PLOS Computational Biology

RESEARCH ARTICLE

# ARTreeFormer: A faster attention-based autoregressive model for phylogenetic inference

**Tianyu Xie**[1], **Yicong Mao**[2], **Cheng Zhang**[1,3]*

**1** School of Mathematical Sciences, Peking University, Beijing, China, **2** School of Public Health, Peking University, Beijing, China, **3** Center for Statistical Science, Peking University, Beijing, China

* chengzhang@math.pku.edu.cn



OPEN ACCESS

**Citation:** Xie T, Mao Y, Zhang C (2025) ARTreeFormer: A faster attention-based autoregressive model for phylogenetic inference. PLoS Comput Biol 21(12): e1013768. https://doi.org/10.1371/journal.pcbi.1013768

**Editor:** Lun Hu, Xinjiang Technical Institute of Physics and Chemistry, CHINA

**Received:** June 25, 2025

**Accepted:** November 19, 2025

**Published:** December 4, 2025



**Data availability statement:** The sequence data for datasets DS1–DS8 are available at https://github.com/tyuxie/ARTreeFormer. For reproducing the TDE task, the short run data and the ground truth data can be found at https://www.doi.org/10.6084/m9.figshare.30272299. The codebase for

## Abstract

Probabilistic modeling over the combinatorially large space of tree topologies remains a central challenge in phylogenetic inference. Previous approaches often necessitate pre-sampled tree topologies, limiting their modeling capability to a subset of the entire tree space. A recent advancement is ARTree, a deep autoregressive model that offers unrestricted distributions for tree topologies. However, its reliance on repetitive tree traversals and inefficient local message passing for computing topological node representations may hamper the scalability to large datasets. This paper proposes ARTreeFormer, a novel approach that harnesses fixed-point iteration and attention mechanisms to accelerate ARTree. By introducing a fixed-point iteration algorithm for computing the topological node embeddings, ARTreeFormer allows for fast vectorized computation, especially on CUDA devices. This, together with an attention-based global message passing scheme, significantly improves the computation speed of ARTree while maintaining great approximation performance. We demonstrate the effectiveness and efficiency of our method on a benchmark of challenging real data phylogenetic inference problems.

## Author summary

Our research introduces novel methods for probabilistic modeling over phylogenetic tree topologies that are useful for various phylogenetic inference tasks such as tree probability estimation and variational Bayesian phylogenetic inference. Our model is based on ARTree, but achieves better scalability by leveraging a fixed-point algorithm for solving linear systems and employing an expressive attention-based architecture that captures long-range dependencies between edges. On benchmark phylogenetic inference datasets, including one with one hundred taxa, we demonstrate that the new model significantly reduces the computational cost of ARTree, while achieving similar or better performance in density estimation and variational approximation. This work represents an important step

reproducing the results of ARTreeFormer is available at https://github.com/tyuxie/ARTreeFormer.

**Funding:** This work was partially supported by National Natural Science Foundation of China (grant no. 12201014, grant no. 12292980 and grant no. 12292983, https://www.nsfc.gov.cn/), as well as National Institutes of Health grant AI162611 (https://www.nih.gov/), to CZ. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing interests:** The authors have declared that no competing interests exist.

toward scaling up variational inference for Bayesian phylogenetics. The techniques introduced here may also inspire future advances in scalable phylogenetic modeling.

## Introduction

Unraveling the evolutionary relationships among species stands as a core problem in the field of computational biology. This complex task, called *phylogenetic inference*, is abstracted as the statistical inference on the hypothesis of shared history, i.e., *phylogenetic trees*, based on collected molecular sequences (e.g., DNA, RNA) of the species of interest and a model of evolution. Phylogenetic inference finds its diverse applications ranging from genomic epidemiology [1–3] to the study of conservation genetics [4]. Classical approaches for phylogenetic inference include maximum likelihood [5], maximum parsimony [6], and Bayesian approaches [7–9], etc. Nevertheless, phylogenetic inference remains a hard challenge partially due to the combinatorially explosive size ($(2N-5)!!$ for unrooted bifurcating trees with $N$ species) of the phylogenetic tree topology space [10,11], which makes many common principles in phylogenetics, e.g., maximum likelihood and maximum parsimony, to be NP-hard problems [12,13].

Recently, the prosperous development of machine learning provides an effective and innovative approach to phylogenetic inference, and many efforts have been made for expressive probabilistic modeling of the tree topologies [14–17]. A notable example among them is ARTree [17], which provides a rich family of tree topology distributions and achieves state-of-the-art performance on benchmark data sets. Given a specific order on the leaf nodes (also called the taxa order), ARTree generates a tree topology by sequentially adding a new leaf node to an edge of the current subtree topology at a time, according to an edge decision distribution modeled by graph neural networks (GNNs), until all the leaf nodes have been added. Compared with previous methods such as conditional clade distribution (CCD) [15] and subsplit Bayesian networks (SBNs) [16], an important advantage of ARTree is that it enjoys unconfined support over the entire tree topology space. However, to compute the edge decision distribution in each leaf node addition step, ARTree requires sequential computations of topological node embeddings via tree traversals, which is hard to vectorize, making it prohibitive for phylogenetic inference for large numbers of species, as observed in [17]. Besides, the message passing in ARTree only updates node features from their neighborhood, ignoring the important global information and would require multiple message passing rounds to obtain adequate information about trees.

To address the computational inefficiencies of ARTree, we propose ARTreeFormer, which enables faster ancestral sampling and probability evaluation by leveraging scalable system-solving algorithms and transformer architectures [18]. More specifically, we replace the time-consuming tree traversal-based algorithm with a fixed-point iteration method for computing the topological node embeddings. We also prove

that, under a specific stopping criterion, the number of iterations required for convergence is independent of both the tree topology and the number of leaves. To further reduce the computational cost, we introduce an attention-based global message passing scheme that captures tree-wide information in a single forward pass. Unlike ARTree, all components of ARTreeFormer can be fully vectorized across multiple tree topologies and nodes, allowing efficient batch-wise generation and evaluation. This design makes ARTreeFormer particularly well suited for large-batch training on CUDA-enabled hardware, where the massive parallelism of modern GPUs can be fully exploited. Our experiments demonstrate that ARTreeFormer achieves comparable or better performance than ARTree, while delivering approximately 10× faster generation and 6× faster training on a benchmark suite covering maximum parsimony reconstruction, tree topology density estimation, and variational Bayesian phylogenetic inference tasks.

## Materials and methods

In this section, we first introduce the necessary background, including the phylogenetic posterior, variational Bayesian phylogenetic inference, and the ARTree model for tree topology generation. We then analyze the computational limitations of ARTree, which motivate the development of ARTreeFormer. Finally, we present the two key components of ARTreeFormer: a fixed-point iteration method for computing topological node embeddings and an attention-based global message passing mechanism.

### Phylogenetic posterior

The common structure for describing evolutionary history is a phylogenetic tree, which consists of a bifurcating tree topology $\tau$ and the associated non-negative branch lengths $\boldsymbol{q}$. The tree topology $\tau$, which contains leaf nodes for the observed species and internal nodes for the unobserved ancestor species, represents the evolutionary relationship among these species. A tree topology can be either rooted or unrooted. In this paper, we only discuss unrooted tree topologies, but the proposed method can be easily adapted to rooted tree topologies. The branch lengths $\boldsymbol{q}$ quantify the evolutionary intensity along the edges on $\tau$. An edge is called a pendant edge if it connects one leaf node to an internal node.

Each leaf node on $\tau$ corresponds to a species with an observed biological sequence (e.g., DNA, RNA, protein). Let $\boldsymbol{Y} = \{Y_1, \ldots, Y_M\} \in \Omega^{N\times M}$ be the observed sequences (with characters in $\Omega$) of $M$ sites over $N$ species. A continuous-time Markov chain is commonly assumed to model the transition probabilities of the characters along the edges of a phylogenetic tree [19]. Under the assumption that different sites evolve independently and identically conditioned on the phylogenetic tree, the likelihood of observing sequences $\boldsymbol{Y}$ given a phylogenetic tree $(\tau, \boldsymbol{q})$ takes the form

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

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), $E$ is the set of edges of $\tau$, $q_{uv}$ is the branch length of the edge $(u, v) \in E$, $P_{jk}(q)$ is the transition probability from character $j$ to $k$ through an edge of length $q$, and $\eta$ is the stationary distribution of the Markov chain. Assuming a prior distribution $p(\tau, \boldsymbol{q})$ on phylogenetic trees, Bayesian phylogenetic inference then amounts to properly estimating the posterior distribution

$$p(\tau, \boldsymbol{q}|\boldsymbol{Y}) = \frac{p(\boldsymbol{Y}|\tau, \boldsymbol{q})p(\tau, \boldsymbol{q})}{p(\boldsymbol{Y})} \propto p(\boldsymbol{Y}|\tau, \boldsymbol{q})p(\tau, \boldsymbol{q}), \tag{2}$$

where $p(\boldsymbol{Y}) = \int \sum_\tau p(\boldsymbol{Y}|\tau, \boldsymbol{q})p(\tau, \boldsymbol{q})\mathrm{d}\boldsymbol{q}$ is the unknown normalizing constant.

### Variational Bayesian phylogenetic inference

By positing a phylogenetic variational family $Q_{\phi,\psi}(\tau, \boldsymbol{q}) = Q_\phi(\tau)Q_\psi(\boldsymbol{q}|\tau)$ as the product of a tree topology model $Q_\phi(\tau)$ and a conditional branch length model $Q_\psi(\boldsymbol{q}|\tau)$, which is not a mean-field approximation as $Q_\psi(\boldsymbol{q}|\tau)$ depends on $\tau$, variational Bayesian phylogenetic inference (VBPI) converts the inference problem (2) into an optimization problem. More specifically, VBPI seeks the best variational approximation by maximizing the following multi-sample lower bound

$$L^K(\phi, \psi) = \mathbb{E}_{Q_{\phi,\psi}(\tau^{1:K}, \boldsymbol{q}^{1:K})} \log\left(\frac{1}{K}\sum_{i=1}^{K} \frac{p(\boldsymbol{Y}|\tau^i, \boldsymbol{q}^i)p(\tau^i, \boldsymbol{q}^i)}{Q_\phi(\tau^i)Q_\psi(\boldsymbol{q}^i|\tau^i)}\right), \tag{3}$$

where $Q_{\phi,\psi}(\tau^{1:K}, \boldsymbol{q}^{1:K}) = \prod_{i=1}^{K} Q_{\phi,\psi}(\tau^i, \boldsymbol{q}^i)$. In addition to the joint probablity $p(\boldsymbol{Y}, \tau, \boldsymbol{q})$ in the numerator of Eq (3), one may also consider the parsimony score defined as the minimum number of character-state changes among all possible sequence assignments for internal nodes, i.e.,

$$\mathcal{S}(\tau; \boldsymbol{Y}) = \sum_{i=1}^{M} \min_{a^i} \sum_{(u,v)\in E} \mathbb{I}(a^i_u \neq a^i_v), \tag{4}$$

where the notations are the same as in Eq (1) [20]. The parsimony score $\mathcal{S}(\tau; \boldsymbol{Y})$ can be efficiently evaluated by the Fitch algorithm [6] in linear time.

The tree topology model $Q_\phi(\tau)$ can take subsplit Bayesian networks (SBNs) [16] which rely on subsplit support estimation for parametrization, or ARTree [17] which is an autoregressive model using graph neural networks (GNNs) that provides distributions over the entire tree topology space. A diagonal lognormal distribution is commonly used for the branch length model $Q_\psi(\boldsymbol{q}|\tau)$ whose locations and scales are parameterized with heuristic features [21] or learnable topological features [22]. More advanced models for branch lengths like normalizing flows [23] or semi-implicit distributions [24] are also applicable. More details about VBPI can be found in Appendix B in S1 Text.

### ARTree for tree topology generation

As an autoregressive model for tree topology generation, ARTree [17] decomposes a tree topology into a sequence of leaf node addition decisions and models the involved conditional probabilities with GNNs. The corresponding tree topology generating process can be described as follows. Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be the set of leaf nodes with a pre-defined order. The generating procedure starts with a simple tree topology $\tau_3 = (V_3, E_3)$ that has the first three nodes $\{x_1, x_2, x_3\}$ as the leaf nodes (which is unique), and keeps adding new leaf nodes according to the following rule. Given an intermediate tree topology $\tau_n = (V_n, E_n)$ that has the first $n < N$ elements in $\mathcal{X}$ as the leaf nodes, i.e., an *ordinal tree topology* of rank $n$ as defined in [17], a probability vector $q_n \in \mathbb{R}^{|E_n|}$ over the edge set $E_n$ is first computed via GNNs. Then, an edge $e_n \in E_n$ is sampled according to $q_n$ and the next leaf node $x_{n+1}$ is attached to it to form an ordinal tree topology $\tau_{n+1}$. This procedure will continue until all the $N$ leaf nodes are added. Although a pre-defined leaf node order is required, [17] shows that the performance of ARTree exhibits negligible dependency on this leaf node order. Fig 1 is an illustration of ARTree. See more details on ARTree in Appendix A in S1 Text.

Although ARTree enjoys unconfined support over the entire tree topology space and provides a more flexible family of variational distributions, it suffers from expensive computation costs (see Appendix E in [17]) which makes it prohibitive for phylogenetic inference when the number of species is large. In the next two subsections, we discuss the computational cost of ARTree and then describe how it can be accelerated using fixed-point iteration and attention-based techniques.

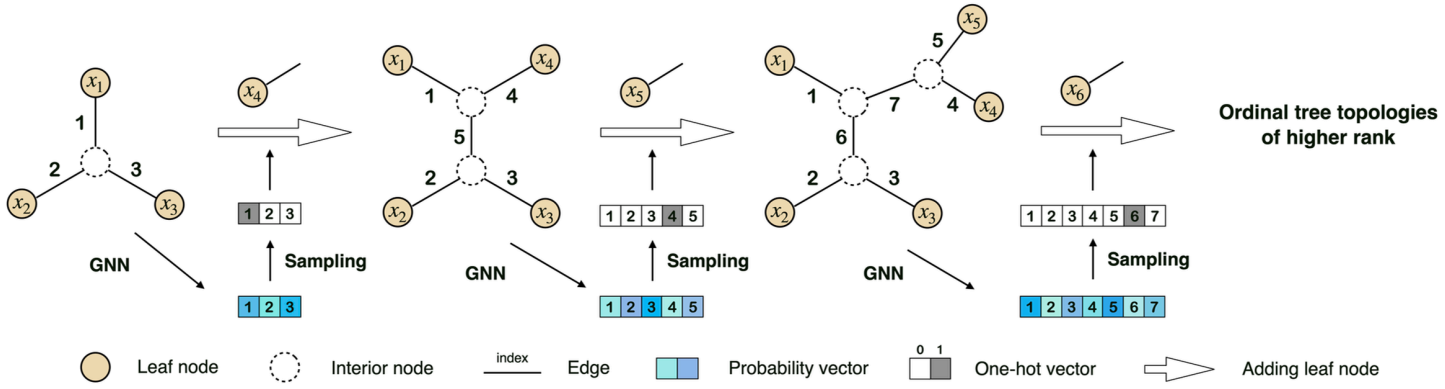


**Fig 1. An illustration of ARTree starting from the star-shaped tree topology with 3 leaf nodes.** This figure is from [17].

https://doi.org/10.1371/journal.pcbi.1013768.g001

### Computational cost of ARTree

In the $n$-th step of leaf node addition, ARTree includes the node embedding module and message passing module for computing the edge decision distribution, as detailed below. Throughout this section, we use "node embeddings" (with dimension $N$) for the node information before message passing and "node features" (with dimension $d$) for those in and after message passing.

**Node embedding module.** The topological node embeddings $\{f_n(u) \in \mathbb{R}^N | u \in V_n\}$ of an ordinal tree topology $\tau_n = (V_n, E_n)$ in [17] are obtained by (i) first assigning one-hot encodings to the leaf nodes, i.e., letting $f_n(x_i) = e_i$, where $e_i$ is a length-$N$ one-hot vector with the only 1 on the $i$-th position; and (ii) minimizing the *Dirichlet energy*

$$\mathscr{E}(f_n, \tau_n) := \sum_{(u,v)\in E_n} \|f_n(u) - f_n(v)\|^2, \tag{5}$$

w.r.t. $\{f_n(u); u \text{ is internal node}\}$, which is typically done by the two-pass algorithm [22] (Algorithm 3 in Appendix A in S1 Text). This algorithm requires a traversal over a tree topology, which is hard to be efficiently vectorized across different nodes and different trees due to its sequential nature and the dependency on the specific tree topology shapes. The complexity of computing the topological node embeddings is $O(Nn)$. Finally, a multi-layer perceptron (MLP) is applied to all the node embeddings to obtain the node features with dimension $d$ enrolled in the computation of the following modules.

**Message passing module.** Assume that the initial node features are $\{f_n^0(u) \in \mathbb{R}^d | u \in V_n\}$, which are transformed from $\{f_n(u) \in \mathbb{R}^N | u \in V_n\}$ using MLPs with complexity $O(nk(N+d))$ where $k$ is the intermediate dimension. In the $l$-th round, these node features are updated by aggregating the information from their neighborhoods through

$$m_n^l(u,v) = F^l_{\text{message}}(f_n^l(u), f_n^l(v)), \tag{6a}$$

$$f_n^{l+1}(v) = F^l_{\text{updating}}\left(\{m_n^l(u,v); u \in \mathcal{N}(v)\}\right), \tag{6b}$$

where the $l$-th message function $F^l_{\text{message}}$ operates by applying an MLP to the concatenated inputs, while the updating function $F^l_{\text{updating}}$ first applies the same MLP to the inputs and then pools the results in a permutation-invariant manner (e.g., sum, mean, or max). After $L$ rounds of message passing, a recurrent neural network implemented by a gated recurrent unit (GRU) [25] is then applied to gather the information from all previously generated tree topologies, i.e.,

$$h_n(v) = \text{GRU}(h_{n-1}(v), f_n^L(v)), \tag{7}$$

where $h_n(v)$ is the hidden state of $v$. Eqs (6) and (7) are applied to the features of all the nodes on $\tau_n$ which require $O(Lnd^2)$ operations and are computationally inefficient especially when the number of leaf nodes is large. Moreover, Eq (6) only updates the features of a node from its neighborhood, ignoring the global information of the full tree topology, and thus is called *local message passing* by us. We summarize the computational complexity of ARTree in Proposition 1 (see Appendix C in S1 Text for proof).

**Proposition 1** (Time complexity of ARTree)**.** *Let L be the number of message passing rounds and B be the number of tree topologies in a batch. For generating B tree topologies with N leaf nodes, the time complexity of ARTree is* $O(BN^3 + BLN^2d^2 + BN^2k(N+d))$. *In the ideal case of vectorization, if we assume perfect linear speedup [26] and sufficiently many threads, the complexity of ARTree is* $O(N^2 + LN)$.

**Remark 1.** *To understand the ideal case of constant time complexity, we need two assumptions, i.e., perfect linear speedup and sufficiently many threads (e.g., on modern GPUs). Let the computation time on one thread be* $T_1$ *and the computation time on p threads be* $T_p$. *The perfect linear speedup (Page 780 in [26]) states that* $T_1/T_p = p$. *So if p exceeds the number of operations in a vectorized computation, the time complexity will become O(1). This ideal constant time complexity is introduced to clearly distinguish the components that cannot be vectorized, and we acknowledge that the constant time complexity generally cannot be attained in practice.*

Fig 2 (left) demonstrates the run time of ARTree as the number of leaf nodes $N$ varies. As $N$ increases, the total run time of ARTree grows rapidly and the node embedding module dominates the total time ($\approx 95\%$ on CUDA and $\approx 60\%$ on CPU), which makes ARTree prohibitive when the number of leaf nodes is large. The reason behind this is that compared to other modules, the node embedding module can not be easily vectorized w.r.t. different tree topologies and different nodes, resulting in great computational inefficiency. It is worth noting that the computation time of the node embedding module on CUDA is even larger than that on CPU, which can be attributed to the inefficiency of CUDA for handling small tensors.

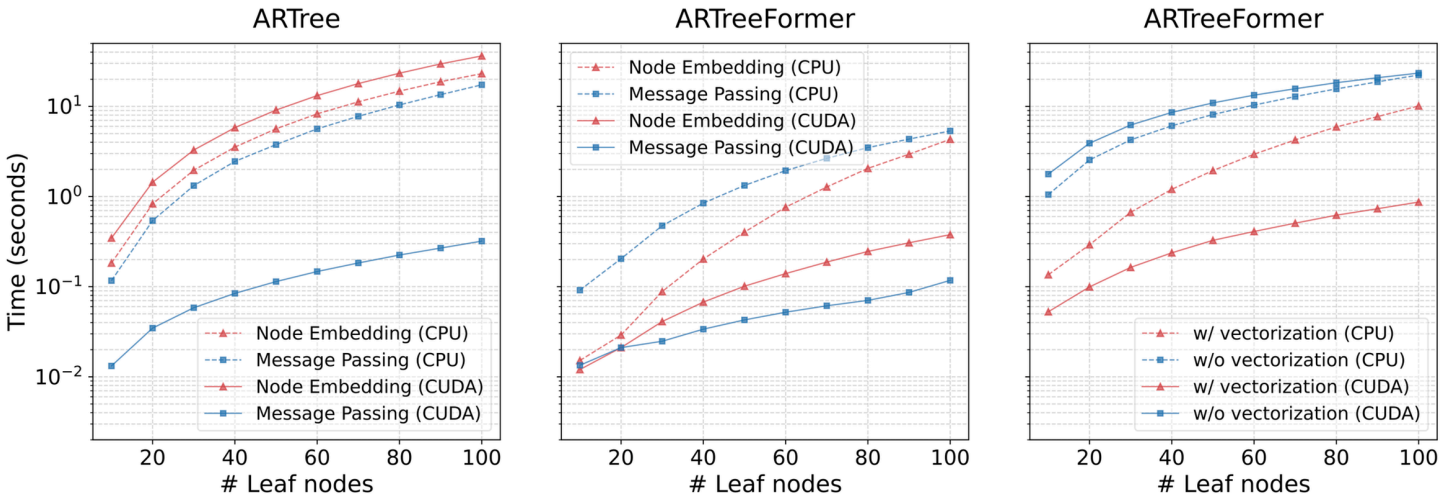


**Fig 2. Time comparison between different models and devices.** Left & Middle: Runtime of the node embedding module and message passing module for generating 128 tree topologies in a single batch using ARTree and ARTreeFormer. Right: The runtime of ARTreeFormer for generating 128 tree topologies with or without vectorization across batched tree topologies. CPU means running on a cluster of 16 2.4 GHz CPUs, and CUDA means running on a single NVIDIA A100 GPU. All these results are averaged over 10 independent trials.

https://doi.org/10.1371/journal.pcbi.1013768.g002

### Accelerated computation of edge decision distributions

In this subsection, we propose ARTreeFormer, which introduces a fast fixed-point iteration algorithm for topological node embeddings and an attention-based global message passing scheme to accelerate the training and sampling in ARTree. In what follows, we present our approach for modeling the edge decision distribution at the $n$-th step.

**Fixed-point iteration for topological node embedding.** Instead of solving the minimization problem of $\ell(f_n, \tau_n)$ in Eq (5) with the time-consuming two-pass algorithm, we reformulate it as a fixed-point iteration algorithm. For a tree topology $\tau_n = (V_n, E_n)$, denote the set of leaf nodes by $\mathcal{X}_n$, the set of internal nodes by $V_n^o$, and the set of internal edges by $E_n^o = \{(u,v)|u,v \in V_n^o\}$. Note that the global minimum of $\ell(f_n, \tau_n)$ satisfies

$$\begin{cases} f_n(u) = \frac{1}{3}\sum_{v\in\mathcal{N}(u)} f_n(v), & u \in V_n^o; \\ f_n(x_i) = \delta_i, & x_i \in \mathcal{X}_n; \end{cases} \tag{8}$$

where $\mathcal{N}(u)$ is the set of neighbors of $u$ and $\delta_i$ is a one-hot vector of length $n$ with the 1 at the $i$-th position. Let $\bar{\mathcal{F}}_n = \{f_n(u) \in \mathbb{R}^n | u \in V_n\} \in \mathbb{R}^{(2n-2)\times n}$ and $\mathcal{F}_n = \{f_n(u) \in \mathbb{R}^n | u \in V_n^o\} \in \mathbb{R}^{(n-2)\times n}$, then $\bar{\mathcal{F}}_n = (I_n, \mathcal{F}_n')'$. Consider a matrix $\bar{A}_n$ satisfying

$$\bar{A}_n = \begin{pmatrix} I_n & 0_{n-2} \\ C_n/3 & A_n/3 \end{pmatrix}, \tag{9}$$

where $C_n(i,j) = \mathbb{I}_{(u_i,x_j)\in E_n}$ and $A_n(i,j) = \mathbb{I}_{(u_i,u_j)\in E_n}$ ($u_i$ denotes the $i$-th node, leaf nodes are indexed as the first $n$ nodes). Note that $A_n$ is exactly the adjacency matrix of $(V_n^o, E_n^o) := \tau_n^o$. We call $A_n$ the *interior adjacency matrix* and $C_n$ the *leaf-interior cross adjacency matrix* of $\tau_n$. The system (8) is then equivalent to $\bar{A}_n\bar{F}_n = \bar{F}_n$, i.e.,

$$\mathcal{F}_n = \frac{A_n}{3}\mathcal{F}_n + \frac{C_n}{3}. \tag{10}$$

This inspires the following fixed-point iteration algorithm:

$$\mathcal{F}_n^{(m+1)} = \frac{A_n}{3}\mathcal{F}_n^{(m)} + \frac{C_n}{3};\ \mathcal{F}_n^{(0)} = \mathcal{F}_n^{(0)}. \tag{11}$$

In practice, we set all the entries to $\mathcal{F}_n^{(0)}$ as $1/n$. Finally, after obtaining the solution $\mathcal{F}_n^*$, we pad $N$–$n$ zeros on its right so that the resulting length-$N$ node embeddings can be fed into the message passing module. Theorem 2 and Corollary 1 prove that the fixed-point iteration (11) will converge to the unique solution of Eq (10) with a uniform speed for all tree topologies $\tau_n$, the number of leaves $n$, and the initial condition $\mathcal{F}_n^{(0)}$.

**Theorem 2.** *For a tree topology $\tau_n$ with n leaf nodes, let $\tau_n^o$ be the subgraph of $\tau_n$ which only contains the internal nodes and the edges among them and $A_n$ be the interior adjacency matrix of $\tau_n$. Let $\rho(\tau_n^o)$ be the spectral radius of $\tau_n^o$ defined as $\rho(\tau_n^o) = \lambda_{\max}(A_n)$, where $\lambda_{\max}(\cdot)$ denotes the largest absolute eigenvalue of a matrix. Then for any $\tau_n$ and n, it holds*

$$\rho(\tau_n^o) \le 2\sqrt{2}.$$

*Proof*: Without loss of generality, we select a node $c$ in $\tau_n^o$ as the "root node" and denote the distance between a node $u$ and $c$ by $d_u$, which induces a hierarchical structure on $\tau_n^o$. Consider a matrix $D = \mathrm{diag}\{2^{d_u/2}, u \in V^o\}$ and it holds that $DA_nD^{-1}$ and $A_n$ share the same eigenvalues. Note that each row of $A$ and $DA_nD^{-1}$ has up to 3 non-zero entries. Now for each row of $DA_nD^{-1}$ and the corresponding node $u$, we make the following analysis.

- If $u = c$, then each non-zero entry in this row equals to $1/\sqrt{2}$.
- If $u$ is a leaf node, then there is only one non-zero entry $\sqrt{2}$ in this row.
- For the remaining cases, as $u$ have one parent node and at most two child nodes, there is a $1\sqrt{2}$ and at most two $1/\sqrt{2}$ entries in this row.

For all these cases, the row sum is less than or equal to $2\sqrt{2}$ which consistently holds for arbitrary topological structures of $\tau_n^o$ and the number of nodes $n$. By the Perron–Frobenius theorem for positive matrices, $\lambda_{\max}(DA_nD^{-1})$ is upper bounded by the largest row sum of $DA_nD^{-1}$. Therefore, we conclude that $\rho(\tau_n^o) \leq 2\sqrt{2}$. This proof is inspired by [27, Section 4.2]. □

**Corollary 1.** *The fixed-point iteration algorithm (11) will converge linearly with rate* $\frac{2\sqrt{2}}{3}$.

*Proof*: Let $\mathcal{F}_n^*$ be the solution to $\mathcal{F}_n^* = A_n\mathcal{F}_n^*/3 + C_n/3$. The existence and uniqueness of $\mathcal{F}_n^*$ are guaranteed by the fact that $I - A_n/3$ is a full-rank matrix. Subtracting $\mathcal{F}_n^*$ from both sides leads to

$$\mathcal{F}_n^{(m+1)} - \mathcal{F}_n^* = (A_n/3)(\mathcal{F}_n^{(m)} - \mathcal{F}_n^*)$$

and thus

$$\|\mathcal{F}_n^{(m+1)} - \mathcal{F}_n^*\|_2 \leq (\|A\|_2/3)\|\mathcal{F}_n^{(m)} - \mathcal{F}_n^*\|_2$$

By Theorem 2, we conclude that $\|\mathcal{F}_n^{(m)} - \mathcal{F}_n^*\|_2 \leq \left(\frac{2\sqrt{2}}{3}\right)^m \|\mathcal{F}_n^{(0)} - \mathcal{F}_n^*\|_2$. □

Unlike the two-pass algorithm for Dirichlet energy minimization, the fixed-point iteration can be easily vectorized over different tree topologies and nodes, making it suitable for fast computation on CUDA. By using $\|\mathcal{F}_n^{(m)} - \mathcal{F}_n^*\|_2/n < \varepsilon$ as the stopping criterion, the required number of iterations $M_\varepsilon$ is a constant independent of the tree topologies. Moreover, by noting that the fixed-point iteration (11) is equivalent to $\bar{\mathcal{F}}_n^{(2^{m+1})} = \bar{A}_n^{2^m}\bar{\mathcal{F}}_n^{(2^m)}$ which repetitively updates $\bar{A}_n^{2^{m+1}} = \left(\bar{A}_n^{2^m}\right)^2$, the number of iterations can be further reduced to $\log_2 M_\varepsilon$ and we call this strategy *the power trick*. With the power trick, the computational complexity of fixed-point iteration over $B$ tree topologies can be reduced to $O(Bn^2 \log_2 M_\varepsilon)$.

**Remark 2.** *For the complexity estimation $O(Bn^2 \log_2 M_\varepsilon)$, the computation over the dimension B and n can be efficiently vectorized, while the computation over* $\log M_\varepsilon$ *is still sequential.*

**Remark 3.** *After adding a new leaf node to $\tau_n$, a local modification can be applied to $A_n$ and $C_n$ to form the $A_{n+1}$ and $C_{n+1}$. Therefore, the time complexity of computing the adjacency matrices of a tree topology is O(1). Algorithm 1 shows the full procedure of the fixed-point iteration when autoregressively building the tree topology.*

**Attention-based global message passing.** After obtaining the topological node embeddings $\mathcal{F}_n^*$ with the fixed-point iteration algorithm, it is fed into a message passing module to form the distribution over edge decisions. Similarly to ARTree, at the start of the module, the dimensionality translation from $N$ to $d$ is performed using MLPs with complexity $O(nk(N+d))$, where $k$ represents the intermediate dimension. To design an edge distribution that captures the global information of the tree topology, we substitute the GNNs with the powerful attention mechanism [18]. Specifically, we first use the attention mechanism to compute a graph representation vector $r_n \in \mathbb{R}^d$, i.e.,

$$\bar{r}_n = F_{\text{graph}}(q_n, L(\mathcal{F}_n^*), L(\mathcal{F}_n^*)), \tag{12a}$$

$$r_n = R_{\text{graph}}(\bar{r}_n), \tag{12b}$$

**Algorithm 1. A fixed-point algorithm for topological node embeddings.**

**Input:** A decision sequence $D=(e_3,\dots,e_{N-1})$ corresponding to $\tau$. A threshold value $\varepsilon$.
**Output:** The topological node embeddings of each subtree $\tau_n$
Initialize the adjacency matrices $A_3$ and $C_3$;
**for** $n=3,\dots,N-1$ **do**
  Compute $\bar{A}_n$ from $A_n$ and $C_n$ using Eq (9);
  Initialize $\mathcal{F}_n^{(1)}=(I_n,1_n/n)$ and $m=1$;
  Compute $\mathcal{F}_n^{(2)}=\bar{A}\mathcal{F}_n^{(1)}$ and $\bar{A}^2=\bar{A}\cdot\bar{A}$;
  **while** $\|\mathcal{F}_n^{(2^m)}-\mathcal{F}_n^{(2^{m-1})}\|_2\geq\varepsilon$ **do**
    Compute $\mathcal{F}_n^{(2^{m+1})}=\bar{A}_n^{2^m}\mathcal{F}_n^{(2^m)}$;
    Compute $\bar{A}_n^{2^{m+1}}=\bar{A}_n^{2^m}\cdot\bar{A}_n^{2^m}$; $m=m+1$;
  **end**
  Pad $N-n$ zeros on the right of each row of $\mathcal{F}_n^{(2^m)}$;
  Add the new leaf node $x_{n+1}$ to the edge $e_n$;
  Locally modify the adjacency matrices $A_n$ and $C_n$ to obtain $A_{n+1}$ and $C_{n+1}$.
**end**

where $F_{\text{graph}}$ is the graph pooling function implemented as a multi-head attention block [18], $R_{\text{graph}}$ is the graph readout function implemented as a 2-layer MLP, $q_n\in\mathbb{R}^d$ is a learnable query vector, and $L:\mathbb{R}^N\to\mathbb{R}^d$ is an embedding map implemented as a 2-layer MLP. Here, the multi-head attention block $M=\text{MHA}(Q,K,V)$ is defined as

$$H_i=\text{softmax}\left(\frac{(QW_i^Q)(KW_i^K)'}{\sqrt{d/h}}\right)\cdot(VW_i^V), \tag{13a}$$

$$M=\text{CONCAT}\,(H_1,\dots,H_h)\,W^O, \tag{13b}$$

where $W_i^Q,W_i^K,W_i^V\in\mathbb{R}^{d\times\frac{d}{h}}$ and $W^O\in\mathbb{R}^{d\times d}$ are learnable matrices, $h$ is the number of heads, and CONCAT is the concatenation operator along the node feature axis. Intuitively, we have used a global vector $q_n$ to query all the node features and obtained a representation vector $r_n$ for the whole tree topology $\tau_n$. We emphasize that Eq (12) enjoys time complexity $O(nd+d^2)$ instead of the $O(n^2d+nd^2)$ of common multi-head attention blocks, as $q_n$ is a one-dimensional vector.

We now compute the edge decision distribution to decide where to add the next leaf node, similarly to ARTree. To incorporate global information into the edge decision, we utilize the global representation vector $r_n$ to compute the edge features. Concretely, the feature of an edge $e=(u,v)$ is formed by

$$p_n(e)=F_{\text{edge}}\left(\{f_n(u),f_n(v)\}\right), \tag{14a}$$

$$r_n(e)=R_{\text{edge}}\left(\text{CONCAT}(p_n(e),r_n)+b_n\right), \tag{14b}$$

where $F_{\text{edge}}$ is an invariant edge pooling function implemented as an elementwise maximum operator, $R_{\text{edge}}$ is the edge readout function implemented as a 2-layer MLP with scalar output, and $b_n$ is the sinusoidal positional embedding [18] of the time step $n$. The time complexity of these MLPs in Eq (14) is $O(nd^2)$.

**Edge decision distribution.** Similarly to ARTree, we build the edge decision distributions in ARTreeFormer in an autoregressive way. That is, we directly read out the representation vector $r_n$ to calculate the edge decision distribution $Q_\phi(\cdot|e_{<n})$ using

$$Q_\phi(\cdot|e_{<n})=\text{Discrete}(\alpha_n),\ \alpha_n=\text{softmax}\left([r_n(e)]_{e\in E_n}\right), \tag{15}$$

and grow $\tau_n$ to $\tau_{n+1}$ by attaching the next leaf node $x_{n+1}$ to the sampled edge (Algorithm 2).

The above node embedding module and message passing module circularly continue until an ordinal tree topology of $N$, $\tau_N$, is constructed, whose ARTreeFormer-based probability is defined as

$$Q_\phi(\tau_N) = \prod_{n=3}^{N-1} Q_\phi(e_n|e_{<n}), \tag{16}$$

where $\phi$ are the learnable parameters and $Q_\phi(e_n|e_{<n})$ is defined in Eq (15). We summarize the time complexity of ARTreeFormer in Proposition 3 (see Appendix C in S1 Text for proof).

**Proposition 3** (Time complexity of ARTreeFormer)**.** *Let B be the number of tree topologies in a batch. For generating B tree topologies with N leaf nodes, the time complexity of ARTreeFormer is* $O(BN^3 \log M_\varepsilon + BN^2d^2 + BN^2k(N+d))$. *In the ideal case of vectorization, if we assume perfect linear speedup [26] and sufficiently many threads, the time complexity of ARTreeFormer is* $O(N(\log M_\varepsilon + 1))$, *where* $\log M_\varepsilon$ *is a constant independent of N.*

Compared to ARTree, the greatly improved computational efficiency of ARTreeFormer mainly comes from two aspects. **First**, the fixed-point iteration algorithm in ARTreeFormer for topological node embeddings can be easily vectorized across different tree topologies and different nodes, since they do not rely on traversals over tree topologies. **Second**, the global message passing in ARTreeFormer forms the global representation only in one pass through the attention mechanism instead of gathering the neighborhood information repetitively with GNNs. We depict the the pipeline of the leaf node addition of ARTreeFormer in Fig 3.

In Fig 2 (left, middle), for the node embedding module on CPU/CUDA, the time consumption of ARTreeFormer is less than 10% of ARTree, and this number is 50% for the message passing module on CPU/CUDA. Moreover, both two modules of ARTreeFormer enjoy a significant time consumption drop on CUDA compared to CPU, since CUDA is more powerful at handling large tensor multiplications. To further verify the vectorization capability of ARTreeFormer, we compare the runtime for generating tree topologies with or without vectorization in Fig 2 (right). Here, the "w/o vectorization" setting performs fixed-point iteration and attention-based message passing sequentially for each tree topology, one at a time. In contrast, the "w/ vectorization" setting applies fixed-point iteration and attention-based message passing simultaneously for a batch of tree topologies, leveraging batched tensor operations for more efficient computation. We see that vectorization greatly improves computational efficiency. The vectorization capability of ARTreeFormer further allows for

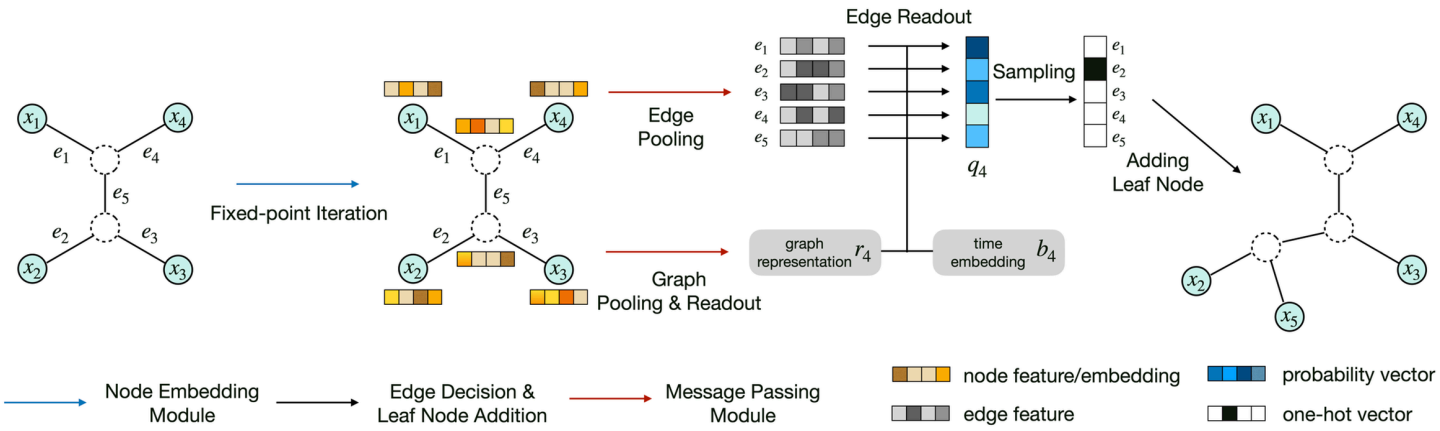


**Fig 3. An illustration of ARTreeFormer for growing an ordinal tree topology $\tau_4$ of rank 4 to an ordinal tree topology $\tau_5$ of rank 5.**

https://doi.org/10.1371/journal.pcbi.1013768.g003

training with a larger batch size (note the batch size is 10 in ARTree), which is a common setting in modern deep learning methods.

## Results

In this section, we demonstrate the effectiveness and efficiency of ARTreeFormer on three benchmark tasks: maximum parsimony, tree topology density estimation (TDE), and variational Bayesian phylogenetic inference (VBPI). Although the pre-selected leaf node order in ARTreeFormer may not be related to the relationships among species, this evolutionary information is already contained in the training data set (for TDE) or the target posterior distribution (for maximum parsimony and VBPI), and thus can be learned by ARTreeFormer. Noting that the main contribution of ARTreeFormer is improving the tree topology model, we select the first two tasks because they only learn the tree topology distribution and can better demonstrate the superiority of ARTreeFormer. The third task, VBPI, is selected as a standard benchmark task for Bayesian phylogenetic inference and evaluates how well ARTreeFormer collaborates with a branch length model. It should be emphasized that we mainly pay attention to the computational efficiency improvement of ARTreeFormer and only expect it to attain similar accuracy to ARTree. Throughout this section, the run times of ARTree are reproduced using its official codebase: https://github.com/tyuxie/ARTree.

**Experimental setup.** For TDE and VBPI, 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 [10,14,15,28–35]. For the Bayesian setting in MrBayes runs [36], we assume a uniform prior on the tree topologies, an i.i.d. exponential prior Exp(10) on branch lengths, and the simple Jukes & Cantor (JC) substitution model [37]. We use the same ARTreeFormer structure across all the data sets for all three experiments. Specifically, we set the dimension of node features to $d = 100$, following [17]. The number of heads in all the multi-head attention blocks is set to $h = 4$. All the activation functions for MLPs are exponential linear units (ELUs) [38]. We add a layer normalization block after each linear layer in MLPs and before each multi-head attention block, which stabilizes training and reduces its sensitivity to optimization tricks [39]. We also add a residual block after the multi-head attention block in the message passing step, which is standard in transformers. For all experiments and data sets, the stopping criterion of the fixed-point iteration algorithm in ARTreeFormer is $\varepsilon = 10^{-5}$. The taxa order is set to the lexicographical order of the corresponding species names. All models are implemented in PyTorch [40] and optimized with the Adam [41] optimizer. All the experiments are run and all the runtimes are measured on a single CUDA-enabled NVIDIA A100 GPU. The learning rate for ARTreeFormer is set to 0.0001 in all the experiments, which is the same as in ARTree [17].

### Maximum parsimony problem

We first test the performance of ARTreeFormer on a variational inference task purely on the tree topologies, whose target distribution is defined as $P(\tau) = \exp(-\mathcal{S}(\tau, \boldsymbol{Y}))/Z$, where $\mathcal{S}(\tau, \boldsymbol{Y})$ is the parsimony score defined in Eq (4) and $Z = \sum_\tau \exp(-\mathcal{S}(\tau, \boldsymbol{Y}))$ is the normalizing constant. To fit a variational distribution $Q_\phi(\tau)$, we maximize the following (annealed) multi-sample lower bound ($K = 10$) in the $t$-th iteration

$$\mathcal{L}(\phi; \beta_t) = \mathbb{E}_{Q_\phi(\tau^{1:K})} \log \left( \frac{1}{K} \sum_{i=1}^{K} \frac{\exp\left(-\beta_t \mathcal{S}(\tau_i, \boldsymbol{Y})\right)}{Q_\phi(\tau_i)} \right), \tag{17}$$

where $Q_\phi(\tau^{1:K}) = \prod_{i=1}^{K} Q_\phi(\tau^i)$ and $\beta_t$ is the annealing schedule. We set $\beta_t = \min\{1, 0.001 + t/200000\}$ and collect the results after 400000 parameter updates. We use the VIMCO estimator [42] to estimate the stochastic gradients of $\mathcal{L}(\phi)$.

Fig 4 shows the performances of different methods for the maximum parsimony problem on DS1. We run the state-of-the-art parsimony analysis software PAUP* [43] to form a collection of tree topologies with low parsimony scores ranging

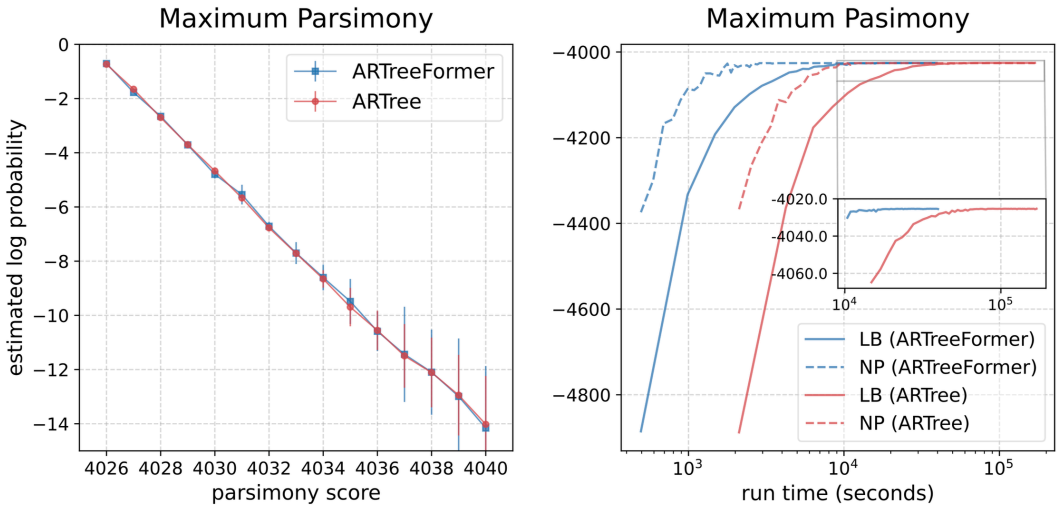


**Fig 4. Performances of ARTree and ARTreeFormer on the maximum parsimony problem.** Left: The estimated log probability $\log Q(\tau)$ versus the parsimony score $\mathcal{S}(\tau, \boldsymbol{Y})$ on DS1. For different tree topologies with the same parsimony score, the mean of the estimated log probabilities is plotted as a dot with the standard deviation as the error bar. Right: The 10-sample lower bound (LB) and the negative parsimony score (NP) as a function of the run time on DS1.

https://doi.org/10.1371/journal.pcbi.1013768.g004

from 4040 to the optimal score 4026. We use the command line `hsearch addseq=random nreps=100 keep=4040` for obtaining all the tree topologies with a parsimony score smaller than or equal to 4040. The version of PAUP* is `4.0a168`. PAUP* is used to form a collection of tree topologies with low parsimony scores for the evaluation of different variational approaches. At the current stage, our main goal is to compare the efficiency between different variational approaches for phylogenetic inference, and we do not compare ARTreeFormer to PAUP* as the latter does not perform Bayesian inference. The left plot of Fig 4 shows that ARTreeFormer and ARTree can provide comparably accurate posterior estimates and identify the most parsimonious tree topology found by PAUP*. In the right plot of Fig 4, the horizontal gap between two curves reflects the ratio of times needed to reach the same lower bound or negative parsimony score. We see that ARTreeFormer is around four times faster than ARTree.

### Tree topology density estimation

We further investigate the ability of ARTreeFormer to model tree topologies on the TDE task. To construct the training data set, we run MrBayes [36] 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, collect the samples every 100 iterations, and discard the first 25%, following [16]. The ground truth distributions are obtained from 10 extremely long single-chain MrBayes runs, each for one billion iterations, where the samples are collected every 1000 iterations, with the first 25% discarded as burn-in. We train ARTreeFormer via maximum likelihood estimation using stochastic gradient ascent. We compare ARTreeFormer to ARTree and SBN baselines: (i) for SBN-EM and SBN-EM-$\alpha$, the SBN model is optimized using the expectation-maximization (EM) algorithm, as done in [16]; (ii) for SBN-SGA and ARTree, the corresponding models are fitted via stochastic gradient ascent, similarly to ARTreeFormer. For SBN-SGA, ARTree, and ARTreeFormer, the results are collected after 200000 parameter updates with a batch size of 10.

The left plot in Fig 5 shows a significant reduction in the training time and evaluation time of ARTreeFormer compared to ARTree on DS1-8. To further demonstrate the benefit of vectorization over different tree topologies, we train ARTreeFormer on DS1 with different batch sizes, and report the Kullback-Leibler (KL) divergences in Fig 5 (right). We see

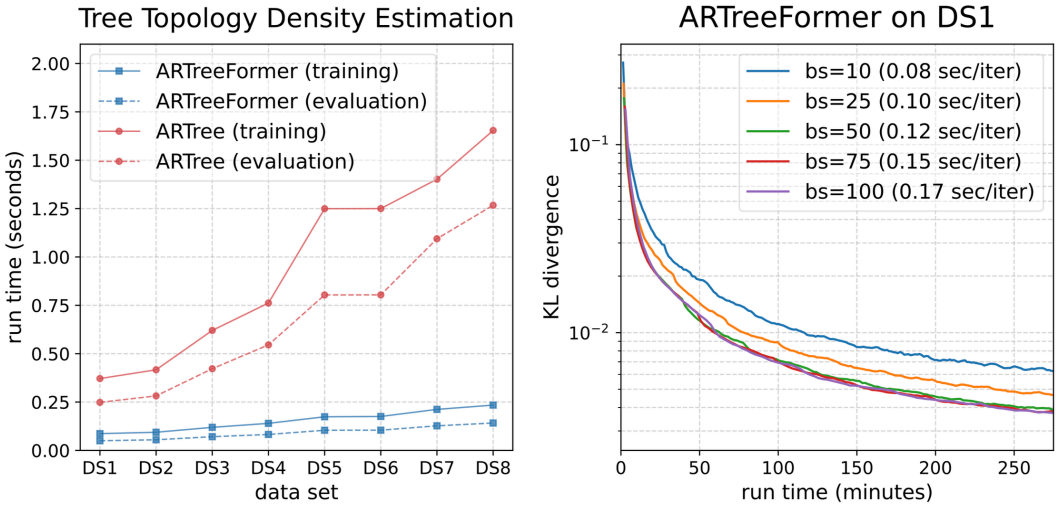


**Fig 5. Performance of ARTree and ARTreeFormer on the TDE task.** Left: The training time (per iteration) and evaluation time (per evaluating the probabilities of 10 tree topologies) of ARTree and ARTreeFormer across eight benchmark data sets for TDE (averaged over 100 trials). Right: The KL divergence to the ground truth on DS1 obtained by ARTreeFormer, as the batch size (bs) varies. The training speed measured by seconds per iteration is reported in the parenthesis.

https://doi.org/10.1371/journal.pcbi.1013768.g005

that a large batch size will only lead to a minor training speed drop, but will significantly benefit the training accuracy. We can also observe a saturated approximation accuracy with a sufficiently large batch size.

The KL divergences between the ground truth and the probability estimation are reported in Table 1. Although ARTreeFormer has only one attention layer for node features, it performs on par or better than ARTree, and consistently outperforms the SBN-related baselines, across all data sets. See the probability estimation on individual tree topologies and an ablation study about the hyperparameters in Appendix D.1 in S1 Text.

### Variational Bayesian phylogenetic inference

Our last experiment is on VBPI, where we examine the performance of ARTreeFormer on tree topology posterior approximation. Following [17], we use the following annealed unnormalized posterior as our target at the $t$-th iteration

$$p(\tau, \boldsymbol{q}|\boldsymbol{Y}, \beta_t) \propto p(\boldsymbol{Y}|\tau, \boldsymbol{q})^{\beta_t} p(\tau, \boldsymbol{q}), \tag{18}$$

**Table 1. KL divergences to the ground truth of different methods across eight benchmark data sets.**

| Data set | # Taxa | # Sites | Sampled trees | GT tree | KL divergence to ground truth | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | SBN-EM | SBN-EM-$\alpha$ | SBN-SGA | ARTree | ARTreeFormer |
| DS1 | 27 | 1949 | 1228 | 2784 | 0.0136 | 0.0130 | 0.0504 | **0.0045** | 0.0065 |
| DS2 | 29 | 2520 | 7 | 42 | 0.0199 | 0.0128 | 0.0118 | **0.0097** | 0.0102 |
| DS3 | 36 | 1812 | 43 | 351 | 0.1243 | 0.0882 | 0.0922 | 0.0548 | **0.0474** |
| DS4 | 41 | 1137 | 828 | 11505 | 0.0763 | 0.0637 | 0.0739 | 0.0299 | **0.0267** |
| DS5 | 50 | 378 | 33752 | 1516877 | 0.8599 | 0.8218 | 0.8044 | 0.6266 | **0.6199** |
| DS6 | 50 | 1133 | 35407 | 809765 | 0.3016 | 0.2786 | 0.2674 | 0.2360 | **0.2313** |
| DS7 | 59 | 1824 | 1125 | 11525 | 0.0483 | 0.0399 | 0.0301 | 0.0191 | **0.0152** |
| DS8 | 64 | 1008 | 3067 | 82162 | 0.1415 | 0.1236 | 0.1177 | 0.0741 | **0.0563** |

The "Sampled trees" column shows the numbers of unique tree topologies in the training sets. The "GT trees" column shows the numbers of unique tree topologies in the ground truth. The results are averaged over 10 replicates. The results of SBN-EM, SBN-EM-$\alpha$ are from [16], and the results of SBN-SGA and ARTree are from [17].

https://doi.org/10.1371/journal.pcbi.1013768.t001

where $\beta_t = \min\{1, 0.001 + t/H\}$ is the annealing weight and $H$ is the annealing period. We use the VIMCO estimator [42] and the reparametrization trick [44] to obtain the gradient estimates for the tree topology parameters and the branch lengths parameters, respectively. The results are collected after 400000 parameter updates.

**VBPI on DS1-8.** In this part we test the performance of VBPI on the eight standard benchmarks DS1-8, as considered in [16,17,21–24,45]. We set $H = 200000$ for the two more difficult dataset DS6 and DS7, and $H = 100000$ for other data sets, following the setting in [17]. We set $K = 10$ for the multi-sample lower bound (3). The results are collected after 400000 parameter updates. To be fair, for all three VBPI-based methods (VBPI-SBN, VBPI-ARTree, and VBPI-ARTreeFormer), we use the same branch length model that is parametrized by GNNs with edge convolutional operator and learnable topological features as done in [22]. We also consider two alternative approaches ($\phi$-CSMC [46], GeoPhy [47]) that provide unconfined tree topology distributions and one MCMC based method (MrBayes) as baselines.

The left plot in Fig 6 shows the lower bound as a function of the number of iterations on DS1. We see that although ARTreeFormer converges more slowly than SBN and ARTree at the beginning, it quickly catches up and reaches a similar lower bound in the end. The middle plot in Fig 6 shows that both ARTree and ARTreeFormer can provide accurate variational approximations to the ground truth posterior of tree topologies, and both of them outperform SBNs by a large margin. In the right plot of Fig 6, we see that the computation time of ARTreeFormer is substantially reduced compared to ARTree. This reduction is especially evident for sampling time since it does not include the branch length generation, likelihood computation, and backpropagation.

Table 2 shows the marginal likelihood estimates obtained by different methods on DS1-8, including the results of the stepping-stone (SS) method [48], which is one of the state-of-the-art sampling based methods for marginal likelihood estimation. We find that VBPI-ARTreeFormer provides comparable estimates to VBPI-SBN and VBPI-ARTree. Compared to other VBPI variants, the methodological and computational superiority of ARTreeFormer is mainly reflected by its unconfined support (compared to SBN) and faster computation speed (compared to ARTree). All VBPI variants perform on par with SS, while the other baselines ($\phi$-CSMC, GeoPhy) tend to provide underestimated results. We also note that the standard deviations of ARTreeFormer can be smaller than those of ARTree and SBN on most data sets, which can be partially attributed to the potentially more accurate approximation. Regarding the efficiency-accuracy trade-off, the simplified architecture in ARTreeformer is enough to maintain or even surpass the performance of ARTree. We also provide more information on the memory and parameter size of different methods for VBPI in Appendix D.2 in S1 Text. Finally, it is worth

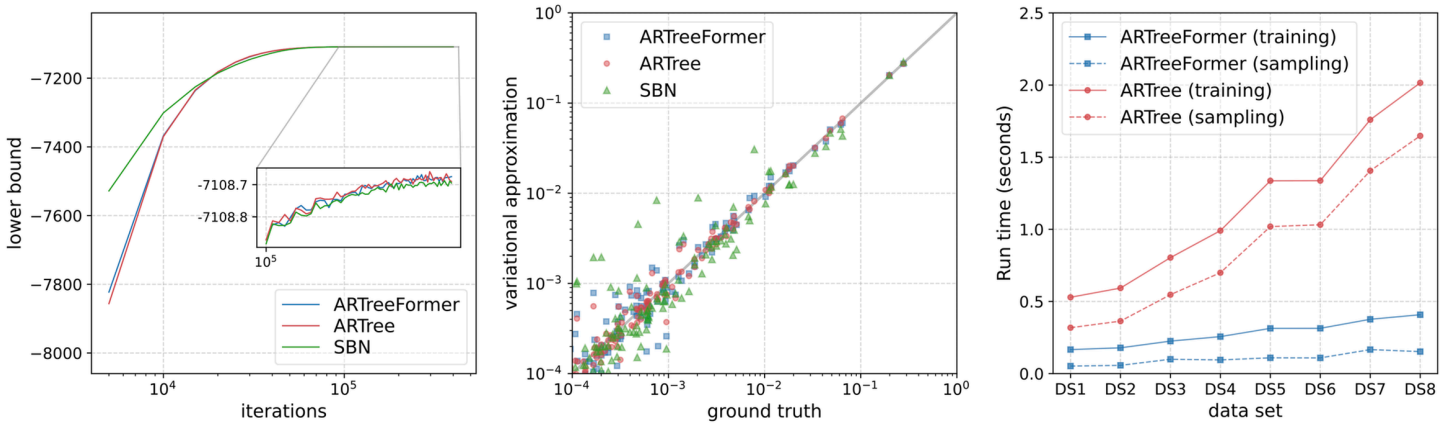


**Fig 6. Performances of different methods for VBPI.** Left: the 10-sample lower bound as a function of the number of iterations on DS1. Middle: the variational approximation v.s. the ground truth of the marginal distribution of tree topologies on DS1. Right: Training time per iteration and sampling time (per sampling 10 tree topologies) across different data sets (averaged over 100 trials).

https://doi.org/10.1371/journal.pcbi.1013768.g006

**Table 2. Marginal likelihood estimates (in units of nats) of different methods across eight benchmark data sets for Bayesian phylogenetic inference.**

| 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 |
| $\phi$-CSMC [46] | –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) |
| GeoPhy [47] | –7111.55(0.07) | –26368.44(0.13) | –33735.85(0.12) | –13337.42(1.32) | –8233.89(6.63) | –6733.91(0.57) | –37350.77(11.74) | –8660.48(0.78) |
| VBPI-SBN [22] | –7108.41(0.14) | **–26367.73(0.07)** | –33735.12(0.09) | –13329.94(0.19) | –8214.64(0.38) | –6724.37(0.40) | **–37332.04(0.26)** | **–8650.65(0.45)** |
| VBPI-ARTree [17] | –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) |
| **VBPI-ARTreeFormer (ours)** | **–7108.43(0.13)** | **–26367.71(0.07)** | **–33735.08(0.08)** | **–13329.93(0.17)** | **–8214.63(0.30)** | **–6724.47(0.35)** | –37331.94(0.31) | –8650.63(0.47) |
| MrBayes SS [48] | –7108.42(0.18) | –26367.57(0.48) | –33735.44(0.50) | –13330.06(0.54) | –8214.51(0.28) | –6724.07(0.86) | –37332.76(2.42) | –8649.88(1.75) |

The "GT trees" column shows the numbers of unique tree topologies in the ground truth, reflecting the diversity of the phylogenetic posterior. The marginal likelihood estimates for ARTreeFormer are obtained by importance sampling with 1000 particles from the variational approximation and are averaged over 100 independent runs with standard deviation in the brackets. A smaller variance is better. The results of MrBayes SS, which serve as the ground truth, are from [21]. The results of other methods are reported in their original papers.

https://doi.org/10.1371/journal.pcbi.1013768.t002

noting that VBPI-mixture [49,50] can provide a better marginal likelihood approximation by employing mixtures of tree models as the variational family.

**VBPI on influenza data.** To further test the scalability and vectorization ability of ARTreeFormer, we consider the influenza data set with an increasing number - $N = 25, 50, 75, 100$ - of nested hemagglutinin (HA) sequences [51]. These sequences were obtained from the NIAID Influenza Research Database (IRD) [52] through the website at `https://www.fludb.org/`, downloading all complete HA sequences that passed quality control, which were then subset to H7 sequences, and further downsampled using the Average Distance to the Closest Leaf (ADCL) criterion [53]. For all the VBPI based methods - SBN, ARTree, and ARTreeFormer, we set $H = 100000$, and the results are collected after 400000 parameter updates. For ARTree and ARTreeFormer, we use the same branch length model that is parametrized by GNNs with edge convolutional operator and learnable topological features as done in [22]; for SBN, we use the bipartition-feature-based branch length model considered in [21].

Table 3 reports the marginal likelihood estimates of different methods on the influenza data set. We see that all three VBPI methods yield very similar marginal likelihood estimates to SS when $N = 25, 50$. For a larger number of sequences $N = 75, 100$, SS tends to provide higher marginal likelihood estimates than VBPI methods, albeit with larger variances which indicates the decreasing reliability of those estimates. On the other hand, the variances of the estimates provided by VBPI methods are much smaller which implies more reliable estimates [51]. Compared to ARTree, ARTreeFormer can provide much better MLL estimates (also closer to SBN) while maintaining a relatively small variance, striking a better balance between approximation accuracy and reliability.

## Discussion

**Comparison with prior works.** The most common approach for Bayesian phylogenetic inference is Markov chain Monte Carlo (MCMC), which relies on random walks to explore the tree space, e.g., MrBayes [36], BEAST [54]. MCMC methods have long been considered the standard practice of systematic biology research and are used to construct the

**Table 3. The marginal likelihood estimates (in units of nats) of different methods on the influenza data with up to 100 taxa.**

| Subset size ($N$) | MrBayes SS | VBPI-SBN | VBPI-ARTree | VBPI-ARTreeFormer |
|---|---|---|---|---|
| 25 | –13378.23(0.24) | –13378.38(0.06) | –13378.39(0.06) | –13378.38(0.06) |
| 50 | –18615.82(1.57) | –18615.40(0.16) | –18615.31(0.18) | –18615.30(0.20) |
| 75 | –23647.14(13.25) | –23681.85(0.27) | –23849.85(0.30) | –23763.58(0.29) |
| 100 | –28176.80(47.16) | –28556.96(0.36) | –29416.42(0.44) | –28650.72(0.32) |

The results of MrBayes SS and VBPI-SBN are reported by [51], and those of VBPI-ARTree and VBPI-ARTreeFormer are produced by us.

https://doi.org/10.1371/journal.pcbi.1013768.t003

ground truth phylogenetic trees in our experiments. However, as the tree space contains both the continuous and discrete components (i.e., the branch lengths and tree topologies), the posterior distributions of phylogenetic trees are often complex multimodal distributions. Furthermore, the involved tree proposals are often limited to local modifications that can lead to low exploration efficiency, which makes MCMC methods require extremely long runs to deliver accurate posterior estimates [10,51].

ARTreeFormer is established in the line of variational inference (VI) [55,56], another powerful tool for Bayesian inference. VI selects the closest member to the posterior distribution from a family of candidate variational distributions by minimizing some statistical distance, usually the KL divergence. Compared to MCMC, VI tends to be faster and easier to scale up to large data by transforming a sampling problem into an optimization problem. The success of VI often relies on the design of expressive variational families and efficient optimization procedures. Besides the variational Bayesian phylogenetic inference (VBPI) introduced before, there exist other VI methods for Bayesian phylogenetic inference. VaiPhy [46] approximates the posterior of multifurcating trees with a novel sequential tree topology sampler based on maximum spanning trees. GeoPhy [47] models the tree topology distribution through a mapping from continuous distributions over the leaf nodes to tree topologies via the Neighbor-Joining (NJ) algorithm [57]. PhyloGen [58] uses pre-trained DNA-based node features for computing the pairwise distance matrix, which will then be mapped to a binary tree topology with the NJ algorithm.

As a classical tool in Bayesian statistics, sequential Monte Carlo (SMC) [59] and its variant combinatorial SMC (CSMC) [60] propose to sample tree topologies through subtree merging and resampling steps for Bayesian phylogenetic inference. VCSMC [61] employs a learnable proposal distribution based on CSMC and optimizes it within a variational framework. $\phi$-CSMC [46] makes use of the parameters of VaiPhy to design the proposal distribution for sampling bifurcating trees. This approach is further developed by H-VCSMC [62] which transfers the merging and resampling steps of VCSMC to the hyperbolic space. The subtree merging operation in SMC based methods is also the core idea of PhyloGFN [20], which instead treats the merging choices as actions within the GFlowNet [63] framework and optimizes the trajectory balance objective [64].

Approximate Bayesian computation (ABC) [65] can also be applied to Bayesian phylogenetic inference [66,67]. It is particularly useful in situations where likelihoods are difficult to compute or involve complex dependencies, such as in tree space with ancestor recombinations or when dealing with models that have many parameters (e.g., substitution models, tree topologies). ARTreeFormer, along with other variational inference methods for phylogenetic models, could potentially be adapted to these challenging scenarios, and exploring this direction is an exciting avenue for future work. In fact, the approach taken by ARTreeFormer is somewhat orthogonal to ABC, with each method offering complementary strengths. While ABC excels in situations where likelihood computations are infeasible, ARTreeFormer provides a powerful variational approximation that can scale more efficiently to larger datasets.

**Potential impact of ARTreeFormer.** Building upon the tree topology construction algorithm of ARTree, ARTreeFormer introduces a more computationally efficient and expressive distribution family for variational Bayesian phylogenetic inference (VBPI). The efficiency gains primarily stem from the use of a fixed-point algorithm in the node embedding module. While fixed-point algorithms can often raise concerns regarding the cost of matrix multiplications and potentially long convergence times—especially when poorly tuned—ARTreeFormer addresses these challenges effectively in several ways. First, matrix multiplications are implemented as tensor operations, which are efficiently accelerated on CUDA-enabled devices (see our open-source implementation). Second, the number of iterations required for convergence is significantly reduced through the use of the power trick, achieving logarithmic scaling. Thirdly, we provide a theoretical guarantee (Corollary 1) that the convergence rate of the fixed-point algorithm is constant, independent of the number of taxa or the shape of the tree topology.

Topological node embeddings (i.e., learnable topological features) [22] provide a general-purpose representation framework for phylogenetic trees and have been employed in various downstream tasks. For example, VBPI-SIBranch [24] uses these embeddings to parametrize semi-implicit branch length distributions, while PhyloVAE [68] leverages them

to obtain low-dimensional representations of tree topologies for tree clustering and diagnostic analysis in phylogenetics. The fixed-point algorithm introduced in this work offers an improved and efficient approach to computing these embeddings, and can be seamlessly integrated into such downstream applications, demonstrating broad potential for impact across phylogenetic modeling tasks.

Another key contribution of ARTreeFormer is the integration of the attention mechanism [18] into phylogenetic inference. Since its introduction, attention has become a foundational component in modern deep learning, powering numerous milestone models such as GPT-4o [69] and DeepSeek-V2 [70]. Despite its widespread success, its potential for modeling phylogenetic tree structures remains underexplored. In this work, we demonstrate that incorporating attention into the message passing module of ARTreeFormer enables comparable or superior performance relative to traditional graph neural networks (GNNs), highlighting its effectiveness in capturing long-range dependencies in tree-structured data.

Phylogenetic inference provides critical insights for making informed public health decisions, particularly during pandemics. Developing efficient Bayesian phylogenetic inference algorithms that can deliver accurate posterior estimates in a timely manner is therefore of immense value, with the potential to save countless lives. VI approaches hold significant promise due to their optimization-based framework. For example, VI methods have been used for rapid analysis of pandemic-scale data (e.g., SARS-CoV-2 genomes) to provide accurate estimates of epidemiologically relevant quantities that can be corroborated via alternative public health data sources [71]. We expect more efficient VI approaches for Bayesian phylogenetics and associated software to be developed in the near future, further advancing this critical field.

**Taxa order in autoregressive modeling.** Both ARTree and ARTreeForemr use a fixed alphabetical order on taxa names. Although Fig 3 (right) in [17] demonstrates that the autoregressive modeling can be robust to the pre-defined taxa order, we acknowledge that this alphabetical order does not have biological interpretation and fixing this order can lead to bias and overfitting. To understand this, letting $\tau_N$ be a tree topology with large posterior probability and $\sigma$ be an arbitrary taxa order, we have the autoregressive decomposition $p(\tau_N|\sigma) = \prod_{n=3}^{N-1} p(e_n|e_{<n}, \sigma)$ similarly to equation (16). If $\sigma$ is inappropriate for $\tau_N$, then some of the conditional distributions $p(e_n|e_{<n}, \sigma)$ can be very difficult to approximate (e.g., they could be multimodal). This increases the likelihood of getting trapped in local modes, which poses a challenge for learning the variational distribution.

The generative framework of ARTreeFormer can be further developed to incorporate the taxa order modeling. To do this, we can augment a tree topology $\tau$ with a taxa order $\sigma$, resulting in an augmented modeling space $(\tau, \sigma)$, where there are $N!$ possible orders. We will then define a joint variational distribution $Q_\phi(\tau, \sigma)$ and perform variational inference over this augmented space of $(\tau, \sigma)$ by optimizing the variational lower bound, similar to the original ARTreeFormer approach. In practice, this can be done by sampling a leaf node from the remaining taxa according to a parameterized distribution, followed by sampling an edge on the current subtree topology to attach the selected leaf node. This process implicitly defines a distribution over taxa orders that can depend on the tree topology, addressing the potential issue of a fixed, biologically meaningless order.

**Scalability.** In this paper, ARTreeFormer is tested on datasets with up to 100 taxa, which we acknowledge as a limitation of the current methodology. For larger trees (e.g., thousands of taxa) or highly heterogeneous datasets (e.g., those with recombination or gene tree discordance), training becomes more challenging due to the increased complexity of the phylogenetic posterior. This challenge is intrinsic to Bayesian phylogenetic inference, as it requires exploring a vast tree topology space. Most Bayesian phylogenetic studies in the literature focus on datasets ranging from dozens to a few hundred taxa. For MCMC-based methods, even on high-performance computers, extremely long runs (weeks or months) are often necessary to produce reliable results on large-scale datasets.

For practical purposes, if the goal is a rough estimate of the posterior in a timely manner, variational approaches like ARTreeFormer provide a useful trade-off between approximation accuracy and computational speed. One way to handle large datasets or heterogeneous data would be to simplify the model architecture. By sacrificing some approximation accuracy, ARTreeFormer can still offer meaningful uncertainty quantification within a more reasonable computational budget. Additionally, even with linear time complexity, the computational cost of ARTreeFormer for very large trees would still

be substantial. An improvement could involve allowing multiple leaf nodes to be added simultaneously, similar to the multiple token generation approach in discrete diffusion models [72,73]. This would enhance generation speed, potentially addressing scalability concerns.

**Future directions.** There are several future practical directions for advancing ARTreeFormer, which we discuss as follows. Firstly, the embedding method for phylogenetic trees in ARTreeFormer can be further explored. For example, PhyloGen [58] use pre-trained DNA-based node features, and GeoPhy [47] and H-VCSMC [62] consider embedding trees in hyperbolic space. As the input to the model, the representation power and generalization ability of the embedding method might have a marked impact on the performance of ARTreeFormer. Secondly, the attention mechanism for the message passing on phylogenetic trees can be more delicately designed. For example, the attention masks can be modified according to the neighborhood structures. [74] provides a comprehensive survey on the design details of graph transformers. Thirdly, the fast computation and scalability of ARTreeFormer offer the possibility of large phylogenetic inference models capable of zero-shot inference on biological sequences. This may require more expressive model designs, especially powerful node embedding schemes, and more high-quality data. Fourthly, one can combine Markov chain Monte Carlo (MCMC) with variational inference (VI) to enhance the approximation accuracy. For example, multiple MCMC transitions could be applied to the variational distribution provided by ARTreeFormer, which can be trained with novel objectives [75]. Last but not the least, variational approximations from ARTreeFormer can be used as an importance distribution for importance sampling. This would allow for more accurate posterior approximations by refining the model's estimates and focusing sampling on regions of higher posterior probability. We hope these discussions could help inspire more advances in variational approaches for phylogenetic inference.

## Conclusion

In this work, we presented ARTreeFormer, a variant of ARTree that leverages the scalable fixed-point iteration algorithm and the attention mechanism to accelerate the autoregressive modeling of tree topologies in phylogenetic inference. In contrast to ARTree, which involves the Dirichlet energy minimization via expensive and non-vectorizable tree traversals to compute the node embeddings, ARTreeFormer introduces a specially designed fixed-point algorithm that facilitates highly vectorizable computation. We also introduce an attention-based global message passing module, which is capable of capturing the crucial global information in only one forward pass, to replace the GNN-based local message passing module. Experiments on various phylogenetic inference problems showed that ARTreeFormer is significantly faster than ARTree in training and evaluation while performing comparably or better in terms of approximation accuracy.

## Supporting information

**S1 Text. Supporting text with appendices.** Appendix A: Details of ARTree. Appendix B: Details of variational Bayesian phylogenetic inference. Appendix C: Proofs for the time complexity results. Appendix D: Additional experimental results. (PDF)

## Acknowledgments

The authors appreciate Zichao Yan, Ming Yang Zhou, and Dinghuai Zhang for their constructive discussion on this project. The authors are grateful for the computational resources provided by the High-performance Computing Platform of Peking University.

## Author contributions

**Conceptualization:** Cheng Zhang.

**Data curation:** Tianyu Xie, Cheng Zhang.

**Formal analysis:** Tianyu Xie, Cheng Zhang.

**Funding acquisition:** Cheng Zhang.

**Investigation:** Tianyu Xie, Yicong Mao.

**Methodology:** Tianyu Xie, Cheng Zhang.

**Project administration:** Cheng Zhang.

**Resources:** Cheng Zhang.

**Software:** Tianyu Xie, Yicong Mao.

**Supervision:** Cheng Zhang.

**Validation:** Cheng Zhang.

**Visualization:** Tianyu Xie, Yicong Mao.

**Writing – original draft:** Tianyu Xie, Yicong Mao.

**Writing – review & editing:** Tianyu Xie, Cheng Zhang.

## References


1. Dudas G, Carvalho LM, Bedford T, Tatem AJ, Baele G, Faria NR, et al. Virus genomes reveal factors that spread and sustained the Ebola epidemic. Nature. 2017;544(7650):309–15. https://doi.org/10.1038/nature22040 PMID: 28405027
2. du Plessis L, McCrone JT, Zarebski AE, Hill V, Ruis C, Gutierrez B, et al. Establishment and lineage dynamics of the SARS-CoV-2 epidemic in the UK. Science. 2021;371(6530):708–12. https://doi.org/10.1126/science.abf2946 PMID: 33419936
3. Attwood SW, Hill SC, Aanensen DM, Connor TR, Pybus OG. Phylogenetic and phylodynamic approaches to understanding and combating the early SARS-CoV-2 pandemic. Nat Rev Genet. 2022;23(9):547–62. https://doi.org/10.1038/s41576-022-00483-8 PMID: 35459859
4. DeSalle R, Amato G. The expansion of conservation genetics. Nat Rev Genet. 2004;5(9):702–12. https://doi.org/10.1038/nrg1425 PMID: 15372093
5. Felsenstein J. Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol. 1981;17(6):368–76. https://doi.org/10.1007/BF01734359 PMID: 7288891
6. Fitch WM. Toward defining the course of evolution: minimum change for a specific tree topology. Systematic Biology. 1971;20(4):406–16. https://doi.org/10.1093/sysbio/20.4.406
7. Yang Z, Rannala B. Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo Method. Mol Biol Evol. 1997;14(7):717–24. https://doi.org/10.1093/oxfordjournals.molbev.a025811 PMID: 9214744
8. Mau B, Newton MA, Larget B. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. Biometrics. 1999;55(1):1–12. https://doi.org/10.1111/j.0006-341x.1999.00001.x PMID: 11318142
9. Larget B, Simon DL. Markov Chasin Monte Carlo algorithms for the bayesian analysis of phylogenetic trees. Molecular Biology and Evolution. 1999;16(6):750–9. https://doi.org/10.1093/oxfordjournals.molbev.a026160
10. Whidden C, Matsen FA 4th. Quantifying MCMC exploration of phylogenetic tree space. Syst Biol. 2015;64(3):472–91. https://doi.org/10.1093/sysbio/syv006 PMID: 25631175
11. Dinh V, Bilge A, Zhang C, Matsen IV FA. Probabilistic Path Hamiltonian Monte Carlo. In: Proceedings of the 34th International Conference on Machine Learning. 2017. p. 1009–18. http://proceedings.mlr.press/v70/dinh17a.html
12. Chor B, Tuller T. Maximum likelihood of evolutionary trees is hard. In: The 9th Annual International Conference on Research in Computational Molecular Biology; 2005.
13. Day WH. Computational complexity of inferring phylogenies from dissimilarity matrices. Bull Math Biol. 1987;49(4):461–7. https://doi.org/10.1007/BF02458863 PMID: 3664032
14. Höhna S, Drummond AJ. Guided tree topology proposals for Bayesian phylogenetic inference. Syst Biol. 2012;61(1):1–11. https://doi.org/10.1093/sysbio/syr074 PMID: 21828081
15. Larget B. The estimation of tree posterior probabilities using conditional clade probability distributions. Syst Biol. 2013;62(4):501–11. https://doi.org/10.1093/sysbio/syt014 PMID: 23479066
16. Zhang C, Matsen IV FA. Generalizing tree probability estimation via Bayesian networks. In: The Thirty-second Conference on Neural Information Processing Systems; 2018.

17. Xie T, Zhang C. ARTree: a deep autoregressive model for phylogenetic inference. In: Thirty-seventh Conference on Neural Information Processing Systems; 2023.
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN. Attention is all you need. Advances in Neural Information Processing Systems. 2017.
19. Felsenstein J. Inferring phylogenies. 2nd ed. Sinauer Associates; 2004.
20. Zhou MY, Yan Z, Layne E, Malkin N, Zhang D, Jain M, et al. PhyloGFN: phylogenetic inference with generative flow networks. In: The Twelfth International Conference on Learning Representations; 2024.
21. Zhang C, Matsen IV FA. Variational Bayesian phylogenetic inference. In: The Seventh International Conference on Learning Representations; 2019.
22. Zhang C. Learnable topological features for phylogenetic inference via graph neural networks. In: The Eleventh International Conference on Learning Representations; 2023.
23. Zhang C. Improved variational bayesian phylogenetic inference with normalizing flows. In: The Thirty-fourth Conference on Neural Information Processing Systems; 2020.
24. Xie T, Matsen IV FA, Suchard MA, Zhang C. Variational Bayesian phylogenetic inference with semi-implicit branch length distributions. arXiv preprint 2024. https://doi.org/arXiv:240805058
25. Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. arXiv preprint 2017. https://doi.org/abs/1704.01212
26. Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. MIT Press; 2022.
27. Spielman DA. Spectral and algebraic graph theory; 2025. http://cs-www.cs.yale.edu/homes/spielman/sagt/sagt.pdf
28. Hedges SB, Moberg KD, Maxson LR. Tetrapod phylogeny inferred from 18S and 28S ribosomal RNA sequences and a review of the evidence for amniote relationships. Mol Biol Evol. 1990;7(6):607–33. https://doi.org/10.1093/oxfordjournals.molbev.a040628 PMID: 2283953
29. Garey JR, Near TJ, Nonnemacher MR, Nadler SA. Molecular evidence for Acanthocephala as a subtaxon of Rotifera. J Mol Evol. 1996;43(3):287–92. https://doi.org/10.1007/BF02338837 PMID: 8703095
30. Yang Z, Yoder AD. 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. 2003;52(5):705–16. https://doi.org/10.1080/10635150390235557 PMID: 14530137
31. Henk DA, Weir A, Blackwell M. Laboulbeniopsis termitarius, an ectoparasite of termites newly recognized as a member of the Laboulbeniomycetes. Mycologia. 2003;95(4):561–4. https://doi.org/10.1080/15572536.2004.11833059 PMID: 21148964
32. Lakner C, van der Mark P, Huelsenbeck JP, Larget B, Ronquist F. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. Syst Biol. 2008;57(1):86–103. https://doi.org/10.1080/10635150801886156 PMID: 18278678
33. Zhang N, Blackwell M. Molecular phylogeny of dogwood anthracnose fungus (Discula destructiva) and the Diaporthales. Mycologia. 2001;93(2):355–65. https://doi.org/10.1080/00275514.2001.12063167
34. Yoder AD, Yang Z. Divergence dates for Malagasy lemurs estimated from multiple gene loci: geological and evolutionary context. Mol Ecol. 2004;13(4):757–73. https://doi.org/10.1046/j.1365-294x.2004.02106.x PMID: 15012754
35. Rossman AY, McKemy JM, Pardo-Schultheiss RA, Schroers H-J. Molecular studies of the Bionectriaceae using large subunit rDNA sequences. Mycologia. 2001;93(1):100–10. https://doi.org/10.1080/00275514.2001.12061283
36. Ronquist F, Teslenko M, van der Mark P, Ayres DL, Darling A, Höhna S, et al. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. Syst Biol. 2012;61(3):539–42. https://doi.org/10.1093/sysbio/sys029 PMID: 22357727
37. Jukes TH, Cantor CR. Evolution of Protein Molecules. 1969.
38. Clevert D, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs). arXiv preprint 2016. http://arxiv.org/abs/1511.07289
39. Xiong R, Yang Y, He D, Zheng K, Zheng S, Xing C, et al. On layer normalization in the transformer architecture. In: The thirty-seventh International Conference on Machine Learning. PMLR; 2020. p. 10524–33.
40. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: an imperative style, high-performance deep learning library. In: The Thirty-third Annual Conference on Neural Information Processing Systems; 2019.
41. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: The third international conference on learning representations; 2015.
42. Mnih A, Rezende DJ. Variational inference for Monte Carlo objectives. In: The Thirty-third International Conference on Machine Learning; 2016.
43. Swofford D. PAUP*: Phylogenetic analysis using parsimony. 2003. http://paupcsitfsuedu/
44. Kingma DP, Welling M. Auto-encoding variational Bayes. In: The second International Conference on Learning Representations; 2014.
45. Xie T, Yuan M, Deng M, Zhang C. Improving tree probability estimation with stochastic optimization and variance reduction. Stat Comput. 2024;34(6):186.https://doi.org/10.1007/s11222-024-10498-2
46. Koptagel H, Kviman O, Melin H, Safinianaini N, Lagergren J. VaiPhy: a variational inference based algorithm for phylogeny. In: Advances in Neural Information Processing Systems. 2022.

47. Mimori T, Hamada M. GeoPhy: differentiable phylogenetic inference via geometric gradients of tree topologies. In: The Thirty-seventh Annual Conference on Neural Information Processing Systems; 2023.
48. Xie W, Lewis PO, Fan Y, Kuo L, Chen M-H. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. Syst Biol. 2011;60(2):150–60. https://doi.org/10.1093/sysbio/syq085 PMID: 21187451
49. Molén R, Kviman O, Lagergren J. Improved variational bayesian phylogenetic inference using mixtures. Transactions on Machine Learning Research. 2024.
50. Hotti A, Kviman O, Molén R, Elvira V, Lagergren J. Efficient mixture learning in black-box variational inference. In: The Forty-first International Conference on Machine Learning; 2024.
51. Zhang C, Matsen FA. A variational approach to Bayesian phylogenetic inference. Journal of Machine Learning Research. 2024;25(145):1–56.
52. Zhang Y, Aevermann BD, Anderson TK, Burke DF, Dauphin G, Gu Z, et al. Influenza research database: an integrated bioinformatics resource for influenza virus research. Nucleic Acids Res. 2017;45(D1):D466–74. https://doi.org/10.1093/nar/gkw857 PMID: 27679478
53. Matsen FA 4th, Gallagher A, McCoy CO. Minimizing the average distance to a closest leaf in a phylogenetic tree. Syst Biol. 2013;62(6):824–36. https://doi.org/10.1093/sysbio/syt044 PMID: 23843314
54. Drummond AJ, Rambaut A. BEAST: Bayesian evolutionary analysis by sampling trees. BMC Evol Biol. 2007;7:214. https://doi.org/10.1186/1471-2148-7-214 PMID: 17996036
55. Jordan MI, Ghahramani Z, Jaakkola TS, Saul LK. An introduction to variational methods for graphical models. Machine Learning. 1999;37(2):183–233. https://doi.org/10.1023/a:1007665907178
56. Blei DM, Kucukelbir A, McAuliffe JD. Variational inference: a review for statisticians. Journal of the American Statistical Association. 2017;112(518):859–77. https://doi.org/10.1080/01621459.2017.1285773
57. Saitou N, Nei M. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol. 1987;4(4):406–25. https://doi.org/10.1093/oxfordjournals.molbev.a040454 PMID: 3447015
58. Duan C, Zang Z, Li S, Xu Y, Li SZ. PhyloGen: language model-enhanced phylogenetic inference via graph structure generation. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems; 2024.
59. Bouchard-Côté A, Sankararaman S, Jordan MI. Phylogenetic inference via sequential Monte Carlo. Syst Biol. 2012;61(4):579–93. https://doi.org/10.1093/sysbio/syr131 PMID: 22223445
60. Wang L, Bouchard-Côté A, Doucet A. Bayesian phylogenetic inference using a combinatorial sequential Monte Carlo method. Journal of the American Statistical Association. 2015;110(512):1362–74. https://doi.org/10.1080/01621459.2015.1054487
61. Moretti AK, Zhang L, Naesseth CA, Venner H, Blei DM, Pe'er I. Variational combinatorial sequential Monte Carlo methods for bayesian phylogenetic inference. In: The Thirty-seventh Conference on Uncertainty in Artificial Intelligence; 2021.
62. Chen A, Chlenski P, Munyuza K, Moretti AK, Naesseth CA, Pe'er I. Variational combinatorial sequential Monte Carlo for Bayesian phylogenetics in hyperbolic space. In: The 28th International Conference on Artificial Intelligence and Statistics; 2025.
63. Bengio E, Jain M, Korablyov M, Precup D, Bengio Y. Flow network based generative models for non-iterative diverse candidate generation. Advances in Neural Information Processing Systems. 2021;34:27381–94.
64. Malkin N, Jain M, Bengio E, Sun C, Bengio Y. Trajectory balance: improved credit assignment in GFlowNets. In: Advances in Neural Information Processing Systems. 2022.
65. Beaumont MA, Zhang W, Balding DJ. Approximate Bayesian computation in population genetics. Genetics. 2002;162(4):2025–35. https://doi.org/10.1093/genetics/162.4.2025 PMID: 12524368
66. Rannala B, Yang Z. Bayesian methods for inferring species trees. Molecular Biology and Evolution. 2003;20(3):474–84.
67. Templeton AR. Statistical hypothesis testing in intraspecific phylogeography: nested clade phylogeographical analysis vs. approximate Bayesian computation. Mol Ecol. 2009;18(2):319–31. https://doi.org/10.1111/j.1365-294X.2008.04026.x PMID: 19192182
68. Xie T, Richman H, Gao J, Matsen IV FA, Zhang C. PhyloVAE: unsupervised learning of phylogenetic trees via variational autoencoders. In: The Thirteenth International Conference on Learning Representations; 2025.
69. OpenAI. Gpt-4o. 2024. https://openai.com/blog/gpt-4o
70. DeepSeek. DeepSeek-V2: A powerful open-source language model. 2024. https://deepseek.com
71. Ki C, Terhorst J. Variational phylodynamic inference using pandemic-scale data. Mol Biol Evol. 2022;39(8):msac154. https://doi.org/10.1093/molbev/msac154 PMID: 35816422
72. Lou A, Meng C, Ermon S. Discrete diffusion modeling by estimating the ratios of the data distribution. arXiv preprint 2023. arXiv:231016834
73. Arriola M, Chiu J, Gokaslan A, Kuleshov V, Marroquin E, Rush A, et al. Simple and effective masked diffusion language models. In: Advances in Neural Information Processing Systems 37. 2024. p. 130136–84. https://doi.org/10.52202/079017-4135
74. Müller L, Galkin M, Morris C, Rampášek L. Attending to graph transformers. Transactions on Machine Learning Research. 2024.
75. Ruiz F, Titsias M. A contrastive divergence for combining variational inference and MCMC. In: Chaudhuri K, Salakhutdinov R, editors. Proceedings of the 36th International Conference on Machine Learning. vol. 97 of Proceedings of Machine Learning Research. PMLR; 2019. p. 5537–45. https://proceedings.mlr.press/v97/ruiz19a.html
76. Wang Y, Sun Y, Liu Z, Sarma SE, Bronstein MM, Solomon JM. Dynamic graph CNN for learning on point clouds. ACM Trans Graph. 2019;38(5):1–12. https://doi.org/10.1145/3326362

77. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint 2014. https://doi.org/10.48550/arXiv.1406.1078

78. Rainforth T, Kosioreck AR, Le TA, Maddison CJ, Igl M, Wood F. Tighter variational bounds are not necessarily better. In: Proceedings of the 36th International Conference on Machine Learning. 2019.

79. Bornschein J, Bengio Y. Reweighted wake-sleep. In: Proceedings of the third International Conference on Learning Representations; 2015.