Evaluating Branch Swapping Methods for Topology Search in Machine Learning-Augmented Phylogenetic Tree Inference

Jan Michael Yap*, Eugene Kasilag, and Camille Comia

Algorithms and Complexity Laboratory, Department of Computer Science, University of the Philippines Diliman, Quezon City, Philippines
*jcyap@up.edu.ph

Abstract. Methods for inferring phylogenetic trees such as maximum likelihood-based methods face scalability challenges due to the computational cost of evaluating candidate trees. To address this, the study evaluates the potential of integrating machine learning models with branch-swapping heuristics for guiding tree search. We assess model performance based on Spearman correlation with true likelihood rankings, as well as the relative position of the predicted best neighbor within the empirical ranking, and vice versa. Our results highlight the potential of machine learning-guided heuristics to enhance the efficiency and accuracy of phylogenetic tree inference, and extend prior work by comparing multiple heuristics beyond Subtree Pruning and Regrafting.

Keywords: phylogenetic tree inference, tree topology search, branch swapping, machine learning

1 Phylogenetic Tree Inference

Phylogenetics is a key discipline in evolutionary biology. By analyzing genetic information, researchers can infer evolutionary relationships and trace the transmission of traits, pathogens, and other biological properties from one organism to another. Phylogenetic trees are among the best tools for illustrating these evolutionary patterns, helping researchers map the flow of genes, traits, and pathogens across groups of species. In addition to establishing evolutionary relationships, phylogenetic trees are also valuable in the study of diseases and the spread of epidemics [2]. A *phylogenetic tree* is a branching diagram that describes a set of species and their common ancestors. Located at the *tips* of the tree are the set of biological sequences of species (e.g DNA, RNA, protein) being observed. *Internal nodes* represent sequences of the ancestors of the tips. The segments that join sequences together are called *branches*. Lastly, we will refer to the sole common ancestor of all the species observed from our data set as the *root* of the tree. Figure 1 provides a visual illustration of these concepts.

Phylogenetic tree inference begins with obtaining DNA, RNA, or protein sequences through experiments or publicly available databases such as Genbank[4],

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2025)*, Atlantis Highlights in Computer Sciences 24,
https://doi.org/10.2991/978-94-6239-638-8_19

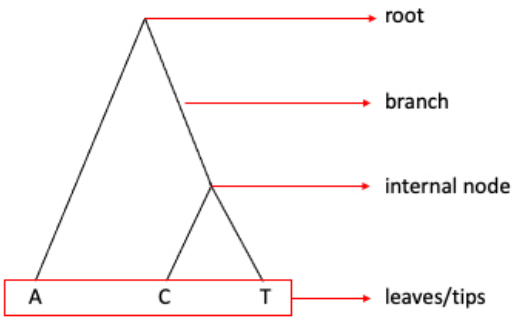


Fig. 1: An illustration of a phylogenetic tree with labeled tips/leaves, internal node, branch, and a root, illustrating ancestral relationships among species

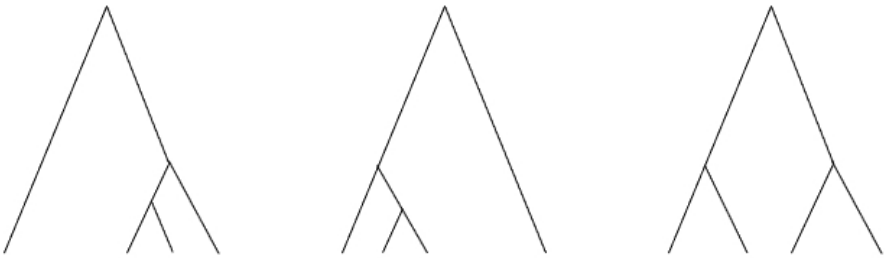


Fig. 2: Trees 1 and 2 have same topologies. All trees have the same number of tips but the branches connecting the two middle tips in Tree 3 cannot be changed to look like Trees 1 and 2 without detaching. Tree 3 has a different topology.

ENA [5], and DDBJ [11]. These sequences are then aligned and trimmed using sequence alignment methods. This is a crucial step in building the tree because it helps researchers to identify the regions of 2 or more sequences which correspond to each other. Once they are aligned, models are selected and optimized to better fit the data. The last step is the construction and assessment of the tree generated.

2 Branch Swapping for Tree Topology Search

Inferring the optimal phylogenetic tree for a given data set under any search criterion, which includes maximum likelihood, is a NP-hard problem [12]. One of the biggest, if not the biggest, challenge lies in identifying the tree topology, among all possible tree structures, that provides the global optimum for a given criterion [15]. *Tree topology* (which we will henceforth simply refer to as topology in this paper) refers to the manner in which the root, internal nodes, and

tips are connected by branches, or simply, the shape of the tree. Two topologies are different if one cannot be restructured into the other without removing a connection between two nodes or a node and a tip. Figure 2 shows trees having same and different topologies. As such brute forcefully assessing each possible tree topology for optimality is impractical because, as the number of taxa (n) increases, the number of topologies of an unrooted binary tree grows combinatorially, given by the equation:

$$\text{Number of topologies} = \frac{(2n-5)!}{2^{n-3}(n-3)!} \tag{1}$$

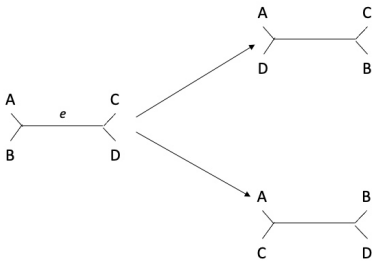
Since the exhaustive approach of computing the likelihood for all trees is not feasible, particularly for large numbers of inputs, branch-swapping techniques are used to reduce the tree space. Here, as discussed in [1], we briefly outline three heuristics commonly used in phylogenetic construction: Nearest Neighbor Interchange, Subtree-Pruning-and-Regrafting, and Tree Bisection and Reconnection.

Nearest Neighbor Interchange (NNI) is the simplest of the three heuristics we will touch on. This method simply swaps subtrees on both ends of an internal edge (See Figure 3a). Subtree-Pruning and Regrafting (SPR) is a technique that begins by selecting a subtree and pruning it with its edge. This subtree is then re-inserted into a branch of the remaining tree, creating a rearrangement of the initial tree. SPR allows for a larger scale change in the tree's topology compared to NNI because it is not restricted to a single internal edge. (See Figure 3b). Tree Bisection and Reconnection (TBR) is similar to SPR where in a branch is broken to create two separate subtrees, t_1 and t_2, but differ in the reconnection step. In SPR, we use the edge that was broken to regraft it to any edge of the remaining tree. In TBR, we can connect the two subtrees by creating a branch between any edge in each of the created subtrees, t_1 and t_2 (See Figure 3c).

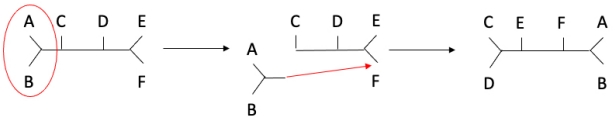
The efficiency and thoroughness of branch-swapping heuristics in phylogenetic inference are influenced by the number of possible tree rearrangements or neighbors they can generate for a given tree. As shown in Table 1, the neighborhood size for each heuristic increases with the number of taxa, but at different rates. The differences in the neighborhood size highlight a key trade-off: while heuristics with larger neighborhoods like SPR and TBR can explore tree space more thoroughly, they also require more computational resources.

3 Machine Learning-Augmented Tree Inference

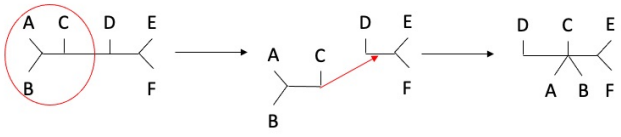
Given the computationally difficult nature of phylogenetic tree inference it is thus no surprise that various non-exact methods have been employed to at least approximate an optimal solution to the problem. One of such methods include machine learning. An important application of machine learning to phylogenetic tree inference is predicting a tree topology. In [10] for example, a neural network was used to determine the optimal topology and evolutionary model to infer



(a)



(b)



(c)

Fig. 3: Depiction of NNI, SPR, and TBR moves. In 3a, two possible NNI moves in reference to edge e can be made from the left-hand side tree. In 3b, the subtree (encircled in the figure) is pruned and re-inserted as a branch connected to F. In 3c, the subtree (encircled in the figure) is pruned at the edge (C, D). The two subtrees are then connected using the edge connecting node C to as a branch in between edge (D, E).

Table 1: The table shows the number of possible tree arrangements using NNI, SPR, and TBR, given a tree with n leaves, based on [1]

# of taxa (n)	Number of neighbors		
	NNI	**SPR**	**TBR**
10	14	182	833
20	34	1 122	10 693
30	54	2 862	41 553
40	74	5 402	105 413
50	94	8 742	214 273
n	$2n-6$	$2(n-3)(2n-7)$	$(2n-3)(n-3)^2$

phylogenetic trees on four taxon alignments at par with maximum likelihood methods.

The machine learning algorithm presented in [3] was trained to predict the ranking of each possible tree arrangement of a given tree, generated by SPR moves. This ranking represents the tendency of that topology to be the highest-scoring neighbor (i.e the one most likely to increase the likelihood). While [10] relied on simulated datasets and a small number of taxa, [3] used thousands of empirical datasets to train its random forest-based model. The training process involved over 200 million data points, where each data point is a possible SPR move of more than 4000 different empirical phylogenies. This approach led to the reduction of the large tree search space which in turn can speed up heuristic tree searches without compromising accuracy.

4 Evaluating Branch Swapping Methods for Tree Topology Search in Machine Learning-Augmented Tree Inference

Multiple phylogeny inference software tools have implemented heuristic search techniques that attempt to find the best tree without traversing the entire tree space. Aside from traditional approaches in optimizing code performance, we recognize the promise of replacing resource-heavy processes with prediction models. The nature of machine learning models use the bulk of resources during the training process which is seen to be advantageous for the phylogenetic tree inference process as long as it can maintain high quality predictions.

While the machine learning model in [3] utilized SPR to generate neighboring trees and predict those more likely to increase the likelihood of a given tree, it lacks sufficient detail on the rationale behind choosing this specific heuristic. Additionally, relying solely on SPR may restrict the search space, as other tree search methods could potentially yield more beneficial tree topologies. We thus explored alternative branch swapping techniques as substitutes for SPR, analyzing their potential to enhance tree search outcomes when used within a machine learning model. We build upon the approach and findings in [3] to explore the

potential of machine learning models for tree topology search in the MLPTI process. We extend the study by considering other branch swapping methods for implementation. The following sections describe the methodology employed in the study.

4.1 Data Collection and Preprocessing

We used the dataset originally assembled from [3], which includes 4,200 multiple sequence alignments (MSAs) collected from various databases like TreeBase[14], Selectome[13], protDB[6], and PloiDB[7]. These datasets vary in size, with 7 to 70 sequences and between 62 to 10,000 aligned positions, providing a diverse set of inference scenarios.

Preprocessing involved initial tree generation and neighboring tree generation. For each MSA, we use PhyML 3.0 [8] to generate an initial tree topology. Neighboring trees are generated using unique scripts for each branch swapping method, and each valid operation produces the candidate neighboring tree with move identifiers. After storing the tree metadata, the corresponding likelihood score is computed using RAxML-NG [9], which evaluates how well the tree explains the observed sequence data given the evolutionary model. The training datasets for SPR and NNI were generated using all 4,200 MSAs. In contrast, the training datasets for TBR and the hybrid methods were generated using a subset of 300 MSAs, selected through stratified sampling based on the number of taxa (e.g., 7–10, 11–20, ..., 61–70). This sampling approach was used to manage the large volume of samples produced while ensuring balanced representation across different tree sizes. Among the TBR and hybrid methods, NNI-SPR was processed first and generated over 100 million samples, leading to insufficient memory during model training. To address this issue, we reduced the dataset to approximately 20 million samples by retaining only 20% of the samples generated per multiple MSA.

4.2 Branch Swapping Method-Based Feature Set Construction

Training the machine learning methods to perform tree topology search involve construction of a feature set based on a branch swapping method. Branch swapping methods used in the study are SPR, NNI, TBR, and two hybrid methods: SPR-NNI and NNI-SPR. We base our features on the 19-element set used in [3], originally designed for SPR. A description of the features used by the prediction model can be seen at Table 2. These features capture global properties of the tree and local characteristics of the applied rearrangement. For SPR, we use the original feature set as-is, since it was designed for this method. For other branch swapping methods, SPR-specific features were retained, others were modified, and a few were removed or consolidated where no direct counterpart existed. For the hybrid methods, the feature set is based on the latter move in the sequence. Specifically, the SPR-NNI method uses the NNI feature set, while the NNI-SPR method uses the SPR feature set.

Table 2: Set of features from [3] that will be utilized in the SPR learning model

# **Features**	**Feature name**	**Represented action**	**Tree considered**
1	Total branch lengths	Shared for pruning and regrafting	Initial tree
2	Longest branch	Shared for pruning and regrafting	Initial tree
3–4	Branch length	Both pruning and regrafting	Initial tree
5	Topology distance from the pruned node	Regrafting only	Initial tree
6	Branch length distance from the pruned node	Regrafting only	Initial tree
7	New branch length	Regrafting only	Initial tree
8–11	Number of species	Both pruning and regrafting	Each of the subtrees in 3b
12–15	Total branch lengths	Both pruning and regrafting	Each of the subtrees in 3b
16–19	Longest branch	Both pruning and regrafting	Each of the subtrees in 3b

4.3 Machine Learning Model Selection

The machine learning models used in this study are the same as those employed in [3] : Random Forest (RF), K-Nearest Neighbors (KNN), Lasso Regression (Lasso), Bayesian Ridge Regression (BR), and Support Vector Machine (SVM). To determine which machine learning model to use in the next step, initial testing on a subset of 20 randomly selected MSAs of varying taxonomic scales and sequence lengths was done. Using this small calibration dataset, we evaluated the aforementioned five machine learning models. Model performance was evaluated using the average Spearman rank correlation coefficient between the predicted and target values across a 10-fold cross-validation procedure, similar to that done in [3]. This measures how well the predicted rankings preserve the relative order of the true target values across all SPR neighbors.

4.4 Branch Swapping Method-Based Feature Set Evaluation

Using the full dataset of 4,200 MSAs, we conducted 10-fold cross-validation for each branch swapping method-based feature set as applied to the best performing machine learning model from the previous step. A subsequent evaluation was

made on a separate validation set not used in training consisting of 50,514 samples derived from 1,000 multiple sequence alignments. In addition to the Spearman's correlation coefficient in the previous subsection, we used two additional metrics for evaluation: (1) the rank of the true best move within the predicted ranking, and (2) the rank of the predicted best move within the true ranking, as determined by full likelihood optimization. The reason for this is that while the Spearman's correlation coefficient provides a broad view of ranking performance, typical hill-climbing heuristics select only the single best neighbor for the next step. Therefore, it is also important to assess the model's ability to identify this optimal move [3].

5 Results and Discussion

5.1 Subtree Pruning and Regrafting

The performance of each machine learning model evaluated is summarized in Table 3. Random Forest exhibited the best performance (0.853), consistent with findings in [3]. We then optimized two key parameters for Random Forest: n_estimators and max_features. For n_estimators, performance increased rapidly from 0.81 (n_estimators=10) to 0.86 (n_estimators=50), with only marginal improvements beyond this point. For max_features, we found optimal performance at 0.33 (the same value used in [3]), with performance declining at both higher and lower values. This confirms that considering approximately one-third of features at each split provides optimal balance between model complexity and generalization.

We conducted 10-fold cross-validation with both the original parameters from [3] (n_estimators=70, max_features=0.33) and our calibrated parameters (n_estimators=100, max_features=0.33). Both configurations achieved impressive Spearman correlations, with our optimized parameters showing only a marginal improvement (0.927 vs 0.926). An interesting asymmetry emerged in the ranking metrics: the models were better at ranking the true best move highly among their predictions ($\tilde{4}$.1-4.2%) than correctly identifying which move would rank highest in the true likelihood ranking ($\tilde{1}$0.7%). This suggests the model excels at narrowing the search space but may require additional computation to identify the absolute best move within this refined pool.

We then tested used both the original and calibrated parameters from [3] in the validation set. As seen in Table 4, the original parameters outperformed the optimized configuration in overall correlation (91.96% vs 89.4%). However, the optimized model showed improvement in one ranking metric: it was better at ranking the true best move highly in its predictions (15.86% vs 12.58%), though both models struggled with having their top predictions align with the actual best moves (4.37% vs 4.02%)

The results suggest that machine learning approaches can achieve remarkably high predictive performance for phylogenetic tree rearrangements (Spearman correlations of 0.92+). While our hyperparameter optimization yielded minimal

Table 3: Performance of different learning models using SPR-based feature set with corresponding hyperparameters and Spearman coefficients on 20 randomly selected MSAs.

Learning Model	**Parameters**	**Spearman Coefficient**
RF	{n_estimator: 70, max_features: 0.33}	0.8529
BR	{}	0.8018
Lasso	{}	NaN
KNN	{n_neighbors: 5}	0.6369
SVM	{kernel: 'rbf', epsilon: 5e-6}	did not finish

Table 4: Performance of Random Forest Regression using SPR-based feature set on 1000 MSAs Unseen Validation Set using original and calibrated parameters

Parameters	**Metric**	**Result**
Original Parameters	Mean Spearman Correlation	0.919
	Best predicted rank in true ranking (%)	4.02
	Best true rank in predicted ranking (%)	12.58
Calibrated Parameters	Mean Spearman Correlation	0.894
	Best predicted rank in true ranking (%)	4.37
	Best true rank in predicted ranking (%)	15.86

improvements and actually resulted in poorer generalization, both configurations demonstrated practical utility in narrowing the search space for optimal tree rearrangements. The models consistently ranked truly optimal moves within their top $\sim 4-5\%$ of predictions, potentially accelerating tree search by orders of magnitude while maintaining high result quality. These findings establish a baseline for comparison with other branch swapping methods in subsequent sections.

5.2 Nearest Neighbor Interchange

To determine the feature subset to be used for the training, we applied a stepwise elimination process based on the model's feature importance scores: after each iteration, the least important feature was removed, and the model was retrained. This process continued until only one feature remained, and we then selected the feature subset that achieved the highest Spearman rank correlation during cross-validation. The other models, each using their respective optimal hyperparameters, were trained on the feature subset selected through the Random Forest model. The average Spearman correlation coefficients for these models are summarized in Table 5. As with SPR, results suggest that Random Forest Regression once again is the best machine learning model for the NNI-based feature set.

The trained model achieved a high Spearman correlation coefficient during cross-validation (Table 6), but a lower coefficient on the validation set (Table 7). In the cross-validation phase, both the rank of the best predicted move within

Table 5: Performance of different Machine Learning models using NNI-based feature set with corresponding hyperparameters and Spearman coefficients.

Learning Model	**Parameters**	**Spearman Coefficient**
RF	{n_estimator: 100, max_features: 0.33}	0.8303
BR	{alpha_1: 1e-06, alpha_2: 0.0001, lambda_1: 0.0001, lambda_2: 1e-06, tol: 0.001}	0.2896
Lasso	{alpha: 0.0001, max_iter: 1000, tol: 0.0001}	0.3506
KNN	{metric: 'euclidean', n_neighbors: 10, p: 1, weights: 'distance'}	0.7333
SVM	{kernel: 'rbf', C: 1.0, epsilon: 0.05, gamma: 'scale'}	0.7310

Table 6: Performance of Random Forest Regression using NNI-based Feature Set on 10-fold cross-validation.

Metric	**Result**
Mean Spearman Correlation	0.8303
Best predicted rank in true ranking (%)	8.0466
Best true rank in predicted ranking (%)	6.6729

Table 7: Performance of Random Forest Regression using NNI-based Feature Set on Validation Set.

Metric	**Result**
Mean Spearman Correlation	0.6914
Best predicted rank in true ranking (%)	13.4791
Best true rank in predicted ranking (%)	12.1453

the true ranking and the rank of the empirically best move within the predicted ranking fell within the 10th percentile. The difference in performance between the 10-fold cross-validation and validation test can be attributed to the smaller number of samples available for validation. For instance, the SPR method generated over 20 million samples, while the NNI method produced only around 200,000 samples—a difference of two orders of magnitude. In the cross-validation phase, both the rank of the best predicted move within the true ranking and the rank of the empirically best move within the predicted ranking fell within the 10th percentile. These metrics remained strong during validation, with ranks falling within the 15th percentile. Additionally, in the cross-validation phase, the histogram in Figure 4, particularly bins a_2 and a_3, show that the best predicted move ranked within the top 10% and 25% of the empirical ranking in 81.02% and 93.88% of the datasets, respectively. Similarly, the empirically best move appeared within the top 10% and 25% of the predicted ranking in 83.50% and 96.55% of the datasets, respectively.

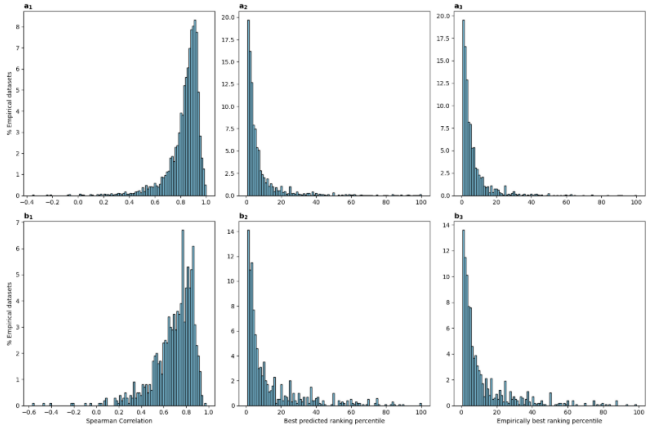


Fig. 4: A histogram showing the distribution of three performance metrics for the learning algorithm evaluated on: (a) the 4,200 starting trees using cross-validation, and (b) the 1,000 validation set starting trees. The Y-axis in both panels represents the percentage of empirical datasets falling into each accuracy score bin. The three accuracy metrics are defined as follows: (1) Spearman correlation coefficient between the predicted ranking and the true ranking of neighboring trees; (2) the percentile rank of the empirically best neighbor within the predicted ranking; (3) the percentile rank of the predicted best neighbor within the empirical ranking.

5.3 Tree Bisection and Reconnection

Initial experiments were conducted on a calibration subset of 20 datasets that were used as well for the evaluation of SPR and Hybrid methods. Contrary to theoretical predictions, TBR demonstrated significantly reduced predictive performance across multiple machine learning algorithms compared to alternative rearrangement methods as seen in Table 8. The results suggest fundamental methodological challenges in applying the current prediction framework to TBR operations. As such, subsequent evaluation did not proceed given the poor performance results for all learning models.

TBR's underperformance could possibly be due to the misalignment between its complex rearrangements and the SPR-based feature set. While some features were adapted, others had no direct TBR equivalent and may have introduced noise or irrelevant signals. TBR's large search space increases structural variance among candidate trees, making it more difficult for models trained on simpler heuristics to learn effective patterns.

5.4 Hybrid Methods

NNI-SPR The preprocessing step in the NNI-SPR pipeline generated 21,557,948 rows of data with valid predictions and labels, providing a substantial dataset for

Table 8: Performance of machine learning models using TBR-based feature set and using parameters from [3]

Learning Model	**Spearman Correlation**
RF	0.334
BR	0.348
Lasso	0.387
KNN	0.259
SVM	0.330

Table 9: Performance of machine learning models using NNI-SPR-based feature set and using parameters from [3].

Learning Model	**Spearman Correlation**
RF	0.8218
BR	0.6392
Lasso	0.7785
KNN	0.6921
SVM	Did not finish (1 day)

Table 10: Performance of Random Forest Regression using NNI-SPR-based feature set with tuned parameters (n_estimators=100 and max_features=0.4).

Metric	**Result**
Mean Spearman Correlation	0.8957
Best predicted rank in true ranking (%)	4.27
Best true rank in predicted ranking (%)	12.73

training. Table 9 shows that Random Forest Regression with default parameter values as in [3], achieved a cross-validation Spearman Correlation score of 0.82 with 10-fold cross-validation. This baseline performance establishes the potential of machine learning to learn effectively from the NNI-SPR generated data.

Subsequent tuning of the Random Forest algorithm parameters were made to attempt improving performance. We observed that increasing n _estimators generally improved performance up to a point, while the max_features parameter showed a more distinct peak in correlation. Table 10 shows that setting n_estimators set 100 and max_features value to 0.4 achieved an improved mean Spearman correlation of approximately 0.8957 on the validation set. The best predicted rank in the true ranking was around 4.27%, and the best true rank in the predicted ranking was about 12.73%.

SPR-NNI Similar to the NNI-SPR hybrid method, the SPR-NNI approach generated a large dataset during its preprocessing step, consisting of 22,822,100 data points. The dataset was trained using 10-fold cross-validation, assuming the optimal hyperparameters previously obtained for each machine learning model in the NNI experiment (see Table 5).

Table 11: Performance of machine learning models using SPR-NNI-based feature set and using parameters in Table 5.

Learning Model	**Spearman Correlation**
RF	0.8323
BR	0.8706
Lasso	0.0208
KNN	0.6626
SVM	Did not finish (1 day)

Table 12: Performance of Bayesian Ridge Regression using SPR-NNI-based feature set and using parameters in Table 5

Metric	**Result**
Mean Spearman Correlation	0.8706
Best predicted rank in true ranking (%)	3.60
Best true rank in predicted ranking (%)	3.19

Bayesian Ridge achieved the highest mean Spearman correlation coefficient among the models as seen in Table 11. Given this, its performance was further evaluated by averaging the percentile rank of the best predicted move within the true ranking, and vice versa. Both the predicted move and the empirically best move are within the top 4% of all possible moves as shown in Table 12.

5.5 Impact of Feature Set Design on Rearrangement Method Performance

The performance of machine learning-guided phylogenetic tree search in this study was determined not only by the rearrangement heuristic but also by the informativeness of the feature sets used to represent candidate moves. The SPR feature set, directly adapted from [3], incorporated detailed statistics about both the pruned subtree and its context within the tree. This comprehensive representation enabled the model to make more accurate predictions of likelihood improvements as indicated by the superior performance of SPR-guided search. In contrast, the NNI and TBR feature sets were widely based on the curated approach (specifically tailored for SPR) from [3], focusing on local and structural properties before and after each rearrangement. For NNI, the feature set captures both global and local properties of the tree. Global features describe the overall structure of the tree while local features focus on the two subtrees directly affected by each NNI move. This combination allows the model to assess both the immediate impact of the rearrangement and its effect within the broader tree context. TBR, while similarly structured, omitted features derived from the bisected subtrees, prioritizing simplicity but sacrificing evolutionary context. This omission likely contributed to increased noise and reduced the model's ability to distinguish beneficial from neutral or detrimental rearrangements, explaining TBR's weaker performance despite its broader search space.

SPR outperformed both TBR and NNI, largely because of its richer, context-aware feature set. Interestingly, we also observed that simpler heuristics like NNI can perform competitively when paired with well-designed features. Overall, the compatibility between the rearrangement method and its feature set is crucial. Features that are effective for SPR do not necessarily translate well to TBR, which underscores the need for method-specific feature engineering.

These findings underscore that the compatibility between the rearrangement method and its feature set is crucial. Context-rich, method-specific features, as demonstrated by SPR, are essential for effective machine learning guidance in phylogenetic inference.

6 Conclusions and Recommendations

This study evaluated the integration of branch swapping methods for machine learning-augmented phylogenetic tree inference. Our results show that while the construction of branch swapping method-based feature sets for machine learning-guided search is promising, optimal performance depends on the alignment between the rearrangement strategy and the feature set. Notably, more complex heuristics like TBR do not guarantee better outcomes if the feature set lacks sufficient context, whereas simpler methods like NNI can perform competitively when paired with well-designed features.

A key challenge identified is the limited applicability of features across different heuristics. Features that were effective for SPR did not translate well to TBR, highlighting the need for tailored feature engineering. Future work should focus on developing richer, context-aware features for each rearrangement method and exploring alternative learning models to further enhance the efficiency and accuracy of machine learning-augmented phylogenetic inference.

7 Acknowledgement

We extend our appreciation to the University of the Philippines Diliman College of Engineering High-Performance Computing Facility team for their technical assistance and to the generous donors for making the HPC facility available to the students and researchers of the academe.

References

1. B.L. Allen. Subtree transfer operations and their induced metrics on evolutionary trees. *Annals of Combinatorics*, 5:1–5, 2001.
2. S. Amiroch, M.S. Pradana, M.I. Irawan, and I. Mukhlash. Maximum likelihood method on the construction of phylogenetic tree for identification the spreading of sars epidemic. In *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, pages 137–141, 2018.

3. D. Azouri, S. Abadi, Y. Mansour, I. Mayrose, and T. Pupko. Harnessing machine learning to guide phylogenetic-tree search algorithms. *Nature Communications*, 12(1983), 03 2021.
4. D.A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and E.W. Sayers. Genbank. *Nucleic Acids Research*, 45(D1):D37–D42, 11 2016.
5. J. Burgin, A. Ahamed, C. Cummins, R. Devraj, K. Gueye, D. Gupta, V. Gupta, M. Haseeb, M. Ihsan, E. Ivanov, S. Jayathilaka, V.B. Kadhirvelu, M. Kumar, A. Lathi, R. Leinonen, M. Mansurova, J. McKinnon, C. O'Cathail, J. Paupério, S. Pesant, N. Rahman, G. Rinck, S. Selvakumar, S. Suman, S. Vijayaraja, P. Waheed, Z.and Woollard, D. Yuan, A. Zyoud, T. Burdett, and G. Cochrane. The european nucleotide archive in 2022. *Nucleic Acids Research*, 51(D1):D121–D125, 11 2022.
6. H. Carroll, W. Beckstead, M. Ebbert T. O'Connor, M. Clement, Q. Snell, and D. McClellan. Dna reference alignment benchmarks based on tertiary structure of encoded proteins. *Bioinformatics*, 23(19):2648–2649, 2007.
7. L. Glick, N. Sabath, T.L. Ashman, E. Goldberg, and I. Mayrose. Polyploidy and sexual system in angiosperms: is there an association? *American Journal of Botany*, 103(7):1223–1235, 2016.
8. S. Guindon, J.F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel. New algorithms and methods to estimate maximum-likelihood phylogenies: Assessing the performance of phyml 3.0. *Systematic Biology*, 59(3):307–321, 05 2010.
9. A.M. Kozlov, D. Darriba, T. Flouri, B. Morel, and A. Stamatakis. Raxml-ng: a fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*, 35(21):4453–4455, 05 2019.
10. N. Kulikov, F. Derakhshandeh, and C. Mayer. Machine learning can be as good as maximum likelihood when reconstructing phylogenetic trees and determining the best evolutionary model on four taxon alignments. *Molecular Phylogenetics and Evolution*, 200:108181, 2024.
11. J. Mashima. Dna data bank of japan (ddbj). *Nucleic acids research vol. 45,D1 (2017): D25-D31*, n.d.
12. D. Money and S. Whelan. Characterizing the phylogenetic tree-search problem. *Systematic Biology*, 61(2):228–228, 03 2012.
13. S. Moretti, B. Laurenczy, W.H. Gharib, B. Castella, A. Kuzniar, H. Schabauer, R.A. Studer, M. Valle, N. Salamin, H. Stockinger, and M. Robinson-Rechavi. Selectome update: quality control and computational improvements to a database of positive selection. *Nucleic Acids Research*, 42(D1):D917–D921, 2014.
14. W.H. Piel, M. Donoghue, and M. Sanderson. Treebase v. 2: a database of phylogenetic knowledge. Presented at e-BioSphere, 2009.
15. H.A. Schmidt and A. von Haeseler. Phylogenetic inference using maximum likelihood methods. In P. Lemey, M. Salemi, and A.M. Vandamme, editors, *The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing*, chapter 6, pages 181–209. Cambridge University Press Cambridge, 2009.