote by jxj the value $\mathit{j}\mathsf{ONES}(\mathsf{x})\mathit{j}$ and $\mathsf{LEAVES}(T)$ the set of leaves of a spanning tree T. Let k be the number of spanning trees T of G with $\mathsf{ONES}(\mathsf{x}) \quad \mathsf{LEAVES}(T)$. Then,

$$\begin{pmatrix} \frac{k}{\epsilon^{n-2}} \quad Z \ p(\mathsf{x}) \quad \frac{k}{\epsilon^{n-2}} + \frac{n^n \ ^2}{\epsilon^{n-3}}, & \mathit{jxj}\%2 = 0 \\ \frac{-k}{\epsilon^{n-2}} + \frac{-n^n \ ^2}{\epsilon^{n-3}} \quad Z \ p(\mathsf{x}) \quad \frac{-k}{\epsilon^{n-2}}, & \mathit{jxj}\%2 = 1 \end{pmatrix}$$

See appendix for proof.

Corollary 1. Let $\epsilon < \frac{1}{2^{n+1} \cdot n^{n-2}}$. The number of spanning trees T with $K \quad \mathsf{LEAVES}(T)$ is given by $\mathit{b}Z \ \epsilon^{n-2} \ p(\mathsf{x})\mathit{e}$, where $x_i = 1$ if and only if $i \ \mathit{2} \ K$ (that is, x is the assignment that assigns 1 to all the variables in K and 0 to all the other variables), $\mathit{b}x\mathit{e}$ denotes the closest integer to x.

Proof. Let k be the number of spanning trees T with $K \quad \mathsf{LEAVES}(T)$. When $\mathit{jxj}\%2 = 0$, $k \quad Z\epsilon^{n-2}p(\mathsf{x}) \quad k + \epsilon n^{n-2} \quad k + \frac{1}{2^{n+1}}$. Thus, $\mathit{b}Z\epsilon^{n-2}p(\mathsf{x})\mathit{e} = k$ as desired. An analogous proof holds for the case of $\mathit{jxj}\%2 = 1$. □

Note that $\frac{P_{uv}}{P_u P_v} \quad 0$, and hence the sign of $p(\mathsf{x})$ depends solely on the parity of jxj. Thus, we can leverage the inclusion-exclusion formula to count spanning trees T with $K = \mathsf{LEAVES}(T)$ using expressions for number of spanning trees T with $K \quad \mathsf{LEAVES}(T)$ given by Corollary 1.

Lemma 4. The number of spanning trees T with $K = \mathsf{LEAVES}(T)$ is given by $\mathit{b}Z\epsilon^{n-2}f(\mathsf{e})\mathit{e}$.

Proof Sketch. From the inclusion-exclusion formula we obtain that the number of spanning trees T with $K = \mathsf{LEAVES}(T)$ (upto sign) is given by

$$\begin{aligned} &\sum_{K \subseteq L} (-1)^{|L|} \sum_{T \in \mathsf{ST}(G)} \mathbf{1}(L \quad \mathsf{LEAVES}(T)) \\ &= \sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} (-1)^{|\mathsf{x}|} \ \mathit{b}Z\epsilon^{n-2}p(\mathsf{x})\mathit{e} \\ &= \mathit{b}Z\epsilon^{n-2} \sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} p(\mathsf{x})\mathit{e} \\ &= \mathit{b}Z\epsilon^{n-2}f(\mathsf{e})\mathit{e} \end{aligned}$$

□

We now obtain that there exists a spanning tree T with $K = \mathsf{LEAVES}(T)$ if and only if $\mathit{b}Z\epsilon^{n-2}f(\mathsf{e})\mathit{e} > 0$. This completes the reduction from SST, as desired. □

It is worth re-emphasizing the strength of this hardness result in the context of marginal computation, in that it eliminates all marginal inference algorithms that are agnostic to parameter values (which is, to the best of our knowledge, all possible known exact marginal inference techniques in literature). This opens up an interesting question about new classes of marginal computation algorithms that are not parameter-value agnostic.

4.2 On the Hardness of MAP Inference

In this section, we prove that maximum-a-posteriori (MAP) inference (i.e., computing the most likely assignment) for the MoAT model is NP-hard via a reduction from the 3-coloring problem (Lovász, 1973).

Theorem 2. MAP inference for MoAT is NP-hard.

Proof. Consider an arbitrary connected graph $G = (V, E)$ on $jVj = n$ vertices. Build a MoAT model M on n discrete random variables $\mathsf{X} = fX_1, X_2, \ldots, X_ng$ (corresponding to the vertices of G) as follows:

- $w_e = \begin{cases} 1, & e = fi, jg \ 2 \ E \\ 0, & \text{otherwise} \end{cases}$
- $val(X_i) = fR, G, Bg.$
- $P_v(R) = P_v(B) = P_v(G) = \frac{1}{3}$ for all $v \ 2 \ V$.
- $P_{uv}(\alpha, \beta) = \begin{cases} 0, & \alpha = \beta \\ \frac{1}{6}, & \alpha \ 6= \beta \end{cases}$

Observe that the weights define a uniform distribution over all possible spaninng trees of G. Furthermore, the univariate marginals P_v and pairwise marginals P_{uv} are consistent and define a valid tree distribution.

Next, observe that a complete assignment x to X corresponds to a coloring of the original graph G. It is easy to check that for any particular spanning tree T,

$$T(\mathsf{x}) = \begin{cases} \frac{1}{3\times 2^{n-1}}, & \mathsf{x} \text{ is a valid 3-coloring of the tree} \\ 0, & \text{otherwise} \end{cases}$$

Now, we show that x is a valid 3 coloring of the given graph G if and only if $M(\mathsf{x}) = \frac{1}{3\times 2^{n-1}}$.

Firstly, if x is a valid 3-coloring of G, then no pair of adjacent vertices in G are assigned the same color. Hence, the probability assigned to x by any of the spanning trees of G is $\frac{1}{3\times 2^{n-1}}$. Hence,

$$M(\mathsf{x}) = \frac{1}{Z} \sum_{T \in \mathsf{ST}(G)} \left(\prod_{e \in T} w_e \right) \left(\frac{\prod_{(u,v)\in T} P_{uv}(x_u, x_v)}{\prod_{v \in V} P_v(x_v)^{\deg v - 1}} \right)$$
$$= \frac{1}{3 \quad 2^{n-1}} \, @ \frac{1}{Z} \sum_{T \in \mathsf{ST}(G)} \left(\prod_{e \in T} w_e \right) A = \frac{1}{3 \quad 2^{n-1}}$$

Conversely, if x is not a valid 3-coloring of G, then there exist at least one pair of neighboring vertices in G which share the same color. Now, any spanning tree that contains the corresponding edge (which always exists) would assign zero likelihood to x and $M(\mathsf{x})$ be strictly less than $\frac{1}{3\times 2^{n-1}}$. Thus, the graph is 3-colorable if and only if the global MAP state of M has a probability of $\frac{1}{3\times 2^{n-1}}$. □

5 EFFICIENT APPROXIMATE INFERENCE

Unlike usual mixture models, all mixture components in MoAT are close to maximum likelihood on the entire dataset (owing to their consistent univariate and pairwise marginals), but are just sufficiently different enough to model complex dependencies. In this section, we explore how this key observation combined with the tractability of tree-shaped models lets us devise fast-converging algorithms for approximate inference on MoAT.

5.1 MoAT as a Latent Variable Model

Interestingly, the MoAT model yields itself to being interpreted as a latent variable model in an extremely natural way with clear semantics. Defining Y to be the latent random variable with $\mathsf{val}(Y) = \mathsf{ST}(G)$, one can view MoAT as a distribution over $fY, X_1, X_2, ..., X_ng$, where Y models the choice of spanning tree, and inference of the form $\Pr_{\text{MoAT}}(\mathsf{x})$ amounts to marginalizing out the latent variable Y. More precisely,

$$\Pr_{\text{MoAT}}(\mathsf{x}) = \sum_{T \in \mathsf{ST}(G)} \frac{\prod_{e\in T} w_e}{Z} \frac{\prod_{(u,v)\in T} \Pr_{uv}(x_u, x_v)}{\prod_{v\in V} P_v(x_v)^{\deg v - 1}}$$
$$= \sum_{y \in \mathsf{val}(Y)} P(y) \qquad P(\mathsf{x} \ j \ y)$$

It is worth emphasizing the distinctiveness of this characterization. Typically in latent variable models, the latent variables act as higher dimensional features over some subset of the variables. However, for the MoAT model, the latent variable controls the sparse dependency structure that is enforced across the same set of variables.

5.2 Efficient Importance Sampling on MoAT

Exact marginals and conditionals are provably tractable on tree distributions owing to classic techniques such as variable elimination. Consequently, tree distributions are extremely amenable to efficient conditional sampling (Koller and Friedman, 2009). We show that MoAT, a mixture over tree distributions, also supports effective conditional sampling even though our theoretical analysis (Section 4.1) suggests that even computation of marginals in MoAT is NP-hard.

Importance Sampling Revisiting the view of MoAT as a latent variable model $P(Y, \mathsf{X})$, we arrive at a very natural choice of proposal distribution $Q(Y, \mathsf{X})$ that leads to an efficient importance sampling algorithm (Tokdar and Kass, 2010). For evidence e, (and abusing notation to have x refer to an assignment to the unobserved variables) we have that:

$$Q(y, \mathsf{x} \ j \ \mathsf{e})$$
$$= P(y)P(\mathsf{x} \ j \ \mathsf{ye}) \qquad P(y \ j \ \mathsf{e})P(\mathsf{x} \ j \ \mathsf{ye}) = P(y, \mathsf{x} \ j \ \mathsf{e})$$

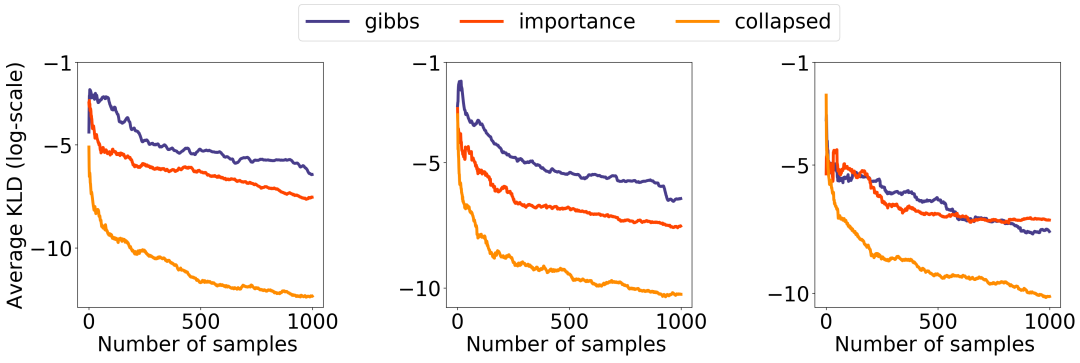


Figure 4: Convergence of various sampling algorithms for posterior marginal inference on NLTCS with different evidence sizes. The reported results are averaged across 5 random seeds.

At a high level, this amounts to sampling a spanning tree unconditionally (Durfee et al., 2017), and then sampling the remaining variables from the chosen tree distribution conditioned on the evidence. More precisely, the weighting function for the samples drawn from the proposal distribution is given by

$$w(y, \mathbf{x} \mid \mathbf{e}) = \frac{P(y, \mathbf{x} \mid \mathbf{e})}{Q(y, \mathbf{x} \mid \mathbf{e})} = \frac{P(y \mid \mathbf{e})}{P(y)} = \frac{P(\mathbf{e} \mid y)}{P(\mathbf{e})}$$

The efficiency of the sampling algorithm (as evaluated through, say, the effective sample size) depends on how close the sample weights are to 1. Intuitively, the ratio $\frac{P(\mathbf{e}|y)}{P(\mathbf{e})}$ captures how much the likelihood of partial evidence in a single spanning tree differs from the corresponding likelihood in the model. As all the mixture components share the same consistent set of univariate and pairwise marginals, it is natural to expect that this ratio does not deviate significantly from 1, thereby leading to high-quality samples. Indeed, our empirical analysis demonstrates that the aforementioned intuition does hold.

Note that we do not actually need to compute $P(\mathbf{e})$ to obtain the sample weights when computing expectations. We can use the unnormalized weight $w'(y, \mathbf{x} \mid \mathbf{e}) = P(\mathbf{e} \mid y)$ as $P(\mathbf{e})$ is a multiplicative constant given $\mathbf{e}$, thereby leading to a self-normalizing importance sampling algorithm. The expectation of any function $f(\mathbf{X})$ over P can be estimated using samples $\mathcal{D} = \{\mathbf{x}y[1], \ldots, \mathbf{x}y[M]\}$ from Q as:

$$\begin{aligned}\hat{E}_{\mathcal{D}}(f) &= \frac{\sum_{m=1}^{M} f(\mathbf{x}[m]) w(\mathbf{x}y[m])}{\sum_{m=1}^{M} w(\mathbf{x}y[m])} \\ &= \frac{\sum_{m=1}^{M} f(\mathbf{x}[m]) P(\mathbf{e} \mid y[m])}{\sum_{m=1}^{M} P(\mathbf{e} \mid y[m])} \\ &= \frac{\sum_{m=1}^{M} f(\mathbf{x}[m]) w'(\mathbf{x}y[m])}{\sum_{m=1}^{M} w'(\mathbf{x}y[m])}\end{aligned}$$

Collapsed Sampling Observe that the sample weights $w(y, \mathbf{x} \mid \mathbf{e}) = \frac{P(\mathbf{e}|y)}{P(\mathbf{e}) = w(y|\mathbf{e})}$ only depend on $\mathbf{e}$ and $\mathbf{y}$ and are independent of $\mathbf{x}$. Given an arbitrary function $f(\mathbf{x})$, this allows to effectively "push the expectation inside" to the tree level, and freely leverage any estimation method available for estimating the expectation of $f(\mathbf{x})$ on a tree distribution. This amounts to a form of collapsed sampling (Koller and Friedman, 2009):

$$\begin{aligned}E_{\mathbf{x}, y \sim P(\cdot \mid \mathbf{e})}(f(\mathbf{x})) &= \sum_{\mathbf{x}y} P(y \mid \mathbf{e}) \; P(\mathbf{x} \mid y\mathbf{e}) \; f(\mathbf{x}) \\ &= \sum_{\mathbf{x}y} w(y, \mathbf{x} \mid \mathbf{e}) \; P(y) \; P(\mathbf{x} \mid y\mathbf{e}) \; f(\mathbf{x}) \\ &= \sum_{y} P(y) \sum_{\mathbf{x}} w(y, \mathbf{x} \mid \mathbf{e}) \; P(\mathbf{x} \mid y\mathbf{e}) \; f(\mathbf{x}) \\ &= \sum_{y} P(y) \; w(y \mid \mathbf{e}) \sum_{\mathbf{x}} (P(\mathbf{x} \mid y\mathbf{e}) \; f(\mathbf{x})) \\ &= \sum_{y} P(y) \; w(y \mid \mathbf{e}) \; E_{\mathbf{x} \sim P(\cdot \mid y\mathbf{e})} f(\mathbf{x})\end{aligned}$$

Our empirical estimator then becomes

$$\hat{E}_{\mathcal{D}}(f) = \frac{\sum_{m=1}^{M} w'(y[m]) E_{\mathbf{x} \sim (\cdot \mid y\mathbf{e})}(f(\mathbf{x}))}{\sum_{m=1}^{M} w'(y[m])}$$

Intuitively, we sample a spanning tree, compute the desired quantity in the corresponding tree distribution, and weight the estimate appropriately. We are thus able to drastically speed up convergence by leveraging the whole suite of exact and approximate techniques available for estimation in tree distributions which have been extensively studied in the literature. For instance, as conditionals of the form $P(X_i = 1 \mid \mathbf{e})$ are tractable in tree distributions, we can efficiently estimate $\Pr_{\text{MoAT}}(X_i = 1 \mid \mathbf{e})$ as

$$\hat{\Pr}_{\text{MoAT}}(X_i = 1 \mid \mathbf{e}) = \frac{\sum_{m=1}^{M} w'(y[m]) P(X_i = 1 \mid y[m]\mathbf{e})}{\sum_{m=1}^{M} w'(y[m])}$$

Empirical Evaluation Empirically, we evaluate our importance sampling algorithm and the collapsed importance sampling algorithm against a standard Gibbs sampling algorithm (Gelfand and Smith, 1990), which is enabled by tractable likelihood computation on the MoAT model. In our experiments, we focus on posterior marginal inference: we fix evidence $\mathbf{e}$ of various sizes, and estimate univariate marginals of the remaining variables conditioned on the evidence $P(X_i \mid \mathbf{e})$. To illustrate speed of convergence to the true value, we require to exactly compute these ground-truth conditionals. To that end, we limit ourselves to a MoAT model on the 16 variable NLTCS dataset from the Twenty Datasets benchmark, where we can exactly compute MoAT marginals and conditionals by exhaustive enumeration. We use average KL-divergence as our metric to assess the speed of convergence:

$$D_{\mathrm{KL}}(P \| \hat{P}) = \sum_{X_i \in \mathbf{X} \setminus E} P(x_i \mid \mathbf{e}) \log \frac{P(x_i \mid \mathbf{e})}{\hat{P}(x_i \mid \mathbf{e})} + P(\overline{x_i} \mid \mathbf{e}) \log \frac{P(\overline{x_i} \mid \mathbf{e})}{\hat{P}(\overline{x_i} \mid \mathbf{e})}$$

As we see Figure 4, the importance sampling and collapsed importance sampling converge orders of magnitude faster than Gibbs sampling. These results are all the more impressive when we account for the superior computational complexity of importance sampling. The bottleneck in the importance sampling algorithm is the spanning tree sampling, leading to a time complexity of $O(n^\omega)$. However, each sample in Gibbs sampling requires n likelihood estimation queries, resulting in a complexity of $O(n \quad n^\omega)$. Further, we observe that the importance sampling algorithm produces very high quality samples as illustrated by the closeness of sample weights to 1 (Figure 5).

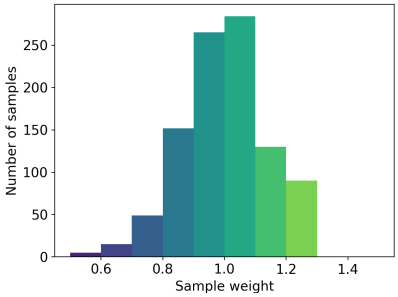


Figure 5: Distribution of sample weights for $|\mathbf{e}| = 4$.

6 CONCLUSION

In this paper, we propose a novel class of generative models called mixture of all trees (MoAT), which strikes a new balance between expressivity and tractability. Despite being a mixture over super-exponentially many tree-shaped distributions, we show that it allows for tractable computation of (normalized) likelihood. Besides, learning a MoAT model does not involve the problem of structure learning, which plagues most probabilistic graphical models.

While we prove hardness of certain classes of queries such as MAP, we demonstrate how MoAT's foundation in tree-shaped models allows us to naturally obtain extremely fast approximate inference algorithms by interpreting it as latent variable model with clear semantics and leveraging tractability of its underlying mixture components. Empirically, we see that MoAT achieves state-of-the-art performance on a variety of density estimation tasks, outperforming powerful probabilistic models such as HCLTs. We leave it to future work to explore MoAT's potential to scale to non-tabular data such as images and text.

We hope that MoAT opens up interesting questions that push the boundaries of tractability and expressive power for probabilistic graphical models.

Acknowledgements

We thank the reviewers for their thoughtful feedback towards improving this paper. This work was funded in part by the DARPA Perceptually-enabled Task Guidance (PTG) Program under contract number HR00112220005, and NSF grants #IIS-1943641, #IIS-1956441, and #CCF-1837129.

References

Anima Anandkumar, Daniel J Hsu, Furong Huang, and Sham M Kakade. Learning mixtures of tree graphical models. *Advances in Neural Information Processing Systems*, 25, 2012.

Fahiem Bacchus, Shannon Dalmao, and Toniann Pitassi. Solving #sat and bayesian inference with backtracking search. *J. Artif. Int. Res.*, 34(1):391–442, mar 2009. ISSN 1076-9757.

Julius Borcea, Petter Brändén, and Thomas Liggett. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 22(2):521–567, 2009.

Seth Chaiken and Daniel J Kleitman. Matrix tree theorems. *Journal of combinatorial theory, Series A*, 24(3):377–381, 1978.

Mark Chavira and Adnan Darwiche. On probabilistic inference by weighted model counting. *Artif. Intell.*, 172(6–7):772–799, apr 2008. ISSN 0004-3702. doi: 10.1016/j.artint.2007.11.002.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968. doi: 10.1109/TIT.1968.1054142.

Adnan Darwiche. A logical approach to factoring belief networks. In *Proceedings of the Eights International*

Conference on Principles of Knowledge Representation and Reasoning, page 409–420, 2002.

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

Rina Dechter. Bucket elimination: A unifying framework for probabilistic inference. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, page 211–219, San Francisco, CA, USA, 1996.

David Durfee, Rasmus Kyng, John Peebles, Anup B. Rao, and Sushant Sachdeva. Sampling random spanning trees faster than matrix multiplication. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, page 730–742, New York, NY, USA, 2017. Association for Computing Machinery.

Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.

Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt. Algebraic model counting. *Journal of Applied Logic*, 22: 46–62, 2017. ISSN 1570-8683.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast mixing Markov chains for strongly rayleigh measures, DPPs, and constrained sampling. In *Advances In Neural Information Processing Systems 29*, pages 4188–4196, 2016.

Anji Liu and Guy Van den Broeck. Tractable regularization of probabilistic circuits. In *Advances in Neural Information Processing Systems*, volume 34, pages 3558–3570, 2021.

H-A Loeliger. An introduction to factor graphs. *IEEE Signal Processing Magazine*, 21(1):28–41, 2004.

László Lovász. Coverings and colorings of hypergraphs. In *Proc. 4th Southeastern Conference of Combinatorics, Graph Theory, and Computing*, pages 3–12. Utilitas Mathematica Publishing, 1973.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: a very deep hierarchy of latent variables for generative modeling. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 6551–6562, 2019.

Vikash Mansinghka, Patrick Shafto, Eric Jonas, Cap Petschulat, Max Gasner, and Joshua B Tenenbaum. Crosscat: a fully bayesian nonparametric method for analyzing heterogeneous, high dimensional data. 2016.

Zelda Mariet, Suvrit Sra, and Stefanie Jegelka. Exponentiated strongly rayleigh distributions. *Advances in neural information processing systems 31*, 2018.

Marina Meilă, Michael I. Jordan, and Pack Kaelbling. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

Joshua Robinson, Suvrit Sra, and Stefanie Jegelka. Flexible modeling of diversity with strongly log-concave distributions. In *Advances in Neural Information Processing Systems 32*, pages 15225–15235, 2019.

Surya T. Tokdar and Robert E. Kass. Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2, 2010.

Jan Van Haaren and Jesse Davis. Markov network structure learning: A randomized feature generation approach. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 1148–1154. AAAI Press, 2012.

Michael Wachter, Rolf Haenni, and Marc Pouly. Optimizing inference in bayesian networks and semiring valuation algebras. In Alexander Gelbukh and Ángel Fernando Kuri Morales, editors, *MICAI 2007: Advances in Artificial Intelligence*, pages 236–247, 2007.

Honghua Zhang, Brendan Juba, and Guy Van den Broeck. Probabilistic generating circuits. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, jul 2021.

Nevin L Zhang and Leonard KM Poon. Latent tree analysis. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Nevin Lianwen Zhang and David Poole. Exploiting causal independence in bayesian network inference. *J. Artif. Int. Res.*, 5(1):301–328, dec 1996. ISSN 1076-9757.

A Complete Proofs

This is section, we present lemmas whose proofs were deferred to the appendix.

Lemma 1. For any distribution $\Pr(\ )$ over binary random variables $X_1, \ldots, X_n$, there exists a set of parameters (i.e., p_v and p_{uv}) in our hypothesis space such that $\Pr(X_u) = P_u(X_u)$ and $\Pr(X_u, X_v) = P_{uv}(X_u, X_v)$ for all $1 \quad u, v \quad n$; i.e., the univariate and pair-wise marginals of Pr are the same as P_u and and P_{uv}.

Proof. Pick MoAT parameters $p_v = \Pr(X_v = 1)$ and $p_{uv} = \Pr(X_u = 1, X_v = 1)$. By construction, the summation constraints are satisifed. Thus, it suffices to check that all the univariate and pairwise marginals are non-negative. For any $v \ 2 \ V$, we have that $P(X_v = 1) = p_v \ 2 \ [0,1]$. Then $P(X_v = 0) = 1 \quad p_v \ 2 \ [0,1]$ as desired. Further, for every $\mathsf{f}u, v\mathsf{g} \ 2 \ \binom{V}{2}$, $P(X_u = 1, X_v = 0) = p_u \quad p_{uv}, P(X_u = 0, X_v = 1) = p_v \quad p_{uv}$, and $P(X_u = 0, X_v = 0) = 1 \quad p_{uv} \quad (p_u \quad p_{uv}) \quad (p_v \quad p_{uv}) = p_{uv} \quad (p_u + p_v \quad 1) \quad 0$ since $P(X_u = 1, X_v = 1) \ 2 \ [max(0, p_u + p_v \quad 1), min(p_u, p_v)]$ Hence, the univariate and pair-wise marginals of Pr are the same as P_u and and P_{uv}, as desired.

. □

Lemma 2. Define SST as the following decision problem: given a connected graph $G = (V, E)$ and a subset $K \quad V$ of the vertices, decide if there exists a spanning tree of G whose leaves are exactly K. SST is NP-hard.

Proof. We proceed via reduction from HAMILTONIAN PATH. Observe that a spanning tree with exactly two leaves is a Hamiltonian path between the two leaves. Given $G = (V, E)$, we iterate over all pairs of vertices $\mathsf{f}i, j\mathsf{g}$, and query the SST oracle for the existence of spanning tree with $K = \mathsf{f}i, j\mathsf{g}$. Then, G has a Hamiltonian path if and only if there exists at least one pair of vertices for which the decision of the SST oracle is YES. □

Lemma 3. Let x be a complete assignment, and denote by ONES(x) the set of variables are are set to 1 in x. Denote by jxj the value jONES(x)j. Denote by LEAVES(T) the set of leaves of a spanning tree T. Let k be the number of spanning trees T of G with ONES(x) LEAVES(T). Then, $\begin{cases} \frac{k}{\epsilon^{n \ 2}} \quad Z \quad p(\mathsf{x}) \quad \frac{k}{\epsilon^{n \ 2}} + \frac{n^{n \ 2}}{\epsilon^{n \ 3}}, & j\mathsf{x}j\%2 = 0 \\ \frac{-k}{\epsilon^{n \ 2}} + \frac{-n^{n \ 2}}{\epsilon^{n \ 3}} \quad Z \quad p(\mathsf{x}) \quad \frac{-k}{\epsilon^{n \ 2}}, & j\mathsf{x}jj\%2 = 1 \end{cases}$

Proof. We will compute the values of the MOAT likelihood function p for any complete assignment x.

- Case 1: ONES(x) LEAVES(T)

$$\left(\prod_{v \in V} P_v(x_v)\right) \left(\prod_{(u,v) \in E} w_{(u,v)} \quad \frac{P_{uv}(x_u, x_v)}{P_u(x_u) \quad P_v(x_v)}\right)$$

$$= \left(\prod_{v \in V} P_v(x_v)\right) \left(\prod_{(u,v) \in E} w_{(u,v)} \quad \frac{P_{uv}(x_u, x_v)}{P_u(x_u) \quad P_v(x_v)}\right) \quad (\quad 1)^{|\mathsf{x}|}$$

(Since $\frac{P_{uv}}{P_u \cdot P_v} \quad 0$ and $P_v(x_v) < 0 \quad (\ ) \quad x_v = 1$)

$$= \frac{\prod_{(u,v) \in E} w_{(u,v)} P_{uv}(x_u, x_v)}{\prod_{v \in V} P_v(x_v)^{\deg v - 1}} \quad (\quad 1)^{|\mathsf{x}|}$$

$$= \frac{\prod_{(u,v) \in E} \epsilon}{\prod_{v \in V} \epsilon^{\deg v - 1}} \quad (\quad 1)^{|\mathsf{x}|}$$

$$= \frac{\prod_{(u,v) \in E} \epsilon}{\prod_{v \in V} \epsilon^{\deg v - 1}} \quad (\quad 1)^{|\mathsf{x}|}$$

$$= \frac{\epsilon^{n-1}}{\epsilon^{2n-3}} \quad (\quad 1)^{|\mathsf{x}|}$$

$$= \frac{1}{\epsilon^{n-2}} \quad (\quad 1)^{|\mathsf{x}|}$$

- Case 2: ONES(x) 6 LEAVES(T)
  In this case $\gamma \quad 1$ internal nodes (nodes with degree more than one) are assigned a value of 1. Then similarly,

– If $|x| \% 2 = 0$, we obtain that

$$
\begin{aligned}
0 & \left(\prod_{v \in V} P_v(x_v) \right) \left(\sum_{(u,v) \in E} w_{(u,v)} \frac{P_{uv}(x_u, x_v)}{P_u(x_u) P_v(x_v)} \right) \\
&= \left(\prod_{v \in V} P_v(x_v) \right) \left(\sum_{(u,v) \in E} w_{(u,v)} \frac{P_{uv}(x_u, x_v)}{P_u(x_u) P_v(x_v)} \right) \\
&= \frac{\prod_{(u,v) \in E} \epsilon}{\prod_{v \in V} P_v(x_v)^{\deg v - 1}} \\
&\quad \frac{\epsilon^{n-1}}{\epsilon^{2n-4}} \\
&\quad \frac{1}{\epsilon^{n-3}}
\end{aligned}
$$

– If $|x| \% 2 = 1$, we obtain that

$$
\begin{aligned}
0 & \left(\prod_{v \in V} P_v(x_v) \right) \left(\sum_{(u,v) \in E} w_{(u,v)} \frac{P_{uv}(x_u, x_v)}{P_u(x_u) P_v(x_v)} \right) \\
&= \left(\prod_{v \in V} P_v(x_v) \right) \left(\sum_{(u,v) \in E} w_{(u,v)} \frac{P_{uv}(x_u, x_v)}{P_u(x_u) P_v(x_v)} \right) \\
&= \frac{\prod_{(u,v) \in E} \epsilon}{\prod_{v \in V} P_v(x_v)^{\deg v - 1}} \\
&\quad \frac{\epsilon^{n-1}}{\epsilon^{2n-4}} \\
&\quad \frac{1}{\epsilon^{n-3}}
\end{aligned}
$$

As the maximum number of spanning trees on a graph with n vertices is n^{n-2}, we obtain the desired bounds:

- If $j\mathsf{x}j\,\%2 = 0$, we obtain that

$$\begin{aligned}
Z\ \ p(\mathsf{x}) &= \sum_{T\in\mathsf{ST}(G)} \Big(\prod_{v\in V} P_v(x_v)\Big) \Big(\prod_{(u,v)\in E} w_{(u,v)}\ \frac{P_{uv}(x_u,x_v)}{P_u(x_u)\ P_v(x_v)}\Big) \\
&= \sum_{\substack{T\in\mathsf{ST}(G)\\ K\subseteq\mathsf{LEAVES}(T)}} \Big(\prod_{v\in V} P_v(x_v)\Big) \Big(\prod_{(u,v)\in E} w_{(u,v)}\ \frac{P_{uv}(x_u,x_v)}{P_u(x_u)\ P_v(x_v)}\Big) \\
&\quad + \sum_{\substack{T\in\mathsf{ST}(G)\\ K\not\subseteq\mathsf{LEAVES}(T)}} \Big(\prod_{v\in V} P_v(x_v)\Big) \Big(\prod_{(u,v)\in E} w_{(u,v)}\ \frac{P_{uv}(x_u,x_v)}{P_u(x_u)\ P_v(x_v)}\Big) \\
&\le \sum_{\substack{T\in\mathsf{ST}(G)\\ K\subseteq\mathsf{LEAVES}(T)}} \frac{1}{\epsilon^{n-2}} + \sum_{\substack{T\in\mathsf{ST}(G)\\ K\not\subseteq\mathsf{LEAVES}(T)}} 0 \\
&\le \frac{k}{\epsilon^{n-2}}
\end{aligned}$$

Similarly $Z\ \ p(\mathsf{x}) \le \sum_{\substack{T\in\mathsf{ST}(G)\\ K\subseteq\mathsf{LEAVES}(T)}} \frac{1}{\epsilon^{n-2}} + \sum_{\substack{T\in\mathsf{ST}(G)\\ K\not\subseteq\mathsf{LEAVES}(T)}} \frac{1}{\epsilon^{n-3}} \le \frac{k}{\epsilon^{n-2}} + \frac{n^{n-2}}{\epsilon^{n-3}}$

- If $j\mathsf{x}j\,\%2 = 1$, we similarly obtain that

$$\begin{aligned}
Z\ \ p(\mathsf{x}) &\le \frac{k}{\epsilon^{n-2}} \\
Z\ \ p(\mathsf{x}) &\le \frac{k}{\epsilon^{n-2}} + \frac{n^{n-2}}{\epsilon^{n-3}}
\end{aligned}$$

Thus, $\begin{cases} \frac{k}{\epsilon^{n-2}} & Z\ \ p(\mathsf{x}) \le \frac{k}{\epsilon^{n-2}} + \frac{n^{n-2}}{\epsilon^{n-3}}, & j\mathsf{x}j\,\%2 = 0 \\ \frac{-k}{\epsilon^{n-2}} + \frac{-n^{n-2}}{\epsilon^{n-3}} & Z\ \ p(\mathsf{x}) \le \frac{-k}{\epsilon^{n-2}}, & j\mathsf{x}jj\%2 = 1 \end{cases}$ as desired.

□

Lemma 4. The number of spanning trees T with $K = \mathsf{LEAVES}(T)$ is given by $\ bZ\epsilon^{n-2}f(\mathsf{e})\mathsf{e}$.

Proof. Since the number of spanning trees T with K $\mathsf{LEAVES}(T)$ is given by $\ bZ\ \ \epsilon^{n-2}\ \ p(\mathsf{x})\mathsf{e}$, from the inclusion-exclusion formula we obtain that the number of spanning trees T with $K = \mathsf{LEAVES}(T)$ (upto sign) is given by

$$\begin{aligned}
&\sum_{K\subseteq L} (\ \ 1)^{|L|} \sum_{T\in\mathsf{ST}(G)} \mathsf{1}(L\ \ \mathsf{LEAVES}(T)) \\
&= \sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} (\ \ 1)^{|\mathsf{x}|}\ \ bZ\epsilon^{n-2}p(\mathsf{x})\mathsf{e} \\
&= \sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} bZ\epsilon^{n-2}p(\mathsf{x})\mathsf{e}
\end{aligned}$$

Let $fxg = jx \quad bxej$. Since the $fZ \quad \epsilon^{n-2} \quad p(x)g \quad \epsilon \quad n^{n-2}$ for all x, we obtain that

$$\begin{aligned}
&\sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} fZ \quad \epsilon^{n-2} \quad p(x)g \\
&2^n \quad \epsilon \quad n^{n-2} \\
<&2^n \quad \frac{1}{2^{n+1} \quad n^{n-2}} \quad n^{n-2} \\
<&\frac{1}{2}
\end{aligned}$$

Thus, we obtain that the number of spanning trees T with $K = \textsc{leaves}(T)$ is given by

$$\begin{aligned}
&\sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} bZ \quad \epsilon^{n-2} \quad p(x)e \\
=&b \sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} Z \quad \epsilon^{n-2} \quad p(x)e \\
=&bZ \quad \epsilon^{n-2} \sum_{val(z_1)} \sum_{val(z_2)} \cdots \sum_{val(z_k)} p(x)e \\
=&bZ \quad \epsilon^{n-2} \quad f(e)e
\end{aligned}$$

□

B Parameterization for Categorical Variables

Consider the MoAT pairwise marginal matrix $P_{uv}^{(k_u \times k_v)}$ defined by $P_{uv}[i][j] = P(X_u = i, X_v = j)$.

B.1 Relation to Doubly Stochastic Matrices

This MoAT pairwise marginal matrix is closely related to the class of matrices called doubly stochastic matrices, where all the entries are between 0 and 1 and the rows and columns sum to 1. Recall that for the MoAT pairwise marginal matrix, we similarly require that the rows and columns sum to the corresponding univariate marginals. The set of $k \quad k$ doubly stochastic matrices (often referred to as the Birkhoff polytope) lies in a $(k \quad 1) \quad (k \quad 1)$ affine subspace of $R^{k \times k}$, and to the best of our knowledge there is no known valid and fully general parameterization for this class of matrices that allows for unconstrained optimization. Similarly, the pairwise marginal matrices which are uniquely defined by the values $P_{uv}[i][j]$ for $(i, j) \; 2 \; f1, 2, \quad , k \quad 1g \quad f1, 2, \quad , k \quad 1g$, also lie in a $(k \quad 1) \quad (k \quad 1)$ affine subspace of $R^{k \times k}$, and there is unfortunately, to the best of our knowledge, no known valid and fully general parameterization for this class of matrices that allows for unconstrained optimization.

However, we instead propose a valid parameterization of MoAT pairwise marginal matrices that is not fully general, but has $\min(k_u, k_v) \quad 1$ free parameters (as opposed to a fully general paramterization with $(k_u \quad 1) \quad (k_v \quad 1)$ parameters) that can be learnt in an unconstrained manner.

B.2 Proposed MoAT Parameterization

First we consider the case of square pairwise marginal matrices $P_{uv}^{(k \times k)}$. We define it inductively, defining a parameterization for the first $l \quad l$ submatrix of P_{uv} (denoted P_l)for $l \; 2 \; [2, k]$. One can interpret this a defining a parameterization for the marginal distribution $P(X_u \; 2 \; [1, l], X_v \; 2 \; [1, l])$, in a way that preserves the relative proportion of univariate marginals.

- **Base case ($l = 2$):**
  This is identical to having binary random variables X_u and X_v with the following univariate marginals.
  - $P_2(X_u = 1) = \frac{P(X_u=1)}{P(X_u=1)+P(X_u=2)}$

- $P_2(X_u = 2) = \frac{P(X_u=2)}{P(X_u=1)+P(X_u=2)}$
- $P_2(X_v = 1) = \frac{P(X_v=1)}{P(X_v=1)+P(X_v=2)}$
- $P_2(X_v = 2) = \frac{P(X_v=2)}{P(X_v=1)+P(X_v=2)}$

The matrix can be parameterized by a single parameter $\lambda_2 = P_2(X_u = 1, X_v = 1)$ as shown in Lemma 1.

- **Inductive case ($l >= 3$):**
Assume we have have a parameterization for first $(l \quad 1) \quad (l \quad 1)$ submatrix P_{l-1} of P_{uv}. Pick $\lambda_l \ 2 \ [max(0, \frac{\sum_{t=1}^{l} P(X_u=t)}{\sum_{t=1}^{l} P(X_u=t)} + \frac{\sum_{t=1}^{l} P(X_v=t)}{\sum_{t=1}^{l} P(X_v=t)} \quad 1), min(\frac{\sum_{t=1}^{l} P(X_u=t)}{\sum_{t=1}^{l} P(X_u=t)}, \frac{\sum_{t=1}^{l} P(X_v=t)}{\sum_{t=1}^{l} P(X_v=t)})]$.
Then, define P_l as follows:

$$P_l[i][j] = \begin{cases} \lambda_l \quad P_{l-1}[i][j], & i < l, j < l \\ \frac{P(X_u=i)}{\sum_{t=1}^{l} P(X_u=t)} \quad \lambda_l \quad \frac{P(X_u=i)}{\sum_{t=1}^{l-1} P(X_u=t)}, & i < l, j = l \\ \frac{P(X_v=j)}{\sum_{t=1}^{l} P(X_v=t)} \quad \lambda_l \quad \frac{P(X_v=j)}{\sum_{t=1}^{l-1} P(X_v=t)}, & i = l, j < l \\ 1 \quad \frac{\sum_{t=1}^{l-1} P(X_u=t)}{\sum_{t=1}^{l} P(X_u=t)} \quad \frac{\sum_{t=1}^{l-1} P(X_v=t)}{\sum_{t=1}^{l} P(X_v=t)} + \lambda_l, & i = l, j = l \end{cases}$$

By choice of λ_l, all the entries of this matrix are non-negative. It now suffices to check that the univariate marginals are in proportion.

- For $i < l$,

$$\begin{aligned} P_l(X_u = i) &= \sum_{j=1}^{} P_l[i][j] \\ &= \sum_{j=1}^{l-1} P_l[i][j] + P_l[i][l] \\ &= \sum_{j=1}^{l-1} (\lambda_l \quad P_{l-1}[i][j]) + \frac{P(X_u = i)}{\sum_{t=1}^{l} P(X_u = t)} \quad \lambda_l \quad \frac{P(X_u = i)}{\sum_{t=1}^{l-1} P(X_u = t)} \\ &= \frac{P(X_u = i)}{\sum_{t=1}^{l} P(X_u = t)} \end{aligned}$$

  as desired.

- For $i = l$,

$$\begin{aligned} P_l(X_u = l) &= \sum_{j=1}^{} P_l[l][j] \\ &= \sum_{j=1}^{l-1} P_l[l][j] + P_l[l][l] \\ &= \sum_{j=1}^{l-1} \frac{P(X_v = j)}{\sum_{t=1}^{l} P(X_v = t)} \quad \lambda_l \quad \frac{P(X_v = j)}{\sum_{t=1}^{l-1} P(X_v = t)} \Bigg! \\ &\quad + 1 \quad \frac{\sum_{t=1}^{l-1} P(X_u = t)}{\sum_{t=1}^{l} P(X_u = t)} \quad \frac{\sum_{t=1}^{l-1} P(X_v = t)}{\sum_{t=1}^{l} P(X_v = t)} + \lambda_l \\ &= \sum_{j=1}^{l-1} \frac{P(X_v = j)}{\sum_{t=1}^{l} P(X_v = t)} \quad + 1 \quad \frac{\sum_{t=1}^{l-1} P(X_u = t)}{\sum_{t=1}^{l} P(X_u = t)} \quad \frac{\sum_{t=1}^{l-1} P(X_v = t)}{\sum_{t=1}^{l} P(X_v = t)} \\ &= 1 \quad \frac{\sum_{t=1}^{l-1} P(X_u = t)}{\sum_{t=1}^{l} P(X_u = t)} \\ &= \frac{P(X_u = l)}{\sum_{t=1}^{l} P(X_u = t)} \end{aligned}$$

as desired.

– By symmetry, the desired results hold for $P_l(X_v = j)$ for all $1 \quad j \quad l$.

Observe that since $\sum_{t=1}^{k} P(X_u = t) = 1$, $P_k(X_u = t) = P(X_u = t)$ for all $1 \quad t \quad k$. Similarly, $P_k(X_v = t) = P(X_v = t)$ for all $1 \quad t \quad k$. Thus, P_k is the desired $k \quad k$ MoAT pairwise marginal matrix, with learnable parameters $\lambda_2 \quad \lambda_k$.

Lastly, observe that this parameterization generalizes to non-square matrices too. Without loss in generality, assume P is $k_u \quad k_v$ with $k_u < k_v$. First, we can parameterize the first $2 \quad k_v \quad k_u \quad 2$ submatrix by a single parameter. Then, we can add $k_u \quad 2$ scaling parameters λ_i as in the case of the square matrix to obtain a parameterization for the whole matrix. Note that the total number of free parameters in this parameterization is $\min(k_u, k_v) \quad 1$.

C Experimental Setup

All experiments were performed on Intel(R) Xeon(R) Gold 5220 CPU @ 2.20GHz. For the experiments on the Twenty Dataset density estimation benchmark, the MoAT model is trained with two sets of hyperparameters: (1) for the datasets with < 500 random variables, the model is trained with batch_size = 1024 and learning rate = 0.05 and (2) for the datasets with $\quad 500$ random variables, the model is trained with batch_size = 64 and learning rate = 0.01. All models are trained for 50 epochs with early stopping: the test log-likelihood corresponding to the epoch with the best validation log-likelihood is reported. The total training time for all datasets takes roughly a day on one NVIDIA RTX A5000 gpu. Complete code and datasets for all the experiments can be found at `https://github.com/UCLA-StarAI/MoAT`.