

ARTree: A Deep Autoregressive Model for Phylogenetic Inference

Tianyu Xie¹, Cheng Zhang^{2,*}

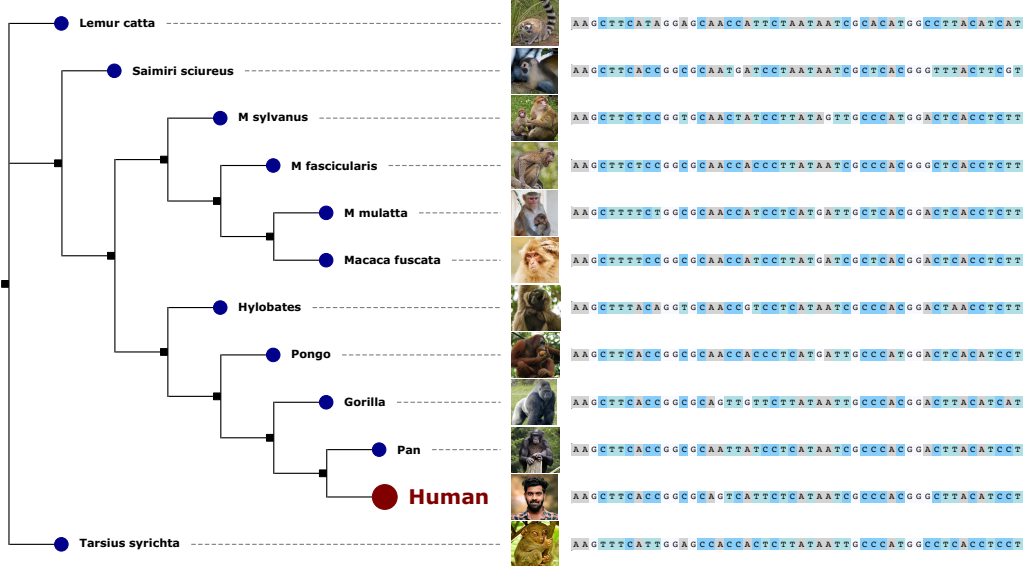
Published at NeurIPS 2023 as a spotlight presentation

¹School of Mathematical Sciences, Peking University

²School of Mathematical Sciences and Center for Statistical Science, Peking University

*Corresponding Author

Phylogenetic Trees



Phylogenetic Trees

Leaf nodes	\longleftrightarrow	Observed species
Internal nodes	\longleftrightarrow	Unobserved ancestor species
Branch length	\longleftrightarrow	Evolutionary time between two species

Phylogenetic Trees

Leaf nodes	\iff	Observed species
Internal nodes	\iff	Unobserved ancestor species
Branch length	\iff	Evolutionary time between two species

- A **phylogenetic tree** is described by a bifurcating tree topology τ and the associated non-negative branch lengths q .

Phylogenetic Trees

Leaf nodes	\iff	Observed species
Internal nodes	\iff	Unobserved ancestor species
Branch length	\iff	Evolutionary time between two species

- A **phylogenetic tree** is described by a bifurcating tree topology τ and the associated non-negative branch lengths \mathbf{q} .
- $\mathbf{Y} = \{Y_1, \dots, Y_N\} \in \Omega^{N \times M}$ are the observed sequences (with characters in Ω) of length M over N species. (e.g. $\Omega = \{A, C, G, T\}$ contain the nucleotides.)

Phylogenetic Trees

[Felsenstein, 2004]

Y_1

Y_2

Y_3

Y_4

$P(\mathbf{Y}|\tau, \mathbf{q})$ follows a continuous-time Markov chain

- Y 's are leaf nodes and X 's are internal nodes.

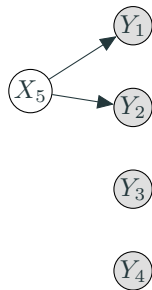
$P(\mathbf{Y}|\tau, \mathbf{q}) =$

Phylogenetic Trees

[Felsenstein, 2004]

$P(\mathbf{Y}|\tau, \mathbf{q})$ follows a continuous-time Markov chain

- Y 's are leaf nodes and X 's are internal nodes.



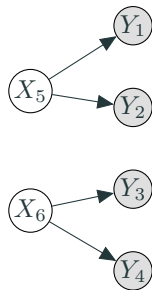
$$P(\mathbf{Y}|\tau, \mathbf{q}) = P(Y_1, Y_2|X_5)$$

Phylogenetic Trees

[Felsenstein, 2004]

$P(\mathbf{Y}|\tau, \mathbf{q})$ follows a continuous-time Markov chain

- Y 's are leaf nodes and X 's are internal nodes.



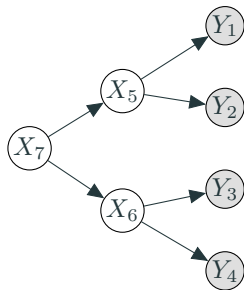
$$P(\mathbf{Y}|\tau, \mathbf{q}) = P(Y_1, Y_2|X_5)P(Y_3, Y_4|X_6)$$

Phylogenetic Trees

[Felsenstein, 2004]

$P(\mathbf{Y}|\tau, \mathbf{q})$ follows a continuous-time Markov chain

- Y 's are leaf nodes and X 's are internal nodes.



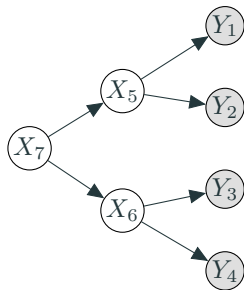
$$P(\mathbf{Y}|\tau, \mathbf{q}) = P(Y_1, Y_2|X_5)P(Y_3, Y_4|X_6)P(X_5, X_6|X_7)$$

Phylogenetic Trees

[Felsenstein, 2004]

$P(\mathbf{Y}|\tau, \mathbf{q})$ follows a continuous-time Markov chain

- Y 's are leaf nodes and X 's are internal nodes.



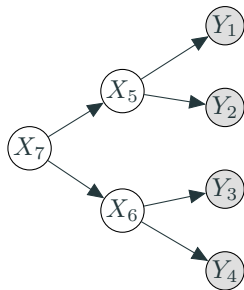
$$P(\mathbf{Y}|\tau, \mathbf{q}) = P(Y_1, Y_2|X_5)P(Y_3, Y_4|X_6)P(X_5, X_6|X_7)\eta(X_7)$$

Phylogenetic Trees

[Felsenstein, 2004]

$P(\mathbf{Y}|\tau, \mathbf{q})$ follows a continuous-time Markov chain

- Y 's are leaf nodes and X 's are internal nodes.



$$P(\mathbf{Y}|\tau, \mathbf{q}) = \sum_X P(Y_1, Y_2|X_5)P(Y_3, Y_4|X_6)P(X_5, X_6|X_7)\eta(X_7)$$

Bayesian Phylogenetic Inference

- The core question in phylogenetic inference is:

Given the biological sequences \mathbf{Y} of observed species, what are the underlying phylogenetic trees?

Bayesian Phylogenetic Inference

- The core question in phylogenetic inference is:

Given the biological sequences \mathbf{Y} of observed species, what are the underlying phylogenetic trees?

- Now, in a Bayesian framework: (i) The **likelihood** function is $P(\mathbf{Y}|\tau, \mathbf{q})$. (ii) The assumed **prior** distribution is $P(\tau, \mathbf{q})$.

Bayesian Phylogenetic Inference

- The core question in phylogenetic inference is:

Given the biological sequences \mathbf{Y} of observed species, what are the underlying phylogenetic trees?

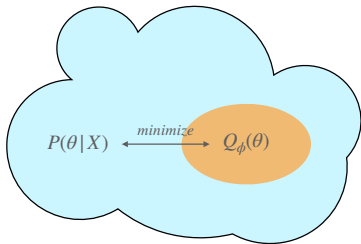
- Now, in a Bayesian framework: (i) The **likelihood** function is $P(\mathbf{Y}|\tau, \mathbf{q})$. (ii) The assumed **prior** distribution is $P(\tau, \mathbf{q})$.

Then the above question turns into:

How can we infer the **posterior** distribution:

$$P(\tau, \mathbf{q}|\mathbf{Y}) = \frac{P(\mathbf{Y}|\tau, \mathbf{q})P(\tau, \mathbf{q})}{p(\mathbf{Y})} \propto P(\mathbf{Y}|\tau, \mathbf{q})P(\tau, \mathbf{q})$$

Variational Inference



$$\begin{aligned}\phi^* &= \arg \min_{\phi} \text{KL} (Q_{\phi}(\theta) \| P(\theta|X)) \\ &= \arg \max_{\phi} \mathbb{E}_{\theta \sim Q_{\phi}(\theta)} \log \left(\frac{P(X|\theta)P(\theta)}{Q_{\phi}(\theta)} \right)\end{aligned}$$

Variational inference (VI) [Blei et al., 2017] turns inference into optimization:

- Specify a variational family $\{Q_{\phi}(\theta)\}_{\phi \in \Phi}$ over the model latent variables.
- Find the optimal variational approximate distribution $Q_{\phi^*}(\theta)$ by minimizing the **KL divergence** or maximizing the **evidence lower bound (ELBO)**.

Variational Bayesian Phylogenetic Inference

[Zhang and Matsen IV, 2019]

- Challenges for Bayesian phylogenetic inference:
 - Combinatorially explosive size ($(2n - 5)!!$) of the tree topology space.
 - The composite structure of discrete (tree topology) and continuous (branch length) components

Variational Bayesian Phylogenetic Inference

[Zhang and Matsen IV, 2019]

- Challenges for Bayesian phylogenetic inference:
 - Combinatorially explosive size ($(2n - 5)!!$) of the tree topology space.
 - The composite structure of discrete (tree topology) and continuous (branch length) components
- Variational family:

$$Q_{\phi, \psi}(\tau, \mathbf{q}) = \overset{\text{branch length}}{Q_{\psi}(\mathbf{q}|\tau)} \cdot \overset{\text{tree topology}}{Q_{\phi}(\tau)}$$

Variational Bayesian Phylogenetic Inference

[Zhang and Matsen IV, 2019]

- Challenges for Bayesian phylogenetic inference:
 - Combinatorially explosive size ($(2n - 5)!!$) of the tree topology space.
 - The composite structure of discrete (tree topology) and continuous (branch length) components
- Variational family:

$$Q_{\phi, \psi}(\tau, \mathbf{q}) = \overset{\text{branch length}}{Q_{\psi}(\mathbf{q}|\tau)} \cdot \overset{\text{tree topology}}{Q_{\phi}(\tau)}$$

- Multi-sample lower bound:

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

Variational Bayesian Phylogenetic Inference

- Variational Bayesian Phylogenetic Inference (VBPI) is done by maximizing the lower bound (1), i.e.

$$\phi^*, \psi^* = \arg \max_{\phi, \psi} L^K(\phi, \psi)$$

Variational Bayesian Phylogenetic Inference

- Variational Bayesian Phylogenetic Inference (VBPI) is done by maximizing the lower bound (1), i.e.

$$\phi^*, \psi^* = \arg \max_{\phi, \psi} L^K(\phi, \psi)$$

- Choices of tree topology model $Q_{\phi}(\tau)$ (**focus of this talk**):
 - Subsplit Bayesian Networks (SBNs) [Zhang and Matsen IV, 2018].
 - **ARTree** (proposed in this paper) [Xie and Zhang, 2023].

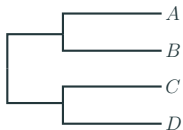
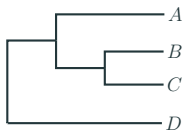
Variational Bayesian Phylogenetic Inference

- Variational Bayesian Phylogenetic Inference (VBPI) is done by maximizing the lower bound (1), i.e.

$$\phi^*, \psi^* = \arg \max_{\phi, \psi} L^K(\phi, \psi)$$

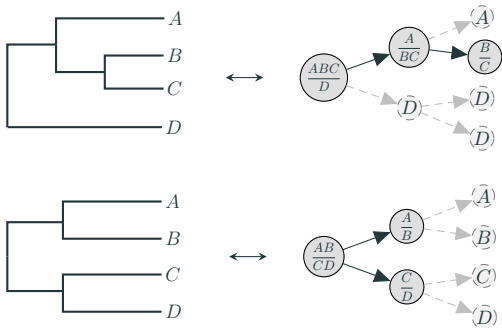
- Choices of tree topology model $Q_\phi(\tau)$ (**focus of this talk**):
 - Subsplit Bayesian Networks (SBNs) [Zhang and Matsen IV, 2018].
 - **ARTree** (proposed in this paper) [Xie and Zhang, 2023].
- Choices of branch model $Q_\psi(\mathbf{q}|\tau)$:
 - Normalizing Flows [Zhang et al., 2020].
 - Graph Neural Networks [Zhang, 2023].

Subsplit Bayesian Networks



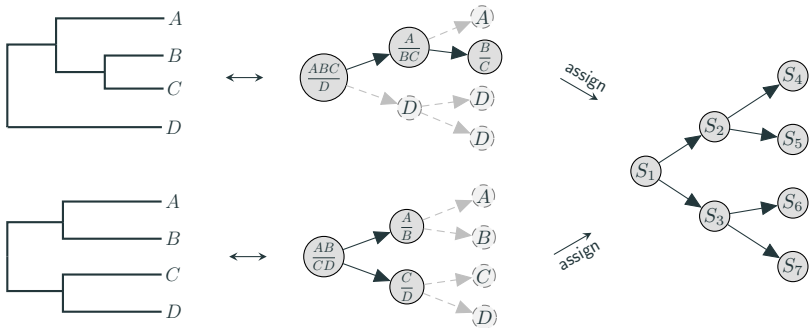
- The basic idea of subsplit Bayesian networks (SBNs) is to decompose the tree into local structures via **Bayesian networks** [Zhang and Matsen IV, 2018].

Subplit Bayesian Networks



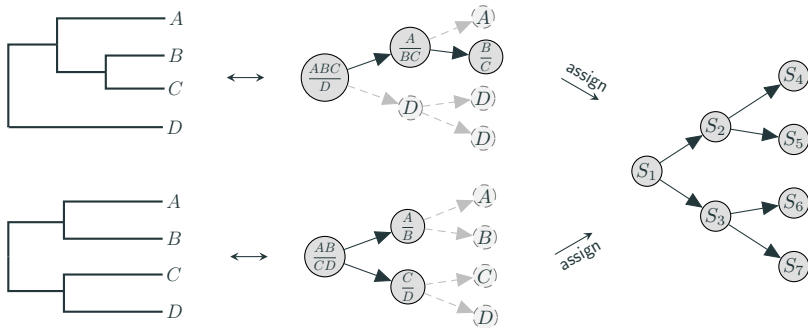
- The basic idea of subplit Bayesian networks (SBNs) is to decompose the tree into local structures via **Bayesian networks** [Zhang and Matsen IV, 2018].

Subsplit Bayesian Networks



- The basic idea of subsplit Bayesian networks (SBNs) is to decompose the tree into local structures via **Bayesian networks** [Zhang and Matsen IV, 2018].

Subsplit Bayesian Networks



- The basic idea of subsplit Bayesian networks (SBNs) is to decompose the tree into local structures via **Bayesian networks** [Zhang and Matsen IV, 2018].
- The SBN-based probability of this tree topology is

$$P_{\text{sbn}}(T = \tau) = P(S_1)P(S_2|S_1)P(S_3|S_1)P(S_4|S_2)P(S_5|S_2)P(S_6|S_7)P(S_7|S_3)$$

Subsplit Bayesian Networks

- SBNs parametrization: conditional probabilities of subsplit pairs.
- Practical compromise: **subsplit support estimation**, e.g. samples from an MCMC run.

Subsplit Bayesian Networks

- SBNs parametrization: conditional probabilities of subsplit pairs.
- Practical compromise: **subsplit support estimation**, e.g. samples from an MCMC run.
- Therefore, SBNs suffer from the following drawbacks [Zhang and Matsen IV, 2022]:
 - **Not standing alone**: SBNs rely on **hand-engineered heuristic features** which would require pre-sampled tree topologies.

Subsplit Bayesian Networks

- SBNs parametrization: conditional probabilities of subsplit pairs.
- Practical compromise: **subsplit support estimation**, e.g. samples from an MCMC run.
- Therefore, SBNs suffer from the following drawbacks [Zhang and Matsen IV, 2022]:
 - **Not standing alone**: SBNs rely on **hand-engineered heuristic features** which would require pre-sampled tree topologies.
 - **Hard support estimation**: The **subsplit support estimation** may become challenging when the phylogenetic posterior is diffuse.

Subsplit Bayesian Networks

- SBNs parametrization: conditional probabilities of subsplit pairs.
- Practical compromise: **subsplit support estimation**, e.g. samples from an MCMC run.
- Therefore, SBNs suffer from the following drawbacks [Zhang and Matsen IV, 2022]:
 - **Not standing alone**: SBNs rely on **hand-engineered heuristic features** which would require pre-sampled tree topologies.
 - **Hard support estimation**: The **subsplit support estimation** may become challenging when the phylogenetic posterior is diffuse.
 - **Confined support**: the **support** of SBNs cannot span the entire tree topology space.

- We propose **ARTree**, a deep autoregressive model for phylogenetic inference enjoys:
 - **Unconstrained support;**
 - **Discarding heuristic features.**

ARTree: Overview

- We propose **ARTree**, a deep autoregressive model for phylogenetic inference enjoys:
 - **Unconstrained support;**
 - **Discarding heuristic features.**
- Notations:
 - $\tau_n = (V_n, E_n)$: tree topology with n leaf nodes
 - V_n, E_n : nodes and edges of τ_n .
 - $\mathcal{X} = \{x_1, \dots, x_N\}$: a pre-selected order for the leaf nodes.

ARTree: Overview

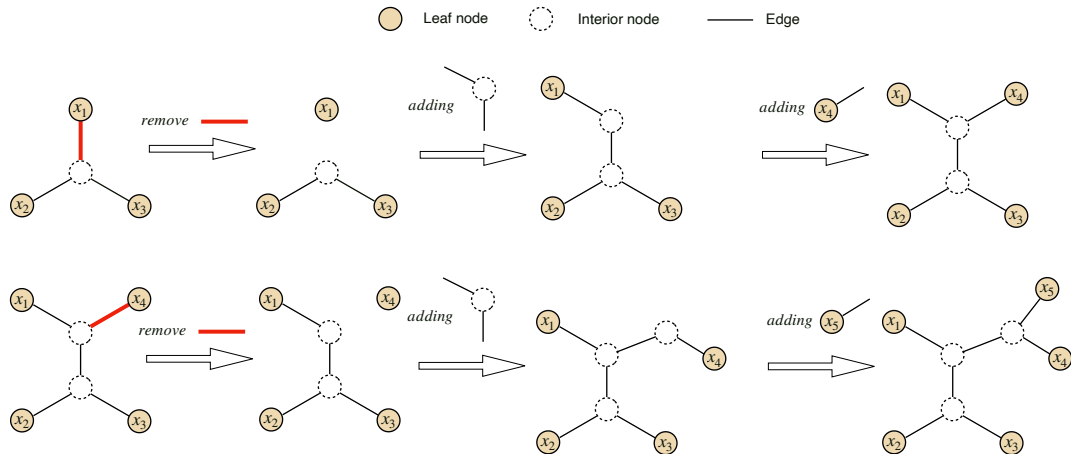
- We propose **ARTree**, a deep autoregressive model for phylogenetic inference enjoys:
 - **Unconstrained support;**
 - **Discarding heuristic features.**
- Notations:
 - $\tau_n = (V_n, E_n)$: tree topology with n leaf nodes
 - V_n, E_n : nodes and edges of τ_n .
 - $\mathcal{X} = \{x_1, \dots, x_N\}$: a pre-selected order for the leaf nodes.
- We consider unrooted tree topologies in ARTree. (But ARTree can be easily adapted to rooted tree topologies.)

ARTree: Overview

- We propose **ARTree**, a deep autoregressive model for phylogenetic inference enjoys:
 - **Unconstrained support;**
 - **Discarding heuristic features.**
- Notations:
 - $\tau_n = (V_n, E_n)$: tree topology with n leaf nodes
 - V_n, E_n : nodes and edges of τ_n .
 - $\mathcal{X} = \{x_1, \dots, x_N\}$: a pre-selected order for the leaf nodes.
- We consider unrooted tree topologies in ARTree. (But ARTree can be easily adapted to rooted tree topologies.)

ARTree: Sequential Generating Process

Construction of ordinal tree topology:



ARTree: Sequential Generating Process

- The sequential **generating process** for phylogenetic tree topologies. Starting from an ordinal tree topology $\tau_n = (V_n, E_n)$, we:
 1. select an edge $e_n = (u, v) \in E_n$ and remove it from E_n ;
 2. add a new node w and two new edges $(u, w), (w, v)$ to the tree topology;
 3. add the leaf node x_{n+1} and an edge (w, x_{n+1}) to the tree topology.Then we obtained an ordinal tree topology $\tau_{n+1} = (V_{n+1}, E_{n+1})$.

ARTree: Sequential Generating Process

- The sequential **generating process** for phylogenetic tree topologies. Starting from an ordinal tree topology $\tau_n = (V_n, E_n)$, we:
 1. select an edge $e_n = (u, v) \in E_n$ and remove it from E_n ;
 2. add a new node w and two new edges $(u, w), (w, v)$ to the tree topology;
 3. add the leaf node x_{n+1} and an edge (w, x_{n+1}) to the tree topology.Then we obtained an ordinal tree topology $\tau_{n+1} = (V_{n+1}, E_{n+1})$.
- During the generating process, the selected edges at each time step form a **decision sequence** $D = (e_3, \dots, e_{N-1})$.

ARTree: Sequential Generating Process

- The sequential **generating process** for phylogenetic tree topologies. Starting from an ordinal tree topology $\tau_n = (V_n, E_n)$, we:
 1. select an edge $e_n = (u, v) \in E_n$ and remove it from E_n ;
 2. add a new node w and two new edges $(u, w), (w, v)$ to the tree topology;
 3. add the leaf node x_{n+1} and an edge (w, x_{n+1}) to the tree topology.Then we obtained an ordinal tree topology $\tau_{n+1} = (V_{n+1}, E_{n+1})$.
- During the generating process, the selected edges at each time step form a **decision sequence** $D = (e_3, \dots, e_{N-1})$.

Theorem (Generating process is bijective)

Let $\mathcal{D} = \{D \mid D = (e_3, \dots, e_{N-1})\}$ be the set of all decision sequences and \mathcal{T} be the set of all ordinal tree topologies. Let the map $g : \mathcal{D} \rightarrow \mathcal{T}$ be the generating process described above. Then g is a bijection between \mathcal{D} and \mathcal{T} .

ARTree: Sequential Generating Process

- By this theorem, for each tree topology $\tau \in \mathcal{T}$, there is a unique decision sequence given by $g^{-1}(\tau)$. We call g^{-1} **decomposition process**.

ARTree: Sequential Generating Process

- By this theorem, for each tree topology $\tau \in \mathcal{T}$, there is a unique decision sequence given by $g^{-1}(\tau)$. We call g^{-1} **decomposition process**.

Lemma

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

ARTree: Sequential Generating Process

- By this theorem, for each tree topology $\tau \in \mathcal{T}$, there is a unique decision sequence given by $g^{-1}(\tau)$. We call g^{-1} **decomposition process**.

Lemma

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

- Decompose $Q(D)$ as the product of conditional distributions:

$$Q(\tau) = Q(D) = \prod_{n=3}^{N-1} Q(e_n | e_3, \dots, e_{n-1}). \quad (2)$$

- The $Q(e_n | e_{<n})$ defines the probability of adding the leaf node x_{n+1} to the edge e_n of τ_n , conditioned on all (τ_3, \dots, τ_n) generated so far.

ARTree: Sequential Generating Process

- By this theorem, for each tree topology $\tau \in \mathcal{T}$, there is a unique decision sequence given by $g^{-1}(\tau)$. We call g^{-1} **decomposition process**.

Lemma

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

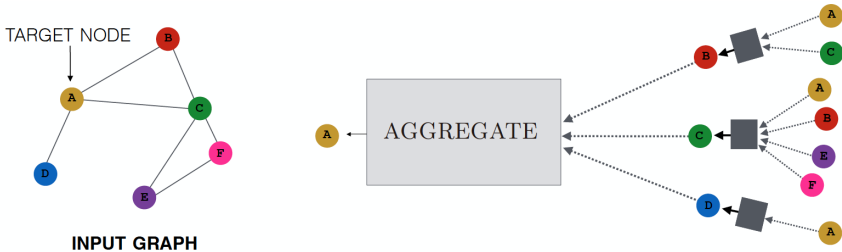
- Decompose $Q(D)$ as the product of conditional distributions:

$$Q(\tau) = Q(D) = \prod_{n=3}^{N-1} Q(e_n | e_3, \dots, e_{n-1}). \quad (2)$$

- The $Q(e_n | e_{<n})$ defines the probability of adding the leaf node x_{n+1} to the edge e_n of τ_n , conditioned on all (τ_3, \dots, τ_n) generated so far.
- We will show how to parametrize $Q(e_n | e_{<n})$ using graph neural networks (GNNs).

ARTree: Graph Neural Networks

Figure 5.1 in [Hamilton, 2020]



A general framework of graph neural networks (GNNs):

- (The first step) **Initialize** the node features.
- **Aggregate** the information from the neighborhood; **Update** the node features.
- (The last step) **Readout** the node features.

Topological node embeddings

- First find the **node embeddings** of $\tau_n = (V_n, E_n)$, which is a set $\{f_n(u) \in \mathbb{R}^N : u \in V_n\}$.

Topological node embeddings

- First find the **node embeddings** of $\tau_n = (V_n, E_n)$, which is a set $\{f_n(u) \in \mathbb{R}^N : u \in V_n\}$.
- For **leaf nodes**, one-hot encoding :

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

where δ is Kronecker delta function.

Topological node embeddings

- First find the **node embeddings** of $\tau_n = (V_n, E_n)$, which is a set $\{f_n(u) \in \mathbb{R}^N : u \in V_n\}$.
- For **leaf nodes**, one-hot encoding :

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

where δ is Kronecker delta function.

- For **interior nodes**, 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].

Message passing networks

- Initialized as topological node embeddings, the node features are updated with the information from their neighborhoods in a convolutional manner [Gilmer et al., 2017].

Message passing networks

- Initialized as topological node embeddings, the node features are updated with the information from their neighborhoods in a convolutional manner [Gilmer et al., 2017].
- l -th round message passing (L round in total):

$$m_n^l(u, v) = F_{\text{message}}^l(f_n^l(u), f_n^l(v)),$$
$$f_n^{l+1}(v) = F_{\text{updating}}^l\left(\{m_n^l(u, v); u \in \mathcal{N}(v)\}\right),$$

where $\mathcal{N}(v)$ is the neighborhood of the node v .

Message passing networks

- Initialized as topological node embeddings, the node features are updated with the information from their neighborhoods in a convolutional manner [Gilmer et al., 2017].
- l -th round message passing (L round in total):

$$m_n^l(u, v) = F_{\text{message}}^l(f_n^l(u), f_n^l(v)),$$
$$f_n^{l+1}(v) = F_{\text{updating}}^l\left(\{m_n^l(u, v); u \in \mathcal{N}(v)\}\right),$$

where $\mathcal{N}(v)$ is the neighborhood of the node v .

- In our implementations, the choices of F_{message}^l and F_{updating}^l follow the edge convolution operator [Wang et al., 2018].

Node hidden states

- The conditional distribution $Q(\cdot|e_{<n})$ has to capture the information from all the previous tree topologies.

Node hidden states

- The conditional distribution $Q(\cdot|e_{<n})$ has to capture the information from all the previous tree topologies.
- After obtaining the final node features of $\{f_n^L(v)\}$, a gated recurrent unit (GRU) [Cho et al., 2014] follows, i.e.

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

where $h_n(v)$ is the **hidden state** of v at the n -th generation step and is initialized to zero.

Time Guided Readout

- A main difference from other graph autoregressive models: **the topological node embedding $f_n^0(v)$ depends the time step n .**

Time Guided Readout

- A main difference from other graph autoregressive models: **the topological node embedding $f_n^0(v)$ depends the time step n .**
- Time guided readout step:

$$p_n(e) = F_{\text{pooling}}(h_n(u) + b_n, h_n(v) + b_n),$$

$$r_n(e) = F_{\text{readout}}(p_n(e) + b_n),$$

where b_n is the sinusoidal positional embedding [Vaswani et al., 2017] of time step n .

Time Guided Readout

- A main difference from other graph autoregressive models: **the topological node embedding $f_n^0(v)$ depends the time step n .**
- Time guided readout step:

$$p_n(e) = F_{\text{pooling}}(h_n(u) + b_n, h_n(v) + b_n),$$

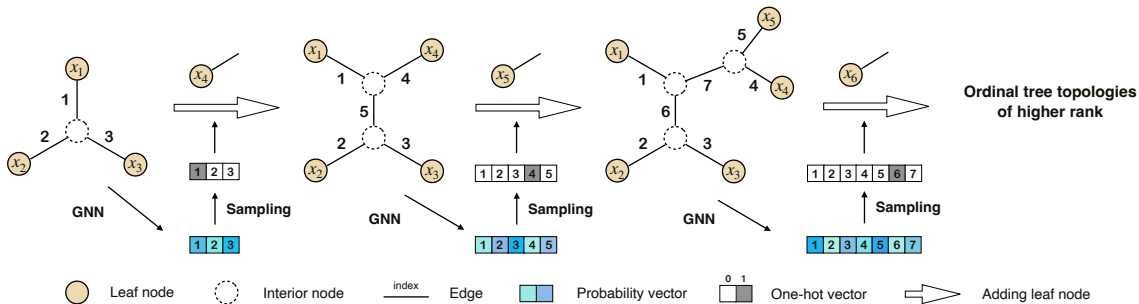
$$r_n(e) = F_{\text{readout}}(p_n(e) + b_n),$$

where b_n is the sinusoidal positional embedding [Vaswani et al., 2017] of time step n .

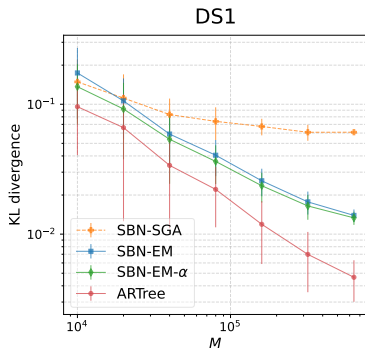
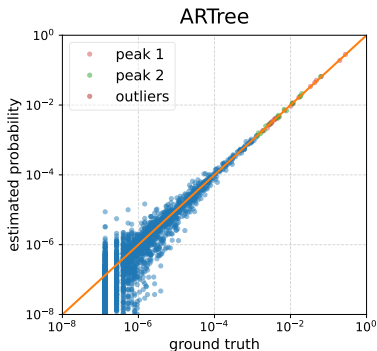
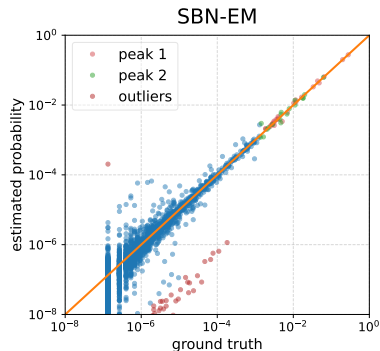
- Edge decision probability:

$$Q(\cdot | e_{<n}) \sim \text{Discrete}(q_n), \quad q_n = \text{softmax}(\{r_n(e)\}_{e \in E_n}),$$

ARTree: Overview



Experiments: Tree Topology Density Estimation



Given a training data set $\mathcal{M} = \{\tau_m\}_{m=1}^M$, we train ARTree via **maximum likelihood estimation**. In each iteration, the stochastic gradient is obtained by $\nabla_{\phi} L(\phi; \mathcal{M}) = \frac{1}{B} \sum_{b=1}^B \nabla_{\phi} \log Q_{\phi}(\tau_{m_b})$, where a minibatch $\{\tau_{m_b}\}_{b=1}^B$ is randomly sampled from \mathcal{M} .

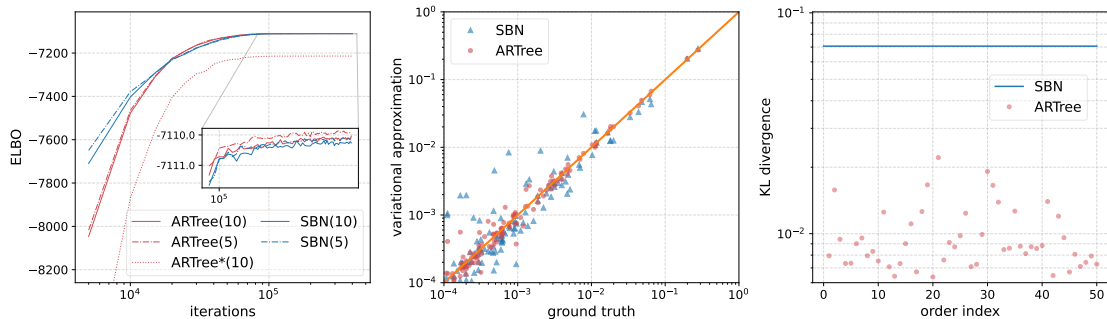
Experiments: Tree Topology Density Estimation

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.

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

Experiments: Variational Bayesian Phylogenetic Inference

For both ARTree and SBN, the collaborative branch lengths are parametrized using learnable topological features with GNNs [Zhang, 2023]. VBPI is done by maximizing the multi-sample lower bound with $K = 10$.



Results: VBPI on DS1. **Left:** The number of particles K are in the brackets. The ARTree* method refers to ARTree without time guidance ($b_n = 0$). **Right:** KL divergences across 50 random taxa orders.

Experiments: Variational Bayesian Phylogenetic Inference

Table 2: Results: VBPI on 8 benchmarks (KL, ELBO, 10-sample lower bound (LB-10), and marginal likelihood (ML)). GT trees row shows the number of unique tree topologies in the ground truth. The ML estimates are obtained via importance sampling using 1000 samples. The results of ϕ -CSMC are from [Koptagel et al., 2022].

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	ϕ -CSMC	-7290.36(7.23)	-30568.49(31.34)	-33798.06(6.62)	-13582.24(35.08)	-8367.51(8.87)	-7013.83(16.99)	N/A	-9209.18(18.03)
	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)



Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017).

Variational inference: A review for statisticians.

Journal of the American statistical Association, 112(518):859–877.



Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014).

Learning phrase representations using RNN encoder-decoder for statistical machine translation.




arXiv preprint arXiv:1406.1078.









Felsenstein, J. (2004).



Inferring Phylogenies.

Sinauer associates, 2 edition.

-  Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017).
Neural message passing for quantum chemistry.
ArXiv, abs/1704.01212.
-  Hamilton, W. L. (2020).
Graph representation learning.
Synthesis Lectures on Artificial Intelligence and Machine Learning, 14(3):1–159.
-  Koptagel, H., Kviman, O., Melin, H., Safinianaini, N., and Lagergren, J. (2022).
VaiPhy: a variational inference based algorithm for phylogeny.
In *Advances in Neural Information Processing Systems*.

-  Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017).
Attention is all you need.
In *Advances in Neural Information Processing Systems*, volume 30.
-  Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2018).
Dynamic graph CNN for learning on point clouds.
ACM Transactions on Graphics (TOG), 38:1 – 12.
-  Xie, T. and Zhang, C. (2023).
ARTree: A deep autoregressive model for phylogenetic inference.
In *Thirty-seventh Conference on Neural Information Processing Systems*.

-  Zhang, C. (2023).
Learnable topological features for phylogenetic inference via graph neural networks.
In International Conference on Learning Representations.
-  Zhang, C., Huelsenbeck, J. P., and Ronquist, F. (2020).
Using parsimony-guided tree proposals to accelerate convergence in Bayesian phylogenetic inference.
Systematic Biology, 69(5):1016–1032.
-  Zhang, C. and Matsen IV, F. A. (2018).
Generalizing tree probability estimation via Bayesian networks.
In Neural Information Processing Systems.

-  Zhang, C. and Matsen IV, F. A. (2019).
Variational Bayesian phylogenetic inference.
In International Conference on Learning Representations.
-  Zhang, C. and Matsen IV, F. A. (2022).
A variational approach to Bayesian phylogenetic inference.

Thank you!

Backup: Subsplit Bayesian Networks

- For a general **rooted** tree topology τ with subsplit assignments $\{s_i\}_{i \geq 1}$, its SBN-based probability is

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

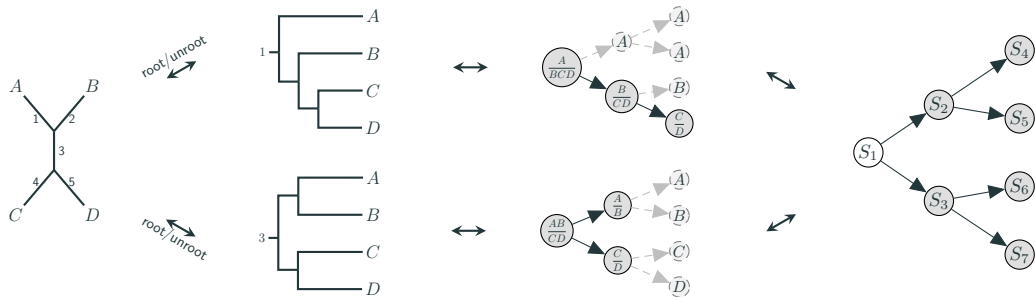
where π_i is the index set of the parents of node i .

- For an **unrooted** tree topology τ^u ,

$$P_{\text{sbn}}(T = \tau^u) = \sum_{\tau \in \mathcal{R}(\tau^u)} P_{\text{sbn}}(T = \tau),$$

where $\mathcal{R}(\tau^u)$ is the resulting tree topologies by adding a root to an edge of τ^u .

Backup: Subsplit Bayesian Networks



For an **unrooted** tree topology τ^u ,

$$P_{\text{sbn}}(T = \tau^u) = \sum_{\tau \in \mathcal{R}(\tau^u)} P_{\text{sbn}}(T = \tau),$$

where $\mathcal{R}(\tau^u)$ is the resulting tree topologies by adding a root to an edge of τ^u .