JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 29, Number 1, 2022

Pp. 74–89
DOI: 10.1089/cmb.2021.0383

Re-evaluating Deep Neural Networks for Phylogeny Estimation: The Issue of Taxon Sampling

PAUL ZAHARIAS,[1,i] MARTIN GROSSHAUSER,[2] and TANDY WARNOW[1,ii]

ABSTRACT

Deep neural networks (DNNs) have been recently proposed for quartet tree phylogeny estimation. Here, we present a study evaluating recently trained DNNs in comparison to a collection of standard phylogeny estimation methods on a heterogeneous collection of datasets simulated under the same models that were used to train the DNNs, and also under similar conditions but with higher rates of evolution. Our study shows that using DNNs with quartet amalgamation is less accurate than several standard phylogeny estimation methods we explore (e.g., maximum likelihood and maximum parsimony). We further find that simple standard phylogeny estimation methods match or improve on DNNs for quartet accuracy, especially, but not exclusively, when used in a global manner (i.e., the tree on the full dataset is computed and then the induced quartet trees are extracted from the full tree). Thus, our study provides evidence that a major challenge impacting the utility of current DNNs for phylogeny estimation is their restriction to estimating quartet trees that must subsequently be combined into a tree on the full dataset. In contrast, global methods (i.e., those that estimate trees from the full set of sequences) are able to benefit from taxon sampling, and hence have higher accuracy on large datasets.

Keywords: deep neural networks, phylogeny estimation and heterotachy.

1. INTRODUCTION

ONE OF THE BASIC CHALLENGES in statistical phylogeny estimation occurs when sequences evolve under processes that violate assumptions of the inferential statistical model. For example, most phylogeny estimation methods are based on sequence evolution models, such as the Generalized Time Reversible (GTR) (Tavaré, 1986) model for nucleotide evolution, that assume that all the sites within the sequences evolve *i.i.d.* (identically and independently) down a model tree. In turn, the sequence evolution models make several simplifying assumptions (such as stationarity, homogeneity, and time-reversibility, jointly referred to as ''SRH'') about sequence evolution to ensure statistical identi ability (i.e., that the model tree is uniquely

[1]Department of Computer Science, University of Illinois, Urbana, Illinois, USA.
[2]Department of Physics, Technical University of Munich, Munich, Germany.
[i]ORCID ID (https://orcid.org/0000-0003-3550-2636).
[ii]ORCID ID (https://orcid.org/0000-0001-7717-3514).

identi ed by the probability distribution it de nes on site patterns) and computational feasibility (i.e., so that tree estimation can be performed by using acceptable efforts).

However, these are unrealistic assumptions, as compositional heterogeneity (indicative of changed substitution rate matrices) across the tree has been observed in many biological datasets [e.g., see discussion in Jermiin et al. (2004)]. Indeed, recent years have shown an increasing awareness of the impact of these violations of the model assumptions on phylogeny estimation (Jermiin et al., 2004; Kolaczkowski and Thornton, 2008; Duchêne et al., 2017; Naser-Khdour et al., 2019; White and Braun, 2019; Crotty et al., 2020), leading Naser-Khdour et al. (2019) to conclude:

> the extent and effects of model violation in phylogenetics may be substantial […] further effort in developing models that do not require SRH assumptions could lead to large improvements in the accuracy of phylogenomic inference.

To address the challenge of model misspeci cation, several investigators have developed parameter-rich models (Steel, 1994; Crotty et al., 2020) and methods for estimating trees under these models; for example, IQ-TREE 2 (Minh et al., 2020) enables estimation under the GHOST (Crotty et al., 2020) model, which incorporates substantial heterogeneity across the tree and across sites. However, two challenges appear with an increased number of parameters: The computational cost increases, and there is a danger of over tting (and hence reduction of accuracy).

An alternative approach that has recently begun to be explored is the use of deep neural networks (DNNs), which are machine-learning models that can be trained on datasets and then used to classify new input datasets. The DNNs have been developed for computing phylogenies on datasets of four sequences (i.e., quartet tree estimation), so that each DNN is trained on sets of four sequences with known true quartet trees, and then used to estimate quartet trees from new sets of four sequences (Suvorov et al., 2020; Zou et al., 2020).

This approach has the bene t of allowing the training to be done under very heterogeneous conditions, including ones that signi cantly violate standard model assumptions. Further, although the training may be computationally intensive, once the training is complete quartet tree estimation is very fast.

Other applications of DNNs to phylogeny estimation are less direct. For example, Bhattacharjee and Bayzid (2020) used machine learning to develop techniques to impute the missing entries in a distance matrix, and so enable the reconstruction of trees from partial distance matrices. Abadi et al. (2020) suggested a machine-learning framework, ModelTeller, for phylogenetic model selection. Leuchtenberger et al. (2020) designed and trained DNNs to distinguish quartet trees exhibiting the properties of the Felsenstein Zone (Felsenstein, 1978) or the Farris Zone (Siddall, 1998), using both simulated and empirical datasets.

This approach does not directly construct quartet trees but does provide guidance to the user in the subsequent choice of method for the phylogeny estimation method for the dataset. Recently, Jiang et al. (2021) developed DEPP, a deep learning-based approach for phylogenetic placement. Even more recently, Azouri et al. (2021) explored the use of a deep-learning algorithm to narrow the tree search space to achieve faster convergence of the maximum likelihood (ML) score.

In this study, we focus on the use of DNNs presented in Suvorov et al. (2020) and Zou et al. (2020), which operate by estimating quartet trees (i.e., unrooted binary trees on just four leaves). To extend the use of these DNNs to estimate larger trees, the quartet trees computed by the DNN must be combined together into a tree on the full dataset, a process referred to as ''quartet amalgamation.''

However, optimization problems for quartet amalgamation, such as maximizing the number of satis ed quartet trees, are NP-hard (Jiang et al., 2001), which necessitates the use of heuristic approaches. Nevertheless, many quartet amalgamation methods have been developed (Strimmer and Von Haeseler, 1996; Ranwez and Gascuel, 2001; Snir et al., 2008; Reaz et al., 2014), with Quartets Max Cut (QMC) now the leading quartet amalgamation method.

We speci cally examine two questions: (1) How accurate are the quartet trees estimated by DNNs in comparison to other standard methods and (2) how accurate are trees computed by DNNs on just slightly larger trees with 20 sequences, in comparison to standard methods? We use the trained DNNs from Zou et al. (2020) for the quartet tree estimators, and QMC for the quartet amalgamation method, thus producing a two-phase approach that we refer to as DNN+QMC.

We simulate sequences using the simulator developed by Zou et al., using the same basic model conditions as they used for training and then extending the set of model conditions to include a wider range

of branch lengths. Signi cantly, we nd that DNNs are less accurate than standard methods for both problems, even when considered on the same data on which they were trained. Indeed, in our study, the commonly used ML methods under standard *i.i.d.* sequence evolution models are more accurate than these DNNs for quartet tree estimation and more accurate than DNN+QMC methods at estimating 20-taxon trees, even though the datasets we examine have evolved under substantial violations of the model assumptions. Further, even neighbor joining (NJ) (Saitou and Nei, 1987) used with p-distances (i.e., normalized Hamming distances) is more accurate than DNN+QMC methods.

Thus, these DNNs do not even match the accuracy of basic phylogeny estimation methods. We provide some insights into why this is likely to be true, focusing on taxon sampling and its bene cial effects in phylogeny estimation. Overall, this study provides a cautionary note about the use of DNNs for phylogenetic gene tree estimation, while indicating some possible directions for their use in phylogenomics.

2. MATERIALS AND METHODS

2.1. Overview

We explore the accuracy of the three DNNs used in Zou et al. (2020). Each of these was trained on a different training set, and they are named after their training datasets; these are DNN1, DNN2, and DNN3. We compare these DNNs with standard phylogeny estimation methods [ML, maximum parsimony (MP) (Foulds and Graham, 1982), NJ, and unweighted pair group method with arithmetic mean (UPGMA) (Sokal and Michener, 1958)] with respect to tree topology accuracy on a range of model conditions. We performed two experiments to understand the performance of DNNs in comparison to standard phylogeny estimation methods. Experiment 1 evaluates the topological accuracy of estimated 20-leaf trees and quartet trees on simulated 200 aa sequence datasets. Experiment 2 performs the same evaluation but on longer sequences (1000 aa). See Appendix A1 for the detailed commands.

2.2. Datasets

We simulated datasets with 20 gap-free amino acid sequences using the *evosimz* simulator from Zou et al. (2020). Three of the model conditions use the same parameter values as the training datasets used in Zou et al. (2020) to train DNN1, DNN2, and DNN3, but we restricted the trees to 20 leaves [while Zou et al. (2020) explored a larger range of tree sizes]. The basic model conditions are called Training1, Training2, and Training3. For the main quartet tree estimation experiments reported in Zou et al. (2020) (i.e., the ''nolba'' conditions, which do not involve long-branch attraction), quartet tree error rates were extremely low: never more than 7%, and most analyses were much better (e.g., MP never had more than 5% error).

This level of quartet tree accuracy is an indication that the tree estimation problem is unusually *easy* under these test model conditions, and hence atypical of most phylogenetic problems. Therefore, in our study, we modi ed the simulator parameter settings to produce more challenging conditions. We achieved this in two ways. First, we modi ed the branch length distributions to increase the rate of evolution. Second, we noted that the main experiments they report were based on sequence lengths drawn from a uniform distribution [100, 3000), so that the median length is about 1550.

Since sequence length has a large impact on phylogeny estimation accuracy, we examined two sequence lengths: 200 aa (a length that is close to typical lengths of single proteins) and 1000 aa [a length that would be much less frequently observed, but still not as unrealistic as the lengths evaluated in Zou et al. (2020)].

We created 12 model conditions, with 4 conditions per basic model condition. Speci cally, for a given basic model condition (Training1, Training2, Training3), we include the original model condition, and we also include three other conditions de ned by modifying the branch length distribution to provide higher upper bounds on the branch lengths, to make for more challenging datasets. For each of the 12 model conditions, 20 trees were generated with 20 leaves each, and gap-free amino acid sequences evolved down the tree under heterogeneous substitution processes that include heterotachy.

With the exception of branch lengths, we set all the numeric simulation parameters identically as for the associated model conditions drawn from Zou et al. (2020). We extended the range of branch lengths as follows: We set the upper bound of the branch length distribution to X times the original upper bound, where X was set to 10, 100, or 1000, and we use the value for X to name the new model conditions [see Zou et al. (2020) for the meaning of the numeric parameters, including the branch length].

We also modi ed the lower bound on the branch length distributions for the Training1-based and Training3-based model conditions. The model parameters for the different model conditions are provided in Supplementary Table S1, and associated empirical statistics of these model conditions are provided in Supplementary Table S3; see also Supplementary Table S2 for information on how to interpret branch lengths.

In Experiment 1, we used all 12 model conditions, and we explored methods on sequences with 200 aa. For Experiment 2, we used 9 of the 12 model conditions (omitting only the three hardest model conditions with the highest rate of evolution) and examined performance on sequences with 1000 aa; thus, Experiment 2 re ects performance on very long protein sequence alignments, a condition that is likely to be much less common than Experiment 1.

2.3. Phylogeny estimation methods

We use a collection of phylogeny estimation methods in this study, starting with the trained DNNs from Zou et al. (2020) and including some standard phylogeny estimation methods.

The DNNs classifiers. We used DNN1 (Epoch 588), DNN2 (Epoch 1272), and DNN3 (Epoch 1098), obtained from Zou et al. (2020), re ecting the nal training state achieved by the authors. DNN1 was trained on quartet trees generated by the rst training set (a superset of Training1), DNN2 was trained on quartet trees generated by the second training set (a superset of Training2), and DNN3 was trained on quartet trees generated by the third training set (a superset of Training3). To compute trees on more than four sequences using DNNs, we combine the quartet trees into a tree on the full dataset using QMC (Snir and Rao, 2008), in default mode. We refer to this two-phase approach as DNN+QMC (i.e., DNN1+QMC, DNN2+QMC, and DNN3+QMC).

Standard methods. For ML analyses, we use the Linux version of IQ-TREE v. 2.0.5 (Minh et al., 2020) in three ways: under the WAG (Whelan and Goldman, 2001) model, under the GHOST (Crotty et al., 2020) model, and using ModelFinder (Kalyaanamoorthy et al., 2017) to select the best tting model according to a BIC criterion. For the MP, NJ, and UPGMA analyses, we use PAUP* v. 4.0a (Swofford, 2002), running these methods in default mode; in particular, both NJ and UPGMA are run by using uncorrected distances (i.e., p-distances, which is the fraction of the sequence length where the two sequences have different amino acids).

Local and global methods for quartet tree estimation. The ''local'' approach for quartet tree estimation is used on four sequences at a time, and it estimates quartet trees by using only the local information and no additional information. For example, the local version of NJ for quartet tree estimation takes four sequences, computes the 4×4 matrix of pairwise distances between them, and then runs NJ on the distance matrix to obtain the quartet tree.

In contrast, the ''global'' approach operates as follows. Given a set of $n > 4$ sequences, a tree is computed on the entire set, and then the individual quartet trees are extracted (by restricting the tree to the desired set of four sequences). Hence, the global version of NJ for quartet tree estimation on 20 sequences would take the 20 sequences, compute the 20×20 matrix of pairwise distances between them, run NJ on the distance matrix to get a tree on the full set of 20 sequences, and then extract the quartet trees from the larger tree. Thus, global approaches have access to the full set of sequences, whereas local approaches do not.

Every phylogeny estimation method we explore, thus, can be used in either a local or global way to estimate quartet trees. The description given earlier for the local and global variants of NJ makes it clear as to how these terms are used for MP, UPGMA, and ML under three models. For the DNNs, the local version is the normal usage of the DNN (i.e., the input is a set of four aligned gap-free sequences, and the output is the quartet tree it computes). The global version of the DNNs is obtained by running DNN+QMC followed by extraction of the tree on the speci ed quartet.

2.4. Criteria

We evaluate methods with respect to tree topology error rates, computed in two different ways: quartet tree error rates and bipartition error rates. For quartet tree error, we record the percentage of the quartet trees computed that are incorrectly reconstructed. For bipartition error (applied for the 20-leaf datasets), we report the Robinson and Foulds (1981) error rates, de ned as follows (see the Appendix A1 for more details). Every edge in a tree de nes a bipartition on the leaf set, so that the Robinson-Foulds (RF) distance between two trees is the number of bipartitions that appear in one tree but not in the other. Finally, the RF error rate for an estimated tree on n leaves is its RF distance to the true (model) tree divided by $2(n-3)$.

3. RESULTS

3.1. Experiment 1: Accuracy on datasets with 200 aa

Here, we explore tree estimation accuracy on datasets with sequences having 200 aa. Experiment 1(a) examines the accuracy of the 20-leaf trees we compute, and Experiment 1(b) examines quartet tree accuracy.

3.1.1. Experiment 1(a): Accuracy on 20-leaf trees. For each training set and method, the most accurate results are obtained on the base model, and error rates then increase as the rate of evolution increases (Table 1 and Fig. 1). These results are expected and re ect the impact of the rate of evolution on phylogenetic estimation dif culty [e.g., see Liu et al. (2011)]. In addition, error rates are lowest on the Training1-based model conditions and highest on Training3-based conditions, so the Training2-based conditions are of intermediate dif culty.

Across all 12 model conditions, UPGMA has the worst accuracy, and the standard phylogeny estimation methods (NJ, ML, and MP) have the best accuracy. Thus, the DNNs only improve on UPGMA. With UPGMA set aside, the relative performance between methods depends on the speci c model condition. The accuracy of ML under IQ-TREE 2 depends on the choice of the model, with an advantage to trees estimated under models selected by ModelFinder (see Supplementary Table S4). The difference between NJ, MP, and the ML methods is generally small, but when there is a difference, it tends to favor the ML methods. Interestingly, MP is the most accurate method of all on the Harder3_1000 condition.

The comparison between the DNNs depends on the model condition, but the differences tend to be small. On the model conditions derived from Training1 and Training2, the three DNNs (DNN1+QMC, DNN2+QMC, DNN3+QMC) are all about the same in terms of accuracy (with perhaps a slight advantage to DNN2+QMC). On the model conditions derived from Training3, DNN2+QMC is slightly better than DNN1+QMC and DNN3+QMC. Thus, overall we see that DNN2+QMC seems to have a small advantage over DNN1+QMC and DNN3+QMC.

It is also worth noting how often a given method comes in rst place (or ties for rst place) in this experiment. ML using Model Finder comes in rst place 11 out of 12 times, followed by MP (7 times), ML under the WAG model (6), ML under Ghost and NJ (both at 6 times), and then DNN2+QMC (1). The remaining DNN+QMC methods never come in rst place in any model condition.

In sum, therefore, the DNN+QMC methods do not match the accuracy of ML under simple models. Further, very simple methods, such as NJ on p-distances and MP, are much more accurate than the DNN+QMC methods.

Table 1. Experiment 1: Tree Error (Median Robinson-Foulds Rates) of 20-Sequence Datasets with 200 aa Estimated Under the Different Model Conditions (20 Replicates)

Scenario	*ML WAG*	*ML GHOST*	*ML MF*	*NJ*	*MP*	*UPGMA*	*DNN1 + QMC*	*DNN2 + QMC*	*DNN3 + QMC*
Training1	**0.00**	0.06	**0.00**	**0.00**	**0.00**	0.12	0.06	0.06	0.06
Harder1_10	**0.00**	0.06	**0.00**	0.06	0.06	0.24	0.06	0.09	0.06
Harder1_100	**0.06**	**0.06**	**0.06**	0.12	**0.06**	0.24	0.18	0.18	0.15
Harder1_1000	0.53	0.41	**0.29**	0.38	0.47	0.47	0.35	0.35	0.50
Training2	0.09	0.12	**0.06**	**0.06**	**0.06**	0.18	0.12	0.09	0.09
Harder2_10	**0.06**	0.09	**0.06**	0.12	**0.06**	0.29	0.15	0.12	0.12
Harder2_100	**0.06**	**0.06**	**0.06**	**0.06**	0.10	0.24	0.12	0.12	0.12
Harder2_1000	0.21	0.24	**0.15**	0.24	0.24	0.32	0.29	0.24	0.18
Training3	**0.12**	**0.12**	**0.12**	**0.12**	**0.12**	0.26	0.18	**0.12**	0.15
Harder3_10	0.12	0.12	**0.09**	0.12	0.12	0.26	0.12	0.12	0.12
Harder3_100	0.21	**0.18**	**0.18**	0.24	**0.18**	0.53	0.35	0.26	0.35
Harder3_1000	0.44	0.35	0.47	0.38	**0.32**	0.68	0.53	0.47	0.53
Times in top place	6	4	11	4	7	0	0	1	0

Boldface values indicate top place.

NJ and UPGMA are run by using uncorrected distances. The ML methods are performed by using IQ-TREE 2 in default mode, under the indicated models.

MF, ModelFinder; MP, maximum parsimony; ML, maximum likelihood; NJ, neighbor joining; QMC, Quartets MaxCut; UPGMA, unweighted pair group method with arithmetic mean.

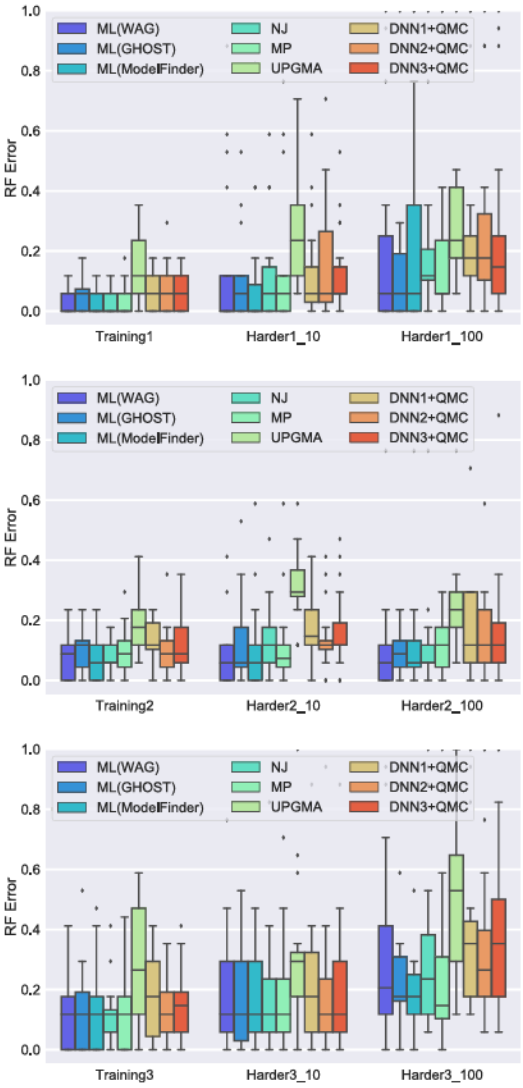

FIG. 1. Experiment 1: Global tree error on 20-leaf datasets with 200 aa.

3.1.2. Experiment 1(b): Accuracy of quartet tree methods. Performance of local quartet tree estimation methods. Table 2 shows quartet tree error rates for the local methods under the 12 model conditions. As expected, the error rates increase for all methods as the rate of evolution increases; however, many of the other trends are surprising. For example, one of the most striking findings in this study is that NJ is the best method, with the lowest median error for all model conditions. This is surprising, given that NJ is used with the simplest distance calculation, p-distances (percentage of aligned sites that are different).

Further, NJ has a consistent advantage over the DNNs, which can be substantial (e.g., Harder1_100, Harder2_100, and Harder3_100). In contrast, UPGMA has the worst accuracy, indicating that the advantage of NJ is not because distance-based methods are generally at an advantage.

Also intriguing is that ML under simpler models is more accurate than ML under more complex models, despite the fact that the simulation protocol introduced very substantial heterogeneity. For example, ML(WAG) generally has better accuracy than ML under GHOST (the most complex model) and under the model selected by ModelFinder.

Another interesting finding is that MP has better accuracy than ML(GHOST) for every model condition, and it is better than ML (MF) for 9 of the 12 model conditions. Indeed, only ML under the simplest model condition, WAG, is more accurate than MP (and even for this, MP improves on ML(WAG) on four model conditions, ties on one condition, and is never more than 2% worse).

Table 2. Experiment 1(B): Error Rates (Medians) of Quartet Trees Estimated Using Local Quartet Methods on Datasets with 200 aa Under Different Model Conditions (20 Replicates)

Scenario	*ML WAG*	*ML GHOST*	*ML MF*	*NJ*	*MP*	*UPGMA*	*DNN1*	*DNN2*	*DNN3*
Training1	**0.04**	0.10	0.08	**0.04**	0.06	0.07	0.05	0.05	0.05
Harder1_10	**0.06**	0.11	0.08	**0.06**	0.07	0.12	0.07	0.08	0.07
Harder1_100	0.18	0.28	0.18	**0.12**	0.17	0.16	0.17	0.17	0.13
Harder1_1000	0.48	0.48	0.42	**0.32**	0.43	0.34	0.36	0.36	0.35
Training2	0.06	0.13	0.09	**0.05**	0.08	0.07	0.07	0.06	0.07
Harder2_10	0.10	0.14	0.10	**0.07**	0.11	0.12	0.09	0.10	0.09
Harder2_100	0.09	0.17	0.11	**0.06**	0.10	0.13	0.09	0.11	0.09
Harder2_1000	0.26	0.38	0.26	**0.19**	0.27	0.24	0.25	0.25	0.23
Training3	0.09	0.11	0.10	**0.06**	0.08	0.14	0.09	0.08	0.08
Harder3_10	0.07	0.13	0.09	**0.06**	0.08	0.12	0.08	0.09	0.09
Harder3_100	0.18	0.21	0.19	**0.13**	0.16	0.19	0.18	0.19	0.19
Harder3_1000	0.33	0.36	0.35	**0.30**	0.33	0.41	0.34	0.33	0.35
Times in top place	2	0	0	12	0	0	0	0	0

Boldface values indicate top place.

NJ and UPGMA are run by using uncorrected distances. ML methods are performed by using IQ-TREE 2 in default mode, under the indicated models.

The three DNNs have similar accuracy across all the model conditions (and no single DNN outperforms the others) and never tie for the rst place. In essence, therefore, the DNNs are not particularly good as local quartet tree estimators, and very simple methods are strictly better.

Performance of global quartet tree estimation methods. In comparing methods as global quartet tree estimation, we also see substantial differences. The key observation is that quartet trees computed using DNN+QMC methods are not as accurate as quartet trees computed using most other global methods, with the only exception being UPGMA, which has the worst accuracy (Table 3). Further, across the 12 model conditions, ML(WAG) came in the rst place in eight conditions, NJ on p-distances and ML(MF) came in the rst place in six conditions, MP came in the rst place in ve conditions, and the three DNN+QMC methods came in the rst place in one to three model conditions. Thus, DNN+QMC methods are not anywhere near the best performing global quartet tree methods.

Comparing local and global quartet tree methods. To better understand the limitations of DNNs as quartet tree methods, we now examine the difference in accuracy between quartet trees computed using

Table 3. Experiment 1B: Median Quartet Error on the 20 Replicates of Each Model Condition Using a Global Quartet Estimation Approach on Datasets with 200 aa.

Scenario	*ML WAG*	*ML GHOST*	*ML MF*	*NJ*	*MP*	*UPGMA*	*DNN1*	*DNN2*	*DNN3*
Training1	**0.00**	0.01	**0.00**	**0.00**	**0.00**	0.05	**0.00**	0.01	0.01
Harder1_10	**0.00**	0.01	**0.00**	0.01	0.01	0.10	0.01	0.01	0.01
Harder1_100	**0.02**	0.04	0.04	0.04	0.05	0.10	0.07	0.06	0.05
Harder1_1000	0.32	0.21	0.18	0.16	0.32	0.26	0.16	**0.15**	0.27
Training2	**0.01**	**0.01**	**0.01**	**0.01**	**0.01**	0.05	0.03	**0.01**	0.03
Harder2_10	**0.01**	0.02	0.02	0.02	0.02	0.11	0.04	0.03	0.04
Harder2_100	**0.01**	**0.01**	**0.01**	**0.01**	0.04	0.09	0.04	0.05	0.03
Harder2_1000	**0.07**	0.10	0.08	0.09	0.13	0.18	0.09	0.10	0.08
Training3	**0.03**	**0.03**	**0.03**	**0.03**	**0.03**	0.09	0.04	0.04	0.04
Harder3_10	0.03	0.04	**0.02**	**0.02**	**0.02**	0.09	**0.02**	**0.02**	**0.02**
Harder3_100	0.11	0.09	0.09	0.13	**0.07**	0.17	0.11	0.11	0.12
Harder3_1000	0.22	**0.19**	0.23	**0.19**	0.22	0.46	0.29	0.20	0.24
Times in top place	8	4	6	6	5	0	2	3	1

Boldface values indicate top place.

NJ and UPGMA are run by using uncorrected distances. ML methods are performed by using IQ-TREE 2 in default mode, under the indicated models.

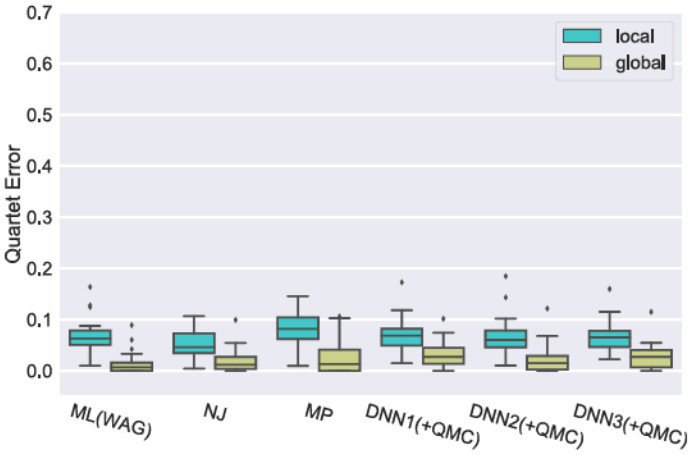

FIG. 2. Experiment 1: Comparison of quartet tree error of local and global versions of each tree estimation method on Training2 datasets with 200 aa. The local approach estimates quartet trees directly on 4-sequence alignments, whereas the global approach estimates quartets by first computing the 20-leaf tree and then each individual quartet is extracted.

local methods and quartet trees computed using global methods. We compare the best local ML method [i.e., ML(WAG)], the three DNNs, NJ, and MP; however, we omit UPGMA because of its generally poor accuracy seen in the previous experiments.

For every model condition and for every method, it is always better to use the global version of the method instead of the local version of the method (see Fig. 2 for one model condition and Supplementary Figs. S1–S3 for the other models). The difference between local and global quartet tree error rates depends on the model condition (smaller differences between these error rates under the slower rate of evolution than under higher rates of evolution) and method. Even under the lowest rate of evolution, the differences are not small: The difference in accuracy ranges from 3% (for NJ) to about 12% (for ML under the GHOST model). Under the highest rate of evolution, the difference for ML methods under all models ranges up to 27% and up to 21% for the DNN methods.

3.2. *Experiment 2: Accuracy on long sequences*

Here, we explore results on 1000 aa sequences. Examining RF error rates on the 20-leaf trees (Table 4), we see that error rates for all methods are much lower than on shorter sequences (200 aa), and that the differences between methods are reduced. Indeed, for three of the nine model conditions all methods recover the true tree exactly, and there are only two model conditions where any method has worse than 6% RF error. Thus, the model conditions we explore in Experiment 2 are much easier than the model conditions explored in Experiment 1.

However, even under these easier conditions, there are still clear differences in RF error between methods. For example, four methods (the three ML methods and MP) come in the first place in eight of the

TABLE 4. EXPERIMENT 2: TREE ERROR (MEDIAN ROBINSON-FOULDS RATES) OF 20-LEAF TREES ESTIMATED UNDER THE DIFFERENT MODEL CONDITIONS (20 REPLICATES) ON LONG SEQUENCES (1000 aa)

Scenario	*ML WAG*	*ML GHOST*	*ML MF*	*NJ*	*MP*	*DNN1 + QMC*	*DNN2 + QMC*	*DNN3 + QMC*
Training1	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Harder1_10	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Harder1_100	**0.00**	**0.00**	**0.00**	0.06	**0.00**	0.06	0.06	0.06
Training2	**0.00**	**0.00**	**0.00**	0.06	**0.00**	0.03	**0.00**	**0.00**
Harder2_10	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Harder2_100	**0.00**	**0.00**	**0.00**	0.06	**0.00**	**0.00**	0.03	**0.00**
Training3	**0.00**	**0.00**	**0.00**	**0.00**	0.01	0.06	**0.00**	**0.00**
Harder3_10	0.03	0.06	0.06	0.09	**0.00**	0.09	0.06	0.06
Harder3_100	**0.06**	**0.06**	**0.06**	0.12	**0.06**	0.15	0.15	0.18
Times in top place	8	8	8	4	8	4	5	6

Boldface values indicate top place.

NJ and UPGMA are run by using uncorrected distances. ML methods are performed by using IQ-TREE 2 in default mode, under the indicated models.

Table 5. Experiment 2: Error Rates (Medians) of Quartet Trees Estimated Using Local Quartet Methods Under the Different Model Conditions (20 Replicates) on Long Sequences (1000 aa)

Scenario	*ML WAG*	*ML GHOST*	*ML MF*	*NJ*	*MP*	*DNN1*	*DNN2*	*DNN3*
Training1	**0.01**	0.03	0.02	0.02	0.02	**0.01**	**0.01**	**0.01**
Harder1_10	**0.02**	0.05	0.03	0.06	0.03	**0.02**	**0.02**	**0.02**
Harder1_100	0.07	0.19	0.06	**0.02**	0.07	0.06	0.07	0.05
Training2	**0.02**	0.06	0.03	0.03	**0.02**	**0.02**	**0.02**	**0.02**
Harder2_10	**0.01**	0.05	0.02	0.04	0.02	**0.01**	0.02	**0.01**
Harder2_100	0.03	0.06	0.03	**0.02**	0.04	0.03	0.03	0.03
Training3	0.02	0.03	0.03	0.02	0.02	0.02	**0.01**	0.02
Harder3_10	**0.04**	0.06	0.05	0.07	0.05	**0.04**	**0.04**	**0.04**
Harder3_100	0.09	0.09	0.09	**0.03**	0.09	0.08	0.08	0.08
Times in top place	5	0	0	3	1	5	5	5

Boldface values indicate top place.

NJ is run by using uncorrected distances. ML methods are performed by using IQ-TREE 2 in default mode, under the indicated models.

nine model conditions, and the remaining methods (NJ and the three DNN+QMC methods) come in the rst place between four and six times. The three ML methods and MP also never have error rates above 6% for any model condition, whereas NJ has error rates between 9% and 12% for two model conditions, and the three DNN+QMC methods have error rates between 15% and 18% for the hardest model condition.

Thus, although these are easier model conditions and differences between the top methods are reduced, the DNN+QMC methods are still not competitive for accuracy with the other methods.

Comparing these methods as local quartet tree methods (Table 5) is also relevant. Here, we see the DNNs producing very good quartet trees, each coming in the rst place in ve of the eight model conditions; only ML(WAG) has this same good performance (and the remaining methods come in the rst place at most three times). Thus, as local quartet tree methods, when analyzing long sequences, the DNNs can provide good value.

Interestingly, we see different relative performance when comparing methods for global quartet tree accuracy (Table 6): The DNN+QMC methods come in the rst place in only four out of the nine conditions, and each of the other methods comes in at least ve times. The best in terms of global quartet tree accuracy is ML(WAG), which is best in eight of the nine conditions, followed by NJ, which is best in seven out of nine conditions, and then by MP, which is best in six out of nine conditions. Thus, as global quartet tree methods, the DNN+QMC approach clearly lags behind the other methods, even for these much easier model conditions.

Table 6. Experiment 2: Error Rates (Medians) of Quartet Trees Estimated Using Global Quartet Estimation Under the Different Model Conditions (20 Replicates) on Long Sequences (1000 aa)

Scenario	*ML WAG*	*ML GHOST*	*ML MF*	*NJ*	*MP*	*DNN1 + QMC*	*DNN2 + QMC*	*DNN3 + QMC*
Training1	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Harder1_10	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	0.01
Harder1_100	**0.00**	0.01	0.01	0.02	**0.00**	0.03	0.03	0.01
Training2	**0.00**	**0.00**	**0.00**	**0.00**	0.01	**0.00**	**0.00**	0.01
Harder2_10	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
Harder2_100	**0.00**	0.22	0.01	**0.00**	**0.00**	0.01	0.03	**0.00**
Training3	**0.00**	**0.00**	**0.00**	**0.00**	0.03	0.01	0.02	**0.00**
Harder3_10	**0.00**	0.03	0.01	0.03	**0.00**	0.01	0.01	0.01
Harder3_100	0.01	0.02	0.04	**0.00**	0.06	0.05	0.05	0.03
Times in top place	8	5	5	7	6	4	4	4

Boldface values indicate top place.

NJ is run is using uncorrected distances. ML methods are performed by using IQ-TREE 2 in default mode, under the indicated models.

4. DISCUSSION

We begin with a summary of the observations in this study. We examined methods for tree construction on either four-leaf datasets or on 20-leaf datasets, and with two sequence lengths and different rates of evolution. When constructing 20-leaf trees, we see very clear advantages to several standard tree estimation methods (notably, ML and MP) compared with DNN+QMC methods. These differences were especially pronounced for our first experiment where we examined accuracy on datasets with 200 aa per sequence, but also evident in our second experiment where we examined performance with longer sequences (1000 aa per sequence).

When used to construct quartet trees, however, the relative accuracy depended on the sequence length. For short sequences, the best method was NJ based on p-distances (i.e., uncorrected distances), an exceedingly simple method, and the DNNs were not particularly accurate. Interestingly, for long sequences (1000 aa), the DNNs were tied with ML (WAG) as the best local quartet tree estimators. Clearly, therefore, sequence length impacts the relative accuracy of DNNs and other methods, used as local quartet tree methods.

However, the most accurate quartet trees were computed by constructing trees on the full set of 20 sequences using one of the standard approaches (i.e., NJ, MP, and ML) and then inducing the quartet trees; further, the global version of quartet tree estimation was always more accurate than the local version. We observed that DNN+QMC methods are also more accurate at estimating quartet trees than the DNNs themselves, showing that the QMC amalgamation method is able, to some extent, to correct errors in the estimating quartet trees through this amalgamation step. Even so, the final quartet tree accuracy of DNN+QMC methods does not match the quartet tree accuracy of the global methods.

Comparison to Zou et al. (2020). The trends reported in Zou et al. (2020) may at first glance seem inconsistent with what we have observed. Although we mainly focused on the accuracy of methods on large (20-leaf) trees, there also seem to be differences between trends we observed for local quartet methods and the trends they observed (i.e., they concluded that DNNs have the best accuracy on simulated datasets, and are ''overall comparable'' to the existing methods on biological datasets).

An important consideration to note is that the majority of simulation conditions Zou et al. (2020) explored were all very easy due to long sequence lengths and reduced rates of evolution. However, even for the model conditions we explored that had the same rates of evolution as their main model conditions (i.e., Training1, Training2, and Training3), our results did not show their DNNs having the best accuracy, even as local quartet methods—instead we saw NJ having the best accuracy. How do we reconcile these differences in observations?

A better understanding of their DNNs on their own test data can be seen by examining results shown in their Figure 2b, which includes a wider range of sequence lengths. This figure demonstrates that sequence length has a large impact on the accuracy of their DNNs (and on other methods), and it also reveals that their DNNs were less accurate than NJ on the shortest sequences they examine (100–200 aa). Thus, the relative performance between their DNNs and NJ seems to depend on the sequence length.

Further, our results on the model conditions with low rates of evolution (i.e., our Training1, Training2, and Training3 model conditions) match their results for similar conditions. Thus, there are no difference in trends observed on simulated data in Zou et al. (2020) and our study when restricting attention to model conditions with the same properties.

It is also helpful to examine the local quartet error rates obtained by their DNNs and the other phylogeny estimation methods on five biological datasets, as reported in Table 2 of Zou et al. (2020) (summarized in our Supplementary Table S5). Analyses of the Mammals (mitochondrial) dataset produced very high error local quartet rates for all methods (ranging from 0.57 to 0.97 error), making comparisons based on this dataset uninformative and potentially misleading. Therefore, we restrict our attention to the remaining four datasets. On the red fluorescent protein dataset, NJ and MP were the best approaches, followed by RAxML.

Then, for the Mammalian genes dataset, NJ performed the best by far, followed by the ML approaches (RAxML and PhyML). On the Mammals (concatenated) dataset, MP and RAxML gave the best results, followed closely by the other non-DNN approaches, and with all the DNNs performing worst. Similarly, for the Plants (concatenated) dataset, NJ was the best approach, followed by all other methods, except for DNN3, which performed the worst.

Overall, therefore, the results reported in Zou et al. (2020) reveal that the DNNs were clearly less reliable than NJ and most other methods on the biological datasets as local quartet tree methods.

These trends suggest that these DNNs can be accurate (and competitive with other methods) for local quartet tree estimation given relatively long sequences that have evolved under suf ciently low rates of evolution but have not been shown in our experiments to provide comparable accuracy to the better standard methods for ''large'' tree estimation (measured using RF error or global quartet error).

It is possible that under other conditions, such as long branch attraction or high levels of heterotachy, DNN methods might be as accurate, or perhaps even more accurate, than standard methods, but the inferior performance of these DNN methods on biological datasets does suggest that these conditions are unlikely to be very general.

Why are DNN+QMC methods not competitive for large-tree estimation? Our study showed that DNN+QMC methods were not as accurate as simple methods for estimating 20-sequence trees; here, we examine the possible explanations for this trend.

The accuracy of these DNN+QMC methods depends on both the accuracy of the quartet trees computed using the DNNs and also the ability of QMC, the quartet amalgamation method we selected, to combine these quartet trees into an accurate tree on the full dataset. Our study clearly shows that the quartet trees, estimated independently of each other (i.e., ''local methods''), are less accurate than quartet trees induced from a tree estimated on the full dataset (i.e., ''global methods'').

This trend holds true for all the standard phylogeny estimation methods we explored, including NJ, ML, and MP. We also noted that QMC, used in conjunction with the DNNs, was able to improve the quartet score, but that this approach to quartet tree estimation was not as accurate as the truly global approach to estimating quartet trees. This seems to be an inherent limitation of DNN+QMC methods.

One might then ask whether QMC itself is a limiting factor, and that substantially better quartet amalgamation methods might exist and yield improved accuracy compared with standard methods. Although QMC is generally regarded as the best of the currently available quartet amalgamation methods, we explored the use of Quartet Puzzling (Strimmer and Von Haeseler, 1996) [the same method used in Zou et al. (2020)] to combine quartet trees computed using DNNs.

Our study found that Quartet Puzzling produced highly unresolved and much less accurate 20-leaf trees than QMC (see Supplementary Table S6). Given QMCs superiority to Quartet Puzzling and the general reception of QMC as a leading quartet amalgamation method, it seems unlikely that better amalgamation methods are available at this time.

Thus, based on our study, we hypothesize that any method that is used to construct a tree in this two-stage approach (i.e., by rst computing quartet trees independently and then merging them with a quartet amalgamation method) is unlikely to be as accurate as a good global method. This hypothesis is based on much prior literature, which has shown that big trees can, in many cases, be easier to estimate with high accuracy than small trees [e.g., see the example in Hillis (1996) and subsequent discussion in Hillis (1998); Pagel and Meade (2008); Zwickl and Hillis (2002); Nabhan and Sarkar (2012)]. This, we posit, is likely to be the main limitation of using any two-stage technique that uses DNNs to estimate small trees (here, four-leaf trees) and then amalgamation methods to combine the smaller trees.

Impact of model complexity. A very interesting trend we observed is that ML under simple models often produces more accurate trees than ML under complex models. For example, on the datasets with 200 aa, ML(GHOST) was not as accurate as ML(WAG), despite GHOST being a more complex model (and most likely a better t to the simulation model) and WAG being an extremely simple model. The advantage of ML(WAG) over ML(GHOST) is also present on the longer sequences when they are used as local quartet tree estimation methods but disappears when they are used as global methods. We conjecture that this may be due to over tting, since the advantage to the simpler model is greatest when the total amount of data is the smallest.

Related to this, we note again the very high accuracy of MP compared with the ML-based methods, when used as a local quartet tree estimation on short sequences (200 aa): MP had better accuracy than ML(GHOST) for every model condition, was better than ML(MF) for 9 of the 12 model conditions, and was only less accurate than ML(WAG) (and even there it was very close in accuracy). These trends may be related to the equivalence between MP and ML under the no-common mechanism model, as proven in Tuf ey and Steel (1997), and the relative accuracy of ML and MP on datasets simulated under heterogeneous models, as provided in Kolaczkowski and Thornton (2004).

5. CONCLUSIONS

The DNNs can be used as quartet tree estimators, and—when combined with quartet amalgamation methods, such as QMC—they can be used to construct larger trees. However, in our study, they were not as accurate at estimating 20-leaf trees as many standard phylogeny estimation methods, including MP, NJ, and ML under simple models of evolution. The failure of DNNs to provide good accuracy that is competitive with even simple methods is noteworthy and important to understand.

Our study shows that the advantage of standard methods (e.g., ML under simple models) over DNN+QMC methods is due mainly to the bene ts inherent in being able to estimate the entire tree at once rather than through a two-stage approach that rst estimates quartet trees (i.e., local quartet estimation) and then combines the quartet trees into a tree on the full dataset. The limitations of these two-stages approaches compared with global methods have been previously noted in other studies [e.g., St. John et al. (2003)], and these are closely related to the well-known bene ts produced by dense taxon sampling.

One way to address this limitation is to design DNNs to estimate much larger trees (e.g., 10-leaf trees rather than 4-leaf trees). However, such an approach would only have limited success, since the estimation of much larger trees would still require amalgamation methods (called ''supertree methods'' when the input trees are not just quartet trees). Further, taxon sampling would still bene t global methods over local methods, even if the local methods were computing 10-leaf trees. Finally, training a DNN requires a large volume of representative datasets, a challenge that is clearly already a problem for training classi ers of 4-leaf trees.

Since the number of 10-leaf trees is already more than 1,000,000, training DNNs to classify 10-taxon trees would likely be prohibitively dif cult. Thus, we predict that trying to address the limitations of this two-stage approach by constructing trees on larger subsets is unlikely to be generally successful.

Given the observed dependency on sequence length, it is possible that the DNN+QMC approach might be best suited to species tree estimation, which is based on multi-locus datasets and in some cases on large portions of whole genomes. However, multi-locus datasets evolve under an array of processes, including incomplete lineage sorting, gene duplication and loss, and gene ow, that results in different loci evolving under different tree topologies (Maddison, 1997); Degnan and Rosenberg, 2009).

Thus, to enable DNNs to be useful on multi-locus datasets, they would need to be trained on data that evolve under complex models that re ect these genome-scale processes, rather than under the models that assume that all sites evolve down a single tree topology. Although this would increase the training complexity, such an approach might well have better results, as a result of having longer sequences and also potentially more biological training data (i.e., true gene trees are rarely known, but true species trees may be reliably known for some sets of four species).

However, since standard methods can be statistically inconsistent in the presence of gene tree heterogeneity (Roch and Steel, 2015), the new DNNs would then need to be compared with phylogenomic species tree estimation methods that explicitly address gene tree heterogeneity [e.g., Heled and Drummond (2009); Mirarab et al. (2014); Richards and Kubatko (2020); Smith et al. (2020)], as these have been shown to provide improved accuracy compared with standard methods and also have strong theoretical guarantees.

However, although the use of DNNs for directly constructing phylogenies has not yet succeeded in matching the accuracy of existing methods (and the challenges to doing this seem very formidable), as discussed in Section 1, DNNs can be used in other ways to inform phylogeny estimation. Thus, although our study suggests substantial limitations for this way of using DNNs in phylogeny estimation methods (i.e., by constructing small trees using DNNs and then combining them into larger trees), there are alternative uses for DNNs that could be highly informative and bene cial in the estimation of phylogenies under realistically complex evolutionary scenarios.

DATA AND CODE AVAILABILITY

The python source code for these experiments is available at https://gitlab.engr.illinois.edu/gmartin6/simulating_quartets. The datasets used in this study are available at https://doi.org/10.13012/B2IDB-8921156_V1.

ACKNOWLEDGMENTS

The authors thank Z. Zou and J. Zhang for assistance in understanding and using their codes.

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no con icting nancial interests.

FUNDING INFORMATION

This work was supported in part by NSF grants 1513629 and 1458652 to T.W. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993), the State of Illinois, and as of December 2019, the National Geospatial-Intelligence Agency. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

SUPPLEMENTARY MATERIAL

Supplementary Figure S1
Supplementary Figure S2
Supplementary Figure S3
Supplementary Table S1
Supplementary Table S2
Supplementary Table S3
Supplementary Table S4
Supplementary Table S5
Supplementary Table S6

REFERENCES

Abadi, S., Avram, O., Rosset, S., et al. 2020. Modelteller: Model selection for optimal phylogenetic reconstruction using machine learning. *Mol. Biol. Evol.* 37, 3338–3352.

Azouri, D., Abadi, S., Mansour, Y., et al. 2021. Harnessing machine learning to guide phylogenetic-tree search algorithms. *Nat. Commun.* 12, 1–9.

Bhattacharjee, A., and Bayzid, M. S. 2020. Machine learning based imputation techniques for estimating phylogenetic trees from incomplete distance matrices. *BMC Genomics* 21, 1–14.

Crotty, S.M., Minh, B.Q., Bean, N.G., et al. 2020. GHOST: Recovering historical signal from heterotachously evolved sequence alignments. *Syst. Biol.* 69, 249–264.

Degnan, J.H., and Rosenberg, N.A. 2009. Gene tree discordance, phylogenetic inference and the multispecies coalescent. *Trends Ecol. Evol.* 24, 332–340.

Duchêne, D.A., Duchêne, S., and Ho, S.Y. 2017. New statistical criteria detect phylogenetic bias caused by compositional heterogeneity. *Mol. Biol. Evol.* 34, 1529–1534.

Felsenstein, J. 1978. Cases in which parsimony or compatibility methods will be positively misleading. *Syst. Zool.* 27, 401–410.

Foulds, L.R., and Graham, R.L. 1982. The Steiner problem in phylogeny is NP-complete. *Adv. Appl. Math.* 3, 43–49.

Heled, J., and Drummond, A.J. 2009. Bayesian inference of species trees from multilocus data. *Mol. Biol. Evol.* 27, 570–580.

Hillis, D.M. 1996. Inferring complex phylogenies. *Nature* 383, 130–131.

Hillis, D.M. 1998. Taxonomic sampling, phylogenetic accuracy, and investigator bias. *Syst. Biol.* 47, 3–8.

Huerta-Cepas, J., Serra, F., and Bork, P. 2016. ETE 3: Reconstruction, analysis, and visualization of phylogenomic data. *Mol. Biol. Evol.*33, 1635–1638.

Jermiin, L.S., Ho, S.Y., Ababneh, F., et al. 2004. The biasing effect of compositional heterogeneity on phylogenetic estimates may be underestimated. *Syst. Biol.* 53, 638–643.

Jiang, T., Kearney, P., and Li, M. 2001. A polynomial time approximation scheme for inferring evolutionary trees from quartet topologies and its application. *SIAM J. Comput.* 30, 1942–1961.

Jiang, Y., Balaban, M., Zhu, Q., et al. 2021. DEPP: Deep learning enables extending species trees using single genes. *bioRxiv.* DOI:10.1101/2021.01.22.427808, accepted to RECOMB 2021.

Kalyaanamoorthy, S., Minh, B.Q., Wong, T.K., et al. 2017. ModelFinder: Fast model selection for accurate phylogenetic estimates. *Nat. Methods* 14, 587–589.

Kolaczkowski, B., and Thornton, J.W. 2004. Performance of maximum parsimony and likelihood phylogenetics when evolution is heterogeneous. *Nature* 431, 980–984.

Kolaczkowski, B., and Thornton, J.W. 2008. A mixed branch length model of heterotachy improves phylogenetic accuracy. *Mol. Biol. Evol.* 25, 1054–1066.

Leuchtenberger, A.F., Crotty, S.M., Drucks, T., et al. 2020. Distinguishing Felsenstein zone from Farris zone using neural networks. *Mol. Biol. Evol.* 37, 3632–3641.

Liu, K., Linder, C.R., and Warnow, T. 2011. RAxML and FastTree: Comparing two methods for large-scale maximum likelihood phylogeny estimation. *PLoS One* 6, e27731.

Maddison, W.P. 1997. Gene trees in species trees. *Syst. Biol.* 46, 523–536.

Minh, B.Q., Schmidt, H.A., Chernomor, O., et al. 2020. Iq-tree 2: New models and ef cient methods for phylogenetic inference in the genomic era. *Mol. Biol. Evol.* 37, 1530–1534.

Mirarab, S., Reaz, R., Bayzid, M.S., et al. 2014. ASTRAL: Genome-scale coalescent-based species tree estimation. *Bioinformatics* 30, i541–i548.

Nabhan, A.R., and Sarkar, I.N. 2012. The impact of taxon sampling on phylogenetic inference: A review of two decades of controversy. *Brief. Bioinform.* 13, 122–134.

Naser-Khdour, S., Minh, B.Q., Zhang, W., et al. 2019. The prevalence and impact of model violations in phylogenetic analysis. *Genome Biol. Evol.* 11, 3341–3352.

Pagel, M., and Meade, A. 2008. Modelling heterotachy in phylogenetic inference by reversible-jump markov chain monte carlo. *Philos. Trans. R. Soc. B Biol. Sci.* 363, 3955–3964.

Ranwez, V., and Gascuel, O. 2001. Quartet-based phylogenetic inference: Improvements and limits. *Mol. Biol. Evol.* 18, 1103–1116.

Reaz, R., Bayzid, M.S., and Rahman, M.S. 2014. Accurate phylogenetic tree reconstruction from quartets: A heuristic approach. *PLoS One* 9, e104008.

Richards, A., and Kubatko, L. 2020. Bayesian weighted triplet and quartet methods for species tree inference. arXiv preprint arXiv:2010.06063.

Robinson, D., and Foulds, L. 1981. Comparison of phylogenetic trees. *Math. Biosci.* 53, 131–147.

Roch, S., and Steel, M. 2015. Likelihood-based tree reconstruction on a concatenation of aligned sequence data sets can be statistically inconsistent. *Theor. Popul. Biol.* 100, 56–62.

Saitou, N., and Nei, M. 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.* 4, 406–425.

Siddall, M.E. 1998. Success of parsimony in the four-taxon case: Long-branch repulsion by likelihood in the Farris zone. *Cladistics* 14, 209–220.

Smith, S.D., Pennell, M.W., Dunn, C.W., et al. 2020. Phylogenetics is the new genetics (for most of biodiversity). *Trends Ecol. Evol.* 35, 415–425.

Snir, S., and Rao, S. 2008. Quartets MaxCut: A divide and conquer quartets algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 7, 704–718.

Snir, S., Warnow, T., and Rao, S. 2008. Short quartet puzzling: A new quartet-based phylogeny reconstruction algorithm. *J. Comput. Biol.* 15, 91–103.

Sokal, R.R., and Michener, C.D. 1958. A statistical method for evaluating systematic relationships. *Univ. Kansas Sci. Bull.* 38 1409–1438.

St. John, K., Warnow, T., Moret, B.M., et al. 2003. Performance study of phylogenetic methods: (Unweighted) quartet methods and neighbor-joining. *J. Algorithms* 48, 173–193.

Steel, M. 1994. Recovering a tree from the leaf colourations it generates under a Markov model. *Appl. Math. Lett.* 7, 19–24.

Strimmer, K., and Von Haeseler, A. 1996. Quartet puzzling: A quartet maximum-likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.* 13, 964–969.

Suvorov, A., Hochuli, J., and Schrider, D.R. 2020. Accurate inference of tree topologies from multiple sequence alignments using deep learning. *Syst. Biol.* 69, 221–233.

Swofford, D.L. 2002. *PAUP: Phylogenetic Analysis Using Parsimony, Version 4.0 b10*. Sinauer Associates, Sunderland, MA.

Tavaré, S. 1986. Some probabilistic and statistical problems in the analysis of DNA sequences. *Lect. Math. Life Sci.* 17, 57–86.

Tuf ey, C., and Steel, M. 1997. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bull. Math. Biol.* 59, 581–607.

Whelan, S., and Goldman, N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol. Biol. Evol.* 18, 691–699.

White, N.D., and Braun, M.J. 2019. Extracting phylogenetic signal from phylogenomic data: Higher-level relationships of the nightbirds (Strisores). *Mol. Phylogenet. Evol.* 141, 106611.

Zou, Z., Zhang, H., Guan, Y., et al. 2020. Deep residual neural networks resolve quartet molecular phylogenies. *Mol. Biol. Evol.* 37, 1495–1507.

Zwickl, D.J., and Hillis, D.M. 2002. Increased taxon sampling greatly reduces phylogenetic error. *Syst. Biol.* 51, 588–598.

Address correspondence to:
Dr. Paul Zaharias
Department of Computer Science
University of Illinois
Urbana, IL 61801
USA

E-mail: zaharias@illinois.edu

Appendix

APPENDIX A1. ADDITIONAL DETAILS ABOUT PROGRAM COMMAND LINES

Phylogeny Estimation

The Linux version of IQ-TREE 2.0.5 was used for the maximum likelihood analyses. IQ-TREE was used in three different con gurations: Using the *WAG* substitution model, using the built-in *ModelFinder* to select a substitution model, and using the complex *GHOST* mixture model.

WAG	iq-tree2 -s <path-to-AA-alignment>	-m wag
ModelFinder	iq-tree2 -s <path-to-AA-alignment>	
GHOST	iq-tree2 -s <path-to-AA-alignment>	-m wag+FO*H4

Maximum Parsimony

We used PAUP* v.4.0a, with the following command:
set criterion = parsimony; hsearch addseq = random nreps = 1000;

By default, the branch-swapping algorithm is tree-bisection-reconnection (TBR)

Neighbor Joining

We used PAUP* v.4.0a, with the following command:
set criterion = distance; nj brlens = yes;

The distance metric used is the p-distance.

Unweighted Pair Group Method With Arithmetic Mean (UPGMA)

We used PAUP* v.4.0a, with the following command:
set criterion = distance; upgma brlens = yes;

The distance metric used is the p-distance.

Phylogenetics by Deep Learning (PhyDL)

DNN1: dnn1.py 588 QUARTET_FOLDER RESULT_FOLDER
DNN2: dnn2.py 1272 QUARTET_FOLDER RESULT_FOLDER
DNN3: dnn3.py 1098 QUARTET_FOLDER RESULT_FOLDER

Quartets Max Cut

nd-cut-Linux-64 qrtt=<path-to- le-with-quartets>
otre=<path-to-output le>

ROBINSON-FOULDS TREE ERROR

Tree estimation error was reported by using RF (Robinson and Foulds, 1981) error rates, using the ETE 3 toolkit from Huerta-Cepas et al. (2016). The:Tree: ''compare'' function with the unrooted=True argument is used to calculate the RF distance between the estimated tree and the reference.