TREESBM: TREE-VALUED SCHRÖDINGER BRIDGES FOR PROTEIN EVOLUTION FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein evolution is a stochastic process in which ancestral sequences diversify into lineages that accumulate mutations and generate complex evolutionary trees. Existing protein language models, diffusion models, and flow-based generators forecast individual sequences, while phylogenetic methods reconstruct historical trees from observed descendants. The inverse problem, however, remains largely unexplored: *given a protein sequence, can we forecast the distribution of future evolutionary lineages that may emerge under selection pressures?* To answer this question, we introduce **Tree**-Valued **S**chrödinger **B**ridge **M**atching (**TreeSBM**), a generative framework for protein evolution forecasting. TreeSBM learns a tree-valued Schrödinger bridge between a reference evolutionary process and a distribution of observed evolutionary trees, transporting a root protein sequence into a distribution over future mutational histories. During inference, TreeSBM generates distributions of evolutionary trees forward in time conditioned on a single ancestral sequence, jointly modeling sequence evolution, lineage diversification, and branching topology. Across viral evolution forecasting benchmarks, TreeSBM supports realistic and high fitness evolutionary tree generation, strong future lineage forecasting, and probabilistic exploration of evolutionary spaces. On SARS-CoV-2 generated variants, TreeSBM recovers pandemic-grade threats and other observed high fitness mutations. These results establish tree-valued Schrödinger bridges as a framework for forecasting branching processes in protein evolution. For reproducibility, we host our code, data-processing scripts, and evaluation pipelines at this anonymous repository: `https://anonymous.4open.science/r/TreeSBM-D7EC`.

1 INTRODUCTION

Protein evolution is fundamentally a branching stochastic process in which ancestral sequences diversify into new lineages that accumulate mutations and form evolutionary trees. Efforts in large-scale sequencing have produced databases of protein sequences spanning viral evolution, antibody maturation, and directed evolution. These datasets have enabled protein language models, evolutionary sequence models, and mutational-effect benchmarks that predict sequence plausibility, estimate fitness landscapes, and generate individual protein sequences (Elnaggar et al., 2022; Lin et al., 2023; Madani et al., 2023; Nijkamp et al., 2023; Notin et al., 2023). Existing computational methods primarily solve two related but distinct problems. Protein generative models produce individual sequences, while phylogenetic methods infer historical trees from observed descendants (Price et al., 2010; Minh et al., 2020; Kozlov et al., 2019; Bouckaert et al., 2014). However, the prospective problem remains largely unexplored: Given an ancestral protein sequence x_0, can we forecast the distribution of future mutational trajectories and evolutionary lineages that may emerge under diversification and selection?

This forecasting setting is directly useful for infectious disease prediction. In viral evolution, one wants to anticipate future escape lineages before they dominate surveillance data for vaccine selection and rapid countermeasure development (Hie et al., 2021; Thadani et al., 2023). The desired output is a distribution over future evolutionary histories, including sequences, lineage survival, and branching topology.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Generative models provide a natural foundation for this problem because they describe how probability mass moves through state space under learned dynamics. Diffusion models, score-based models, flow matching, and discrete generative models learn stochastic or deterministic processes that transport simple distributions toward complex target distributions through successive updates in continuous or discrete state spaces (Ho et al., 2020; Song et al., 2021; Lipman et al., 2023; Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024; Gat et al., 2024; Stark et al., 2024). Schrödinger bridge formulations provide a complementary stochastic-control view, where generation is defined by the minimum-relative-entropy path measure connecting prescribed endpoint distributions (Schrödinger, 1932; Léonard, 2014; De Bortoli et al., 2021; Tong et al., 2024; Shi et al., 2023). Recent variants extend Schrödinger bridge matching to discrete peptide sequence design, branching trajectories, and interacting multi-particle systems (Goel & Chatterjee, 2026; Tang et al., 2026; 2025). Protein evolution adds a distinct challenge: the state is an evolving tree, and uncertainty exists over both mutations and lineage diversification. This requires generative models that operate directly on tree-valued state spaces.

In this work, we introduce **TreeSBM**, **Tree**-Valued **S**chrödinger **B**ridge **M**atching for protein evolution forecasting. TreeSBM learns a tree-valued Schrödinger bridge between a reference evolutionary process and an empirical distribution of observed evolutionary trees. During inference, TreeSBM generates distributions of future evolutionary trees conditioned on a single root sequence, jointly modeling sequence evolution, lineage diversification, and branching topology (Figure 1).

We summarize our main contributions as follows.

- We propose a framework for a **Tree-Valued Schrödinger Bridge**, a process for transport between a reference process and set of observed trees. We denote mechanisms for learning tree topology, branching dynamics, mutation rates, and protein specific site specifications.
- We evaluate this framework on SARS the CoV-2 spike protein, Influenza H1N1 pdm09 HA, Influenza H3N2 HA, and HIV envelope protein, demonstrating TreeSBM's advantage in escape mutation and pandemic lineage recovery over autoregressive and pLM baselines.
- We use external fitness evaluators and structural validation to show that TreeSBM mutations result in high-fitness escape variants and occur at known antigenic sites.

2 Preliminaries

2.1 Protein sequences and evolutionary trees

Let $\mathcal{X}$ denote the space of protein sequences and let

$$T = (V, E, r, \ell, X) \tag{1}$$

denote a rooted evolutionary tree, where V and E are nodes and edges, r is the root, ℓ assigns branch lengths, and $X = \{x_v\}_{v \in V}$ assigns a protein sequence to each node. The root sequence is denoted $x_r = x_0$. A dataset consists of pairs $\{(x_0^{(i)}, T^{(i)})\}_{i=1}^N$ where $x_0^{(i)}$ is an ancestral sequence and $T^{(i)}$ is an observed or reconstructed evolutionary tree rooted at that sequence. Protein evolution seeks to model the conditional distribution $p(T_{\text{future}} \mid x_0)$ where T_{future} contains future mutational trajectories, branching topology, branch lengths, and descendant sequences.

2.2 Tree-valued stochastic processes

In order to model a tree-valued branching stochastic process, we propose the following framework. Let $\mathfrak{T}$ denote the space of rooted evolutionary trees and let $\mathcal{T}_t \in \mathfrak{T}$ be a tree-valued stochastic process initialized from the root-only tree $\mathcal{T}_0 = \{x_0\}$. The process evolves through mutation, branching, lineage extinction, and branch-length updates according to transition rates $R_t(\mathcal{T}, \mathcal{T}')$. If $\mu_t(\mathcal{T})$ denotes the distribution over tree states, then its evolution is governed by the tree-valued master equation

$$\frac{d\mu_t(\mathcal{T})}{dt} = \sum_{\mathcal{T}'} \mu_t(\mathcal{T}') R_t(\mathcal{T}', \mathcal{T}) - \mu_t(\mathcal{T}) \sum_{\mathcal{T}'} R_t(\mathcal{T}, \mathcal{T}'). \tag{2}$$

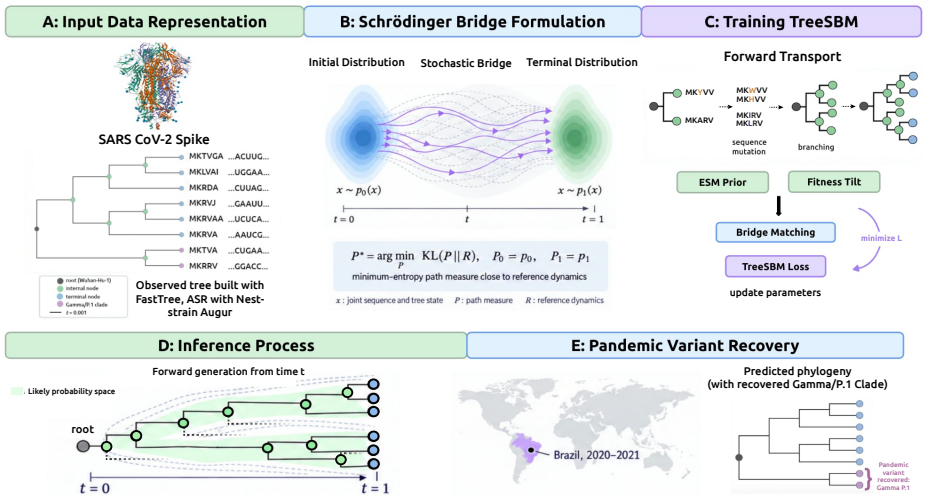

Figure 1: Overview of TreeSBM training and inference on the SARS-CoV-2 Spike protein. Panel A shows the input data representation, a FastTree phylogeny. Panel B shows the Tree-Valued Schrödinger bridge, the transport process between initial and terminal distributions. We minimize the KL divergence between the path and the reference process. Panel C shows the training process, where rateheads learn mutation, stopping, and branching events and update weights accordingly. Panel D shows the inference process, where we simulate a held out root's evolution forward in time. Panel E shows TreeSBM's capability of pandemic grade variant recovery for the Gamma variant of concern in a population in Brazil.

2.3 Schrödinger bridges on evolutionary trees

Let P^0 denote a reference evolutionary process on tree space and let $\Omega = D([0,1], \mathfrak{T})$ denote the corresponding path space. Given endpoint distributions π_0 and π_1, the Schrödinger bridge is the path measure

$$P^\star = \arg\min_P \mathrm{KL}(P\|P^0) \quad \text{s.t.} \quad P \circ \mathcal{T}_0^{-1} = \pi_0 \quad | \quad P \circ \mathcal{T}_1^{-1} = \pi_1 \tag{3}$$

For protein evolution forecasting, the initial distribution is the root-only tree $\pi_0 = \delta_{\{x_0\}}$ and the terminal distribution is the empirical distribution of evolutionary trees rooted at x_0, namely $\pi_1(T) = p_{\mathrm{data}}(T \mid x_0)$. The resulting bridge defines the minimum-control evolutionary process that transforms an ancestral sequence into an observed distribution of evolutionary trees.

2.4 Doob transforms

For Markov jump processes, the optimal bridge modifies the reference transition rates through a Doob transform. If $R_t^0(\mathcal{T}, \mathcal{T}')$ denotes the reference transition rate and $\varphi_t(\mathcal{T})$ is a positive Schrödinger potential, then the controlled dynamics are given by

$$R_t^\star(\mathcal{T}, \mathcal{T}') = R_t^0(\mathcal{T}, \mathcal{T}') \frac{\varphi_t(\mathcal{T}')}{\varphi_t(\mathcal{T})}. \tag{4}$$

The transformed process preserves the local support of the reference evolutionary dynamics while reweighting trajectories toward the desired terminal tree distribution. TreeSBM learns a parameterized approximation to these controlled dynamics and uses them to generate distributions of future mutational trees from a single ancestral protein sequence.

3 TREESBM

3.1 REFERENCE EVOLUTIONARY PROCESS

TreeSBM begins from a biologically grounded reference evolutionary process $P^0(\mathcal{T}_{0:1} \mid x_0)$ defined on the space of tree-valued trajectories. Mutation dynamics are governed by a protein-language-model, ESM-2 (Lin et al., 2023), $Q^0(x, x')$ which assigns probabilities to local mutational edits. To bias evolution toward biologically viable sequences, we incorporate exponential tilting via a fitness predictor $F(x)$ and define

$$Q_F^0(x, x') \propto Q^0(x, x') \exp(\beta F(x')).$$

Together, this pLM grounding and fitness-based tilting induce a reference path measure over evolutionary histories (Figure 1).

3.2 TREE REPRESENTATIONS

The evolutionary tree T is encoded with a frozen pretrained protein language model. The encoder uses ESM-2 (8M), mean-pooled over residue tokens to a 320-dimensional vector. Node features additionally include structural indicators and Laplacian positional encodings of the current topology. These tree representations are processed using a graph transformer.

Edge features used by the graph transformer include branch length via a learned per-head edge-weight MLP. Attention is restricted by a temporal causal mask. A stack of $L = 4$ sparse graph-transformer layers with $H = 8$ attention heads produces contextualized node and tree representations. These embeddings parameterize transition rates, branching probabilities, and stopping probabilities throughout the bridge process. Additionally, bridge time $t \in [0, 1]$ is injected by adding a projected scalar embedding to every node before attention. The tree readout is a mean-pooled graph embedding passed through a small MLP, while rate prediction uses per-node embeddings.

3.3 TREE-VALUED SCHRÖDINGER BRIDGES

Let P^0 denote the reference evolutionary process and let $p_{\mathrm{data}}(T \mid x_0)$ denote the empirical distribution of observed evolutionary trees rooted at x_0. TreeSBM defines the optimal evolutionary process as the solution to the Schrödinger bridge problem

$$P^\star = \arg\min_P \mathrm{KL}(P \| P^0) \quad \text{s.t.} \quad P \circ \mathcal{T}_0^{-1} = \delta_{\{x_0\}},\ P \circ \mathcal{T}_1^{-1} = p_{\mathrm{data}}(T \mid x_0). \tag{5}$$

The resulting bridge defines the minimum-control evolutionary process that transforms an ancestral sequence into the observed distribution of evolutionary trees. Direct computation of Schrödinger potentials is intractable on the space of evolutionary trees. Instead, TreeSBM learns a parameterized transition-rate model $R_\theta(\mathcal{T}, \mathcal{T}', t)$ through bridge matching. Given an observed tree T_1 rooted at x_0, we sample intermediate bridge states $\mathcal{T}_t \sim P^0(\mathcal{T}_t \mid \mathcal{T}_0, \mathcal{T}_1)$. Conditioned on $\mathcal{T}_t$, the model predicts local mutation, branching, and termination rates.

Separate prediction heads (MLPs) parameterize nonnegative rates for mutation, branching, branch-length updates, and termination from active-leaf embeddings h_v. Let $R_t^0(T, T')$ be the reference rate for a valid local event. The optimal Schrödinger bridge has a Doob-transformed rate where φ_t is a positive Schrödinger potential. We approximate the log-potential difference with a neural controller and parameterize

$$R_\theta(T, T', t) = R_t^0(T, T') \exp\big(g_\theta(T', t) - g_\theta(T, t)\big). \tag{6}$$

For mutations, this is implemented sitewise as

$$\log R_\theta(i, a) = \log R_0(i, a) + c_\theta(i, a), \qquad c_\theta \in \mathbb{R}^{L \times 20}, \tag{7}$$

where c_θ is predicted by the mutation head. TreeSBM models single substitution mutations only. Branching uses a Softplus Poisson intensity $\lambda_\theta(v)$, branch length uses a Softplus magnitude, and stopping uses a sigmoid probability.

In the mutation head, because h_v is a single averaged vector broadcasted across all L sites, we inject site identity to c_θ through learned positional embedding, current amino-acid embeddings, and per column Shannon entropy injected into the hidden state. To ensure that mutations occur at known antigenic regions, positions are partitioned by comparing the bridge-state amino acid a_t to the terminal target x_1, and labeling conserved and mutated sites.

Thus, the mutation term of the rate loss is

$$\mathcal{L}_{\text{rate}} = \lambda_{\text{mut}}\mathcal{L}_{\text{mut}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}. \tag{8}$$

In addition to $\mathcal{L}_{\text{rate}}$, training loss includes a weighted sum of Poisson NLL of predicted branching intensity vs. T_1 child count, MSE of predicted branch length vs. mean T_1 child branch length, BCE of stop probability vs. "no children in T_1", negative mean $\log R_0$ of current bridge amino acids (ESM fitness regularizer).

To improve long-horizon consistency and composability, we enforce a semigroup regularizer

$$\mathcal{L}_{\text{semi}} = \mathbb{E}_{0\le s<r<t\le t_{\max}}\left[d(G_{s,t}, G_{r,t}\circ G_{s,r})\right], \tag{9}$$

with complete objective

$$\mathcal{L}_{\text{TreeSBM}} = \mathcal{L}_{\text{bridge}} + \lambda_{\text{semi}}\mathcal{L}_{\text{semi}}. \tag{10}$$

3.4 Future tree generation

At inference, TreeSBM receives a single ancestral sequence x_0. Generation begins from the root-only tree. The learned transition process is then simulated forward in time,

$$\mathcal{T}_{t+\Delta t} \sim R_\theta(\mathcal{T}_t, \cdot, t),$$

producing mutation events, lineage branching events, and branch-length updates. Sampling continues until a predefined evolutionary horizon or stopping condition is reached. Repeated sampling generates a distribution of plausible future evolutionary histories (Figure 1).

4 Results

4.1 Experimental setup

To determine whether TreeSBM could recover evolutionary-tree structure and forecast unseen viral lineages, we evaluated the forward-generation process summarized in Figure 1 using held-out ancestral sequences paired with observed evolutionary trees. We assigned observed descendants to the leaves and reconstructed ancestral intermediates to internal nodes. The main experiments shown in Table 1 include temporally split influenza H3N2 and geographically split SARS-CoV-2 Spike and influenza H1N1 HA datasets. We also trained outbreak aware split models for each SARS-CoV-2 and flu models, designing splits in accordance with epidemiological history. Split details are fully discussed in Appendix A. Outbreak splits were not performed for HIV Envelope protein (Env) given its low-infectivity and modality of mutating largely within a single host.

For each held-out root, we sampled K future trees and compared TreeSBM with sequence-level mutation baselines, branching neutral models, an autoregressive tree-edit model (Xie et al., 2025a), neural tree generators, and a VAE (Xie et al., 2025b). See Appendix A for baseline details. Tables 2 and 3 define the tree-level and lineage-level metrics respectively. This design separated fidelity to observed evolutionary structure from temporal and geographic forecasting performance.

Table 1: Datasets and evaluation tasks.

Dataset	Domain	Root definition	# Trees	# Seqs	Evaluation split	Data Source
SARS-CoV-2 Spike	Viral evolution	Sequence at time t	404	104,260	Geographic	NCBI Viral Database
Influenza HA H3N2	Viral evolution	Seasonal ancestor	295	118,000	Temporal (pre-2024)	NCBI Viral Database
Influenza HA H1N1	Viral evolution	Seasonal ancestor	301	55,755	Geographic	NCBI Viral Database
HIV Env	Within host/population evolution	Early isolate	147	43,363	Geographic	NCBI Viral Database

4.2 TreeSBM generates realistic evolutionary trees

We first asked whether TreeSBM could generate realistic evolutionary-tree distributions before testing its forecasting utility. On held-out 2024 North American H3N2 roots, we compared generated and observed trees using Robinson-Foulds (RF) distance, quartet distance, branch-length Wasserstein distance, and terminal-sequence edit distance. TreeSBM achieved the lowest value for every metric, including an RF distance of 0.9814, a quartet distance of 0.6336, a branch-length Wasserstein distance of 4.45×10^{-4}, and a terminal edit distance of 0.0136 (Table 2). We therefore found that bridge matching improved topology, branch-length calibration, and terminal-sequence agreement within the same generated tree distribution.

Table 2: Tree topology evaluation on held-out 2024 North American Influenza A H3N2 roots.

Method	RF ↓	Quartet ↓	Branch W_1 ↓	Terminal edit ↓
Neutral CTMC + birth-death	0.9824	0.7110	7.22×10^{-4}	0.0148
JTT + birth-death	0.9839	0.7119	7.22×10^{-4}	0.0151
pLM mutation prior (ESM-2)	0.9823	0.7128	7.22×10^{-4}	0.0151
ARTreeFormer (topology) Xie et al. (2025a)	0.9891	0.7126	8.61×10^{-4}	0.0151
PhyloVAE (topology) Xie et al. (2025b)	0.9900	0.7160	8.12×10^{-4}	0.0151
TreeSBM (temporal split)	**0.9814**	**0.6336**	$\mathbf{4.45\times10^{-4}}$	**0.0136**

4.3 TreeSBM forecasts future lineages from a single ancestral sequence

Having established that TreeSBM generated realistic H3N2 trees, we next asked whether those trees contained lineages that emerged after the training period. We trained on sequences collected through 2023, sampled $K = 10$ or $K = 100$ trees from each held-out root, and measured the fraction of observed 2024 sequences lying within $\varepsilon = 2$ mutations of at least one generated leaf. With 100 sampled trees, TreeSBM recovered 72.5% of observed future sequences and achieved the highest mutation recall and lowest median minimum distance (Table 3). The autoregressive baseline recovered more complete clades, with clade recall of 0.300 compared with 0.100 for TreeSBM. These results showed that TreeSBM broadened sequence-level coverage of future evolution, while recovery of complete clade structure remained more difficult. Across SARS-CoV-2 Spike and HIV Env, expanded forecasting results showed modest improvements in coverage, mutation recall, and minimum distance, with slightly lower clade recall (Table 4).

Table 3: Future-lineage forecasting on H3N2. Coverage@K is the fraction of observed 2024 sequences with at least one generated leaf within $\varepsilon = 2$ mutations among K sampled trees.

Method	Cov.@10 ↑	Cov.@100 ↑	Mut. recall ↑	Clade recall ↑	Min. dist. ↓
Neutral CTMC + birth-death	0.662	0.662	0.020	0.000	2.65
pLM mutation prior	0.662	0.675	0.004	0.000	2.65
Autoregressive tree-edit (Xie et al., 2025a)	0.662	0.675	0.079	**0.300**	2.40
TreeSBM (temporal)	**0.675**	**0.725**	**0.156**	0.100	**2.24**
TreeSBM (epidemic)	**0.825**	**0.825**	**0.062**	0.167	**1.21**

We then sought to explore whether TreeSBM could transfer across geographic populations and recover biologically consequential viral substitutions. We forecast lineages in unseen regions for SARS-CoV-2 Spike, influenza H1N1 HA, influenza H3N2 HA, and the HIV Envelope protein. The evaluation included many mutation recovery and antigenicity metrics, distance to observed tree, EVEScape score difference, and conservation retention. Antigenic site recall measures per virus antigenic region recall, testing RBD for SARS-CoV-2, HA head region for flu, V-loop for HIV Env. TreeSBM achieved the highest coverage, mutation-site recall, and AA|hit and the smallest minimum distance for most viruses; it also achieved the highest antigenic-site recall for H1N1 and EVEScape differences closest to the observed mutations (Table 4). Conservation retention remained below the autoregressive baseline for all viruses, and SARS-CoV-2 antigenic-site recall remained below the autoregressive result. We therefore found that TreeSBM improved recovery of the correct mutation sites and identities across geographic populations, with a tradeoff in sequence conservation.

Table 4: Viral evolution forecasting on held-out roots. Coverage@100 is ε=2 AA. EVEscape Δ = model − GT mean official EVEscape over scored mutations. H3N2 has no pathogen-matched EVEscape tensor (—). HIV uses geo and temporal splits only (no epidemic-wave split).

Virus	Model	Coverage @100 ↑	Mutation site recall ↑	Antigenic recall ↑	AA|hit ↑	Min. distance ↓	EVEscape Δ ↑	Conservation retention ↑
SARS-CoV-2 Spike	pLM (geo)	0.775	0.048	0.000	0.017	1.425	−0.424	0.985
	Autoregressive (geo)	0.775	0.006	**0.017**	0.094	1.425	−0.371	**0.993**
	TreeSBM (geo)	0.788	0.219	0.005	**0.375**	**1.388**	−0.048	0.840
	pLM (epidemic)	**0.800**	0.329	0.015	0.088	1.725	−0.109	0.710
	Autoregressive (epidemic)	**0.800**	0.189	0.009	0.144	1.700	−0.091	0.912
	TreeSBM (epidemic)	**0.800**	**0.357**	0.013	0.110	1.725	−0.004	0.753
Influenza H1N1 HA	pLM (geo)	0.813	0.071	0.115	0.136	1.462	−0.354	0.945
	Autoregressive (geo)	0.813	0.006	0.000	0.008	1.462	−0.297	**0.998**
	TreeSBM (geo)	0.825	0.241	0.163	0.618	1.388	−0.138	0.952
	pLM (epidemic)	**0.925**	0.000	0.056	—	0.838	−0.862	0.984
	Autoregressive (epidemic)	0.175	0.008	0.011	0.333	3.225	−0.686	0.994
	TreeSBM (epidemic)	0.912	**0.274**	**0.295**	**0.701**	**0.800**	−0.554	0.879
Influenza H3N2 HA	pLM (temporal)	0.675	0.059	0.013	0.111	2.650	—	0.958
	Autoregressive (temporal)	0.675	0.006	0.116	0.404	2.400	—	**0.999**
	TreeSBM (temporal)	0.725	0.253	0.204	0.439	2.238	—	0.918
	pLM (epidemic)	**1.000**	0.012	0.000	0.000	**0.578**	—	0.985
	Autoregressive (epidemic)	0.225	0.021	0.130	0.375	5.860	—	0.992
	TreeSBM (epidemic)	0.825	**0.329**	**0.343**	**0.606**	1.275	—	0.866
HIV-1 Env	pLM (geo)	**0.188**	**0.614**	0.168	0.075	63.9	−0.053	0.414
	Autoregressive (geo)	**0.188**	0.098	0.174	0.098	63	−0.003	0.938
	TreeSBM (geo)	**0.188**	0.373	0.134	0.337	**63.1**	−0.273	0.893
	pLM (temporal)	0.000	0.597	0.167	0.077	182	−0.034	0.424
	Autoregressive (temporal)	0.000	0.128	**0.190**	0.131	182	+0.005	**0.952**
	TreeSBM (temporal)	0.000	0.320	0.099	**0.359**	182	−0.220	0.870

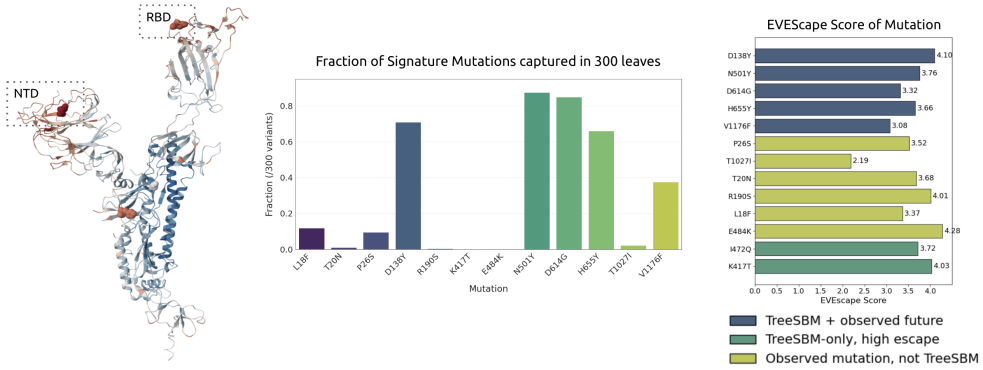

Figure 2: TreeSBM-generated mutations exhibit high predicted immune escape. Left: generated mutations mapped to known antigenic regions of SARS-CoV-2 Spike. Middle: recovery frequency of known Gamma escape mutations across $n = 300$ generated leaves. Right: EVEScape scores for mutations observed along the historical and TreeSBM-generated evolutionary trajectories.

Finally, to determine whether the substitutions represent plausible immune-escape events, we mapped generated SARS-CoV-2 mutations to antigenic regions in and evaluate predicted escape with EVEscape. TreeSBM recovered multiple substitutions observed in subsequent viral evolution, including mutations in antigenic regions with high predicted immune-escape scores Figure 2.

4.4 TreeSBM can recover SARS-CoV-2 Pandemic Grade Variants of Concern

We next ask whether TreeSBM forecasts recover the Spike substitutions that historically defined named SARS-CoV-2 variants of concern (VOC) and interest (VOI). For each lineage we take an ancestral Spike sequence from a local tree rooted in that lineage's geographic origin near its documented emergence window, generate $K=10$ descendant trees with the epidemic-split COVID model and score leaves only against that lineage's most signature mutation (Table A.2). We compare against the previous best mut-recovery COVID checkpoint under an identical protocol.

We used the following metrics to evaluate VOC results. Exact mutational recall is the fraction of signature substitutions present on at least one generated leaf. Acquired recall counts only substitutions that are absent from the ancestral root (the primary prospective KPI). Defining-bundle hit indicates whether any leaf realizes the lineage's small co-occurring defining set. All defining bundles are listed in Table A.2). Lastly, site recall measures any amino-acid change at signature sites relative to Wuhan. Across the six selected ancestors, the epidemic model improves exact recall (0.841 vs. 0.713), acquired recall (0.403 vs. 0.277), and defining-bundle hit rate (0.867 vs. 0.567) relative to the prior best checkpoint (Table 5).

Table 5: VOC recovery on the selected ancestor for each population (mean over 10 generated trees). *Epidemic* is the epidemic-split COVID TreeSBM model;

		Epidemic split model			Population temporal split best model		
VOC	Origin	Exact	Acquired	Bundle hit	Exact	Acquired	Bundle hit
Alpha	United Kingdom	0.600	0.314	0.900	0.443	0.157	0.300
Beta	South Africa	0.843	0.143	0.900	0.743	0.043	1.000
Delta	India	0.917	0.417	0.500	0.867	0.367	0.500
Epsilon	United States	1.000	0.750	0.900	0.875	0.625	0.400
Iota	United States	0.850	0.517	1.000	0.617	0.283	0.200
Mu	Colombia	0.833	0.278	1.000	0.733	0.189	1.000

Alpha (B.1.1.7), United Kingdom. The selected UK ancestor carries only D614G and P681H. The epidemic model recovers the high-EVEscape RBD mark N501Y (1.062) in all 10 generated trees (prior best: 4 of 10) and realizes the N501Y+P681H defining bundle in 9 of 10 trees (bundle hit 0.90 vs. 0.30). Acquired recall is 0.314. Internal marks S982A and A570D remain rare, consistent with their weaker escape annotation (unscored outside the RBD matrix).

Beta (B.1.351), South Africa. This ancestor already carries the full RBD triple K417N (1.355), E484K (1.775), and N501Y (1.062), so acquired recall is necessarily low (0.143 epidemic / 0.043 prior best). The remaining prospective NTD marks D80A and D215G each appear in 5 of 10 epidemic trees versus 1–2 of 10 for the prior best model. We treat Beta primarily as a calibration case: high exact recall without high acquired recall does not imply de novo discovery of the RBD triple.

Delta (B.1.617.2), India. The ancestor already has L452R (1.555), D614G, and P681R. The outstanding historical step is T478K (EVEscape **1.977**, the highest in this panel), recovered in 5 of 10 epidemic trees; the defining-bundle hit rate is 0.50. Exact and acquired recall are 0.917 and 0.417, with T19R and D950N present in all 10 epidemic trees.

Epsilon (B.1.427/429), United States. The root in this case contains only a D614G mutation. The epidemic model attains exact recall 1.000, acquired recall 0.750, and bundle hit 0.900, recovering S13I, W152C, and L452R (1.555) in all 10 trees. The prior best model reaches the L452R+W152C bundle in only 4 of 10 trees. This is the strongest single-population demonstration in the panel, showing strong recovery of the pandemic grade epsilon variant.

Iota (B.1.526), United States. Iota is an E484K (1.775)-centered New York lineage. The epidemic model places E484K on a leaf in all 10 trees (bundle hit 1.0) versus 2 of 10 for the prior best model (bundle hit 0.2), with acquired recall 0.517 vs. 0.283. Recovering this high-EVEscape immune-escape substitution from a non-E484K ancestor is direct evidence that generated trajectories intersect the historically selected Iota path.

Mu (B.1.621), Colombia. Mu's distinctive RBD mark R346K (1.307) is recovered in only 1 of 10 epidemic trees and in none for the prior best model, despite high exact recall (0.833) driven by E484K and N501Y already present on the ancestor. We report Mu as a negative control: bundle hits inherited from the root must not be read as discovery of R346K.

Overall, epidemic-aware training improves recovery of Spike mutations relative to the prior best COVID checkpoint, with Epsilon and Iota as the clearest prospective successes and Mu's missing R346K as an explicit failure mode.

4.5 Protein language model reference processes affect forecasting performance

TreeSBM separates the reference evolutionary process from the learned bridge control. We therefore evaluated different pLMs as mutation priors in Table 6 comparing pLM references with and without fitness weighting. TreeSBM default remained the strongest for mutation recall, with ESM-C (EvolutionaryScale, 2024) remaining a strong alternative based on other metrics.

Table 6: Reference process ablation. We compare different mutation priors and fitness-weighted variants. TreeSBM uses each reference inside the same bridge-matching framework.

Reference process	Fitness weight	Tree-KL ↓	Coverage@100 ↑	Mut. recall ↑	pLM NLL ↓
JTT/WAG/LG substitution	no	0.693	0.775	0.023	0.471
ESM-2-650M	no	0.693	0.775	0.018	0.445
ESM-2-650M	yes	0.693	0.775	0.001	0.441
ESM-C	no	0.693	0.775	0.035	0.461
ESM-C	yes	0.693	0.775	0.036	0.427
ProGen2	yes	0.693	0.775	0.012	0.454
TreeSBM default	yes	0.693	0.775	0.038	0.446

We next ablated the components of TreeSBM that distinguish it from a reference simulator or an autoregressive tree generator. To do this, we removed bridge matching, fitness weighting, sequence-dependent branching, branch-length modeling, internal sequence modeling, and tree-context encoding. As can be seen in Table 7, the full model shows the highest accuracy for mutation prediction, demonstrating that all of the aforementioned components are critical to its performance.

Table 7: Core TreeSBM ablations. The table reports sequence and forecasting metrics on held out SARS-CoV-2 roots to separate effects of bridge control, reference processes, and tree representation.

Variant	Tree-KL ↓	Cons. retention ↑	AA | hit ↑	Coverage@100 ↑	Mut. recall ↑	pLM NLL ↓
TreeSBM without bridge matching	0.693	0.876	0.087	0.775	0.008	0.447
TreeSBM without sequence-dependent branching	0.693	0.824	0.247	0.775	0.045	0.454
TreeSBM without branch-length head	0.693	0.839	0.375	0.775	0.08	0.457
TreeSBM without tree-context encoder	0.693	0.845	0.264	0.775	0.084	0.456
TreeSBM without per site entropy	0.693	0.835	0.312	0.775	0.084	0.457
TreeSBM full	0.693	0.839	0.375	0.775	0.08	0.457

5 Discussion and Limitations

TreeSBM generates future evolutionary histories from one ancestral protein by combining a pLM reference process with tree-valued bridge matching over mutation, branching, branch lengths, termination, and uncertainty. It improves topology and sequence agreement on held-out H3N2 viral trajectories, future-lineage coverage and mutation recovery, and geographic transfer for SARS-CoV-2 and H1N1. TreeSBM-generated SARS-CoV-2 trees also recovered known pandemic-grade trajectories. Because training phylogenies come from unevenly sampled sequences, however, topology, ancestral-reconstruction, and collection-time errors all may affect results. Our current process also omits recombination, reassortment, large insertions/deletions, host-specific selection, and uncertainty in reconstructed trees. As such, future work will develop recombination-aware graph histories, propagate phylogenetic uncertainty, and incorporate experimentally calibrated fitness and immune-escape measurements. In total, we envision TreeSBM supporting both prospective surveillance as well as development of vaccines and therapeutics robust to forecasted viral diversification.

6 AI USE STATEMENT

AI assistance was limited to minor language editing. Attempts to obtain broader assistance were restricted by the tools' safety policies due to the sensitive nature of the work. The authors reviewed all AI-assisted edits and take responsibility for the manuscript.

7 REPRODUCIBILITY STATEMENT

Our code, data-processing scripts, and evaluation pipelines are available at our anonymous repository: `https://anonymous.4open.science/r/TreeSBM-D7EC`. The manuscript and appendix describe dataset construction, evaluation splits, model hyperparameters, and sampling procedures.

8 ETHICS STATEMENT

This work aims to support viral surveillance and pandemic preparedness through computational forecasting of protein evolution. We recognize that forecasts of viral mutations may have dual-use implications and require careful consideration when sharing models and generated sequences. Predicted fitness and immune escape remain computational estimates and do not establish infectivity, transmissibility, or pathogenicity. Public-health applications require independent validation and appropriate institutional oversight.

REFERENCES

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Remco Bouckaert, Joseph Heled, Denise Kühnert, Tim Vaughan, Chieh-Hsi Wu, Dong Xie, Marc A. Suchard, Andrew Rambaut, and Alexei J. Drummond. BEAST 2: A software platform for bayesian evolutionary analysis. *PLOS Computational Biology*, 10(4):e1003537, 2014. doi: 10.1371/journal.pcbi.1003537.

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17695–17709, 2021.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7112–7127, 2022. doi: 10.1109/TPAMI.2021.3095381.

EvolutionaryScale. ESM Cambrian: Revealing the mysteries of proteins with unsupervised learning, 2024. URL `https://www.evolutionaryscale.ai/blog/esm-cambrian`. Blog post; optional citation for ESM-C software/model use.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *Advances in Neural Information Processing Systems*, volume 37, pp. 133345–133385, 2024.

Shrey Goel and Pranam Chatterjee. Minimal-action discrete Schrödinger bridge matching for peptide sequence design. In *ICLR 2026 2nd Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*, 2026. URL `https://openreview.net/forum?id=94KB26ww3D`.

James Hadfield, Colin Megill, Sidney M. Bell, John Huddleston, Barney Potter, Charlton Callender, Pavel Sagulenko, Trevor Bedford, and Richard A. Neher. Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics*, 34(23):4121–4123, 2018. doi: 10.1093/bioinformatics/bty407.

Brian Hie, Ellen D. Zhong, Bonnie Berger, and Bryan Bryson. Learning the language of viral evolution and escape. *Science*, 371(6526):284–288, 2021. doi: 10.1126/science.abd7331.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Alexey M. Kozlov, Diego Darriba, Tomáš Flouri, Benoit Morel, and Alexandros Stamatakis. RAxML-NG: A fast, scalable and user-friendly tool for maximum likelihood phylogenetic inference. *Bioinformatics*, 35(21):4453–4455, 2019. doi: 10.1093/bioinformatics/btz305.

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete and Continuous Dynamical Systems - Series A*, 34(4):1533–1574, 2014. doi: 10.3934/dcds.2014.34.1533.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Naeem Z. Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41:1099–1106, 2023. doi: 10.1038/s41587-022-01618-2.

Bui Quang Minh, Heiko A. Schmidt, Olga Chernomor, Dominik Schrempf, Michael D. Woodhams, Arndt von Haeseler, and Robert Lanfear. IQ-TREE 2: New models and efficient methods for phylogenetic inference in the genomic era. *Molecular Biology and Evolution*, 37(5):1530–1534, 2020. doi: 10.1093/molbev/msaa015.

Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, 2023. doi: 10.1016/j.cels.2023.10.002.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Jonathan Marchena Hurtado, Aidan N. Gomez, Debora S. Marks, and Yarin Gal. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/cac723e5ff29f65e3fcbb0739ae91bee-Abstract-Datasets_and_Benchmarks.html.

Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. FastTree 2: Approximately maximum-likelihood trees for large alignments. *PLOS ONE*, 5(3):e9490, 2010. doi: 10.1371/journal.pone.0009490.

Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M. Rush, Yair Schiff, Justin T. Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'Institut Henri Poincaré*, volume 2, pp. 269–310, 1932.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, pp. 103131–103167, 2024.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion Schrödinger bridge matching. In *Advances in Neural Information Processing Systems*, volume 36, pp. 62183–62223, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=PxTIG12RRHS.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=RjIYn7ai2i.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Entangled Schrödinger bridge matching. In *2nd edition of Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. URL https://openreview.net/forum?id=1XUYOBebyR.

Sophia Tang, Yinuo Zhang, Alexander Tong, and Pranam Chatterjee. Branched Schrödinger bridge matching. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ctq8BfUXWz.

Nikhil N. Thadani, Sofia Gurev, Pascal Notin, et al. Learning from prepandemic data to forecast viral escape. *Nature*, 622:818–825, 2023. doi: 10.1038/s41586-023-06617-0.

Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free Schrödinger bridges via score and flow matching. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 1279–1287. PMLR, 2024. URL https://proceedings.mlr.press/v238/tong24a.html.

Tianyu Xie, Yicong Mao, and Cheng Zhang. ARTreeFormer: A faster attention-based autoregressive model for phylogenetic inference. *PLOS Computational Biology*, 21(12):e1013768, 2025a. doi: 10.1371/journal.pcbi.1013768.

Tianyu Xie, Harry Richman, Jiansi Gao, Frederick A. IV Matsen, and Cheng Zhang. PhyloVAE: Unsupervised learning of phylogenetic trees via variational autoencoders. In *Proceedings of the International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=Z8TglKXDWm.

Appendix

A Additional Experimental Details

A.1 Dataset construction from Newick trees

We parse each FastTree-generated Newick tree into topology, branch lengths, and leaf labels. Root sequences are assigned from known ancestral sequences, earliest sampled sequences, inferred family ancestors, germline antibody sequences, or wild-type directed-evolution parents, depending on the dataset. Table A.1 summarizes how roots, leaves, branch lengths, and internal node sequences are assigned.

Table A.1: Dataset construction and tree processing.

Dataset	Root assignment	Leaf sequences	Internal sequences	Branch lengths	Filtering
SARS-CoV-2 Spike	earliest sequence / time root	observed variants	ASR	FastTree	date, lineage, completeness
Influenza H3N2/H1N1	seasonal ancestor	observed isolates	ASR	FastTree	season, subtype
HIV Env	founder / early isolate	observed descendants	ASR	FastTree	patient / clade

A.2 Evaluation splits

We collected sequences and sampling dates from the NCBI Viral Database by population and time interval. We then aligned those sequences using MAFFT and construct bifurcating trees using FastTree, and rooted these trees using Nextstrain Augur library and conduct ancestral sequence reconstruction by sampling date using Nextstrain Treetime (Hadfield et al., 2018).

The data splits were performed in the following manner:

- **Geographic holdout** assigns whole populations in different locations to train/val/test so evaluation probes spatial generalization. HIV Env uses a greedy 80/10/10 assignment over geographic units.
- **Temporal holdout** keeps the same within-location, date-ordered grouping but cuts by collection year so test trees are strictly future relative to training (COVID calendar years train $\leq$2022 / val 2023 / test 2024–2025; H3N2 train $\leq$2022 / val 2023 / test 2024; H1N1 train $\leq$2023 / val 2024 / test 2025; HIV Env train $\leq$2014 / val 2015 / test $\geq$2016).
- **Epidemic-aware holdout** retain those same temporal cutoffs but redefine each tree as a biologically coherent outbreak unit rather than an arbitrary date chunk: COVID trees are (country $\times$ Nextclade clade), falling back to (country $\times$ year) when clade labels are unavailable; H3N2 trees are Northern-hemisphere flu seasons (Oct–Sep), with high volume seasons further date-chunked; H1N1 trees are (location $\times$ flu season). Large pools are date-sorted and chunked to a target group size (typically 300 leaves; groups below a minimum size are dropped), and groups never straddle a temporal cutoff. Models are trained only on the corresponding train trees (with the held-out val set for early stopping/selection). Reported forecasting performance reflects generalization across space, time, or both, never reconstruction of leaves seen during training.

A.3 Baseline Implementation Methods

TreeSBM is a forward generator from a root sequence to a joint sequence–tree object. Several published baselines do not expose this interface, so we adapt them to the same evaluation protocol (target leaf count N, root-to-leaf diversity H, and shared metrics). Neutral and empirical substitution models (uniform / JTT–WAG–LG CTMCs) are paired with a birth–death topology sampler fitted on training trees (NeutralBD / EmpiricalBD). The protein language model (pLM) baseline uses the same birth–death topologies but substitutes amino acids along branches with ESM-2 site-conditional rates rather than an empirical matrix. ARTreeFormer Xie et al. (2025a) is an autoregressive model of phylogenetic topology density for fixed labeled taxa; it does not take a novel ancestral sequence as input or emit sequences and branch lengths. We therefore use it as an unconditional topology prior:

topologies are sampled from a released ARTreeFormer checkpoint, after which branch lengths and sequences are assigned by shared adapters identical to those used for other topology-only methods (branch lengths matched to H; sequences evolved from the evaluation root with a common substitution adapter). This ARTreeFormer-adapted construction attributes only tree shape to ARTreeFormer and prevents sequence quality from being credited to the topology model. PhyloVAE Xie et al. (2025b) is adapted analogously.

A.4 Defining Mutations for Analyzed Variants of Concern

Table A.2: VOC/VOI panel: origin populations, pandemic signatures, and defining bundles. EVEscape values (Spike RBD matrix) are shown for scored substitutions.

WHO label	Pango lineage	Origin / emergence window	Defining bundle (EVEscape)
Alpha	B.1.1.7	United Kingdom, Sep 2020–Jan 2021	N501Y (1.062), P681H
Beta	B.1.351	South Africa, Aug 2020–Jan 2021	K417N (1.355), E484K (1.775), N501Y (1.062)
Delta	B.1.617.2	India, Oct 2020–Jun 2021	L452R (1.555), T478K (1.977)
Epsilon	B.1.427/429	United States, Jul 2020–Mar 2021	L452R (1.555), W152C
Iota	B.1.526	United States, Nov 2020–Apr 2021	E484K (1.775)
Mu	B.1.621	Colombia, Jan–Aug 2021	R346K (1.307), E484K (1.775), N501Y (1.062)

B Implementation Details

B.1 Model hyperparameters

Table B.1: Shared TreeSBM architecture and optimization.

Hyperparameter	Value
Tree encoder layers	4
Hidden dimension	128
Attention heads	8
pLM reference	ESM-2 8M
Mutation / branching / BL head depth	2 / 2 / 2
Dropout	0.1
Optimizer	AdamW
Learning rate	1×10^{-4}
Batch size	1
Scheduled epochs / patience	100 / 50
Bridge c / λ_{mut} / λ_{cons}	1.0 / 12.0 / 0.5

Table B.2: Per-virus sequence length and best-checkpoint epoch (early-stopped within 100 scheduled epochs).

Virus / split	Max length L	Best epoch
SARS-CoV-2 Spike (geo)	1280	14
SARS-CoV-2 Spike (epidemic)	1280	50
Influenza H1N1 HA (geo)	566	74
Influenza H1N1 HA (epidemic)	566	88
Influenza H3N2 HA (temporal)	566	57
Influenza H3N2 HA (epidemic)	566	99
HIV-1 Env (geo)	900	14
HIV-1 Env (temporal)	900	98

B.2 SAMPLING SETTINGS

Table B.3: TreeSBM sampling settings for coverage and enrichment evaluations.

Sampling parameter	Coverage@100	Enrichment
Evolutionary horizon	50	100
Maximum leaves	400	300
Maximum branching factor	2	2
Mutation temperature	1.0	1.0
Branching temperature	1.0	1.0
Branch-rate scale	6.0	6.0
Mutation-rate scale	0.5	0.5
Number of generated trees K	100	20

C ALGORITHMS

Here, we provide pseudo-code for the core TreeSBM procedures. Algorithm 1 describes bridge-matching training from root-sequence/tree pairs. Algorithm 2 describes sampling intermediate tree states from an observed terminal tree. Algorithm 3 describes simulation under the reference evolutionary process. Algorithm 4 describes future tree generation from a single ancestral sequence. Algorithm 6 describes future-lineage coverage evaluation.

Algorithm 1 `TrainTreeSBM`: Tree-valued Schrödinger bridge matching

Input: Dataset $\mathcal{D} = \{(x_0^{(i)}, T_1^{(i)})\}_{i=1}^{N}$, reference process P^0, TreeSBM rates R_θ, tree encoder Enc_θ, sampling steps N_{steps}
Initialize model parameters θ
while not converged **do**
  Sample minibatch $\mathcal{B} \subset \mathcal{D}$
  for $(x_0, T_1) \in \mathcal{B}$ **do**
    $\mathcal{T}_0 \leftarrow \{x_0\}$ ▷ Root-only evolutionary tree
    $t \sim \mathcal{U}(0,1)$
    $\mathcal{T}_t \leftarrow \texttt{SampleBridgeState}(\mathcal{T}_0, T_1, t, P^0)$ ▷ Intermediate tree on a reference bridge path
    $H_t \leftarrow \mathrm{Enc}_\theta(\mathcal{T}_t, x_0, t)$ ▷ Contextual node, edge, and graph embeddings
    $\hat{R}_\theta(\mathcal{T}_t, \cdot, t) \leftarrow \texttt{RateHeads}_\theta(H_t)$ ▷ Mutation, branching, branch-length, and stopping rates
    $R_t^{0|T_1}(\mathcal{T}_t, \cdot) \leftarrow \texttt{ConditionalReferenceRates}(\mathcal{T}_t, T_1, t, P^0)$ ▷ Local target bridge rates
    $\mathcal{L}_{\mathrm{rate}} \leftarrow D_{\mathrm{rate}}\Big(R_t^{0|T_1}(\mathcal{T}_t, \cdot), \hat{R}_\theta(\mathcal{T}_t, \cdot, t)\Big)$
    $\mathcal{L}_{\mathrm{top}} \leftarrow \texttt{TopologyLoss}(\hat{R}_\theta, \mathcal{T}_t, T_1)$ ▷ Branching and lineage-event supervision
    $\mathcal{L}_{\mathrm{seq}} \leftarrow \texttt{MutationLoss}(\hat{R}_\theta, \mathcal{T}_t, T_1)$ ▷ Residue-edit supervision along branches
    $\mathcal{L}_{\mathrm{br}} \leftarrow \texttt{BranchLengthLoss}(\hat{R}_\theta, \mathcal{T}_t, T_1)$ ▷ Continuous branch-length regression
  end for
  $\mathcal{L}_{\mathrm{TreeSBM}} \leftarrow \mathcal{L}_{\mathrm{rate}} + \lambda_{\mathrm{top}}\mathcal{L}_{\mathrm{top}} + \lambda_{\mathrm{seq}}\mathcal{L}_{\mathrm{seq}} + \lambda_{\mathrm{br}}\mathcal{L}_{\mathrm{br}}$
  Update θ using $\nabla_\theta \mathcal{L}_{\mathrm{TreeSBM}}$
end while
return trained TreeSBM model R_θ

Algorithm 2 `SampleBridgeState`: Sample an intermediate tree state from an observed evolutionary tree

Input: Root-only tree $\mathcal{T}_0 = \{x_0\}$, terminal tree T_1, time t, reference process P^0
Parse $T_1 = (V, E, r, \ell, X)$
Sample retained branches $E_t \subseteq E$ according to branch survival probabilities at time t ▷ Keeps early branches more often than late branches
Construct partial topology V_t, E_t induced by retained branches
For each retained edge $e = (u, v)$, set partial branch length $\ell_t(e) \leftarrow t\,\ell(e)$ ▷ Intermediate branch length
for each node $v \in V_t$ **do**
    Sample partial mutation set $M_t(v) \subseteq M_1(v)$ according to mutation-time schedule ▷ Only mutations whose sampled event time is before t are retained
    $x_v(t) \leftarrow \texttt{ApplyEdits}(x_0, M_t(v))$
end for
$\mathcal{T}_t \leftarrow (V_t, E_t, r, \ell_t, \{x_v(t)\}_{v \in V_t})$
return $\mathcal{T}_t$

Algorithm 3 `ReferenceRollout`: Simulate the reference evolutionary process

Input: Root sequence x_0, pLM mutation prior p_{pLM}, fitness model F, branching intensity λ, horizon T, step size Δt
$\mathcal{T}_0 \leftarrow \{x_0\}$
for $t = 0, \Delta t, \ldots, T - \Delta t$ **do**
    for each active leaf node v in $\mathcal{T}_t$ **do**
        $b_v \sim \mathrm{Poisson}(\lambda(x_v)\Delta t)$ ▷ Number of new descendant branches
        for $j = 1, \ldots, b_v$ **do**
            Sample edit proposal $x' \sim p_{\mathrm{pLM}}(\cdot \mid x_v)$
            Accept x' with probability proportional to $\exp(\beta F(x'))$ ▷ Fitness-weighted mutation prior
            Add child node v_j with sequence $x_{v_j} = x'$ and branch length Δt
        end for
        Update existing branch lengths by Δt
        Stop lineage v with probability $p_{\mathrm{stop}}(x_v, t)\Delta t$
    end for
end for
return reference tree $\mathcal{T}_T$

Algorithm 4 `GenerateFutureTree`: TreeSBM generation from one ancestral sequence

Input: Root sequence x_0, trained rates R_θ, horizon T, step size Δt, maximum nodes $N_{\max}$
$\mathcal{T}_0 \leftarrow \{x_0\}$
for $t = 0, \Delta t, \ldots, T - \Delta t$ **do**
    $H_t \leftarrow \mathrm{Enc}_\theta(\mathcal{T}_t, x_0, t)$
    $\hat{R}_\theta(\mathcal{T}_t, \cdot, t) \leftarrow \texttt{RateHeads}_\theta(H_t)$
    for each active leaf node v **do**
        Sample event $a_v \sim \hat{R}_\theta(\mathcal{T}_t, \cdot, t)$ ▷ Mutation, branching, extension, or stopping
        if a_v is mutation **then**
            Sample residue edit $e \sim p_\theta(e \mid \mathcal{T}_t, v, t)$
            $x_v \leftarrow \texttt{ApplyEdit}(x_v, e)$
        else if a_v is branching **then**
            Sample number of children $b_v \sim p_\theta(b \mid \mathcal{T}_t, v, t)$
            for $j = 1, \ldots, b_v$ **do**
                Sample child sequence $x_{v_j} \sim p_\theta(x_{\text{child}} \mid x_v, \mathcal{T}_t, t)$
                Add child node v_j with sequence x_{v_j} and initial branch length 0
            end for
        else if a_v is branch extension **then**
            $\ell(e_v) \leftarrow \ell(e_v) + \Delta t$
        else if a_v is stopping **then**
            Mark lineage v as terminal
        end if
    end for
    if $|\mathcal{T}_t| > N_{\max}$ **then**
        break ▷ Prevent uncontrolled tree growth
    end if
end for
return sampled future tree $T \sim p_\theta(T \mid x_0)$

Algorithm 5 `BatchGenerateTrees`: Generate a forecast distribution

Input: Root sequence x_0, trained TreeSBM model R_θ, number of samples K
$\mathcal{G} \leftarrow \emptyset$
for $k = 1, \ldots, K$ **do**
    $T_k \leftarrow \texttt{GenerateFutureTree}(x_0, R_\theta)$
    $\mathcal{G} \leftarrow \mathcal{G} \cup \{T_k\}$
end for
return generated tree set $\mathcal{G} = \{T_1, \ldots, T_K\}$

Algorithm 6 `FutureLineageCoverage`: Evaluate whether realized lineages fall inside the generated distribution

Input: Generated trees $\mathcal{G} = \{T_1, \ldots, T_K\}$, observed future lineages $\mathcal{L}_{\text{real}}$, distance threshold ε
`covered` $\leftarrow 0$
for each realized lineage $L_{\text{real}} \in \mathcal{L}_{\text{real}}$ **do**
    $d_{\min} \leftarrow \infty$
    for each generated tree $T_k \in \mathcal{G}$ **do**
        for each lineage $L \in T_k$ **do**
            $d_{\min} \leftarrow \min(d_{\min}, d(L_{\text{real}}, L))$ ▷ Edit distance, mutation-set distance, or pLM embedding distance
        end for
    end for
    if $d_{\min} \leq \varepsilon$ **then**
        `covered` $\leftarrow$ `covered` $+ 1$
    end if
end for
$\text{Coverage}_{\varepsilon} \leftarrow$ `covered`$/|\mathcal{L}_{\text{real}}|$
return $\text{Coverage}_{\varepsilon}$