Training and Deploying Neural Networks to Accelerate Inference of Viral Transmission Dynamics

Ben Carr[1], Julianna Hays[1], Tayana Roychowdhury[1] , Jackson Wells[1], Paula Weidemüller[2], Nicola F. Müller[2]

[1]Bioinformatics and Genomics Master's Program - KCGIP, University of Oregon, [2]Division of HIV, ID and Global Medicine, University of California San Francisco

1. Introduction

Background: Human population structure influences viral transmission patterns. *Phylodynamics** uses mathematical models to reconstruct transmission histories of outbreaks with information about where a virus was sampled (*state*), its genome sequence data (*lineage*), and the time of sampling. *MASCOT* is a state-of-the-art phylodynamics software that solves sets of ordinary differential equations (*ODEs*) using a step-wise algorithm to infer viral transmission histories.

Problem: MASCOT is computationally intensive and can take weeks or months to run on larger datasets, limiting its utility in quantifying ongoing outbreak dynamics.

Goal: We aim to train neural networks that quickly infer ODE solutions and implement them in *nnMascot*, an updated version of MASCOT, for efficient inference on viral data.

**Refer to QR code for glossary of italicized words.*

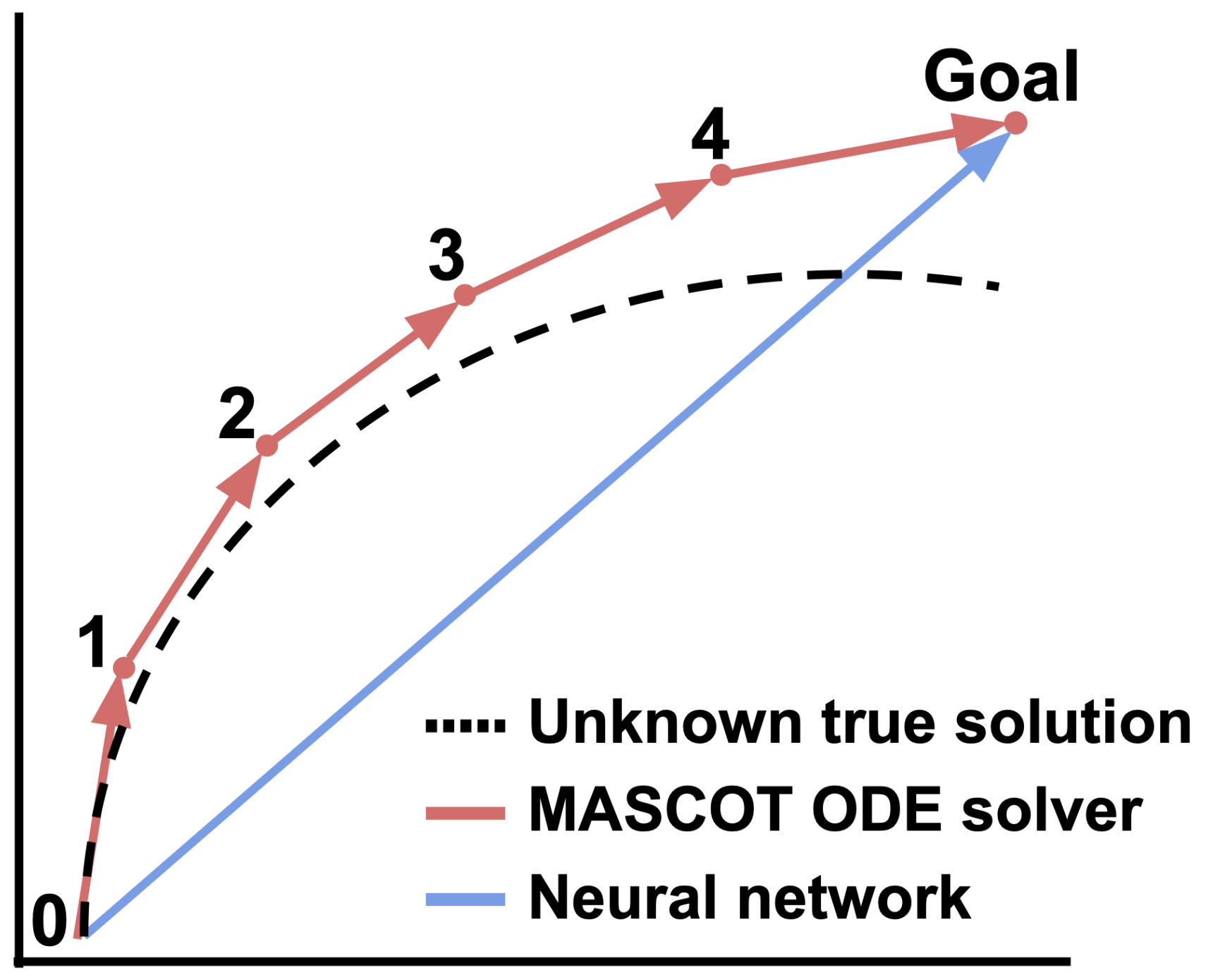

Figure 1: MASCOT algorithm versus neural network strategy. MASCOT (red) performs multiple calculations to approximate the true solution (black). A neural network (blue) can predict the solution from the input values in one step, increasing the speed.

2. Workflow

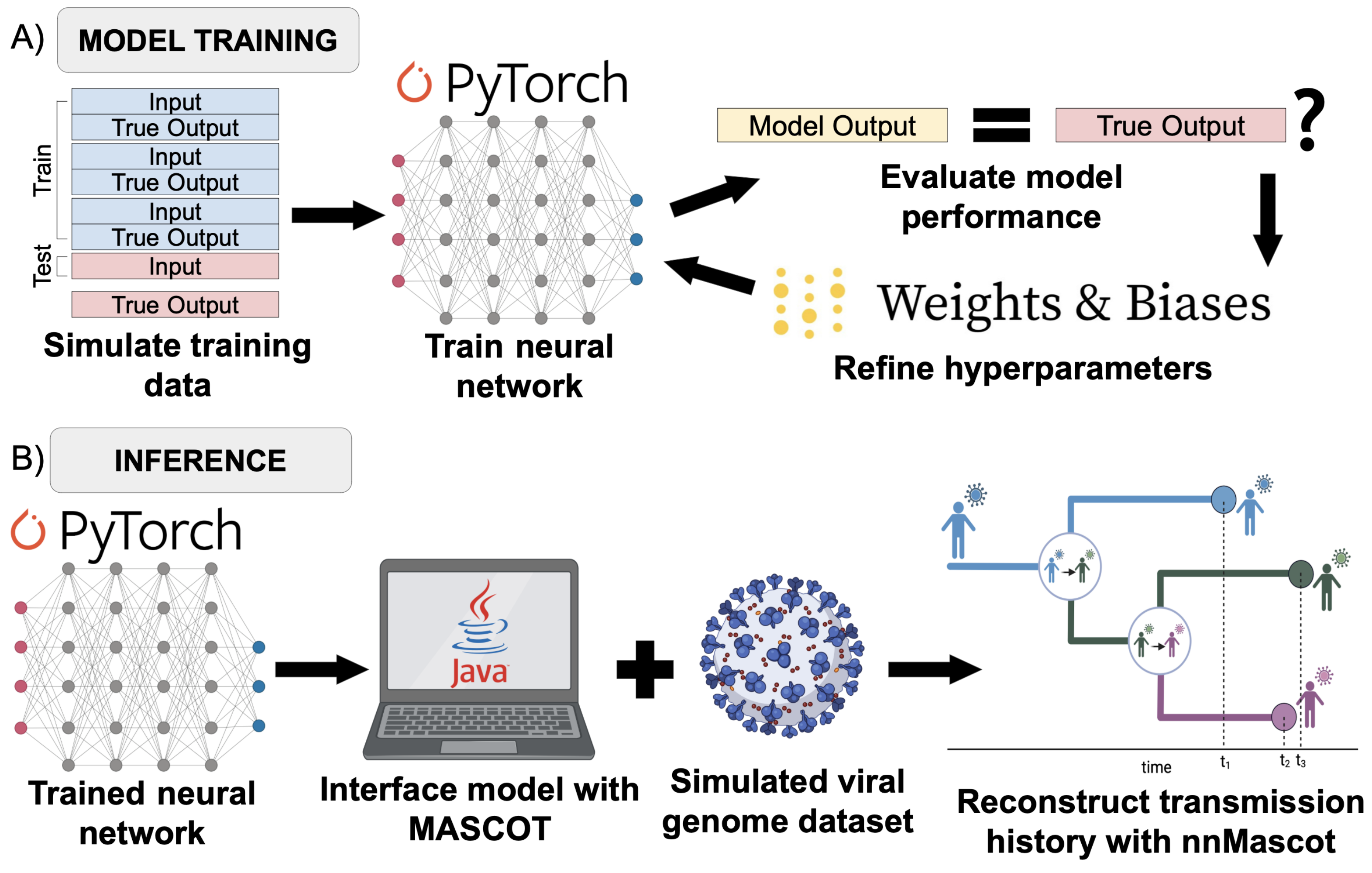

Figure 2: Workflow for deploying neural networks in ODE-solving step of Java-based MASCOT algorithm. (A) Models are trained on simulated lineage-state probabilities in PyTorch[2]; *hyperparameters* are tuned in Weights & Biases API[3]. (B) Models are implemented in Java in nnMascot, and accuracy of inference is assessed on simulated viral data.

3. Progression of Model Design to Minimize Error of Lineage-State Predictions

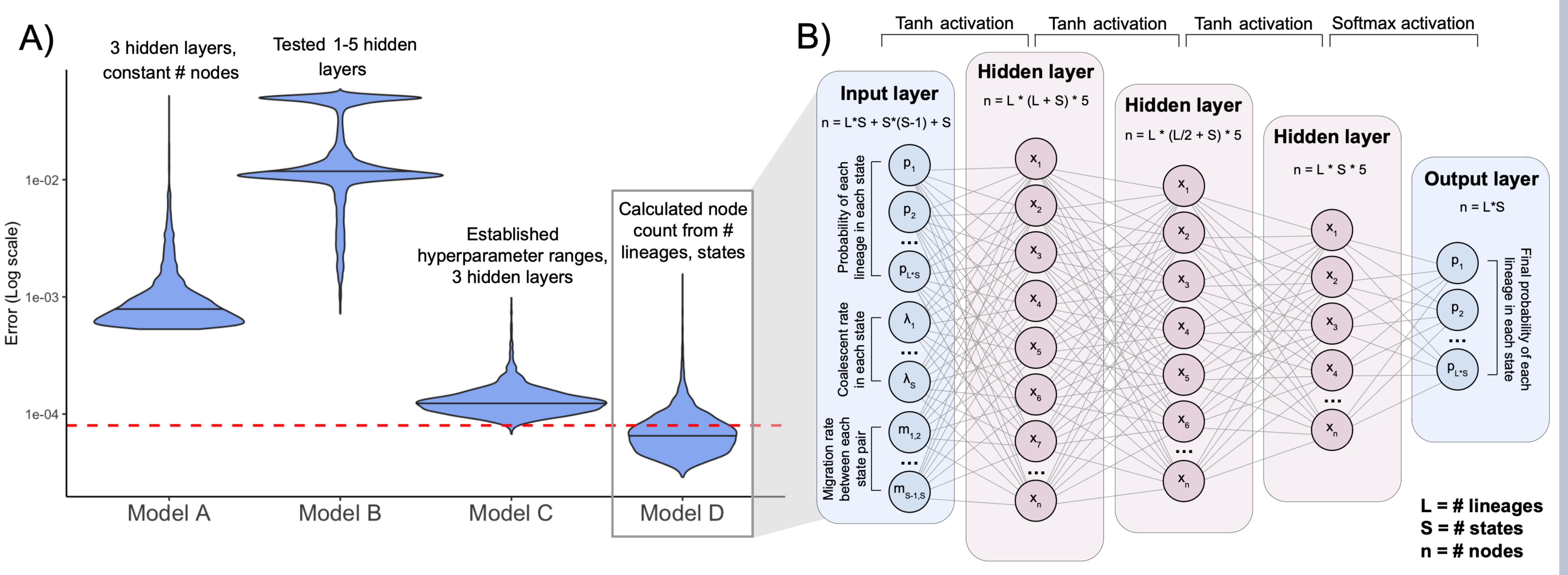

Figure 3: Impact of design modifications on error of neural network predictions, and diagram of best model architecture. (A) The y-axis represents the error (*loss*) distribution of hundreds of training runs grouped by four distinct model architectures. Black lines represent medians, and the dotted red line represents target error level. (B) Model D architecture optimized on data with 8 lineages and 4 states. The input layer contains *nodes* receiving the initial lineage-state probabilities, coalescent rates, and migration rates. Each output layer node predicts the final probability of a lineage being in a state.

4. Accuracy of nnMascot State Inference

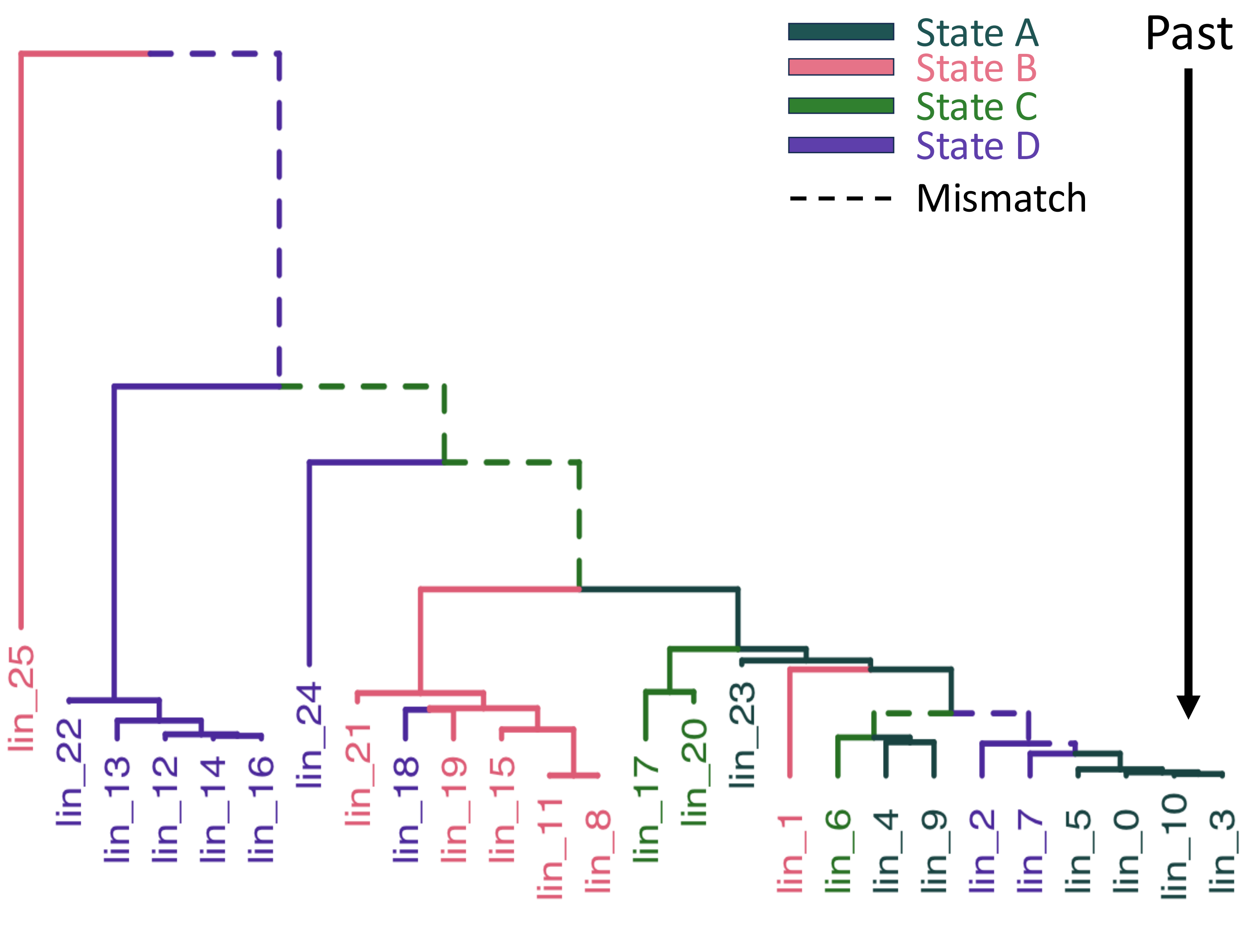

Figure 4: nnMascot versus MASCOT inference on simulated viral phylogeny with 26 lineages and 4 states. Branches are distinct lineages. Branch colors represent the state of the lineage inferred by nnMascot at the given time. Solid branches indicate nnMascot inferred the same state as MASCOT; dashed branches show discrepancies between nnMascot and MASCOT output.

5. Conclusions

- Final model predictions on training data were 10x more accurate than the initial model predictions.
- nnMascot successfully used predictions from multiple neural networks to infer state information for all lineages in simulated viral data. However, nnMascot is currently 10x slower than MASCOT.
- Transmission history inferred by nnMascot was less accurate than MASCOT output.

6. Future Work

- Train simpler model architectures to increase speed of neural network computations on real data with more lineages and states.
- Hybrid approach: use MASCOT for small number of lineages and nnMascot for >100 lineages.

7. References & Acknowledgements

We would like to acknowledge Jason Sydes for his mentorship. This work benefited from access to the University of Oregon high performance computing cluster, Talapas.