ARTree: A Deep Autoregressive Model for Phylogenetic Inference

Anonymous Author(s)
Affiliation
Address
email

Abstract

Designing flexible probabilistic models over tree topologies is important for developing efficient phylogenetic inference methods. To do that, previous works often leverage the similarity of tree topologies via hand-engineered heuristic features which would require domain expertise and may suffer from limited approximation capability. In this paper, we propose a deep autoregressive model for phylogenetic inference based on graph neural networks (GNNs), called ARTree. By decomposing a tree topology into a sequence of leaf node addition operations and modeling the involved conditional distributions based on learnable topological features via GNNs, ARTree can provide a rich family of distributions over tree topologies that have simple sampling algorithms, without using heuristic features. We demonstrate the effectiveness and efficiency of our method on a benchmark of challenging real data tree topology density estimation and variational Bayesian phylogenetic inference problems.

1 Introduction

Reconstructing the evolutionary relationships among species has been one of the central problems in computational biology, with a wide range of applications such as genomic epidemiology (Dudas et al., 2017; du Plessis et al., 2021; Attwood et al., 2022) and conservation genetics (DeSalle & Amato, 2004). Based on molecular sequence data (e.g. DNA, RNA, or protein sequences) of the observed species and a model of evolution, this has been formulated as a statistical inference problem on the hypotheses of shared history, i.e., *phylogenetic trees*, where maximum likelihood and Bayesian approaches are the most popular methods (Felsenstein, 1981; Yang & Rannala, 1997; Mau et al., 1999; Larget & Simon, 1999; Huelsenbeck et al., 2001). However, phylogenetic inference can be challenging due to the composite structure of tree space which contains both continuous and discrete components (e.g., the branch lengths and the tree topologies) and the large search space of tree topologies that explodes combinatorially as the number of species increases (Whidden & Matsen IV, 2015; Dinh et al., 2017).

Recently, several efforts have been made to improve the efficiency of phylogenetic inference algorithms by designing flexible probabilistic models over the tree topology space (Höhna & Drummond, 2012; Larget, 2013; Zhang & Matsen IV, 2018). One typical example is subsplit Bayesian networks (SBNs) (Zhang & Matsen IV, 2018), which is a powerful probabilistic graphical model that provides a flexible family of distributions over tree topologies. Given a sample of tree topologies (e.g., sampled tree topologies from an MCMC run), SBNs have proved effective for accurate tree topology density estimation that generalizes beyond observed samples by leveraging the similarity of hand-engineered subsplit structures among tree topologies. Moreover, SBNs also allow fast ancestral sampling and hence were later on integrated into a variational Bayesian phylogenetic inference (VBPI) framework to provide variational posteriors over tree topologies (Zhang & Matsen IV, 2019). However, due to

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

the limited parent-child subsplit patterns in the observed samples, SBNs can not provide distributions whose support spans the entire tree topology space (Zhang & Matsen IV, 2022). Furthermore, when used as variational distributions over tree topologies in VBPI, SBNs often rely on subsplit support estimation for variational parameterization, which requires domain expertise and would become challenging when the posterior is diffuse.

While SBNs suffer from the aforementioned limitations due to their hand-engineered design, a number of deep learning methods have been proposed for probabilistic modeling of graphs (Jin et al., 2018; You et al., 2018a; Cao & Kipf, 2018; Simonovsky & Komodakis, 2018). Instead of using hand-engineered features, these approaches use neural networks to define probabilistic models for the connections between graph nodes which allow for learnable distributions over graphs. Due to the flexibility of neural networks, the resulting models are capable of learning complex graph patterns automatically. Among these deep graph models, graph autoregressive models (You et al., 2018b; Li et al., 2018; Liao et al., 2019; Dai et al., 2020; Shi et al., 2020) are designed to learn flexible graph distributions that also allow easy sampling procedures by sequentially adding nodes and edges. Therefore, they serve as an ideal substitution of SBNs for phylogenetic inference that can provide more expressive distributions over tree topologies while not requiring domain expertise.

In this paper, we propose a novel deep autoregressive model for phylogenetic inference, called ARTree, which allows for more flexible distributions over tree topologies without using heuristic features than SBNs. With a pre-selected order of leaf nodes (i.e., species or taxa), ARTree generates a tree topology by recursively adding new leaf nodes to the edges of current tree topology, starting from a star-shaped tree topology with the first three leaf nodes (Figure 1). The edge to which a new leaf node connects is determined according to a conditional distribution based on learnable topological features of current tree topology via GNNs (Zhang, 2023). This way, probability distributions provided by ARTree all have full support that spans the entire tree topology space. Unlike SBNs, ARTree can be readily used in VBPI without requiring subsplit support estimation for parameterization. In experiments, we show that ARTree outperforms SBNs on a benchmark of challenging real data tree topology density estimation and variational Bayesian phylogenetic inference problems.

2 Background

Phylogenetic likelihoods A phylogenetic tree is commonly described by a bifurcating tree topology τ and the associated non-negative branch lengths $\boldsymbol{q}$. The tree topology τ represents the evolutionary relationship of the species and the branch lengths $\boldsymbol{q}$ quantify the evolutionary intensity along the edges of τ. The leaf nodes of τ correspond to the observed species and the internal nodes of τ represent the unobserved ancestor species. A continuous time Markov model is often used to describe the transition probabilities of the characters along the edges of the tree (Felsenstein, 2004). Concretely, let $\boldsymbol{Y} = \{Y_1, \ldots, Y_M\} \in \Omega^{N\times M}$ be the observed sequences (with characters in Ω) of length M over N species. Under the assumption that different sites evolve independently and identically, the likelihood of $\boldsymbol{Y}$ given $\tau, \boldsymbol{q}$ takes the form

$$p(\boldsymbol{Y}|\tau, \boldsymbol{q}) = \prod_{i=1}^{M} p(Y_i|\tau, \boldsymbol{q}) = \prod_{i=1}^{M} \sum_{a^i} \eta(a_r^i) \prod_{(u,v)\in E(\tau)} P_{a_u^i a_v^i}(q_{uv}), \tag{1}$$

where a^i ranges over all extensions of Y_i to the internal nodes with a_u^i being the character assignment of node u (r represents the root node), $E(\tau)$ is the set of edges of τ, q_{uv} is the branch length of the edge $(u, v) \in E(\tau)$, $P_{jk}(q)$ is the transition probability from character j to k through a branch of length q, and η is the stationary distribution of the Markov model.

Subsplit Bayesian networks Let $\mathcal{X}$ be the set of leaf labels representing the existing species. A non-empty subset of $\mathcal{X}$ is called a *clade* and the set of all clades $\mathcal{C}(\mathcal{X})$ is equipped with a total order $\succ$ (e.g., lexicographical order). An ordered clade pair (W, Z) satisfying $W \cap Z = \emptyset$ and $W \succ Z$ is called a *subsplit*. A *subsplit Bayesian network* (SBN) is then defined as a Bayesian network whose nodes take subsplit values or singleton clade values that describe the local topological structures of tree topologies. For a rooted tree topology, one can find the corresponding node assignment of SBNs by following its splitting processes (Figure 4 in Appendix A). The SBN based probability of a rooted tree topology τ then takes the following form

$$p_{\mathrm{sbn}}(T = \tau) = p(S_1 = s_1) \prod_{i>1} p(S_i = s_i|S_{\pi_i} = s_{\pi_i}), \tag{2}$$

where S_i denotes the subsplit- or singleton-clade-valued random varaibles at node i (node 1 is the root node), π_i is the index set of the parents of node i and $\{s_i\}_{i\geq 1}$ is the corresponding node assignment. For unrooted tree topologies, we can also define their SBN based probabilities by viewing them as rooted tree topologies with unobserved roots and integrating out the positions of the root node as follows: $p_{\text{sbn}}(T^{\text{u}} = \tau) = \sum_{e\in E(\tau)} p_{\text{sbn}}(\tau^e)$, where τ^e is the resulting rooted tree topology when the rooting position is on edge e. In practice, the conditional probability tables (CPTs) of SBNs are often parameterized based on a sample of tree topologies (e.g., the observed data for density estimation (Zhang & Matsen IV, 2018) or fast bootstrap/MCMC samples (Minh et al., 2013; Zhang, 2020) for VBPI). As a result, the supports of SBN-induced distributions are often limited by the splitting patterns in the observed samples and could not span the entire tree topology space (Zhang & Matsen IV, 2022). More details on SBNs can be found in Appendix A.

Variational Bayesian phylogenetic inference Given a prior distribution $p(\tau, \boldsymbol{q})$, the phylogenetic posterior distribution takes the form

$$p(\tau, \boldsymbol{q}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\tau, \boldsymbol{q})p(\tau, \boldsymbol{q})}{p(\boldsymbol{Y})} \propto p(\boldsymbol{Y}|\tau, \boldsymbol{q})p(\tau, \boldsymbol{q}). \tag{3}$$

Let $Q_{\boldsymbol{\phi}}(\tau)$ and $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ be variational families over the spaces of tree topologies and branch lengths respectively. The VBPI approach uses $Q_{\boldsymbol{\phi},\boldsymbol{\psi}}(\tau, \boldsymbol{q}) = Q_{\boldsymbol{\phi}}(\tau)Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ to approximate the posterior $p(\tau, \boldsymbol{q}|\boldsymbol{Y})$ by maximizing the following multi-sample lower bound

$$L^K(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathbb{E}_{\{(\tau^i,\boldsymbol{q}^i)\}_{i=1}^K \overset{\text{i.i.d.}}{\sim} Q_{\boldsymbol{\phi},\boldsymbol{\psi}}} \log\left(\frac{1}{K}\sum_{i=1}^K \frac{p(\boldsymbol{Y}|\tau^i, \boldsymbol{q}^i)p(\tau^i, \boldsymbol{q}^i)}{Q_{\boldsymbol{\phi}}(\tau^i)Q_{\boldsymbol{\psi}}(\boldsymbol{q}^i|\tau^i)}\right). \tag{4}$$

The tree topology distribution $Q_{\boldsymbol{\phi}}(\tau)$ is often SBNs which in this case rely on subsplit support estimation for parameterization that requires domain expertise and would become challenging for diffuse posteriors (Zhang & Matsen IV, 2022). The branch lengths distribution $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ can be diagonal lognormal distribution parametrized via heuristic features or learnable topological features of τ (Zhang & Matsen IV, 2019; Zhang, 2020, 2023). See more details on VBPI in Appendix B.

Graph autoregressive models By decomposing a graph as a sequence of components (nodes, edges, motifs, etc), graph autoregressive models generate the full graph by adding one component at a time, until some stopping criteria are satisfied (You et al., 2018b; Jin et al., 2018; Liao et al., 2019). In previous works, recurrent neural networks (RNNs) for graphs are usually utilized to predict new graph components conditioned on the sub-graphs generated so far. The key of graph autoregressive models is to find a way to efficiently sequentialize graph structures, which is often domain-specific.

3 Proposed method

In this section, we propose ARTree, a deep autoregressive model for phylogenetic inference that can provide flexible distributions whose support spans the entire tree topology space and can be naturally parameterized without using heuristic approaches such as subsplit support estimation. We first describe a particular autoregressive generating process of phylogenetic tree topologies. We then develop powerful GNNs to parameterize learnable conditional distributions of this generating process. We consider unrooted tree topologies in this section, but the method developed here can be easily adapted to rooted tree topologies.

3.1 A sequential generating process of tree topologies

To better illustrate our approach, we begin with some notations. Let $\tau_n = (V_n, E_n)$ be a tree topology with n leaf nodes and V_n, E_n are the sets of nodes and edges respectively. Note that $|V_n| = 2n - 2$ and $|E_n| = 2n - 3$ due to the unrooted and bifurcating structure of τ_n. The leaf nodes in V_n are treated as labeled nodes and the interior nodes in V_n are treated as unlabeled nodes. Let us assume a pre-selected order for the leaf nodes $\mathcal{X} = \{x_1, \ldots, x_N\}$, which is called taxa order for short by us. Now, consider a sequential generating process for all possible tree topologies that have leaf nodes $\mathcal{X}$. We start with a definition below.

Definition 1 (Ordinal Tree Topology)**.** *Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be a set of $N (N \geq 3)$ leaf nodes. Let $\tau_n = (V_n, E_n)$ be a tree topology with $n (n \leq N)$ leaf nodes in $\mathcal{X}$. We say τ_n is an ordinal tree topology of rank n, if its leaf nodes are the first n elements of $\mathcal{X}$, i.e., $V_n \cap \mathcal{X} = \{x_1, \ldots, x_n\}$.*

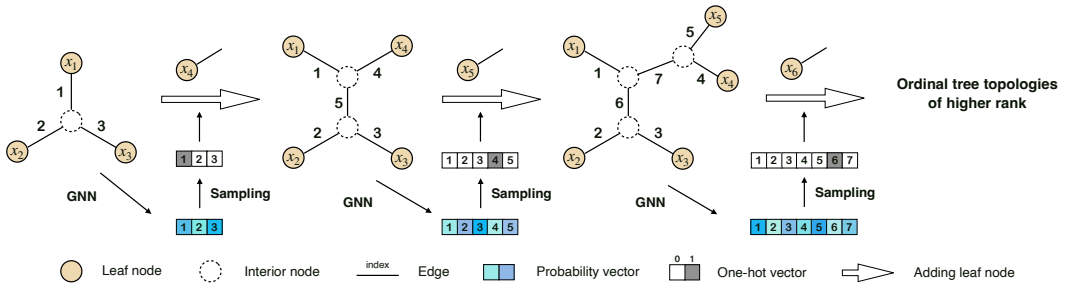

Figure 1: An overview of ARTree for autoregressive tree topology generation. The left plot is the starting ordinal tree topology of rank 3. This tree topology is then fed into GNNs which output a probability vector over edges. We then sample from the corresponding edge decision distribution and attach the next leaf node to the sampled edge. This process continues until an ordinal tree topology of rank N is reached.

We now describe a procedure that constructs ordinal tree topologies of rank N recursively by adding one leaf node at a time as follows. We first start from τ_3, the ordinal tree topology of rank 3, which is the smallest ordinal tree topology and is unique due to its unrooted and bifurcating structure. Suppose now we have an ordinal tree topology $\tau_n = (V_n, E_n)$ of rank n. To add the leaf node x_{n+1} to τ_n, we i) select an edge $e_n = (u, v) \in E_n$ and remove it from E_n; ii) add a new node w and two new edges $(u, w), (w, v)$ to the tree topology; iii) add the leaf node x_{n+1} and an edge (w, x_{n+1}) to the tree topology. This way, we obtain an ordinal tree topology τ_{n+1} of rank $n + 1$. Intuitively, the leaf node x_{n+1} is added to the tree topology τ_n by attaching it to an existing edge $e_n \in E_n$. The position of the selected edge represents the evolutionary relationship between this new species and others. After performing this procedure for $n = 3, \ldots, N - 1$, we finally obtain an ordinal tree topology $\tau = \tau_N$ of rank N. See Figure 1 for an illustration.

During the generating process described above, the selected edges at each time step form a sequence $D = (e_3, \ldots, e_{N-1})$. This sequence D of length $N - 3$ records all the decisions we have made for autoregressively generating a tree topology τ and thus we call D a decision sequence. In fact, there is a one-to-one mapping between decision sequences and ordinal tree topologies of rank N, which is formalized in Theorem 1. Note that a similar process is also used in online phylogenetic sequential Monte Carlo (OPSMC) (Dinh et al., 2016), where the leaf node addition operation is incorporated into the design of the proposal distributions.

Theorem 1. *Let $\mathcal{D} = \{D | D = (e_3, \ldots, e_{N-1}),\ e_n \in E_n, \forall\ 3 \leq n \leq N - 1\}$ be the set of all decision sequences of length $N - 3$ and $\mathcal{T}$ be the set of all ordinal tree topologies of rank N. Let the map*

$$\begin{array}{rccc} g: & \mathcal{D} & \to & \mathcal{T} \\ & D & \mapsto & \tau \end{array}$$

be the generating process described above. Then g is a bijection between $\mathcal{D}$ and $\mathcal{T}$.

According to Theorem 1, for each tree topology $\tau \in \mathcal{T}$, there is a unique decision sequence given by $g^{-1}(\tau)$. We call this process of finding the decision sequences of tree topologies the *decomposition process*. See more details on the decomposition process in Appendix C. The following lemma shows that one can find $g^{-1}(\tau)$ in linear time.

Lemma 1. *The time complexity of the decomposition process induced by $g^{-1}(\cdot)$ is $O(N)$.*

The proofs of Theorem 1 and Lemma 1 can be found in Appendix D. Based on the bijection g defined in Theorem 1, we can model the distribution $Q(D)$ over the space of decision sequences $\mathcal{D}$ instead of modeling the distribution $Q(\tau)$ over $\mathcal{T}$. Due to the sequential nature of D, we can decompose $Q(D)$ as the product of conditional distributions over the elements:

$$Q(D) = \prod_{n=3}^{N-1} Q(e_n | e_3, \ldots, e_{n-1}). \tag{5}$$

In what follows, we simplify $Q(e_n | e_3, \ldots, e_{n-1})$ as $Q(e_n | e_{<n})$ and let $e_{<3}$ be the empty set.

Algorithm 1: ARTree: An autoregressive model for phylogenetic tree topologies

Input: a set $\mathcal{X} = \{x_1, \ldots, x_N\}$ of leaf nodes.
Output: an ordinal tree topology τ of rank N; the ARTree probability $Q(\tau)$ of τ.
$\tau_3 = (V_3, E_3) \leftarrow$ the unique ordinal tree topology of rank 3;
for $n = 3, \ldots, N-1$ **do**
  Calculate the probability vector $q_n \in \mathbb{R}^{|E_n|}$ using the current GNN model;
  Sample an edge decision e_n from Discrete (q_n) and assume $e_n = (u, v)$;
  Create a new node w;
  $E_{n+1} \leftarrow (E_n \backslash \{e_n\}) \cup \{(u, w), (w, v), (w, x_{n+1})\}$;
  $V_{n+1} \leftarrow V_n \cup \{w, x_{n+1}\}$;
  $\tau_{n+1} \leftarrow (V_{n+1}, E_{n+1})$;
end
$\tau \leftarrow \tau_N$;
$Q(\tau) \leftarrow q_3(e_3)q_4(e_4)\cdots q_{N-1}(e_{N-1})$.

3.2 Graph neural networks for edge decision distribution

By Theorem 1, the sequence $e_{<n}$ corresponds to a sequence of ordinal tree topologies of increasing ranks $(\tau_3, \ldots, \tau_n)$ (the empty set $e_{<3}$ corresponds to the unique ordinal tree topology τ_3 of rank 3). Therefore, the discrete distribution $Q(e_n|e_{<n})$ in equation (5) defines the probability of adding the leaf node x_{n+1} to the edge e_n of τ_n, conditioned on all the ordinal tree topologies $(\tau_3, \ldots, \tau_n)$ generated so far. In what follows, we will show step by step how to use graph neural networks (GNNs) to parameterize such a conditional distribution given tree topologies.

Topological node embeddings At the n-th time step of the generating process, we first find the node embeddings of the current tree topology $\tau_n = (V_n, E_n)$, which is a set $\{f_n(u) \in \mathbb{R}^N : u \in V_n\}$ that assigns each node with an encoding vector in $\mathbb{R}^N$. Following Zhang (2023), we first assign one hot encoding to the leaf nodes, i.e.

$$[f_n(x_i)]_j = \delta_{ij}, 1 \le i \le n, 1 \le j \le N, \tag{6}$$

where δ is Kronecker delta function; we then get the embeddings for the interior nodes by minimizing the Dirichlet energy $\ell(f_n, \tau_n) := \sum_{(u,v)\in E_n} ||f_n(u) - f_n(v)||^2$ using the efficient two-pass algorithm described in Zhang (2023). One should note that the embeddings for interior nodes may change as new leaf nodes are added to the ordinal tree topologies, which is a main difference between our model and other graph autoregressive models.

Message passing networks Using these topological node embeddings as the initial node features, GNNs apply message passing steps to compute the representation vector of nodes that encode topological information of τ_n, where the node features are updated with the information from their neighborhoods in a convolutional manner (Gilmer et al., 2017). More concretely, the l-th round of message passing is implemented by

$$m_n^l(u, v) = M_l(f_n^l(u), f_n^l(v)), \tag{7a}$$

$$f_n^{l+1}(v) = U_l\left(\{m_n^l(u, v); u \in \mathcal{N}(v)\}\right), \tag{7b}$$

where M_l and U_l are the message function and updating function in the l-th round, and $\mathcal{N}(v)$ is the neighborhood of the node v. In our implementations, the message function takes the form

$$M_l(x, y) = \tilde{M}_l\left(c(y, x - y)\right), \tag{8}$$

where $c(\cdot, \cdot)$ is the concatenation operator and $\tilde{M}_l$ is a single-layer perceptron; the updating function U_l is the elementwise maximum operator. Our choices of M_l and U_l follow the edge convolution operator (Wang et al., 2018), while other variants of GNNs can also be applied. The final node features of τ_n are given by $\{f_n^L(v) : v \in V_n\}$ after L rounds of message passing.

Node hidden states The conditional distribution $Q(\cdot|e_{<n})$ is highly complicated as it has to capture how x_{n+1} can be added to τ_n based on how previous leaf nodes are added to form the tree topologies.

A common approach is to use RNNs to model this complex distribution that strikes a good balance between expressiveness and scalability (You et al., 2018b; Liao et al., 2019). In our model, after obtaining the final node features of τ_n, a gated recurrent unit (GRU) (Cho et al., 2014) follows, i.e.

$$h_n(v) = \mathrm{GRU}(h_{n-1}(v), f_n^L(v)), \tag{9}$$

where $h_n(v)$ is the hidden state of v at the n-th generation step and is initialized to zero for the newly added nodes including those in τ_3. The node hidden states $\{h_n(v); v \in V_n\}$, therefore, contain the information of all the tree topologies generated so far which can be used for conditional distribution modeling.

Time guided readout We now construct the distribution $Q(\cdot|e_{<n})$ over edge decisions based on the node hidden states. As mentioned before, a main difference between our model and other graph autoregressive models is that the node embedding $f_n^0(v)$ of a node v may vary with the time step n. We, therefore, incorporate time embeddings into the readout step which first forms the edge features $o_n(e) \in \mathbb{R}$ of $e = (u, v)$ using

$$p_n(e) = P\left(h_n(u) + b_n, h_n(v) + b_n\right), \tag{10a}$$

$$o_n(e) = R\left(p_n(e) + b_n\right), \tag{10b}$$

where $b_n = \mathrm{MLP}^b(\mathrm{emb}(n))$, $\mathrm{emb}(n)$ is the sinusoidal positional embedding of time step n that is widely used in Transformers (Vaswani et al., 2017), MLP^b is 2-layer multi-layer perceptrons (MLPs), P is the pooling layer implemented as 2-layer MLPs followed by an elementwise maximum operator, and R is the readout function implemented as 2-layer MLPs with a scalar output. Then the conditional distribution for edge decision is

$$Q(\cdot|e_{<n}) \sim \mathrm{Discrete}\left(q_n\right), \quad q_n = \mathrm{softmax}\left(\{o_n(e)\}_{e\in E_n}\right), \tag{11}$$

where probability vector $q_n \in \mathbb{R}^{|E_n|}$ for parametrizing $Q(\cdot|e_{<n})$ is obtained by applying a softmax function to all the time guided edge features in equation (10b).

As the last step, we sample an edge $e_n \in E_n$ from the discrete distribution in equation (11) and add the leaf node x_{n+1} to e_n as described in Section 3.1. This way, we update the ordinal tree topology from τ_n of rank n to τ_{n+1} of rank $n+1$. We can repeat this procedure until an ordinal tree topology $\tau = \tau_N$ of rank N is reached. The probability of τ then takes the form

$$Q_{\boldsymbol{\phi}}(\tau) = Q_{\boldsymbol{\phi}}(D) = \prod_{n=3}^{N-1} Q_{\boldsymbol{\phi}}(e_n|e_{<n}), \tag{12}$$

where D is the decision sequence and $\boldsymbol{\phi}$ are the learnable parameters in the model. We call this autoregressive model for tree topologies ARTree and summarize it in Algorithm 1. Note that equation (12) can also be used for tree topology density estimation tasks where the decision sequence $D = g^{-1}(\tau)$ is obtained from the decomposition process that enjoys a linear time complexity (Lemma 1). Compared to SBNs, ARTree does not rely on heuristic features for parameterization and can provide distributions whose support spans the entire tree topology space. Although different taxa orders may affect the performance of ARTree, we find this effect is negligible in our experiments.

4 Experiments

In this section, we test the effectiveness and efficiency of ARTree for phylogenetic inference on two benchmark tasks: tree topology density estimation (TDE) and variational Bayesian phylogenetic inference (VBPI). In all experiments, we report the inclusive KL divergence from posterior estimates to the ground truth to measure the approximation error of different methods. We will use "KL divergence" for inclusive KL divergence throughout this section unless otherwise specified.

Experimental setup We perform experiments on eight data sets which we will call DS1-8. These data sets, consisting of sequences from 27 to 64 eukaryote species with 378 to 2520 site observations, are commonly used to benchmark phylogenetic MCMC methods (Hedges et al., 1990; Garey et al., 1996; Yang & Yoder, 2003; Henk et al., 2003; Lakner et al., 2008; Zhang & Blackwell, 2001; Yoder & Yang, 2004; Rossman et al., 2001; Höhna & Drummond, 2012; Larget, 2013; Whidden & Matsen IV,

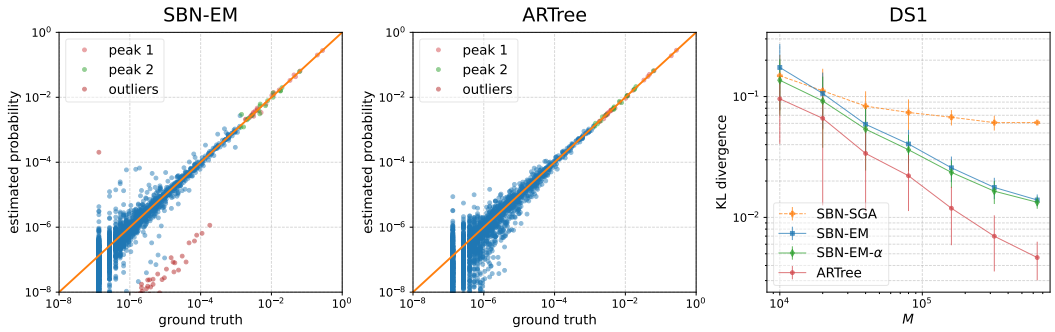

Figure 2: Performances of different methods for TDE on DS1. **Left/Middle**: Comparison of the ground truth and the estimated probabilities using SBN-EM and ARTree. A tree topology is marked as an outlier if it satisfies $|\log(\text{estimated probability}) - \log(\text{ground truth})| > 2$. **Right**: The KL divergence as a function of the sample size. The results are averaged over 10 replicates with one standard deviation as the error bar.

2015). For the Bayesian setting, we focus on the joint posterior distribution of the tree topologies and the branch lengths and assume a uniform prior on the tree topologies, an i.i.d. exponential prior $\text{Exp}(10)$ on branch lengths, and the simple JC substitution model (Jukes et al., 1969). For each of these data sets, we run 10 single-chain MrBayes (Ronquist et al., 2012) for one billion iterations, collect samples every 1000 iterations, and discard the first 25% samples as burn-in. These samples form the ground truth of the marginal distribution of tree topologies to which we will compare the posterior estimates obtained by different methods. All GNNs have $L = 2$ rounds in the message passing step. All the activation functions in MLPs are exponential linear units (ELUs) (Clevert et al., 2015). To stabilize outputs and accelerate training, we add a layer normalization block after each linear transformation in $\tilde{M}_l$, P, and R. The taxa order is set to the lexicographical order of the corresponding species names in all experiments except the ablation studies. All the experiments are run on an Intel Xeon Platinum 9242 processor. All models are implemented in PyTorch (Paszke et al., 2019) and trained with the Adam (Kingma & Ba, 2015) optimizer. The learning rate is 0.001 for SBNs, 0.0001 for ARTree, and 0.001 for the branch length model.

4.1 Tree topology density estimation

We first investigate the performance of ARTree for tree topology density estimation given the MCMC posterior samples on DS1-8. Following Zhang & Matsen IV (2018), we run MrBayes on each data set with 10 replicates of 4 chains and 8 runs until the runs have ASDSF (the standard convergence criteria used in MrBayes) less than 0.01 or a maximum of 100 million iterations. The training data sets are formed by collecting samples every 100 iterations and discarding the first 25%. Now, given a training data set $\mathcal{M} = \{\tau_m\}_{m=1}^M$, we train ARTree via maximum likelihood estimation using stochastic gradient ascent. In each iteration, the stochastic gradient is obtained as follows

$$\nabla_{\boldsymbol{\phi}} L(\boldsymbol{\phi}; \mathcal{M}) = \frac{1}{B} \sum_{b=1}^{B} \nabla_{\boldsymbol{\phi}} \log Q_{\boldsymbol{\phi}}(\tau_{m_b}), \tag{13}$$

where a minibatch $\{\tau_{m_b}\}_{b=1}^B$ is randomly sampled from $\mathcal{M}$. We compare ARTree to SBN baselines including SBN-EM, SBN-EM-α, and SBN-SGA. For SBN-EM and SBN-EM-α, we use the same setting as previously done in Zhang & Matsen IV (2018) (see Appendix A for more details). In addition to these EM variants, a gradient based method for SBNs called SBN-SGA is considered, where SBNs are reparametrized with the latent parameters initialized as zero (see equation (19) in Appendix B) and optimized via stochastic gradient ascent, similarly to ARTree. For both ARTree and SBN-SGA, the results are collected after 200000 parameter updates with batch size $B = 10$.

The left and middle plots of Figure 2 show a comparison between ARTree and SBN-EM on DS1, which has a peaky posterior distribution. Compared to SBN-EM, ARTree provides more accurate probability estimates for tree topologies on the peaks and significantly reduces the large biases in the low probability region (the crimson dots). The right plot of Figure 2 shows the KL divergence of different methods as a function of the sample size of the training data. We see that ARTree

Table 1: KL divergences to the ground truth of different methods across 8 benchmark data sets. Sampled trees column shows the numbers of unique tree topologies in the training sets formed by MrBayes runs. The results are averaged over 10 replicates. The results of SBN-EM, SBN-EM-α are from Zhang & Matsen IV (2018).

Data set	#Taxa	#Sites	Sampled trees	KL divergence to ground truth			
				SBN-EM	SBN-EM-α	SBN-SGA	ARTree
DS1	27	1949	1228	0.0136	0.0130	0.0504	**0.0045**
DS2	29	2520	7	0.0199	0.0128	0.0118	**0.0097**
DS3	36	1812	43	0.1243	0.0882	0.0922	**0.0548**
DS4	41	1137	828	0.0763	0.0637	0.0739	**0.0299**
DS5	50	378	33752	0.8599	0.8218	0.8044	**0.6266**
DS6	50	1133	35407	0.3016	0.2786	0.2674	**0.2360**
DS7	59	1824	1125	0.0483	0.0399	0.0301	**0.0191**
DS8	64	1008	3067	0.1415	0.1236	0.1177	**0.0741**

consistently outperforms SBN based methods for all Ms. Moreover, as the sample size M increases, ARTree keeps providing better approximation while SBNs start to level off when M is large. This indicates the superior flexibility of ARTree over SBNs for tree topology density estimation.

Table 1 shows the KL divergences of different methods on DS1-8. We see that ARTree outperforms SBN based methods on all data sets. The gradient based method SBN-SGA is better than SBN-EM on most of the data sets because SBN-EM is well initialized (Zhang & Matsen IV, 2018) and more likely to get trapped in local modes. From this point of view, the comparison between ARTree and SBN-SGA is fair because they both use a uniform initialization that facilitates exploration.

4.2 Variational Bayesian phylogenetic inference

Our second experiment is on VBPI, where we compare ARTree to SBNs for tree topology variational approximations. Both methods are evaluated on the aforementioned benchmark data sets DS1-8. Following Zhang & Matsen IV (2019), we use the simplest SBN and gather the subsplit support from 10 replicates of 10000 ultrafast maximum likelihood bootstrap trees (Minh et al., 2013). For both ARTree and SBNs, the collaborative branch lengths are parametrized using the learnable topological features with the edge convolution operator (EDGE) for GNNs (Zhang, 2023). We set $K = 10$ for the multi-sample lower bound (4) and use the following annealed unnormalized posterior at the i-th iteration

$$p(\boldsymbol{Y}, \tau, \boldsymbol{q}; \beta_i) = p(\boldsymbol{Y}|\tau, \boldsymbol{q})^{\beta_i} p(\tau, \boldsymbol{q}) \tag{14}$$

where $\beta_i = \min\{1.0, 0.001 + i/H\}$ is the inverse temperature that goes from 0.001 to 1 after H iterations. For ARTree, a long annealing period $H = 200000$ is used for DS6 and DS7 due to the highly multimodal posterior distributions on these two data sets (Whidden & Matsen IV, 2015) and $H = 100000$ is used for the other data sets. For SBNs, we set $H = 100000$ for all data sets. The Monte Carlo gradient estimates for the tree topology parameters and the branch lengths parameters are obtained via VIMCO (Mnih & Rezende, 2016) and the reparametrization trick (Zhang & Matsen IV, 2019) respectively. The results are collected after 400000 parameter updates.

The left plot in Figure 3 shows the evidence lower bound (ELBO) as a function of the number of iterations on DS1. Although the larger support of ARTree adds to the complexity of training for tree topology variational approximation, we see that by the time SBN based methods converge, ARTree based methods achieve comparable (if not better) lower bounds and finally surpass the SBN baselines in the end. We also find that using fewer particles $(K = 5)$ in the training objective tends to provide larger ELBO. Moreover, time guidance turns out to be crucial for ARTree, as evidenced by the significant performance drop when it is turned off. As shown in the middle plot, compared to SBNs, ARTree can provide a more accurate variational approximation of the tree topology posterior. To investigate the effect of taxa orders on ARTree, we randomly sample 50 taxa orders and report the KL divergence for each order in the right plot of Figure 3. We find that ARTree exhibits weak randomness as the taxa order varies and consistently outperforms SBNs by a large margin.

Table 2 shows the KL divergences to the ground truth, evidence lower bound (ELBO), 10-sample lower bound (LB-10), and marginal likelihood (ML) estimates obtained by different methods on

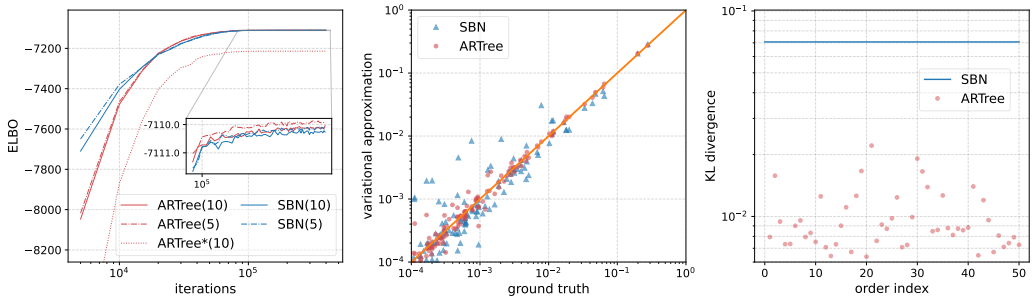

Figure 3: Performances of ARTree and SBN as tree topology variational approximations for VBPI on DS1. **Left**: the evidence lower bound (ELBO) as a function of iterations. The number of particles used in the training objective are in the brackets. The ARTree* method refers to ARTree without time guidance, i.e. $b_n = 0$ for all n in the readout step. **Middle**: variational approximations vs ground truth posterior probabilities of the tree topologies. **Right**: KL divergences across 50 random taxa orders. The KL divergence of SBNs is averaged over 10 independent trainings.

Table 2: KL divergences to the ground truth, evidence lower bound (ELBO), 10-sample lower bound (LB-10), and marginal likelihood (ML) estimates of different methods across 8 benchmark data sets. GT trees row shows the number of unique tree topologies in the ground truth. The marginal likelihood estimates are obtained via importance sampling using 1000 samples. The KL results are averaged over 10 independent trainings. For ELBO, LB-10, and ML, the results are averaged over 100, 100, and 1000 independent runs respectively with standard deviation in the brackets. For ELBO and LB-10, a larger mean is better; for ML, a smaller standard deviation is better.

	Data set	DS1	DS2	DS3	DS4	DS5	DS6	DS7	DS8
	# Taxa	27	29	36	41	50	50	59	64
	# Sites	1949	2520	1812	1137	378	1133	1824	1008
	GT trees	2784	42	351	11505	1516877	809765	11525	82162
KL	SBN	0.0707	0.0144	0.0554	0.0739	1.2472	0.3795	0.1531	0.3173
	ARTree	**0.0097**	**0.0004**	**0.0064**	**0.0219**	**0.8979**	**0.2216**	**0.0123**	**0.1231**
ELBO	SBN	-7110.24(0.03)	-26368.88(0.03)	-33736.22(0.02)	-13331.83(0.03)	-8217.80(0.04)	**-6728.65(0.06)**	-37334.85(0.04)	-8655.05(0.05)
	ARTree	**-7110.09(0.04)**	**-26368.78(0.07)**	**-33736.17(0.08)**	**-13331.82(0.05)**	**-8217.68(0.04)**	**-6728.65(0.06)**	**-37334.84(0.13)**	**-8655.03(0.05)**
LB-10	SBN	-7108.69(0.02)	-26367.87(0.02)	-33735.26(0.02)	-13330.29(0.02)	-8215.42(0.04)	**-6725.33(0.04)**	-37332.58(0.03)	-8651.78(0.04)
	ARTree	**-7108.68(0.02)**	**-26367.86(0.02)**	**-33735.25(0.02)**	**-13330.27(0.03)**	**-8215.34(0.03)**	**-6725.33(0.04)**	**-37332.54(0.03)**	**-8651.73(0.04)**
ML	SBN	**-7108.41(0.15)**	-26367.71(0.08)	**-33735.09(0.09)**	-13329.94(0.20)	-8214.62(0.40)	**-6724.37(0.43)**	-37331.97(0.28)	-8650.64(0.50)
	ARTree	-7108.41(0.19)	**-26367.71(0.07)**	**-33735.09(0.09)**	**-13329.94(0.17)**	**-8214.59(0.34)**	-6724.37(0.46)	**-37331.95(0.27)**	**-8650.61(0.48)**

DS1-8. We find that ARTree achieves smaller KL divergences than SBNs across all data sets and performs on par or better than SBNs for lower bound and marginal likelihood estimation. Compared to SBNs, the ELBOs provided by ARTree tend to have larger variances, especially on DS2, DS3, and DS7, which is partly due to the larger support of ARTree that spans the entire tree topology space (see more discussions in Appendix E).

5 Conclusion

In this paper, we introduced ARTree, a deep autoregressive model over tree topologies for phylogenetic inference. Unlike SBNs that rely on hand-engineered features for parameterization, ARTree is built on top of learnable topological features (Zhang, 2023) via GNNs which allows for a rich family of distributions over phylogenetic tree topologies without requiring domain expertise. Therefore, ARTree does not have to rely on hand-engineered design and can provide more flexible distributions whose support spans the entire tree topology space. Moreover, as an autoregressive model, ARTree also allows simple forward sampling procedures, which makes it readily usable for variational Bayesian phylogenetic inference. In experiments, we showed that ARTree outperforms SBNs on a benchmark of challenging real data tree topology density estimation and variational Bayesian phylogenetic inference problems, especially in terms of tree topology posterior approximation accuracy.

References

Stephen W. Attwood, Sarah C. Hill, David M. Aanensen, Thomas R. Connor, and Oliver G. Pybus. Phylogenetic and phylodynamic approaches to understanding and combating the early sars-cov-2 pandemic. *Nature Reviews. Genetics*, 23:547 – 562, 2022.

Nicola De Cao and Thomas Kipf. Molgan: An implicit generative model for small molecular graphs, 2018. URL https://arxiv.org/abs/1805.11973.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv: Learning*, 2015.

Hanjun Dai, Azade Nazi, Yujia Li, Bo Dai, and Dale Schuurmans. Scalable deep generative modeling for sparse graphs. In *International Conference on Machine Learning*, pp. 2302–2312, 2020.

Rob DeSalle and George Amato. The expansion of conservation genetics. *Nat. Rev. Genet.*, 5(9): 702–712, September 2004. ISSN 1471-0056. doi: 10.1038/nrg1425. URL http://dx.doi.org/10.1038/nrg1425.

Vu Dinh, Arman Bilge, Cheng Zhang, and Frederick A Matsen IV. Probabilistic Path Hamiltonian Monte Carlo. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1009–1018, July 2017. URL http://proceedings.mlr.press/v70/dinh17a.html.

Vu C. Dinh, Aaron E. Darling, and Frederick A. Matsen IV. Online bayesian phylogenetic inference: Theoretical foundations via sequential monte carlo. *Systematic Biology*, 67:503 – 517, 2016.

Louis du Plessis, John T McCrone, Alexander E Zarebski, Verity Hill, Christopher Ruis, Bernardo Gutierrez, Jayna Raghwani, Jordan Ashworth, Rachel Colquhoun, Thomas R Connor, Nuno R Faria, Ben Jackson, Nicholas J Loman, Áine O'Toole, Samuel M Nicholls, Kris V Parag, Emily Scher, Tetyana I Vasylyeva, Erik M Volz, Alexander Watts, Isaac I Bogoch, Kamran Khan, COVID-19 Genomics UK (COG-UK) Consortium†, David M Aanensen, Moritz U G Kraemer, Andrew Rambaut, and Oliver G Pybus. Establishment and lineage dynamics of the SARS-CoV-2 epidemic in the UK. *Science*, January 2021. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.abf2946. URL https://science.sciencemag.org/content/early/2021/01/07/science.abf2946.

Gytis Dudas, Luiz Max Carvalho, Trevor Bedford, Andrew J Tatem, Guy Baele, Nuno R Faria, Daniel J Park, Jason T Ladner, Armando Arias, Danny Asogun, Filip Bielejec, Sarah L Caddy, Matthew Cotten, Jonathan D'Ambrozio, Simon Dellicour, Antonino Di Caro, Joseph W Diclaro, Sophie Duraffour, Michael J Elmore, Lawrence S Fakoli, Ousmane Faye, Merle L Gilbert, Sahr M Gevao, Stephen Gire, Adrianne Gladden-Young, Andreas Gnirke, Augustine Goba, Donald S Grant, Bart L Haagmans, Julian A Hiscox, Umaru Jah, Jeffrey R Kugelman, Di Liu, Jia Lu, Christine M Malboeuf, Suzanne Mate, David A Matthews, Christian B Matranga, Luke W Meredith, James Qu, Joshua Quick, Suzan D Pas, My V T Phan, Georgios Pollakis, Chantal B Reusken, Mariano Sanchez-Lockhart, Stephen F Schaffner, John S Schieffelin, Rachel S Sealfon, Etienne Simon-Loriere, Saskia L Smits, Kilian Stoecker, Lucy Thorne, Ekaete Alice Tobin, Mohamed A Vandi, Simon J Watson, Kendra West, Shannon Whitmer, Michael R Wiley, Sarah M Winnicki, Shirlee Wohl, Roman Wölfel, Nathan L Yozwiak, Kristian G Andersen, Sylvia O Blyden, Fatorma Bolay, Miles W Carroll, Bernice Dahn, Boubacar Diallo, Pierre Formenty, Christophe Fraser, George F Gao, Robert F Garry, Ian Goodfellow, Stephan Günther, Christian T Happi, Edward C Holmes, Brima Kargbo, Sakoba Keïta, Paul Kellam, Marion P G Koopmans, Jens H Kuhn, Nicholas J Loman, N'faly Magassouba, Dhamari Naidoo, Stuart T Nichol, Tolbert Nyenswah, Gustavo Palacios, Oliver G Pybus, Pardis C Sabeti, Amadou Sall, Ute Ströher, Isatta Wurie, Marc A Suchard, Philippe Lemey, and Andrew Rambaut. Virus genomes reveal factors that spread and sustained the ebola epidemic. *Nature*, April 2017. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature22040. URL http://dx.doi.org/10.1038/nature22040.

J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17:268–276, 1981.

Joseph Felsenstein. *Inferring phylogenies*. Sinauer associates, 2 edition, 2004.

J. R. Garey, T. J. Near, M. R. Nonnemacher, and S. A. Nadler. Molecular evidence for Acanthocephala as a subtaxon of Rotifera. *Mol. Evol.*, 43:287–292, 1996.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *ArXiv*, abs/1704.01212, 2017.

S. B. Hedges, K. D. Moberg, and L. R. Maxson. Tetrapod phylogeny inferred from 18S and 28S ribosomal RNA sequences and review of the evidence for amniote relationships. *Mol. Biol. Evol.*, 7:607–633, 1990.

D. A. Henk, A. Weir, and M. Blackwell. Laboulbeniopsis termitarius, an ectoparasite of termites newly recognized as a member of the Laboulbeniomycetes. *Mycologia*, 95:561–564, 2003.

Sebastian Höhna and Alexei J. Drummond. Guided tree topology proposals for Bayesian phylogenetic inference. *Syst. Biol.*, 61(1):1–11, January 2012. ISSN 1063-5157. doi: 10.1093/sysbio/syr074. URL `http://dx.doi.org/10.1093/sysbio/syr074`.

J. P. Huelsenbeck, F. Ronquist, R. Nielsen, and J. P. Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294:2310–2314, 2001.

Wengong Jin, Regina Barzilay, and T. Jaakkola. Junction tree variational autoencoder for molecular graph generation. *ArXiv*, abs/1802.04364, 2018.

Thomas H Jukes, Charles R Cantor, et al. Evolution of protein molecules. *Mammalian protein metabolism*, 3:21–132, 1969.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

C. Lakner, P. van der Mark, J. P. Huelsenbeck, B. Larget, and F. Ronquist. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. *Syst. Biol.*, 57:86–103, 2008.

Bret Larget. The estimation of tree posterior probabilities using conditional clade probability distributions. *Syst. Biol.*, 62(4):501–511, July 2013. ISSN 1063-5157. doi: 10.1093/sysbio/syt014. URL `http://dx.doi.org/10.1093/sysbio/syt014`.

Bret R. Larget and D. L. Simon. Markov chasin monte carlo algorithms for the bayesian analysis of phylogenetic trees. *Molecular Biology and Evolution*, 16:750–750, 1999.

Y. Li, O. Vinyals, C. Dyer, R. Pascanu, and P. Battaglia. Learning deep generative models of graphs, 2018. URL `https://arxiv.org/abs/1803.03324`.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Charlie Nash, William L. Hamilton, David Kristjanson Duvenaud, Raquel Urtasun, and Richard S. Zemel. Efficient graph generation with graph recurrent attention networks. *ArXiv*, abs/1910.00760, 2019.

B. Mau, M. Newton, and B. Larget. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. *Biometrics*, 55:1–12, 1999.

Bui Quang Minh, Minh Anh Nguyen, and Arndt von Haeseler. Ultrafast approximation for phylogenetic bootstrap. *Molecular Biology and Evolution*, 30:1188 – 1195, 2013.

Andriy Mnih and Danilo Jimenez Rezende. Variational inference for monte carlo objectives. In *International Conference on Machine Learning*, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems*, 2019.

Fredrik Ronquist, Maxim Teslenko, Paul Van Der Mark, Daniel L Ayres, Aaron Darling, Sebastian Höhna, Bret Larget, Liang Liu, Marc A Suchard, and John P Huelsenbeck. Mrbayes 3.2: efficient bayesian phylogenetic inference and model choice across a large model space. *Systematic biology*, 61(3):539–542, 2012.

A. Y. Rossman, J. M. Mckemy, R. A. Pardo-Schultheiss, and H. J. Schroers. Molecular studies of the Bionectriaceae using large subunit rDNA sequences. *Mycologia*, 93:100–110, 2001.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1esMkHYPr.

Martin Simonovsky and Nikos Komodakis. Graphvae: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pp. 412–422, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38:1 – 12, 2018.

Chris Whidden and Frederick A Matsen IV. Quantifying MCMC exploration of phylogenetic tree space. *Syst. Biol.*, 64(3):472–491, May 2015. ISSN 1063-5157, 1076-836X. doi: 10.1093/sysbio/syv006. URL http://dx.doi.org/10.1093/sysbio/syv006.

Z. Yang and A. D. Yoder. Comparison of likelihood and Bayesian methods for estimating divergence times using multiple gene loci and calibration points, with application to a radiation of cute-looking mouse lemur species. *Syst. Biol.*, 52:705–716, 2003.

Ziheng Yang and Bruce Rannala. Bayesian phylogenetic inference using dna sequences: a markov chain monte carlo method. *Molecular biology and evolution*, 14(7):717–724, 1997.

A. D. Yoder and Z. Yang. Divergence datas for Malagasy lemurs estimated from multiple gene loci: geological and evolutionary context. *Mol. Ecol.*, 13:757–773, 2004.

Jiaxuan You, Bowen Liu, Zhitao Ying, Vijay Pande, and Jure Leskovec. Graph convolutional policy network for goal-directed molecular graph generation. In *Advances in Neural Information Processing Systems*, pp. 6410–6421, 2018a.

Jiaxuan You, Rex Ying, Xiang Ren, William L. Hamilton, and Jure Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International Conference on Machine Learning*, 2018b.

Cheng Zhang. Improved variational bayesian phylogenetic inference with normalizing flows. In *Neural Information Processing Systems*, 2020.

Cheng Zhang. Learnable topological features for phylogenetic inference via graph neural networks. In *International Conference on Learning Representations*, 2023.

Cheng Zhang and Frederick A Matsen IV. Generalizing tree probability estimation via bayesian networks. In *Neural Information Processing Systems*, 2018.

Cheng Zhang and Frederick A Matsen IV. Variational bayesian phylogenetic inference. In *International Conference on Learning Representations*, 2019.

Cheng Zhang and Frederick A Matsen IV. A variational approach to Bayesian phylogenetic inference, 2022. URL https://arxiv.org/abs/2204.07747.

N. Zhang and M. Blackwell. Molecular phylogeny of dogwood anthracnose fungus (Discula destructiva) and the Diaporthales. *Mycologia*, 93:355–365, 2001.

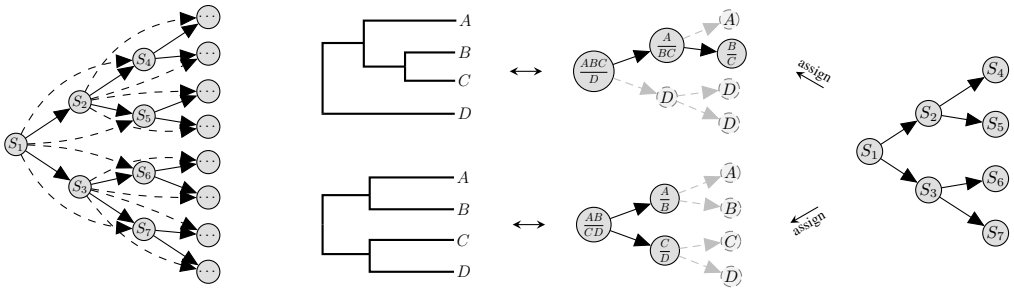

Figure 4: Subsplit Bayesian networks and a simple example for a leaf set of 4 taxa (denoted by A, B, C, D respectively). **Left:** General subsplit Bayesian networks. The solid full and complete binary tree network is $B^*_{\mathcal{X}}$. The dashed arrows represent the additional dependence for more expressiveness. **Middle Left:** Examples of (rooted) phylogenetic trees that are hypothesized to model the evolutionary history of the taxa. **Middle Right:** The corresponding subsplit assignments for the trees. For ease of illustration, subsplit (Y, Z) is represented as $\frac{Y}{Z}$ in the graph. **Right:** The SBN for this example, which is $\mathcal{B}^*_{\mathcal{X}}$ in this case.

A Details of subsplit Bayesian networks

One recent and expressive graphical model that provides a flexible family of distributions over tree topologies is the subsplit Bayesian network, as proposed by Zhang & Matsen IV (2018). Let $\mathcal{X}$ be the set of N labeled leaf nodes. A non-empty set C of $\mathcal{X}$ is referred to as a *clade* and the set of all clades of $\mathcal{X}$, denoted by $\mathcal{C}(\mathcal{X})$, is a totally ordered set with a partial order $\succ$ (e.g., lexicographical order) defined on it. An ordered pair of clades (W, Z) is called a *subsplit* of a clade C if it is a bipartition of C, i.e., $W \succ Z$, $W \cap Z = \emptyset$, and $W \cup Z = C$.

Definition 2 (Subsplit Bayesian Network)**.** *A subsplit Bayesian network (SBN) $\mathcal{B}_{\mathcal{X}}$ on a leaf node set $\mathcal{X}$ of size N is defined as a Bayesian network whose nodes take on subsplit or singleton clade values of $\mathcal{X}$ and has the following properties: (a) The root node of $\mathcal{B}_{\mathcal{X}}$ takes on subsplits of the entire labeled leaf node set $\mathcal{X}$; (b) $\mathcal{B}_{\mathcal{X}}$ contains a full and complete binary tree network $B^*_{\mathcal{X}}$ as a subnetwork; (c) The depth of $B_{\mathcal{X}}$ is $N-1$, with the root counted as depth* 1.

Due to the binary structure of $B^*_{\mathcal{X}}$, the nodes in SBNs can be indexed by denoting the root node with S_1 and two children of S_i with S_{2i} and S_{2i+1} recursively where S_i is an internal node (see the left plot in Figure 4). For any rooted tree topology, by assigning the corresponding subsplits or singleton clades values $\{S_i = s_i\}_{i \geq 1}$ to its nodes, one can uniquely map it into an SBN node assignment (see the middle and right plots in Figure 4).

As Bayesian networks, the SBN based probability of a rooted tree topology τ takes the following form

$$p_{\mathrm{sbn}}(T = \tau) = p(S_1 = s_1) \prod_{i>1} p(S_i = s_i | S_{\pi_i} = s_{\pi_i}), \tag{15}$$

where π_i is the index set of the parents of node i. For unrooted tree topologies, we can also define their SBN based probabilities by viewing them as rooted tree topologies with unobserved roots and integrating out the positions of the root node as follows:

$$p_{\mathrm{sbn}}(T^{\mathrm{u}} = \tau) = \sum_{e \in E(\tau)} p_{\mathrm{sbn}}(\tau^e) \tag{16}$$

where τ^e is the resulting rooted tree topology when the rooting position is on edge e.

In practice, SBNs are parameterized according to the *conditional probability sharing* principle where the conditional probability for parent-child subsplit pairs are shared across the SBN network, regardless of their locations. The set of all conditional probabilities are called conditional probability tables (CPTs). Parameterizing SBNs, therefore, often requires finding an appropriate support of CPTs. For tree topology density estimation, this can be done using the sample of tree topologies that is given as the data set. For variational Bayesian phylogenetic inference, as no sample of tree topologies is available, one often resorts to fast bootstrap or MCMC methods (Minh et al., 2013; Zhang, 2020). Let $\mathbb{S}_{\mathrm{r}}$ denotes the root subsplits and $\mathbb{S}_{\mathrm{ch|pa}}$ denotes the child-parent subsplit pairs in the support. The parameters of SBNs are then $p = \{p_{s_1}; s_1 \in \mathbb{S}_{\mathrm{r}}\} \cup \{p_{s|t}; s|t \in \mathbb{S}_{\mathrm{ch|pa}}\}$ where

$$p_{s_1} = p(S_1 = s_1), \quad p_{s|t} = p(S_i = s | S_{\pi_i} = t), \ \forall i > 1. \tag{17}$$

As a result, the supports of SBN-induced distributions are often limited by the splitting patterns in the observed samples and could not span the entire tree topology space (Zhang & Matsen IV, 2022).

The SBN-EM Algorithm For unrooted tree topologies, the SBN based probability (16) can be viewed as a hidden variable model where the root subsplit is the hidden variable. In this case, SBNs can be trained using the expectation-maximization (EM) algorithm, as proposed by Zhang & Matsen IV (2018). Given a training set $\{\tau_k\}_{k=1}^M$, we first initialize the parameter estimates as $\hat{p}^{EM,(0)}$ (i.e., the simple average estimates as in Zhang & Matsen IV (2018)). In the i-th step, we run the E-step and M-step as follows

- **E-step**: $\forall 1 \le k \le M$, compute $q_k^{(i)}(s_1) = \frac{p\left(\tau_k, s_1 | \hat{p}^{\mathrm{EM},(i)}\right)}{\sum_{s_1 \sim \tau_k} p\left(\tau_k, s_1 | \hat{p}^{\mathrm{EM},(i)}\right)}$ where $s_1 \sim \tau_k$ means the subsplit s_1 can be achieved by placing a "virtual root" on an edge of τ.
- **M-step**: update the parameter estimates by

$$\hat{p}_{s_1}^{\mathrm{EM},(i+1)} = \frac{\bar{m}_{s_1}^{(i)} + \alpha \tilde{m}_{s_1}}{K + \alpha \sum_{s_1 \in \mathbb{S}_{\mathrm{r}}} \tilde{m}_{s_1}}, \quad \bar{m}_{s_1}^{(\mathrm{i})} = \sum_{k=1}^{M} \sum_{e \in E(\tau_k)} q_k^{(i)}(s_1) \mathbb{I}\left(s_{1,k}^e = s_1\right)$$

$$\hat{p}_{s|t}^{\mathrm{EM},(i+1)} = \frac{\bar{m}_{s,t}^{(i)} + \alpha \tilde{m}_{s,t}}{\sum_s \left(\bar{m}_{s,t}^{(i)} + \alpha \tilde{m}_{s,t}\right)}, \quad \bar{m}_{s,t}^{(i)} = \sum_{k=1}^{M} \sum_{e \in E(\tau_k)} q_k^{(i)}\left(s_{1,k}^e\right) \sum_{j>1} \mathbb{I}\left(s_{j,k}^e = s, s_{\pi_j,k}^e = t\right)$$

  where $\mathbb{I}$ is the indicator function, $s_{j,k}^e$ is the node value of S_j in τ_k^e, $\tilde{m}_s$ and $\tilde{m}_{s,t}$ are equivalent counts and α is the regularization coefficient that encourages generalization.

When $\alpha > 0$, this algorithm is called SBN-EM-α.

B Details of variational Bayesian phylogenetic inference

With two variational families $Q_{\boldsymbol{\phi}}(\tau)$ and $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ over the space of tree topologies and branch lengths, the variational Bayesian phylogenetic inference (VBPI) approach forces $Q_{\boldsymbol{\phi},\boldsymbol{\psi}}(\tau, \boldsymbol{q}) = Q_{\boldsymbol{\phi}}(\tau) Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ to approximate the posterior $p(\tau, \boldsymbol{q}|\boldsymbol{Y})$ by maximizing the following multi-sample lower bound

$$L^K(\boldsymbol{\phi}, \boldsymbol{\psi}) = \mathbb{E}_{\{(\tau^i, \boldsymbol{q}^i)\}_{i=1}^K \overset{\text{i.i.d.}}{\sim} Q_{\boldsymbol{\phi},\boldsymbol{\psi}}} \log\left(\frac{1}{K}\sum_{i=1}^{K} \frac{p(\boldsymbol{Y}|\tau^i, \boldsymbol{q}^i) p(\tau^i, \boldsymbol{q}^i)}{Q_{\boldsymbol{\phi}}(\tau^i) Q_{\boldsymbol{\psi}}(\boldsymbol{q}^i|\tau^i)}\right). \tag{18}$$

Gradients of the objective (18) w.r.t. $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ can be estimated by the VIMCO estimator (Mnih & Rezende, 2016) and the reparameterization trick respectively. In the following, we introduce some common choices of $Q_{\boldsymbol{\phi}}(\tau)$ and $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$.

Choice of $Q_{\boldsymbol{\phi}}(\tau)$ Before the proposed ARTree framework in this article, SBNs is the common choice of $Q_{\boldsymbol{\phi}}(\tau)$. As introduced in Appendix A, SBNs provide a probability distribution over unrooted tree topologies in equation (16). Given a subsplit support of CPTs, SBNs can be parameterized as follows

$$p_{s_1} = \frac{\exp(\phi_{s_1})}{\sum_{s' \in \mathbb{S}_{\mathrm{r}}} \exp(\phi_{s'})},\ s_1 \in \mathbb{S}_{\mathrm{r}}; \quad p_{s|t} = \frac{\exp(\phi_{s|t})}{\sum_{s': s'|t \in \mathbb{S}_{\mathrm{ch|pa}}} \exp(\phi_{s'|t})},\ s|t \in \mathbb{S}_{\mathrm{ch|pa}}. \tag{19}$$

The parameters $\boldsymbol{\phi} = \{\phi_{s_1}; s_1 \in \mathbb{S}_{\mathrm{r}}\} \cup \{\phi_{s|t};\ s|t \in \mathbb{S}_{\mathrm{ch|pa}}\}$ are called latent parameters of SBNs.

Choice of $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ The distribution $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ is often taken to be a diagonal lognormal distribution, which can be parametrized using some heuristic features (Zhang & Matsen IV, 2019) or the recently proposed learnable topological features (Zhang, 2023) of τ as follows. For each edge $e = (u, v)$ in τ, one can first obtain the edge features using $h_e = f(h_u, h_v)$ where h_u is the GNN output at node u and f is a permutation invariant function. Then the mean and standard deviation parameters are given by

$$\mu(e, \tau) = \mathrm{MLP}^{\mu}(h_e), \quad \sigma(e, \tau) = \mathrm{MLP}^{\sigma}(h_e)$$

where MLP^{μ} and MLP^{σ} are two multi-layer perceptrons (MLPs).

Algorithm 2: Tree topology decomposition process

Input: a tree topology τ with all of the N leaf nodes.
Output: a decision sequence D.
$\tau_N = (V_N, E_N) \leftarrow$ the tree topology τ;
for $n = N-1, \ldots, 3$ **do**
  Determine the unique neighbor w of the leaf node x_{n+1};
  Determine the two neighbors u and v (except x_{n+1}) of w;
  $V_n \leftarrow V_{n+1} \backslash \{w, x_{n+1}\}$;
  $E_n \leftarrow (E_{n+1} \cup \{(u,v)\}) \backslash \{(w, x_{n+1}), (w,u), (w,v)\}$;
  $\tau_n \leftarrow (V_n, E_n)$;
  $e_n \leftarrow (u, v)$;
end
$D \leftarrow (e_3, \ldots, e_{N-1})$.

C Details of tree topology decomposition process

The tree topology decomposition process, which maps a tree topology to a corresponding decision sequence, is indeed the inverse operation of Algorithm 1. Also, the decomposition process is implemented in a recursive way starting from the tree topology τ_N of rank N. Intuitively, given an ordinal tree topology τ_{n+1} of rank $n+1$, one can detach the leaf node x_{n+1} as well as its unique neighbor w and reconnect the two neighbors of w. The remaining graph, denoted by τ_n, is an ordinal tree topology of rank n; the edge decision e_n is exactly the reconnected edge. This process continues until the unique ordinal tree topology τ_3 of rank 3 is reached. We summarize the sketch of tree topology decomposition process in Algorithm 2.

Given a tree topology τ with all of the N leaf nodes, we can evaluate its ARTree based probability by first mapping it to a decision sequence $D = (e_3, \ldots, e_{N-1})$ following Algorithm 2 and then calculate the probability as the product of conditionals

$$Q(\tau) = Q(D) = \prod_{n=3}^{N-1} Q(e_n | e_{<n}).$$

D The proofs of Theorem 1 and Lemma 1

Theorem 1. *Let $\mathcal{D} = \{D | D = (e_3, \ldots, e_{N-1}),\ e_n \in E_n, \forall\ 3 \le n \le N-1\}$ be the set of all decision sequences of length $N-3$ and $\mathcal{T}$ be the set of all ordinal tree topologies of rank N. Let the map*

$$\begin{array}{rccc} g: & \mathcal{D} & \to & \mathcal{T} \\ & D & \mapsto & \tau \end{array}$$

be the generating process used in ARTree. Then g is a bijection between $\mathcal{D}$ and $\mathcal{T}$.

Proof of Theorem 1 It is obvious that g is a well-defined map from $\mathcal{D}$ to $\mathcal{T}$. To prove it is a bijection, it suffices to show g is injective and surjective.

We first show that g is injective. Assume there are two decision sequences $D^{(1)}, D^{(2)} \in \mathcal{D}$ and $D^{(1)} \neq D^{(2)}$. Let k be the first position where $D^{(1)}$ and $D^{(2)}$ begin to differ, i.e. $e_i^{(1)} = e_i^{(2)}$ for all $i < k$ and $e_k^{(1)} \neq e_k^{(2)}$. If $g(D^{(1)}) = g(D^{(2)}) =: \tau$, one can take the subtree topology of rank k and $k+1$ of τ, τ_k and τ_{k+1}. Noting that e_k refers to the edge in τ where to add the new node x_{n+1}, the equation $e_k^{(1)} \neq e_k^{(2)}$ implies they will induce different τ_{k+1}s. This contradicts the uniqueness of τ_{k+1}. Therefore, we conclude that g is injective.

Next, we prove that g is surjective. For a tree topology τ with rank N, we denote its subtree topology of rank k by τ_k, where $k = 3, \ldots, n$. In fact, for each k, the tree topology τ_{k+1} corresponds to adding the leaf node x_{k+1} to an edge in τ_k, and we denote this edge by e_k. It is easy to verify that the constructed $D = (e_3, \ldots, e_{N-1})$ is a preimage of τ.

Lemma 1. *The time complexity of the decomposition process induced by $g^{-1}(\cdot)$ is $O(N)$.*

Proof of Lemma 1 Assume the tree topology τ is stored as a binary tree data structure, where each node other than the root node also has a parent node pointer. Before decomposing τ, we first build a dictionary of the (n, x_n) mappings by a traversal across τ that maps n to the leaf node x_n in τ, $\forall n \leq N$. The time complexity of building this dictionary is $O(N)$. In the decomposition process, given an ordinal tree topology of rank $n+1$, it costs $O(1)$ time to locate x_{n+1} and determine the unique parent node w of x_{n+1} and the neighbors of w. It is obvious that the time complexity of detaching and reconnecting operations is $O(1)$. One can repeat this procedure for $n = N-1, \ldots, 3$, resulting in a time complexity of $O(N)$. Therefore, the time complexity of the decomposition process induced by $g^{-1}(\cdot)$ is $O(N)$.

E Limitations

As an autoregressive model for phylogenetic tree topology, ARTree provides reliable approximations for target distributions over tree topologies, as evidenced by our real data experiments. However, when incorporated with the branch length model for VBPI, the ELBOs provided by ARTree tend to have larger variances, which we find is caused by the occasional occurrence of "outliers" among samples. In fact, as the support of ARTree spans over the entire tree topology space, it adds to the difficulty of fitting the conditional distribution $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$, compared to SBNs. When combined with ARTree, the approximation accuracy of $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ might be related to the cluster structure of peaks in the tree topology posterior. See Figure 3 in Whidden & Matsen IV (2015) for cluster subtree-prune-and-regraft (SPR) graphs of DS1-8. We also examine the approximation accuracies of $Q_{\boldsymbol{\psi}}(\boldsymbol{q}|\tau)$ trained along with ARTree for those τ in the support of SBNs and find a significant enhancement in ELBO and reduction of variances.

This phenomenon raises an important topic: proper design and optimization of branch length model when the support of tree topology model spans the entire space. We leave this for future work.