ARGFLOW: AMORTIZED POSTERIOR SAMPLING OF ANCESTRAL RECOMBINATION GRAPHS WITH GENERATIVE FLOW NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ancestral recombination graphs (ARGs) describe the genealogical and recombination history of sampled genomes, underpinning inference of demography, selection, and mutation age. Despite decades of work, posterior inference over ARGs remains challenging. The space combines a variable number of discrete events with continuous event times and is sharply multimodal because structurally different histories can explain the same sequence data almost equally well. In this paper we adopt generative flow networks (GFlowNets) to tackle Bayesian inference over ARGs. Because GFlowNets sample combinatorial structures in proportion to reward, they are a natural choice for a posterior whose mass is spread across many distinct histories, and where posterior sampling requires sustained exploration of alternative histories. Across three simulated datasets spanning different recombination rates, ARGFlow achieves competitive reconstruction against ARGInfer and SINGER while representing uncertainty in coalescence times and local genealogical structure.

1 INTRODUCTION

Ancestral recombination graphs (ARGs) provide a unified representation of the genealogical history of recombining genomes. At each genomic position, sampled sequences are related by a local genealogy, while historical recombination events induce changes in genealogical relationships along the genome. An ARG connects these local genealogies into a single representation containing both coalescence and recombination events. This representation has enabled genealogical approaches to population-genetic inference, including the estimation of coalescence times, allele ages, recombination histories, demographic processes, and signals of natural selection (Lewanski et al., 2024; Nielsen et al., 2025).

However, inferring ARGs from sequence data remains computationally challenging. Even in the absence of recombination, the number of possible phylogenetic tree topologies grows super-exponentially with the number of sampled sequences. ARGs extend this already large space of trees by incorporating both coalescence and recombination. Backward in time, ancestral lineages merge through coalescence and split through recombination, allowing different genomic regions to have different genealogies. Recombination therefore introduces additional unknowns, including the number of ancestral events and the genomic breakpoints at which ancestral material is partitioned. Inference must jointly resolve how ancestral lineages merge and split, when these events occur, and which genomic regions they affect. Moreover, distinct ancestral histories may explain the same observed sequence data nearly equally well, making uncertainty an intrinsic feature of ARG inference. Exact inference over this space is generally intractable, motivating approximations and specialized search or sampling algorithms (Hudson & Kaplan, 1988; Li & Stephens, 2003; McVean & Cardin, 2005).

Over the past decade, Bayesian approaches such as ARGweaver (Rasmussen et al., 2014) and SINGER (Deng et al., 2025) have enabled uncertainty quantification by sampling ancestral histories rather than returning a single reconstruction. These methods use MCMC, generating histories through successive updates to a current ancestral graph. Despite substantial advances in sampling efficiency, obtaining additional effective samples requires continued chain simulation, with attention

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.

to burn-in, autocorrelation, and convergence. This motivates a complementary approach in which a generative policy is learned for a fixed dataset, allowing its training cost to be amortized over repeated, independent draws from the learned distribution.

Generative flow networks (GFlowNets) provide a framework for learning such a policy (Bengio et al., 2023). They construct structured objects through a sequence of decisions and are trained to sample completed objects with probability proportional to a specified reward. When this reward is an unnormalized posterior density, the resulting policy targets the corresponding Bayesian posterior. This formulation has been applied to Bayesian structure learning (Deleu et al., 2022) and phylogenetic inference (Zhou et al., 2024). For ARG inference, an ancestral history can be constructed backward in time from sampled genomes through a variable-length sequence of coalescence and recombination events. This introduces a challenge for learning a construction policy because recombination can increase the number of active lineages, leaving the trajectory length unknown in advance. Defining the reward of each completed history as the product of its sequence likelihood and population-genetic prior provides a training target for learning to sample from the posterior over ancestral histories. As genomic sequence length and sample size increase, construction trajectories can become longer, making it harder to learn how individual decisions contribute to this terminal reward.

In this work, we introduce ARGFlow, a GFlowNet framework for amortized posterior sampling of ancestral recombination graphs. We formulate ARG inference as a sequential process that constructs genealogically valid ancestral histories backward in time under Hudson's coalescent-with-recombination model (Hudson, 1983). For a fixed dataset, ARGFlow learns a generative policy that targets the posterior over these histories, amortizing training cost over repeated, independent draws from the learned distribution.

Our contributions are as follows:

1. **Sequential formulation of ARG inference.** We formulate ARG construction as an acyclic sequential decision process over backward-time coalescence and recombination events, accommodating a variable number of ancestral events, recombination locations, and continuous event times.
2. **Structured neural policy for ancestral histories.** We develop a state-dependent policy that represents evolving ancestral lineages through their genetic information and genomic material and parameterizes the structured decisions required to construct valid ARGs.
3. **Posterior-proportional GFlowNet training.** We define the terminal reward from the sequence likelihood and Hudson coalescent-with-recombination prior and train the model using Subtrajectory Balance, connecting sequential ARG construction to sampling from the Bayesian posterior.
4. **Empirical evaluation of posterior inference.** We evaluate ARGFlow against ARGInfer and SINGER across simulated recombination regimes, considering both reconstruction of the generating ancestry and agreement in posterior summaries such as pairwise coalescence times and local-clade probabilities.

2 Related Work

ARGweaver (Rasmussen et al., 2014) combines a sequentially Markov coalescent approximation with MCMC threading operations. SINGER (Deng et al., 2025) advances scalable Bayesian sampling of genome-wide genealogies. Both sample through transitions between ancestral histories; ARGFlow generates each history through a sequence of construction actions under a trained policy.

Relate, tsinfer/tsdate, and ARG-Needle use specialized approximations and reconstruction strategies to infer genealogies for large cohorts (Speidel et al., 2019; Kelleher et al., 2019; Zhang et al., 2023). Our evaluation instead focuses on posterior sampling, comparing both reconstruction and the distributions of genealogical summaries.

PhyloGFN (Zhou et al., 2024) applies GFlowNets to phylogenetic inference by constructing trees through successive subtree merges. ARG construction additionally requires recombination, which splits ancestral material and increases the number of active lineages. The policy must therefore track which genomic regions each lineage carries and allow the number of construction steps to vary

across histories for the same sampled genomes. These requirements distinguish ARGFlow's state and action space from phylogenetic-tree construction.

3 BACKGROUND

3.1 ANCESTRAL RECOMBINATION GRAPH

An ancestral recombination graph (ARG) describes the joint genealogical history of a set of sampled genomes along a recombining genomic region. Let $Y = \{y_1, \ldots, y_n\}$ denote n sampled haplotypes observed over a region of length L. Unlike a phylogenetic tree, which assumes a single genealogy for all sites, an ARG allows the genealogy to vary along the genome as a consequence of historical recombination. At each genomic position $x \in [0, L)$, the ARG G induces a local genealogy $T_x(G)$. Together, these genealogies describe how the sampled genomes are related across the region, while G records the underlying history of coalescence and recombination.

We consider ARGs under Hudson's coalescent-with-recombination (CwR) model (Hudson, 1983). Tracing ancestry backward in time, ancestral lineages undergo two types of events: *coalescence*, in which two lineages merge into a common ancestral lineage, and *recombination*, in which a lineage splits into two ancestral lineages carrying the material on either side of a genomic breakpoint. Consequently, an ARG can be represented by a variable-length sequence of timed coalescence and recombination events. Let $\phi = (N_e, \mu, r)$ denote the population-genetic parameters, where N_e is the effective population size, μ is the per-base mutation rate, and r is the per-base recombination rate. Under the CwR model, these parameters induce a prior density over ancestral histories, $p_{\mathrm{CwR}}(G \mid \phi)$. Given an ARG G, the observed sequence data are generated along its local genealogies according to a mutation model, defining the likelihood $p(Y \mid G, \phi)$. Bayesian ARG inference seeks the posterior distribution over ancestral histories conditioned on the observed haplotypes,

$$p(G \mid Y, \phi) = \frac{p(Y \mid G, \phi)\, p_{\mathrm{CwR}}(G \mid \phi)}{p(Y \mid \phi)}. \tag{1}$$

The normalizing constant is the marginal likelihood, obtained by integrating over the space $\mathcal{G}$ of ancestral histories:

$$p(Y \mid \phi) = \int_{\mathcal{G}} p(Y \mid G, \phi)\, p_{\mathrm{CwR}}(G \mid \phi)\, dG. \tag{2}$$

Because $p(Y \mid \phi)$ does not depend on G, the posterior is proportional to the product of the likelihood and prior:

$$p(G \mid Y, \phi) \propto p(Y \mid G, \phi)\, p_{\mathrm{CwR}}(G \mid \phi). \tag{3}$$

The integral in Eq. 2 combines summation over discrete choices of event types, ancestral lineages, and recombination breakpoints with integration over continuous event times. The number of events also varies across histories. This rapidly growing structured space makes direct normalization and exhaustive posterior enumeration intractable except for very small instances. In Section 4, we formulate ARG construction as a sequential generative process and use a GFlowNet to learn a sampling policy whose terminal distribution approximates Eq. 1.

3.2 GENERATIVE FLOW NETWORKS

Generative flow networks (GFlowNets) are generative models for learning stochastic policies over compositional state spaces (Bengio et al., 2021). A GFlowNet constructs an object through a sequence of actions, starting from an initial state s_0 and terminating at a state s_T corresponding to a complete object x. Given a non-negative reward function $R(x)$, the GFlowNet objective is to learn a forward policy P_F whose terminal distribution satisfies

$$P_F^{\top}(x) = \frac{R(x)}{Z}, \qquad Z = \sum_{x' \in \mathcal{X}} R(x'), \tag{4}$$

where $\mathcal{X}$ denotes the set of terminal objects and Z is the normalizing constant. GFlowNets therefore allocate probability mass across terminal objects in proportion to their reward.

A GFlowNet can be viewed as defining a flow over a directed acyclic graph whose nodes are intermediate states and whose edges correspond to actions. The forward policy $P_F(a_t \mid s_t)$ determines how a partial object is extended at each step. Training enforces consistency between the flows assigned to states and transitions so that the resulting terminal distribution approaches the reward-proportional target. This construction makes GFlowNets particularly suitable for distributions whose probability mass is spread across multiple distinct modes.

Several objectives have been developed for training GFlowNets. Trajectory Balance (TB) enforces a balance constraint over complete trajectories from the initial state to a terminal object, improving credit assignment across long sequences of decisions (Malkin et al., 2022). Subtrajectory Balance (SubTB) generalizes this principle by applying balance constraints to contiguous portions of a trajectory, providing learning signals at intermediate states (Madan et al., 2023). We use SubTB for ARGFlow and describe the resulting objective in Section 4.4.

ARG inference naturally fits this framework. A complete ARG can be constructed through a variable-length sequence of coalescence and recombination events. Choosing its unnormalized posterior density as the terminal reward makes posterior sampling the GFlowNet training target. Section 4 develops this construction and defines the reward. Figure 1 provides an overview of the sequential ARG construction process and ARGFlow's forward policy.

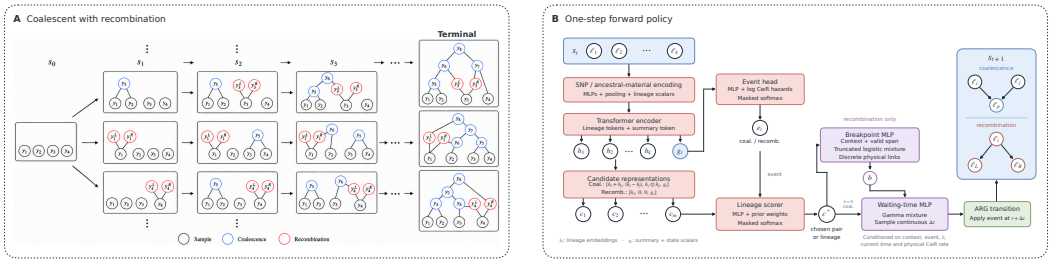


Figure 1: **Left (A):** Sequential ARG construction. Starting from sampled haplotypes, coalescence merges active lineages and recombination splits their ancestral material until every genomic position has a common ancestor. **Right (B):** ARGFlow's forward policy. A Transformer encodes SNP and ancestral-material features of active lineages. MLP heads select the event and participating lineages, sample a recombination breakpoint when needed, and draw a continuous waiting time before updating the ARG state.

4 Ancestral Recombination Graph Inference with GFlowNets

4.1 Sequential ARG Construction

We formulate ARG inference as a sequential generative process that constructs an ancestral history backward in time. Starting from the n observed haplotypes, a state s_t retains the chronological history of ancestral events constructed so far, the current backward time u_t, and the set of active ancestral lineages $\mathcal{L}_t = \{\ell_1, \ldots, \ell_{k_t}\}$. Unlike a subtree in phylogenetic-tree construction, an active ancestral lineage may carry only part of the genome, with descendant relationships that vary across genomic positions. We therefore track both the ancestral material carried by each lineage and its descendant information along the genome, allowing subsequent decisions to depend jointly on genealogical relationships and genomic coverage. Figure 1A illustrates how coalescence and recombination update the active lineages and their ancestral material during backward-time construction.

From a state s_t, ARGFlow extends the partial history through one of two ancestral events. A *coalescence* action selects two active lineages and replaces them with their common ancestral lineage. A *recombination* action selects an active lineage and a genomic breakpoint, splitting its ancestral material between two ancestral lineages. Each event additionally specifies a positive waiting time $\Delta t > 0$, advancing backward time as

$$u_{t+1} = u_t + \Delta t. \tag{5}$$

The resulting trajectories have variable length: coalescence reduces the number of active lineages by one, whereas recombination increases it by one. At each state, we allow only transitions that preserve a valid ancestral history, so genealogical constraints are enforced during construction.

Generation terminates once the ancestry of every genomic position has reached a common ancestor of all sampled haplotypes. Appendix A.1 formalizes the states, allowed actions, and termination condition. A complete construction trajectory

$$\tau = (s_0, a_0, s_1, \ldots, a_{T-1}, s_T) \tag{6}$$

therefore specifies a complete terminal ARG $G = s_T$.

Each transition appends one event to the history retained in the state, so the construction cannot return to an earlier state. The resulting state graph is therefore acyclic (Appendix A.2). Every non-initial state also has a unique predecessor obtained by reversing its most recent event (Appendix A.3).

4.2 Structured Forward Policy

The set of available actions changes throughout ARG construction as ancestral lineages are created, merged, and split. We therefore factorize the forward policy according to the structure of an ancestral event:

$$P_F(C, i, j, \Delta t \mid s_t) = P_F(C \mid s_t)\, P_F(i, j \mid C, s_t)\, p_F(\Delta t \mid C, i, j, s_t), \tag{7}$$

$$P_F(R, i, b, \Delta t \mid s_t) = P_F(R \mid s_t)\, P_F(i \mid R, s_t) \times P_F(b \mid R, i, s_t)\, p_F(\Delta t \mid R, i, b, s_t). \tag{8}$$

Here, C and R denote coalescence and recombination, respectively; i, j index active ancestral lineages, b denotes a recombination breakpoint, and $\Delta t > 0$ is the waiting time to the next ancestral event. This factorization decomposes each transition into an event type, the ancestral lineages involved, a recombination breakpoint when applicable, and a continuous event time, avoiding direct enumeration of their joint action space.

Compatibility constraints. We incorporate genealogical validity directly into the state-dependent action space. Under the infinite-sites mutation model, coalescence actions that are incompatible with the observed derived-allele pattern are masked from the forward policy. Consequently, the policy assigns probability only to transitions permitted by the current partial ancestral history and the observed sequence data. The compatibility criterion and its preservation under construction are detailed in Appendices A.1 and A.2.

4.3 Neural Architecture

ARGFlow parameterizes the structured forward policy using a Transformer encoder over the active ancestral lineages (Figure 1B). Each lineage embedding combines SNP features describing observed alleles and their compatibility with the partial ancestry, ancestral-material features describing genomic coverage and descendant relationships, and summary attributes such as lineage age. Encoding ancestral material separately preserves information about genomic regions without observed variants, which is needed to model recombination.

The Transformer jointly processes the lineage embeddings to capture interactions among active lineages. Since these lineages form an unordered set, we omit positional encodings of lineage order while retaining genomic coordinates in the input features. The encoder produces permutation-equivariant lineage embeddings and a permutation-invariant summary, which is combined with global features to represent the current state.

These representations feed prediction heads corresponding to the policy factorization in Eqs. 7–8. The heads specify distributions over the event type, participating lineage or lineage pair, recombination breakpoint when applicable, and continuous waiting time.

To estimate the state flow $F_\theta(s_t)$ used in training, a separate trainable flow head receives representations from a copy of the encoder that is frozen at initialization. Appendix C provides the feature definitions and architectural details.

4.4 Posterior Reward and GFlowNet Training

4.4.1 Posterior Reward

We train ARGFlow to sample complete ancestral histories in proportion to their Bayesian posterior probability. For a terminal ARG G, we define the unnormalized reward

$$R(G; Y, \phi) = p(Y \mid G, \phi)\, p_{\mathrm{CwR}}(G \mid \phi). \tag{9}$$

The first term is the likelihood of the observed haplotypes under the mutation model on the local genealogies induced by G, while the second is the prior density of the complete ancestral history under Hudson's CwR model. By Eq. 3, this reward is proportional to the Bayesian posterior over ARGs. Under the support, normalization, and almost-sure termination assumptions stated in Appendix A.4, exact flow balance induces the terminal distribution

$$P_F^{\top}(G \mid Y, \phi) = \frac{R(G; Y, \phi)}{\int R(G'; Y, \phi)\, dG'} = p(G \mid Y, \phi). \tag{10}$$

This identity defines the posterior-matching target for ARGFlow training; the learned policy approximates it through the balance objective below.

We evaluate $p_{\mathrm{CwR}}(G \mid \phi)$ from the sequence of timed coalescence and recombination events using the physical CwR event rates. We evaluate $p(Y \mid G, \phi)$ under the infinite-sites mutation model by marginalizing over branches on which each observed mutation could have occurred. Likelihood contributions are accumulated incrementally as local genealogies become resolved, avoiding explicit sampling of mutation placements. The complete likelihood and prior calculations are provided in Appendix B.

4.4.2 Subtrajectory Balance

We train the forward policy and state-flow model using Subtrajectory Balance (SubTB). Because the chronological construction gives every non-initial state a unique predecessor, the backward transition is deterministic (Appendix A.3). For a contiguous subtrajectory $s_m \to \cdots \to s_n$, the log-balance residual therefore takes the form

$$\delta_{m:n} = \log F_\theta(s_m) + \sum_{t=m}^{n-1} \log P_F(a_t \mid s_t) - \log F_\theta(s_n). \tag{11}$$

Terminal flows are fixed to the reward,

$$F_\theta(s_T) = R(G; Y, \phi). \tag{12}$$

We minimize the weighted squared residuals over subtrajectories,

$$\mathcal{L}_{\mathrm{SubTB}} = \mathbb{E}_\tau \left[\sum_{m<n} w_{m,n}\, \delta_{m:n}^2 \right], \tag{13}$$

where $w_{m,n}$ controls the relative contribution of subtrajectories of different lengths. SubTB provides intermediate training signals along the ARG construction trajectory while retaining the terminal posterior reward as the global learning target.

5 Experiments

We evaluate ARGFlow's ability to reconstruct ancestral histories and represent posterior uncertainty through two complementary analyses. First, we assess reconstruction accuracy against known simulated ancestry. Second, we compare posterior summaries of coalescence times and local genealogical structure with those obtained from ARGInfer and SINGER to assess agreement with established posterior samplers.

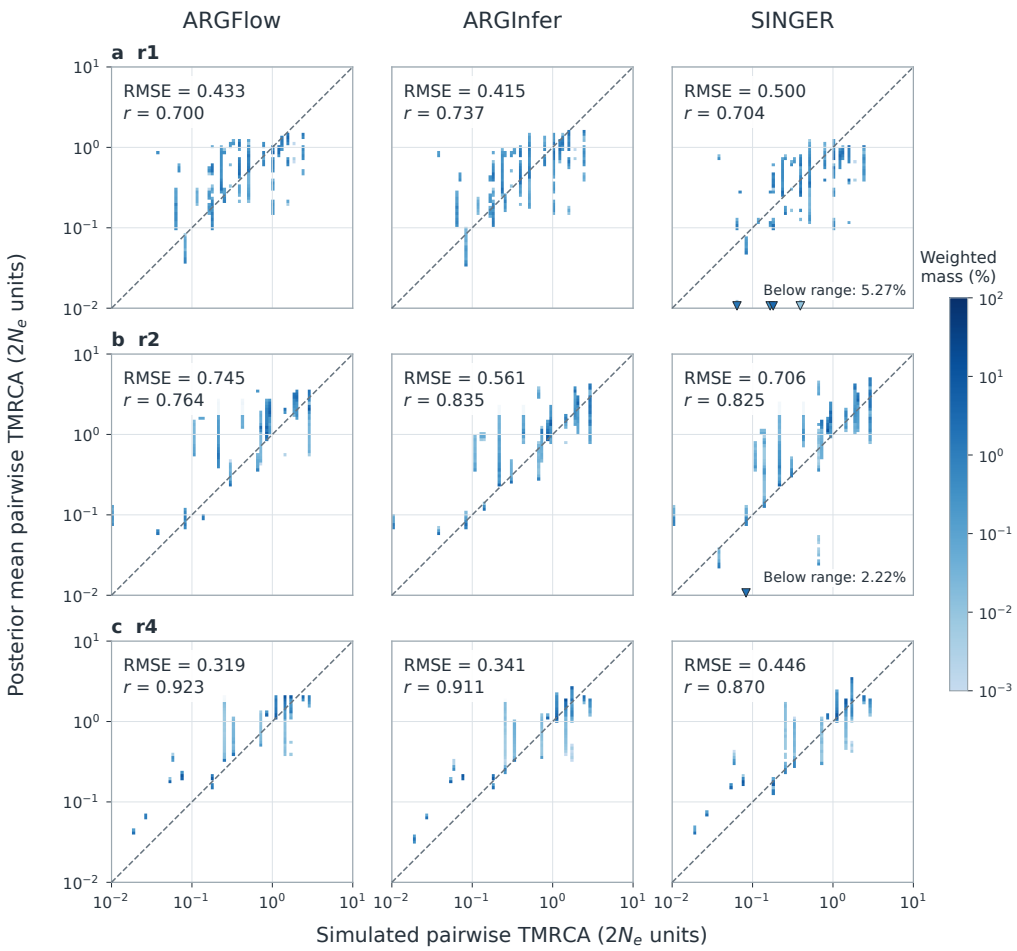


Figure 2: Pairwise TMRCA reconstruction against simulated truth. Posterior-mean pairwise TMRCA is plotted against the generating TMRCA for ARGFlow, ARGInfer, and SINGER across the three recombination regimes. Heatmaps summarize 45 haplotype pairs over genomic spans on shared logarithmic axes, with color indicating genomic-span-weighted mass per bin. The dashed diagonal denotes exact agreement. RMSE (in $2N_e$ units) and span-weighted Pearson correlation are computed on untransformed times and reported within each panel. Downward triangles mark estimates below the plotting range; their mass is stated in the affected panels and included in both statistics.

We simulate 10 haplotypes from a constant-size population with effective population size $N_e = 10{,}000$ over 25-kb regions under the infinite-sites mutation model. We fix the per-base mutation rate at $\mu = 10^{-8}$ and consider three recombination regimes, denoted r1, r2, and r4, with $\mu/r \in \{1, 2, 4\}$. These correspond to population-scaled mutation rate $\theta = 10$ and recombination rates $\rho \in \{10, 5, 2.5\}$, respectively. Thus, the three settings vary the amount of genealogical change due to recombination while keeping the expected mutational information fixed.

For each recombination regime, we generate one dataset and retain its complete generating ARG, including the local genealogies and coalescence times. All methods receive the same observed haplotypes and population-genetic parameters. The true ARG used to simulate each dataset is retained only for evaluation. Table 1 summarizes the simulation settings.

We compare ARGFlow with ARGInfer, whose coalescent-with-recombination and infinite-sites formulation provides a closely matched Bayesian baseline, and SINGER, a scalable posterior sampler for genome-wide genealogies. ARGFlow samples are obtained from the frozen policy after train-

Table 1: Simulation settings. Each setting contains one independently simulated ARG and corresponding sequence dataset.

Dataset	μ/r	r	θ	ρ
r1	1	1.0×10^{-8}	10	10
r2	2	5.0×10^{-9}	10	5
r4	4	2.5×10^{-9}	10	2.5

Table 2: Reconstruction accuracy and posterior uncertainty against the simulated ancestry. TMRCA RMSE and interval width are in units of $2N_e$ generations. Coverage and width refer to 90% pairwise-TMRCA credible intervals. Lower RMSE, rooted RF, and Brier scores are better; nominal coverage is 90%. Bold identifies ARGFlow, not the best value.

Dataset	Method	TMRCA RMSE	Rooted RF	Clade Brier	90% coverage	Interval width
r1	**ARGFlow**	**0.433**	**10.026**	**0.00567**	**90.63%**	**1.284**
	ARGInfer	0.415	10.141	0.00555	91.38%	1.179
	SINGER	0.500	10.484	0.00828	75.57%	0.663
r2	**ARGFlow**	**0.745**	**4.146**	**0.00184**	**96.95%**	**2.847**
	ARGInfer	0.561	3.942	0.00163	94.67%	1.766
	SINGER	0.706	5.082	0.00338	84.77%	1.912
r4	**ARGFlow**	**0.319**	**4.063**	**0.00204**	**99.26%**	**1.307**
	ARGInfer	0.341	4.052	0.00200	88.92%	1.142
	SINGER	0.446	4.189	0.00209	78.26%	1.092

ing. Sampling and baseline inference details are provided in Appendix D. The evaluated ensembles contain $1{,}000$ draws for each method, with equal weights within each ensemble.

5.1 Accuracy and Uncertainty

We first evaluate each method against the known generating ARG. We consider complementary aspects of genealogical reconstruction: pairwise TMRCA RMSE measures error in inferred coalescence times, rooted Robinson–Foulds (RF) distance measures disagreement in local-tree topology, and the clade Brier score evaluates the probabilities assigned to local clades. To assess temporal uncertainty, we additionally report the empirical coverage of 90% posterior credible intervals for pairwise TMRCA and their mean width. All metrics are weighted by genomic span; lower values are better for RMSE, RF, and Brier score. Full metric definitions are provided in Appendix D.

Table 2 shows competitive reconstruction with variation across settings. In r1, ARGFlow has the lowest rooted RF distance and coverage close to the nominal 90%. In r2, its temporal error exceeds both baselines and its intervals are substantially wider, although its topology metrics improve on SINGER. In r4, ARGFlow has the lowest TMRCA RMSE, but 99.26% coverage indicates overcoverage. These results do not establish uniformly superior accuracy or calibrated uncertainty across settings.

Figure 2 resolves these temporal errors across haplotype pairs and genomic spans. In r1, the three methods show similar broad patterns, with appreciable scatter around the diagonal. In r2, ARGFlow overestimates some shorter coalescence times, consistent with its larger RMSE in Table 2. In r4, its posterior means track the diagonal more closely, with the highest correlation (0.923) and lowest RMSE (0.319) among the three methods. These plots assess posterior-mean reconstruction; interval coverage is evaluated separately in Table 2.

5.2 Posterior-Summary Agreement

We assess agreement between ARGFlow and established posterior samplers by comparing uncertainty in coalescence times and local genealogical structure. Figure 3 compares pairwise TMRCA distributions and posterior probabilities of local clades with those obtained from ARGInfer and

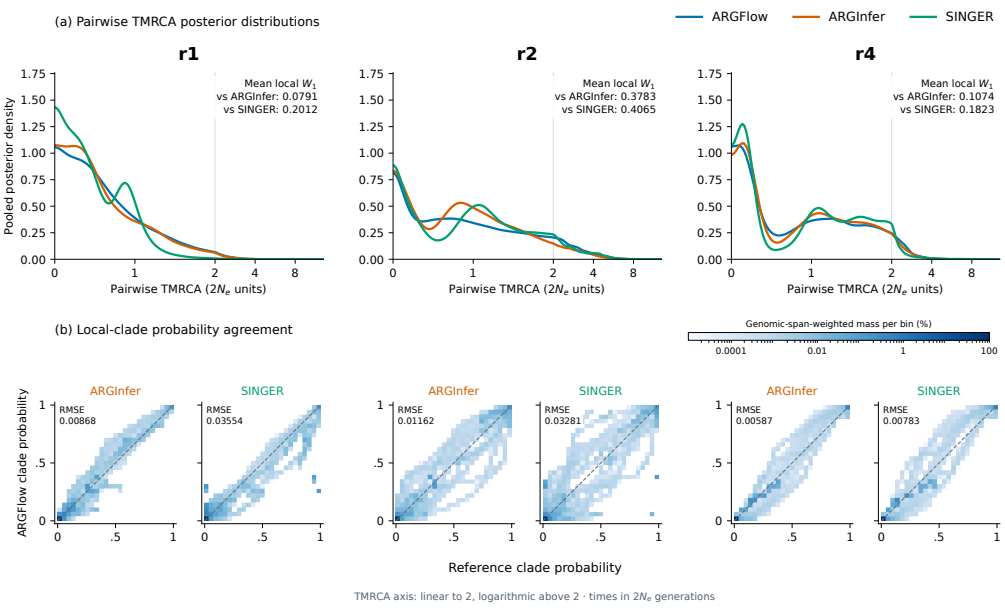


Figure 3: Posterior-summary agreement with ARGInfer and SINGER. (a) Pooled pairwise TMRCA distributions, with mean local Wasserstein distances between ARGFlow and each reference sampler. (b) Posterior local-clade probabilities, with clade-probability RMSE. The dashed diagonal indicates equal probabilities. TMRCA is expressed in units of $2N_e$ generations, with a linear axis up to 2 and a logarithmic axis thereafter.

SINGER. We quantify temporal agreement using the mean local Wasserstein distance and topological agreement using clade-probability RMSE.

Across all three settings, ARGFlow agrees more closely with ARGInfer than with SINGER under both metrics. Temporal agreement with ARGInfer is strongest in r1 and weakest in r2, with Wasserstein distances of 0.0791 and 0.3783, respectively. The larger discrepancy in r2 accompanies the increased temporal reconstruction error observed in Section 5.1. For local-clade probabilities, agreement with both samplers is strongest in r4, where RMSE is 0.00587 against ARGInfer and 0.00783 against SINGER.

These comparisons show that ARGFlow captures posterior summaries similar to those of existing samplers, particularly ARGInfer, which uses a closely matched population-genetic model. The agreement provides evidence about the evaluated temporal and topological summaries, while leaving differences in the full posterior over ancestral histories unresolved.

6 Conclusion

ARGFlow enables amortized posterior sampling of ancestral recombination graphs under Hudson's CwR model. After one training phase per dataset, its policy generates independent approximate posterior samples without MCMC burn-in or thinning. Among the evaluated methods, only ARGFlow amortizes sampling. ARGInfer also uses CwR but relies on MCMC, while SINGER uses a sequentially Markov approximation. Across three simulated datasets, ARGFlow achieves competitive reconstruction and posterior summaries closer to ARGInfer than to SINGER.

Scalability remains a limitation. Longer genomic regions and larger samples of haplotypes can require longer construction trajectories. Future work could combine the learned policy with local search to refine ancestry within selected genomic regions of an existing ARG, shortening refinement trajectories and enabling inference on larger datasets.

AI USE STATEMENT

The research idea and scientific direction originated with the authors. Large language models were used to assist with code development and manuscript writing. The authors retain responsibility for the methodology, implementation, experimental results, and claims presented in this paper.

ETHICS STATEMENT

The simulation experiments use synthetic genetic data and do not expose identifiable personal information. When applied to real human genomic data, ancestral inference can reveal sensitive information about individuals and their relatives. Such applications should respect participant consent, data-access restrictions, and appropriate privacy safeguards.

REPRODUCIBILITY STATEMENT

Our implementation is provided in the supplementary archive `code_supplement.zip`. Appendix F provides commands for training, posterior sampling, baseline inference, and evaluation. Our implementation includes configuration files specifying the datasets, model architecture, training hyperparameters, random seeds, and evaluation settings. Saved checkpoints and sampling manifests record the provenance of the evaluated ARG samples. The scripts in `paper/scripts/paper_datasets/` reproduce Table 2 and Figures 2 and 3 using a shared configuration file. The accompanying README documents the commands, metric definitions, sample selection, and time-unit conventions. Evaluation outputs include input-file hashes and software versions to support reproducibility.

REFERENCES

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation, November 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. GFlowNet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023. URL `https://jmlr.org/papers/v24/22-0364.html`.

Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In James Cussens and Kun Zhang (eds.), *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pp. 518–528. PMLR, 2022. URL `https://proceedings.mlr.press/v180/deleu22a.html`.

Yun Deng, Rasmus Nielsen, and Yun S. Song. Robust and accurate Bayesian inference of genome-wide genealogies for hundreds of genomes. *Nature Genetics*, 57(9):2124–2135, September 2025. ISSN 1546-1718. doi: 10.1038/s41588-025-02317-9.

R. R. Hudson. Properties of a neutral allele model with intragenic recombination. *Theoretical Population Biology*, 23(2):183–201, 1983. doi: 10.1016/0040-5809(83)90013-8.

R. R. Hudson and N. L. Kaplan. The coalescent process in models with selection and recombination. *Genetics*, 120(3):831–840, November 1988. ISSN 0016-6731. doi: 10.1093/genetics/120.3.831.

Jerome Kelleher, Yan Wong, Anthony W. Wohns, Chaimaa Fadil, Patrick K. Albers, and Gil McVean. Inferring whole-genome histories in large population datasets. *Nature Genetics*, 51 (9):1330–1338, September 2019. ISSN 1546-1718. doi: 10.1038/s41588-019-0483-y.

Alexander L. Lewanski, Michael C. Grundler, and Gideon S. Bradburd. The era of the ARG: An introduction to ancestral recombination graphs and their significance in empirical evolutionary genomics. *PLOS Genetics*, 20(1):e1011110, January 2024. ISSN 1553-7404. doi: 10.1371/journal.pgen.1011110.

Na Li and Matthew Stephens. Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. *Genetics*, 165(4):2213–2233, December 2003. ISSN 1943-2631. doi: 10.1093/genetics/165.4.2213.

Kanika Madan, Jarrid Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Cristian Nica, Tom Bosc, Yoshua Bengio, and Nikolay Malkin. Learning GFlowNets from partial episodes for improved convergence and stability. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23467–23483. PMLR, 2023. URL https://proceedings.mlr.press/v202/madan23a.html.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in GFlowNets. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 5955–5967. Curran Associates, Inc., 2022. doi: 10.52202/068431-0431.

Gilean A.T McVean and Niall J Cardin. Approximating the coalescent with recombination. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1459):1387–1393, July 2005. ISSN 0962-8436. doi: 10.1098/rstb.2005.1673.

Rasmus Nielsen, Andrew H. Vaughn, and Yun Deng. Inference and applications of ancestral recombination graphs. *Nature Reviews Genetics*, 26(1):47–58, January 2025. ISSN 1471-0064. doi: 10.1038/s41576-024-00772-4.

Matthew D. Rasmussen, Melissa J. Hubisz, Ilan Gronau, and Adam Siepel. Genome-Wide Inference of Ancestral Recombination Graphs. *PLOS Genetics*, 10(5):e1004342, May 2014. ISSN 1553-7404. doi: 10.1371/journal.pgen.1004342.

Leo Speidel, Marie Forest, Sinan Shi, and Simon R. Myers. A method for genome-wide genealogy estimation for thousands of samples. *Nature Genetics*, 51(9):1321–1329, September 2019. ISSN 1546-1718. doi: 10.1038/s41588-019-0484-x.

Brian C. Zhang, Arjun Biddanda, Árni Freyr Gunnarsson, Fergus Cooper, and Pier Francesco Palamara. Biobank-scale inference of ancestral recombination graphs enables genealogical analysis of complex traits. *Nature Genetics*, 55(5):768–776, May 2023. ISSN 1546-1718. doi: 10.1038/s41588-023-01379-x.

Mingyang Zhou, Zichao Yan, Elliot Layne, Nikolay Malkin, Dinghuai Zhang, Moksh Jain, Mathieu Blanchette, and Yoshua Bengio. PHYLOGFN: PHYLOGENETIC INFERENCE WITH GENERATIVE FLOW NETWORKS. 2024.

A Properties of the ARGFlow Construction

A.1 Formal State and Action Definitions

Unlike a subtree in phylogenetic-tree construction, an ancestral lineage in an ARG need not carry the entire genome, and the samples descending from it can vary along the genome. We first define this information, then describe how coalescence and recombination update it. Throughout, the observed haplotypes Y and population-genetic parameters ϕ are fixed.

Ancestral lineages. Let $[n] = \{1, \dots, n\}$ index the sampled haplotypes and let $[0, L)$ be the genomic region. An active lineage ℓ_i carries ancestral material $A_i \subseteq [0, L)$, represented as a finite union of half-open intervals. At position x, its descendant set $d_i(x) \subseteq [n]$ identifies the samples whose ancestry it carries; set $d_i(x) = \emptyset$ outside A_i. Thus, A_i records *where* the lineage carries ancestry, while $d_i(x)$ records *whose* ancestry it carries. Descendant sets are constant within intervals, but may differ between intervals: for example, a lineage can carry descendants $\{1, 2\}$ to the left of a breakpoint and $\{1\}$ to its right. Material is tracked across the entire region, including positions without observed variants.

State and initial condition. After t events, the construction state is

$$s_t = (H_t, u_t, \mathcal{L}_t), \qquad \mathcal{L}_t = \{\ell_1, \ldots, \ell_{k_t}\}. \tag{14}$$

Here, H_t retains the chronological event history and the resulting partial ARG, including lineage creation times and ancestry connections. The active lineages form the frontier extended by the next event. The integer t counts events, whereas u_t is backward time, measured in units of $2N_e$ generations. Initially, $u_0 = 0$ and the history contains only the sampled lineages, with $A_i = [0, L)$ and $d_i(x) = \{i\}$. At every position, the nonempty descendant sets of active lineages partition $[n]$: each sample's ancestry is carried by exactly one active lineage there. The likelihood summaries used in Appendices B and C are computed from this state and the observations; they are not additional sampled variables.

Coalescence and recombination actions. An action specifies an event and a continuous waiting time $\Delta t > 0$. Both event types advance time to $u_{t+1} = u_t + \Delta t$, create the new ancestral lineages at that time, and append the event to H_t. A coalescence action $a_t = (C, i, j, \Delta t)$, with $i < j$, replaces two active lineages by one lineage $\ell_\star$ satisfying

$$A_\star = A_i \cup A_j, \qquad d_\star(x) = d_i(x) \cup d_j(x). \tag{15}$$

Where their material overlaps, the new lineage joins the two local ancestries; elsewhere it carries the ancestry of the sole contributing lineage. Coalescence is also allowed between lineages with disjoint material, subject to the compatibility condition below.

A recombination action $a_t = (R, i, b, \Delta t)$ replaces ℓ_i by two lineages carrying its material to the left and right of breakpoint b:

$$A_{\mathrm{L}} = A_i \cap [0, b), \qquad A_{\mathrm{R}} = A_i \cap [b, L). \tag{16}$$

Each new descendant map is the restriction of d_i to its respective material set. Both sets must be nonempty. We use a discrete genome with integer length L and integer breakpoint links between bases, while observed SNP positions may be continuous. The eligible links are

$$\mathcal{B}_i(s) = \big\{b \in \{1, \ldots, L-1\} : A_i \cap [0, b) \neq \emptyset,\ A_i \cap [b, L) \neq \emptyset\big\}, \qquad B_i = |\mathcal{B}_i(s)|. \tag{17}$$

These include links inside gaps in ancestral material, since such a split still separates material on either side. For $r > 0$, define the recombinable lineages as $\mathcal{R}(s) = \{i : B_i > 0\}$; for $r = 0$, set $\mathcal{R}(s) = \emptyset$. Each ancestry connection in the updated graph carries the material inherited along that connection. Given the state and action, these updates are deterministic.

Compatibility with observed alleles. Under the infinite-sites model, the derived alleles at each SNP must be explainable by a single mutation on a branch of its local genealogy. Consequently, a merge cannot join some derived carriers to ancestral carriers while other derived carriers remain outside the merged group: that would prevent the derived carriers from forming a clade. Let p_q be the position of SNP q and $D_q \subset [n]$ its observed derived-carrier set. For a SNP carried by both selected lineages, write $U_q = d_i(p_q) \cup d_j(p_q)$. The merge is admissible at this SNP when

$$U_q \cap D_q = \emptyset \quad \text{or} \quad U_q \subseteq D_q \quad \text{or} \quad D_q \subseteq U_q. \tag{18}$$

These cases respectively merge only ancestral carriers, only derived carriers, or a group that already contains all derived carriers. Define $\mathcal{C}(s)$ as the set of unordered pairs satisfying this condition at every shared SNP. At a SNP carried by only one selected lineage, the descendant set is unchanged, so no additional check is needed. The forward policy selects coalescence pairs from $\mathcal{C}(s)$ and recombination lineages and links from $\mathcal{R}(s)$ and $\mathcal{B}_i(s)$. This mask restricts policy actions; it does not remove incompatible pairs from the physical CwR rates used to evaluate the prior (Appendix B.1).

Completion and terminal histories. Construction stops at the first state in which every genomic position is covered by exactly one active lineage, whose descendant set is $[n]$. This means that every local genealogy has reached its most recent common ancestor. It does not require a single active lineage for the entire genome: several lineages may remain if they carry disjoint regions. Material whose local ancestry is already complete remains represented while the remaining regions are resolved. Terminal states have no forward actions.

We identify the terminal state $G = s_T$ with its complete chronological ARG history, retaining the events rather than only the simplified local genealogies used for evaluation. These definitions specify the construction underlying the GFlowNet. The following subsections address its validity and acyclicity, deterministic backward transitions, and the connection between terminal rewards and posterior sampling.

A.2 Validity and Acyclicity of Sequential ARG Construction

Two graphs enter the construction: the ancestral graph being generated, whose edges connect younger and older lineages, and the GFlowNet state graph, whose edges correspond to construction actions. We establish validity of the ancestry first, then acyclicity of the state graph.

Proposition (Validity and acyclicity). Starting from the initial state in Appendix A.1, every finite sequence of allowed actions preserves the partition of sample ancestry at each genomic position, strictly ordered times along ancestry edges, and compatibility with the observed infinite-sites allele patterns. If the sequence terminates, it defines complete rooted local genealogies compatible with those observations. Moreover, the graph of reachable construction states is acyclic.

Proof: preservation of ancestry. Fix a genomic position x. Initially, the active descendant sets are the singletons $\{1\}, \ldots, \{n\}$, which partition $[n]$. Suppose this property holds before an event. If both coalescing lineages carry x, Eq. 15 replaces two disjoint sets by their union. If only one carries x, its descendant set is unchanged. A recombination instead passes the descendant set to exactly one parent, according to whether $x < b$ or $x \geq b$ (Eq. 16). All other sets remain unchanged. Thus, the partition is preserved by induction: no sample's ancestry is lost or duplicated. This argument also covers disjoint-material merges and breakpoints inside material gaps, since it applies separately at every position.

Temporal and local-genealogy validity. Every active lineage was created at or before u_t, whereas each new parent is created at $u_t + \Delta t > u_t$. Hence, every ancestry edge has positive temporal length, and following edges backward in time cannot form a directed cycle. At a fixed position, recombination provides only one parental continuation; coalescence joins two local branches when both children carry that position, and otherwise provides a single continuation. The partial local genealogy therefore remains a forest. At termination, the partition consists of the single set $[n]$, so this forest has become one rooted genealogy containing all samples. Unary ancestry may remain in the graph representation and is simplified for evaluation.

Infinite-sites compatibility. Consider a segregating SNP q, with nonempty proper derived-carrier set $D_q \subset [n]$. Initially, every descendant set is either contained in D_q or disjoint from it. The mask in Eq. 18 ensures that every newly formed group is disjoint from, contained in, or contains D_q. Recombination and events involving only one lineage at p_q preserve its local descendant set. In particular, consider the first local merge that joins a derived carrier to an ancestral carrier. Its derived child must already contain all of D_q: otherwise the merged group would contain ancestral and derived carriers while excluding another derived carrier, violating the mask. Before that merge, the derived child thus subtends exactly D_q. This child exists initially for a singleton D_q, or is formed by earlier coalescences for a larger D_q. In a completed genealogy, such a merge must occur because $D_q \neq [n]$. The branch below it has positive length and subtends precisely D_q, permitting a single mutation that explains the SNP. The argument applies independently at every observed SNP.

Acyclicity of the construction-state graph. Let $h(s)$ be the number of ancestral events retained in the history component H of state s. Every allowed transition appends exactly one event, so

$$h(s_{t+1}) = h(s_t) + 1. \tag{19}$$

A directed cycle would return to the same state and hence the same history length, contradicting this strict increase. For example, splitting and then rejoining material can restore an earlier active-material configuration, but cannot restore the earlier state: the retained history contains two additional events. This completes the argument.

Acyclicity does not by itself guarantee termination: recombination can continue to extend a history. The proposition concerns finite prefixes and completed trajectories, and does not assert almost-sure termination under an arbitrary policy. Appendix A.3 addresses the backward transitions of this chronological construction.

A.3 Deterministic Backward Transitions

The backward transition undoes the latest construction event. For a noninitial state s', remove the last event from its retained history and reconstruct the state from the remaining prefix. Positive waiting times order the events strictly, so there is a unique latest event. If it is a coalescence, its new parent is removed and the two child lineages are restored to the active set. If it is a recombination, both new parents are removed together and their shared child is restored. The two recombination parents share a creation time because they belong to one event; they do not represent two possible backward choices. The previous global event time and all history-derived quantities are restored as well. This operation reverses a construction step rather than specifying a biological model for ancestry evolving toward the present.

Proposition (Unique predecessor). Every reachable noninitial state s' has exactly one predecessor, denoted $\mathrm{pred}(s')$, under the construction in Appendix A.1.

Proof. The retained history specifies the last action, including its waiting time and participating lineages. Any predecessor must therefore have the history obtained by deleting that action. Starting from the fixed initial state, the deterministic updates of Appendix A.1 map this prefix to exactly one state. Reapplying the deleted action recovers s', establishing both existence and uniqueness.

Consequently, the backward transition kernel is the point mass

$$P_B(\cdot \mid s') = \delta_{\mathrm{pred}(s')}. \tag{20}$$

It assigns probability one to the unique predecessor, so its log-probability contribution along every construction trajectory is zero. This explains the absence of a backward-policy term in Eq. 11. The point mass is a deterministic kernel, not a density equal to one over a continuous state space. The source state has no predecessor. Uniqueness relies on retaining the chronological history: active-lineage features or simplified local genealogies alone can omit events and need not determine a unique preceding state.

A.4 Posterior Proportionality of the Terminal Distribution

We now connect balance along construction trajectories to the posterior over complete histories. Condition throughout on fixed Y and ϕ. Let $\mathcal{G}$ be the space of complete chronological histories using the breakpoint and stopping conventions in Appendix A.1. On this mixed discrete–continuous space, dG denotes counting measure over discrete event choices and Lebesgue measure over positive waiting times, summed over finite trajectory lengths. We use the same measure for the target and sampling densities. Accordingly, each factor $P_F(a_t \mid s_t)$ below includes both the discrete probabilities and the continuous waiting-time density in Eqs. 7–8.

Proposition (Posterior proportionality). Suppose the target reward $R(G; Y, \phi)$ in Eq. 9 is defined on $\mathcal{G}$ with these same conventions, the forward policy covers its positive-reward support, and $0 < Z = \int_{\mathcal{G}} R(G; Y, \phi)\, dG < \infty$. Assume that the policy is normalized at every nonterminal state and terminates almost surely. If a flow F, with finite positive source flow, satisfies exact balance on the reachable support and has terminal boundary $F(s_T) = R(G; Y, \phi)$, then its terminal density is $P_F^{\top}(G \mid Y, \phi) = R(G; Y, \phi)/Z = p(G \mid Y, \phi)$.

Proof. By Appendix A.3, the predecessor is deterministic. In the history coordinates above, one-step balance therefore reads

$$F(s_t) P_F(a_t \mid s_t) = F(s_{t+1}), \tag{21}$$

with density equalities understood almost everywhere. Applying this relation along the unique trajectory producing G and substituting the terminal boundary gives

$$F(s_0) \prod_{t=0}^{T-1} P_F(a_t \mid s_t) = R(G; Y, \phi). \tag{22}$$

Because the complete history retains every action, it has exactly one construction trajectory. The product is thus its terminal sampling density; no sum over alternative construction orders is needed.

Almost-sure termination implies that this density integrates to one. Integrating Eq. 22 over $\mathcal{G}$ therefore yields $F(s_0) = Z$, and hence

$$P_F^{\top}(G \mid Y, \phi) = \frac{R(G; Y, \phi)}{Z} = \frac{p(Y \mid G, \phi)\, p_{\mathrm{CwR}}(G \mid \phi)}{\int_{\mathcal{G}} p(Y \mid G', \phi)\, p_{\mathrm{CwR}}(G' \mid \phi)\, dG'} = p(G \mid Y, \phi). \tag{23}$$

Support and represented objects. The compatibility mask does not exclude positive-likelihood histories within this construction: a forbidden descendant group overlaps the derived-carrier set without containing it or being contained in it. Such a group cannot coexist with the derived-carrier clade in a completed local tree, and subsequent events cannot remove that group from the retained local ancestry. Any completed history containing it therefore has zero infinite-sites likelihood. The proposition concerns the posterior on $\mathcal{G}$, not an unspecified alternative ARG representation. Local genealogies and other summaries inherit their distributions by mapping sampled histories to those quantities. Several distinct histories can give the same summary, whose probability aggregates their contributions.

Connection to training. Exact SubTB constraints include the one-step relations above. Minimizing the empirical objective encourages these constraints but does not establish that they hold throughout the support; the trained policy is therefore an approximation to the stated posterior. The implementation uses $\widetilde{R} = e^C R$ (Eq. 57), which replaces Z by $\widetilde{Z} = e^C Z$ and leaves $\widetilde{R}/\widetilde{Z} = R/Z$ unchanged. Finally, the almost-sure termination assumption is separate from the acyclicity result in Appendix A.2. A computational event limit does not prove this assumption, and incomplete histories are not terminal samples covered by the proposition.

B Likelihood and Coalescent-with-Recombination Prior

B.1 CwR Event Density

We evaluate the prior on the chronological histories defined in Appendix A.1. Time is measured in units of $2N_e$ generations. In these units, every unordered pair of active lineages coalesces at rate one, and every eligible recombination link on a lineage has rate $\gamma = 2N_e r$. For a nonterminal state s with k active lineages, let $B_i = |\mathcal{B}_i(s)|$ denote the number of eligible integer links on lineage i. The physical event rates are

$$\lambda_C(s) = \binom{k}{2}, \qquad \lambda_R(s) = \gamma \sum_{i=1}^{k} B_i, \qquad \Lambda(s) = \lambda_C(s) + \lambda_R(s). \tag{24}$$

As in Eq. 17, a link is eligible whenever material lies on both sides, including links inside gaps in ancestral material. Thus, recombination rates depend on the number of physical links separating material, not just on the total length of occupied intervals. With $\rho = 4N_e r L$, the per-link rate is also $\gamma = \rho/(2L)$.

The waiting time has density $\Lambda(s)e^{-\Lambda(s)\Delta t}$. Conditional on an event, a specified coalescence pair has probability $1/\Lambda(s)$, and a specified lineage–link pair has probability $\gamma/\Lambda(s)$. Multiplying by the waiting-time density gives

$$p_{\mathrm{CwR}}(C, i, j, \Delta t \mid s) = e^{-\Lambda(s)\Delta t}, \qquad i < j, \tag{25}$$

$$p_{\mathrm{CwR}}(R, i, b, \Delta t \mid s) = \gamma e^{-\Lambda(s)\Delta t}, \qquad b \in \mathcal{B}_i(s). \tag{26}$$

These are densities with respect to continuous waiting time and counting measure over discrete event choices. Summing over choices and integrating over $\Delta t > 0$ yields one. If $r = 0$, recombination has zero density and only coalescence events remain.

The prior includes all $\binom{k}{2}$ pairs, including disjoint-material pairs and pairs excluded by the observed-allele compatibility mask. That mask restricts the learned policy, not the physical prior. In particular, replacing $\lambda_C(s)$ by $|\mathcal{C}(s)|$ would change both the waiting-time density and the target posterior.

For a completed history $G = (s_0, a_0, \ldots, a_{T-1}, s_T)$, define $h(a_t) = 1$ for coalescence and $h(a_t) = \gamma$ for recombination. Then

$$p_{\mathrm{CwR}}(G \mid \phi) = \prod_{t=0}^{T-1} h(a_t) e^{-\Lambda(s_t)\Delta t_t}, \qquad \log p_{\mathrm{prior}}(s_t) = \sum_{v=0}^{t-1} \big[\log h(a_v) - \Lambda(s_v)\Delta t_v\big]. \tag{27}$$

The empty sum at the initial state is zero. Construction stops at the first state with complete local ancestry everywhere. There is no additional waiting-time or survival factor after that stopping event: the represented history ends there, rather than at a fixed later time.

B.2 Infinite-Sites Likelihood

We condition on contemporaneous, phased haplotypes, known ancestral alleles, and observation of the entire region $[0, L)$. Let M be the number of segregating sites, with positions p_q and nonempty proper derived-carrier sets $D_q \subset [n]$. Mutation positions are continuous; this does not require recombination breakpoints to be continuous. In the time units above, the mutation rate per base is $\kappa = 2N_e\mu$.

For the local genealogy at position x, let $\mathcal{E}_x(G)$ contain its branches below the local MRCA. A branch e has duration ℓ_e and descendant set $D_e(x)$. Define the total genomic branch exposure and the branch length compatible with SNP q by

$$E(G) = \int_0^L \sum_{e \in \mathcal{E}_x(G)} \ell_e \, dx, \tag{28}$$

$$W_q(G) = \sum_{e \in \mathcal{E}_{p_q}(G)} \ell_e \, \mathbf{1}\{D_e(p_q) = D_q\}. \tag{29}$$

Exposure has units of base pairs times scaled time; W_q has units of scaled time. Unary subdivisions may contribute several compatible segments, whose lengths add. Connections above a local MRCA have all samples as descendants and are excluded from both quantities, even if the chronological ARG retains them while other regions are unresolved.

Under the infinite-sites model, each observed SNP is explained by one mutation. Conditional on its position, integrating the mutation's location along all compatible branches supplies the factor $\kappa W_q(G)$. The Poisson no-additional-mutation factor over the observed genealogy is $e^{-\kappa E(G)}$. Hence the polarized mutation-pattern density, with fixed data-only factors suppressed, is

$$p(Y \mid G, \phi) = e^{-\kappa E(G)} \prod_{q=1}^{M} \kappa W_q(G), \qquad \log p(Y \mid G, \phi) = -\kappa E(G) + \sum_{q=1}^{M} \log\big(\kappa W_q(G)\big). \tag{30}$$

This is a density for the observed positions and allele patterns, not a likelihood conditioned on the number of SNPs. The exposure integral includes invariant regions, so branches in regions without SNPs still affect the likelihood. Mutation placements are marginalized rather than sampled as additional latent actions.

If any $W_q(G) = 0$, the corresponding derived-carrier pattern cannot be explained by a single mutation and the likelihood is zero. For $M = 0$, the product is empty and the likelihood reduces to $e^{-\kappa E(G)}$. If $\mu = 0$, a dataset with any SNP has zero likelihood, whereas a dataset without SNPs has likelihood one. All likelihood statements below use the same convention for fixed data-only factors.

B.3 Incremental Likelihood Computation

The implementation computes the same likelihood during construction using local messages and accumulated exposure. Messages are maintained only at SNPs carried by a lineage. For lineage i at SNP q, let (a_{iq}, d_{iq}, m_{iq}) denote its message at the lineage's creation time. The indicators a_{iq} and d_{iq} specify whether its sampled descendants are uniformly ancestral or uniformly derived. The quantity m_{iq} is the total length of single-mutation placements in the constructed local subtree that explain its observed descendant alleles when the allele at its root is ancestral. It excludes the mutation-rate factor and the no-additional-mutation factor. At a sampled leaf with observed allele $y_{iq} \in \{0, 1\}$,

$$(a_{iq}, d_{iq}, m_{iq}) = (1 - y_{iq}, y_{iq}, 0). \tag{31}$$

Propagation and merging. Suppose a child lineage created at time u_i is attached to a parent at time $u > u_i$. Its branch duration is $\delta_i = u - u_i$, which can span several global events; it is not necessarily the latest event's waiting time. Propagating the message along this branch gives

$$a'_{iq} = a_{iq}, \qquad d'_{iq} = d_{iq}, \qquad m'_{iq} = m_{iq} + \delta_i d_{iq}. \tag{32}$$

The added term accounts for a mutation on the new branch, which explains the observations only if every descendant is derived. For two children carrying the same SNP, their propagated messages merge as

$$a_q = a'_{1q}a'_{2q}, \qquad d_q = d'_{1q}d'_{2q}, \qquad m_q = m'_{1q}a'_{2q} + a'_{1q}m'_{2q}. \tag{33}$$

With one mutation in total, it must occur in one child's ancestry while the other child's descendants remain ancestral; summing these two possibilities gives the recurrence for m_q. If only one child carries the SNP, its propagated message is copied. Equivalently, an absent child contributes the algebraic identity $(1, 1, 0)$, with absence determined from material coverage rather than message values.

At recombination, each parent receives the child's propagated messages restricted to its own half-open material intervals. Every SNP is assigned to exactly one parent. No mutation placement is sampled or duplicated. When a newly created lineage first contains all sampled descendants at SNP q, its message m_q is recorded as W_q. This value is recorded only once, so subsequent retained ancestry above the local MRCA does not alter the SNP contribution.

Accumulating exposure. Let E_s be the exposure of the ancestry connections already constructed in state s, initially zero. For each new parent–child connection from child i to parent v, let A_{vi} be the material inherited along that connection. Its exposure increment is

$$\Delta E_{vi} = (u_v - u_i) \int_{A_{vi}} \mathbf{1}\{d_i(x) \neq [n]\}\, dx. \tag{34}$$

Here $d_i(x)$ is the descendant set from Appendix A.1, distinct from the scalar derived-allele indicator d_{iq}. At coalescence, both children's increments are added; overlapping material contributes once for each distinct child branch. At recombination, the two parent connections partition the child's material, so their increments count its branch exposure exactly once. The indicator excludes stems above locally completed ancestry. Since the integral covers physical material rather than SNP locations, it also includes exposure in invariant intervals. Unattached portions of active lineages are accounted for when their parent connections are created, rather than being extended after every global event.

Partial potential and terminal equality. Let $Q_{\mathrm{resolved}}(s)$ contain the SNPs whose local ancestry has completed. The likelihood potential used in the state-flow model is

$$\ell_{\mathrm{partial}}(s) = -\kappa E_s + \sum_{q \in Q_{\mathrm{resolved}}(s)} \log\big(\kappa W_q\big). \tag{35}$$

It is zero initially. Before termination, it is a deterministic potential of the constructed history, not a marginal likelihood obtained by integrating over unresolved ancestry. In particular, unresolved SNPs do not yet contribute their logarithmic factors.

At termination, every local genealogy is complete, all SNPs have been recorded, and every branch below a local MRCA has been counted once. The message recurrences sum precisely the compatible mutation-placement lengths in Eq. 29; the exposure increments sum to Eq. 28. Consequently,

$$E_{s_T} = E(G), \qquad \ell_{\mathrm{partial}}(s_T) = \log p(Y \mid G, \phi). \tag{36}$$

For example, with two samples coalescing at scaled time u over a region of length L and one singleton SNP, the two leaf branches give $E(G) = 2Lu$ and $W_1(G) = u$. Both the incremental and completed-tree calculations therefore give $-2\kappa Lu + \log(\kappa u)$. Combining the terminal potential with the accumulated event log prior gives the log reward in Eq. 9, before adding the constant training offset in Eq. 57.

C Architecture and Training

C.1 Lineage and State Representations

ARGFlow encodes each partial ancestral history using the genetic information and ancestral material carried by its active lineages. Let $\mathcal{L}(s) = \{\ell_1, \ldots, \ell_K\}$ denote the active lineages in state s, and let n, M, and L denote the number of sampled haplotypes, observed variant sites, and genomic

sequence length, respectively. The encoder first constructs an embedding for each lineage and then jointly processes these embeddings to obtain contextual lineage representations and a global state representation.

Each lineage is represented by two collections of features: *SNP features*, which describe the observed variation and its compatibility with the partial ancestry, and *ancestral-material features*, which describe genomic coverage and descendant relationships. This separation allows the representation to account for ancestral material in regions without observed variants.

For each SNP j carried by lineage ℓ_i, we construct

$$\mathbf{x}_{ij} = \left[a_{ij},\, d_{ij},\, \log(1 + m_{ij}),\, \frac{p_j}{L},\, \frac{p_j - p_{j-1}}{L},\, \frac{p_{j+1} - p_j}{L},\, c_{ij},\, \mathbf{y}_j,\, \mathbf{b}_{ij} \right] \in \mathbb{R}^{7+2n}. \tag{37}$$

Here, p_j is the physical SNP position, with boundary values $p_0 = 0$ and $p_{M+1} = L$. The vector $\mathbf{y}_j \in \{0,1\}^n$ contains the observed alleles across sampled haplotypes, and $\mathbf{b}_{ij} \in \{0,1\}^n$ identifies the samples descending from lineage ℓ_i at that position. The indicator c_{ij} equals one when this descendant set contains all sampled haplotypes. The quantities (a_{ij}, d_{ij}, m_{ij}) are the incremental infinite-sites messages described in Appendix B.3: a_{ij} and d_{ij} indicate whether the descendants are uniformly ancestral or derived, respectively, while m_{ij} accumulates branch length compatible with a single mutation explaining their observed alleles.

Ancestral material is partitioned into intervals on which the descendant set is constant. For an interval $[u_{ir}, v_{ir})$, its feature vector is

$$\mathbf{z}_{ir} = \left[\frac{u_{ir}}{L},\, \frac{v_{ir}}{L},\, \frac{v_{ir} - u_{ir}}{L},\, c_{ir},\, \mathbf{b}_{ir} \right] \in \mathbb{R}^{4+n}, \tag{38}$$

where $\mathbf{b}_{ir}$ is the descendant indicator vector and c_{ir} indicates whether the interval has ancestry from all sampled haplotypes. These features explicitly encode physical interval boundaries, including coverage between SNPs.

Separate multilayer perceptrons map SNP and interval features into d-dimensional embeddings. Within each lineage, SNP embeddings are aggregated using their elementwise mean and maximum. Interval embeddings are aggregated using a length-weighted mean and an elementwise maximum:

$$\mathbf{u}_i^{\mathrm{SNP}} = \left[\mathrm{mean}_j\, \phi_{\mathrm{SNP}}(\mathbf{x}_{ij}),\, \max_j\, \phi_{\mathrm{SNP}}(\mathbf{x}_{ij}) \right], \tag{39}$$

$$\mathbf{u}_i^{\mathrm{mat}} = \left[\frac{\sum_r w_{ir} \phi_{\mathrm{mat}}(\mathbf{z}_{ir})}{\sum_r w_{ir}},\, \max_r\, \phi_{\mathrm{mat}}(\mathbf{z}_{ir}) \right], \tag{40}$$

where $w_{ir} = (v_{ir} - u_{ir})/L$. Both SNP pooling outputs are set to zero for lineages carrying no observed variants.

The pooled features are concatenated with seven lineage-level scalars: the logarithmically transformed creation time and elapsed time since creation, total material length and enclosing genomic span normalized by L, logarithmically transformed counts of descendant intervals and carried SNPs, and an indicator for the presence of at least one SNP. All logarithmic transformations use $\log(1+x)$, and times are expressed in units of $2N_e$ generations. A learned projection produces the initial lineage embedding,

$$\mathbf{h}_i^{(0)} = \phi_{\mathrm{lin}}\left(\left[\mathbf{u}_i^{\mathrm{SNP}},\, \mathbf{u}_i^{\mathrm{mat}},\, \mathbf{q}_i \right] \right) \in \mathbb{R}^d, \tag{41}$$

where $\mathbf{q}_i$ denotes the vector of lineage-level scalar features.

A learned summary token is prepended to the lineage embeddings, and the resulting sequence is processed by a Transformer:

$$\left[\widetilde{\mathbf{h}}_s,\, \mathbf{h}_1, \ldots, \mathbf{h}_K \right] = \mathrm{Transformer}\left(\left[\mathbf{h}_{\mathrm{summary}},\, \mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_K^{(0)} \right] \right). \tag{42}$$

No positional encoding is applied to lineage order. Consequently, reordering active lineages reorders their output embeddings while leaving the summary representation unchanged. Genomic position remains available through the SNP and interval features. Padding is masked when batching states with different numbers of active lineages.

Finally, the summary output is concatenated with a 12-dimensional vector of global state features and projected to obtain $\mathbf{h}_s \in \mathbb{R}^d$. These features comprise the current time, active-lineage count,

total active material, fraction of resolved SNPs, accumulated mutation exposure, accumulated log prior, partial log likelihood, total CwR event rate, relative coalescence and recombination rates, and scaled mutation and recombination parameters. Coordinates and material lengths are normalized by L; the log prior and partial log likelihood are normalized by L and $\max(M, 1)$, respectively. The time, lineage count, mutation exposure, and total event rate use $\log(1 + x)$ transformations. The resulting lineage embeddings $\{\mathbf{h}_i\}_{i=1}^{K}$ and state embedding $\mathbf{h}_s$ provide the representations used by the prediction heads described in Appendix C.2.

C.2 Policy and Flow Heads

The forward policy is parameterized by prediction heads operating on the contextual lineage embeddings $\mathbf{h}_i$ and state embedding $\mathbf{h}_s$ defined in Appendix C.1. We use embeddings of dimension $d = 64$, produced by a six-layer Transformer with four attention heads and feedforward width 128. The policy first selects an event type, then the participating lineage or lineage pair, followed by a breakpoint for recombination events and a positive waiting time.

Event-type head. The event-type head predicts two residual logits, $r_{\mathrm{coal}}(s)$ and $r_{\mathrm{recomb}}(s)$. These modify baseline weights derived from the available CwR actions:

$$P_\theta(e \mid s) = \frac{H_e(s) \exp(r_e(s))}{\sum_{e'} H_{e'}(s) \exp(r_{e'}(s))}. \tag{43}$$

Let $\mathcal{C}(s)$ denote the set of compatible coalescence pairs, $\mathcal{R}(s)$ the set of recombinable lineages, and B_i the number of valid physical breakpoint links for lineage i. In units of $2N_e$ generations, the baseline weights are

$$H_{\mathrm{coal}}(s) = |\mathcal{C}(s)|, \qquad H_{\mathrm{recomb}}(s) = 2N_e r \sum_{i \in \mathcal{R}(s)} B_i. \tag{44}$$

An event type with zero available weight receives zero probability. These weights incorporate the policy's compatibility constraints; the physical rates used to evaluate the CwR prior remain those defined in Appendix B.1.

Candidate-selection head. For a coalescence candidate (i, j), we construct the symmetric context

$$\mathbf{c}_{ij} = [\mathbf{h}_i + \mathbf{h}_j,\ |\mathbf{h}_i - \mathbf{h}_j|,\ \mathbf{h}_i \odot \mathbf{h}_j,\ \mathbf{h}_s]\,. \tag{45}$$

For a recombination candidate i, we use

$$\mathbf{c}_i = [\mathbf{h}_i,\ \mathbf{0},\ \mathbf{0},\ \mathbf{h}_s]\,. \tag{46}$$

Both contexts have dimension $4d = 256$ and are scored by the same scalar-valued network f_θ. Conditional candidate probabilities are

$$P_\theta(i, j \mid s, \mathrm{coal}) \propto \exp(f_\theta(\mathbf{c}_{ij}))\,, \qquad (i, j) \in \mathcal{C}(s), \tag{47}$$

$$P_\theta(i \mid s, \mathrm{recomb}) \propto B_i \exp(f_\theta(\mathbf{c}_i))\,, \qquad i \in \mathcal{R}(s). \tag{48}$$

Normalization is performed over the corresponding candidate set. The coalescence context is invariant to exchanging i and j, while the recombination baseline accounts for differences in the number of eligible links. The final layers of the event and candidate heads are initialized to zero, so their initial probabilities follow these baseline weights.

Breakpoint head. Conditional on a selected recombination lineage, the breakpoint head defines a four-component mixture of discretized, truncated logistic distributions. Let $\{a, \ldots, z\}$ be its valid integer breakpoint links and $B = z - a + 1$. The head receives $\mathbf{c}_i$ together with the span features

$$\left[\frac{a}{L},\ \frac{z}{L},\ \frac{B}{\max(L - 1, 1)}\right]. \tag{49}$$

It predicts mixture logits u_k, location parameters v_k, and scale parameters w_k, transformed as

$$\pi_k = \mathrm{softmax}(\mathbf{u})_k, \qquad \xi_k = a - \frac{1}{2} + B\sigma(v_k), \qquad s_k = 0.1 + B\sigma(w_k), \tag{50}$$

where σ is the sigmoid function. Writing

$$F_k(x) = \sigma\left(\frac{x - \xi_k}{s_k}\right), \tag{51}$$

the probability of breakpoint b is

$$P_\theta(b \mid s, i, \text{recomb}) = \sum_{k=1}^{4} \pi_k \frac{F_k(b + \frac{1}{2}) - F_k(b - \frac{1}{2})}{F_k(z + \frac{1}{2}) - F_k(a - \frac{1}{2})}. \tag{52}$$

This distribution is normalized over all eligible physical links, including links between observed SNPs and within gaps in ancestral material.

Waiting-time head. The waiting-time head receives the selected candidate context concatenated with four scalars: $\log(1 + t)$, $\log \Lambda(s)$, a recombination indicator, and the normalized breakpoint b/L, set to zero for coalescence. Here, $\Lambda(s)$ is the total physical CwR event rate before mutation-compatibility filtering. The head predicts a four-component Gamma mixture,

$$p_\theta(\Delta t \mid s, e, \text{event parameters}) = \sum_{k=1}^{4} \omega_k \, \text{Gamma}\left(\Delta t; \alpha_k, \beta_k\right), \qquad \Delta t > 0, \tag{53}$$

using the shape–rate convention. Under the configured parameterization,

$$\omega_k = \text{softmax}(\mathbf{u}^t)_k, \qquad \alpha_k = \exp(v_k^t), \qquad \beta_k = \Lambda(s) \exp(g_k^t)\alpha_k. \tag{54}$$

Thus, component k has mean $\exp(-g_k^t)/\Lambda(s)$, allowing its mean and shape to vary separately. Initially, mixture weights are uniform, shapes are $1, 2, 3, 4$, and all component means equal $1/\Lambda(s)$. Mixture-component selection is internal to sampling; policy densities marginalize over all components. The target waiting-time prior remains exponential with rate $\Lambda(s)$.

State-flow head. The state-flow model uses a separate, permanently frozen copy of the complete encoder at initialization. Denote its state embedding by $\overline{\mathbf{h}}_s$ and the 12 global state features from Appendix C.1 by $\mathbf{v}(s)$. A trainable scalar head receives their concatenation. For nonterminal states, the log flow is

$$\begin{aligned} \log F_\phi(s) = C + \log p_{\text{prior}}(s) + \ell_{\text{partial}}(s) + \eta(s)(b_0 - C) \\ + \sigma_F f_\phi\big([\overline{\mathbf{h}}_s, \mathbf{v}(s)]\big), \end{aligned} \tag{55}$$

where $\log p_{\text{prior}}(s)$ is the accumulated event log density and $\ell_{\text{partial}}(s)$ is the incremental likelihood potential from Appendix B.3. The material-dependent coefficient is

$$\eta(s) = \frac{A(s)/L - 1}{n - 1}, \tag{56}$$

with $A(s)$ the total material length summed over active lineages. The fixed initialization center b_0 and empirical output scale σ_F are described in Appendix C.3.

We use the constant log-reward offset $C = 3000$, corresponding to the scaled reward

$$\widetilde{R}(G) = e^C R(G), \tag{57}$$

which preserves the normalized target distribution. Terminal flows are set exactly to $\log \widetilde{R}(G)$. The source flow is evaluated by the same state-flow model. Flow gradients update the flow head, while the frozen encoder remains unchanged.

Head architectures. The prediction-head architectures are summarized in Table 3. Hidden widths list each hidden layer; every head ends with a linear output layer. Dropout is disabled throughout the encoder and prediction heads.

Table 3: Architecture of the ARGFlow policy and state-flow heads.

Head	Input dim.	Hidden widths	Activation	Output dim.
Event type	64	128	SiLU	2
Shared candidate scorer	256	128	SiLU	1
Breakpoint mixture	259	128, 128, 128, 128, 64	SiLU	12
Waiting-time mixture	260	128, 128, 128, 128	ReLU	12
State-flow residual	76	128	SiLU	1

C.3 Training Objective and Optimization

We jointly train the forward policy and state-flow head using Subtrajectory Balance (SubTB), supplemented by a full-trajectory balance loss. Let $\tau = (s_0, a_0, s_1, \ldots, a_{T-1}, s_T)$ denote a complete ARG construction trajectory. For a contiguous subtrajectory from s_m to s_n, the log-balance residual is

$$\delta_{m,n}(\tau) = \log F_\phi(s_m) + \sum_{t=m}^{n-1} \log P_\theta(a_t \mid s_t) - \log F_\phi(s_n). \tag{58}$$

The backward-policy contribution vanishes because every non-initial state has a unique chronological predecessor, giving $P_B(s_t \mid s_{t+1}) = 1$. Each forward-policy term includes the discrete action probabilities and continuous waiting-time density described in Appendix C.2. Terminal flows are fixed to the scaled posterior reward,

$$\begin{aligned} \log F_\phi(s_T) &= \log \widetilde{R}(G) \\ &= C + \log p(Y \mid G, \phi) + \log p_{\mathrm{CwR}}(G \mid \phi), \end{aligned} \tag{59}$$

where the constant C preserves the normalized target distribution.

Subtrajectory Balance. The SubTB objective includes every nonempty contiguous subtrajectory:

$$\mathcal{L}_{\mathrm{SubTB}}(\tau) = \frac{\displaystyle\sum_{0 \le m < n \le T} \lambda^{n-m-1} \delta_{m,n}(\tau)^2}{\displaystyle\sum_{0 \le m < n \le T} \lambda^{n-m-1}}, \qquad \lambda = 0.9. \tag{60}$$

Geometric weighting emphasizes shorter subtrajectories while retaining constraints spanning longer portions of the construction. We normalize the loss within each trajectory and then average equally across trajectories, preventing longer histories from receiving greater total weight simply because they contain more subtrajectories. The objective is evaluated using a linear-time recurrence over one-step balance residuals.

We additionally impose full-trajectory balance through

$$\mathcal{L}_{\mathrm{TB}}(\tau) = \left[\log F_\phi(s_0) + \sum_{t=0}^{T-1} \log P_\theta(a_t \mid s_t) - \log \widetilde{R}(G) \right]^2. \tag{61}$$

For a batch $\mathcal{B}$, the combined training objective is

$$\mathcal{L}(\theta, \phi) = \frac{1}{|\mathcal{B}|} \sum_{\tau \in \mathcal{B}} \left[\mathcal{L}_{\mathrm{SubTB}}(\tau) + 0.25\, \mathcal{L}_{\mathrm{TB}}(\tau) \right]. \tag{62}$$

Although full trajectories are already included in SubTB, the additional term provides direct source-to-terminal supervision whose weight does not decay geometrically with trajectory length.

Flow initialization. Before initial training, we calibrate the flow center and output scale using complete trajectories sampled from the initialized policy. For each initialization trajectory τ_q, we compute

$$z_q = \log \widetilde{R}(G_q) - \sum_{t=0}^{T_q-1} \log P_{\theta_0}\left(a_t^{(q)} \mid s_t^{(q)} \right). \tag{63}$$

The constants in the flow parameterization of Appendix C.2 are set to

$$b_0 = \frac{1}{N_{\text{init}}} \sum_q z_q, \tag{64}$$

$$\sigma_F = \max\left(1,\ \sqrt{\frac{1}{N_{\text{init}}} \sum_q (z_q - b_0)^2}\right). \tag{65}$$

These statistics remain fixed during optimization. The flow head's final linear layer is initialized to zero, and the policy and flow head are trained jointly from the first optimizer update.

Fresh and replay trajectories. Training batches combine newly sampled trajectories with stored complete trajectories. Once the replay buffer reaches its minimum size, all settings draw 15% of each batch from replay and 85% from the current policy. Before this threshold, the entire batch consists of fresh policy samples. No additional exploration-proposal trajectories are used, and the sampling temperature remains one.

The replay buffer allocates half its capacity to reservoir sampling over newly admitted trajectories and half to high-reward trajectories with a quota per topology signature. Signatures are constructed from local-tree topologies evaluated on a fixed genomic grid. Replay draws select either component with equal probability. Within the reservoir, trajectories are sampled uniformly; within the high-reward component, sampling first selects a grid position, then a represented local topology, and finally a trajectory carrying that topology. This procedure retains both historical samples and high-reward histories while encouraging topological diversity. Stored action sequences are replayed through the environment, and their policy scores and state flows are recomputed under the current model. Fresh and replayed trajectories enter the same balance objective without importance reweighting.

Optimization. We optimize the model using Adam with separate parameter groups for the policy encoder and policy heads, and for the state-flow head. The flow encoder remains permanently frozen, as described in Appendix C.2. Sampled actions, including continuous waiting times, are treated as fixed during differentiation; gradients propagate through their evaluated policy scores and the learned state flows. We clip the global gradient norm before each optimizer step and use `float64` arithmetic for log-probability accumulation and balance-loss evaluation.

Both trainable parameter groups follow a cosine learning-rate schedule without warm-up, decreasing to a fixed minimum fraction of their base learning rates and remaining constant thereafter. All three configurations start training from initialization with `resume_checkpoint: null`, a flow-head base learning rate of 10^{-3}, and replay fraction 0.15. Numerical hyperparameters are reported in Table 4, and training commands appear in Appendix F.

Table 4: Architecture and training hyperparameters used for ARGFlow.

Category	Hyperparameter	Value
Architecture	Embedding dimension	64
Architecture	Transformer layers	6
Architecture	Attention heads per layer	4
Architecture	Transformer feedforward width	128
Architecture	SNP/material encoder hidden width	128
Architecture	Breakpoint mixture components	4
Architecture	Breakpoint head hidden widths	128, 128, 128, 128, 64
Architecture	Waiting-time mixture components	4
Architecture	Waiting-time head hidden widths	128, 128, 128, 128
Architecture	Dropout, including attention dropout	0
Flow model	Encoder	Frozen copy of the initial encoder
Flow model	Initialization trajectories	200
Flow model	Output scale	Empirical standard deviation, lower-bounded by 1
Flow model	Log-reward offset C	3,000
Objective	SubTB weighting parameter λ	0.9
Objective	Auxiliary trajectory-balance weight	0.25
Optimization	Optimizer	Adam
Optimization	Adam (β_1, β_2)	$(0.9, 0.999)$
Optimization	Adam ϵ	10^{-8}
Optimization	Weight decay	0
Optimization	Trajectories per update	128
Optimization	Gradient accumulation steps	1
Optimization	Global gradient-norm limit	10
Optimization	Policy and encoder base learning rate	10^{-4}
Optimization	Flow-head base learning rate	10^{-3}
Optimization	Learning-rate schedule	Cosine decay
Optimization	Cosine schedule duration	6,000 updates
Optimization	Minimum learning-rate multiplier	0.1
Optimization	Learning-rate warm-up updates	0
Optimization	Separate flow-head pretraining updates	0
Replay	Fraction of trajectories drawn from replay	0.15
Replay	Buffer capacity	2,048 entries across both components
Replay	Reservoir / high-reward capacity allocation	50% / 50%
Replay	Genomic positions used for topology signatures	16
Replay	Maximum entries per signature in the high-reward component	4
Replay	Minimum unique buffer entries before replay	128
Sampling	Additional exploration-proposal fraction	0
Sampling	Policy temperature	1
Training limits	Maximum events per trajectory	10,000
Training limits	Configured total optimizer-update limit	10,000
Reproducibility	Training seed	7

D Experimental Details

D.1 Simulation Setup

We simulate phased haplotypes using `msprime` version 1.3.4 under a constant-size coalescent-with-recombination model. Each dataset contains $n = 10$ haplotypes, corresponding to five diploid individuals, over a genomic region of length $L = 25{,}000$ bp. We fix the diploid effective population size to $N_e = 10{,}000$ and the mutation rate to $\mu = 10^{-8}$ per base per generation. We consider three recombination rates, $r \in \{10^{-8}, 5\times10^{-9}, 2.5\times10^{-9}\}$, corresponding to mutation-to-recombination ratios $\mu/r \in \{1, 2, 4\}$.

Ancestry is simulated on a discrete genome, with recombination breakpoints between bases. Mutations are subsequently generated under an infinite-sites model with continuous genomic positions and nucleotide states. Each segregating site therefore contains a single mutation. We retain all

Table 5: Simulation settings. All datasets contain 10 haplotypes over 25 kb, with $N_e = 10{,}000$ and $\mu = 10^{-8}$ per base per generation.

μ/r	r	SNPs	Ancestry seed	Mutation seed
1	10^{-8}	16	10001	10002
2	5×10^{-9}	45	20001	20002
4	2.5×10^{-9}	31	40001	40002

simulated variants and record their ancestral alleles, phased genotypes, and exact positions. The generating genealogies are retained for evaluation.

We generate one independent dataset per recombination setting. Table 5 reports the simulation seeds and realized numbers of segregating sites. Because each setting uses an independently simulated ancestry and mutation realization, differences between settings reflect both the recombination rate and variation in the realized dataset.

ARGFlow uses the exact continuous SNP positions. The baseline inputs use the integer coordinates obtained when exporting the simulated data to VCF by rounding SNP positions. No additional coordinate adjustment beyond rounding was required for these datasets. All methods are evaluated against the original simulated genealogies.

D.2 Baseline Configuration

We compare ARGFlow with ARGInfer and SINGER. Both baselines receive the observed phased haplotypes and the population size, mutation rate, and recombination rate used to generate each dataset. Population-genetic parameters are held fixed during inference.

ARGInfer receives the haplotype matrix, ancestral alleles, and SNP positions extracted from the simulated VCF. The ancestral allele at each site is taken from the known simulated ancestral state. The ARGInfer protocol discards the first $200{,}000$ iterations and retains an ARG every $1{,}000$ iterations thereafter. We evaluate the first $1{,}000$ retained ARGs per dataset, corresponding to iterations $201{,}000$ through $1{,}200{,}000$. The reproduction launcher runs $1{,}200{,}001$ iterations to produce this ensemble.

For SINGER, we use version 0.1.9-beta with the phased VCF, mutation rate $\mu = 10^{-8}$, and recombination-to-mutation ratios $r/\mu \in \{1, 0.5, 0.25\}$. The inference interval spans the complete 25 kb region. We set the ancestral-state polarization probability to 0.99 and the random seed to 7. The sampling protocol requests $1{,}100$ saved ARGs with a thinning interval of 500 MCMC iterations. We discard the first 100 saved ARGs and retain the remaining $1{,}000$.

Baseline outputs are converted to tree sequences for a common evaluation procedure. Sample identities, genomic lengths, and observed allele patterns are checked during conversion. Local genealogies are simplified to the sampled haplotypes, removing representation-specific unary ancestry. Times are interpreted in generations and subsequently divided by $2N_e$ for reporting.

D.3 Posterior Sampling

ARGFlow is trained separately on each observed dataset. We evaluate $1{,}000$ complete ARGs per dataset, generated with the trained policy parameters fixed. Each ARG is constructed by sampling from the initial state until ancestry is complete. Sampling uses temperature one and batches of at most 64 trajectories. Each trajectory is generated independently conditional on the fixed policy, without MCMC burn-in or thinning.

The reported ARGFlow summaries use direct policy samples with equal weights. For each generated ARG, we verify complete local ancestry and compatibility with the observed derived-allele sets. We also recompute the infinite-sites likelihood from the completed tree sequence and check its agreement with the incremental likelihood accumulated during construction.

The comparisons use the first $1{,}000$ eligible draws per method for each dataset, in generation order for ARGFlow and saved-iteration order after burn-in for the baselines, with equal weights within

each evaluated ensemble. These counts describe the evaluated ensemble sizes. The MCMC samples may retain autocorrelation.

D.4 Evaluation Metrics

We evaluate reconstruction against the generating genealogies and agreement between posterior summaries. Let $P = \binom{n}{2} = 45$ denote the number of unordered haplotype pairs. For pair p at genomic position x, let $t_p^\star(x)$ denote the true time to the most recent common ancestor (TMRCA), and let $t_p^{(k)}(x)$ denote its value in posterior draw k. All TMRCA quantities below are expressed in units of $2N_e$ generations.

Genomic averages are computed using exact interval lengths. We align the tree-sequence breakpoints required by each comparison and integrate the resulting piecewise-constant quantities over $[0, L)$. Each haplotype pair receives equal weight, and each retained draw receives equal weight within its method.

The posterior-mean TMRCA is

$$\bar{t}_p(x) = \frac{1}{S} \sum_{k=1}^{S} t_p^{(k)}(x), \tag{66}$$

where S is the number of retained draws. Reconstruction error is measured by

$$\mathrm{RMSE}_{\mathrm{TMRCA}} = \left[\frac{1}{LP} \sum_{p=1}^{P} \int_0^L \left(\bar{t}_p(x) - t_p^\star(x) \right)^2 \, dx \right]^{1/2}. \tag{67}$$

The square root is taken after averaging squared errors across pairs and genomic positions. For the reconstruction plots, we also report Pearson correlation between posterior-mean and true TMRCA using the same genomic-span and pair weights.

Topological reconstruction is evaluated using rooted Robinson–Foulds (RF) distance. Let $\mathcal{C}^{(k)}(x)$ and $\mathcal{C}^\star(x)$ denote the sets of nontrivial descendant clades in a sampled and true local genealogy, respectively. Singleton clades and the full sample set are excluded, and duplicate clades induced by unary nodes are counted once. We report

$$\mathrm{RF} = \frac{1}{LS} \sum_{k=1}^{S} \int_0^L \left| \mathcal{C}^{(k)}(x) \triangle \mathcal{C}^\star(x) \right| \, dx, \tag{68}$$

where $\triangle$ denotes symmetric difference. This is the raw rooted RF distance, whose maximum for two fully binary trees with 10 leaves is 16.

To evaluate posterior clade probabilities, define

$$q_c(x) = \frac{1}{S} \sum_{k=1}^{S} \mathbf{1}\left\{ c \in \mathcal{C}^{(k)}(x) \right\}, \qquad y_c(x) = \mathbf{1}\{ c \in \mathcal{C}^\star(x) \}. \tag{69}$$

Let $\mathcal{U}$ contain all subsets of the sampled haplotypes with sizes from 2 to $n-1$. Its cardinality is $|\mathcal{U}| = 2^n - n - 2 = 1{,}012$. The clade Brier score is

$$\mathrm{Brier} = \frac{1}{L|\mathcal{U}|} \int_0^L \sum_{c \in \mathcal{U}} \left(q_c(x) - y_c(x) \right)^2 \, dx. \tag{70}$$

Using this fixed clade universe gives every method the same normalization. Clades absent from both the sampled genealogies and the truth contribute zero.

Temporal uncertainty is summarized using equal-tail 90% credible intervals. Let $\ell_p(x)$ and $u_p(x)$ be the empirical 5th and 95th percentiles of the sampled TMRCA values, computed by linear interpolation between ordered draws. We report

$$\mathrm{Coverage}_{0.90} = \frac{1}{LP} \sum_{p=1}^{P} \int_0^L \mathbf{1}\left\{ \ell_p(x) \le t_p^\star(x) \le u_p(x) \right\} \, dx, \tag{71}$$

$$\mathrm{Width}_{0.90} = \frac{1}{LP} \sum_{p=1}^{P} \int_0^L \left(u_p(x) - \ell_p(x) \right) \, dx. \tag{72}$$

Coverage and width are interpreted jointly. Because genomic positions and haplotype pairs are dependent and only one dataset is simulated per setting, coverage describes these realized datasets.

For posterior-summary agreement, we compare ARGFlow separately with each baseline. Let $\widehat{\Pi}^A_{p,x}$ and $\widehat{\Pi}^B_{p,x}$ be the empirical TMRCA distributions from two methods for the same pair and position. We compute

$$\overline{W}_1(A,B) = \frac{1}{LP}\sum_{p=1}^{P}\int_0^L W_1\left(\widehat{\Pi}^A_{p,x}, \widehat{\Pi}^B_{p,x}\right)\, dx, \tag{73}$$

where

$$W_1(\Pi_1,\Pi_2) = \int_0^1 |Q_1(v) - Q_2(v)|\, dv \tag{74}$$

and Q_1, Q_2 are the corresponding quantile functions. The empirical distance is evaluated exactly, including when the methods have different numbers of retained draws. The density curves in the posterior-comparison figure pool across pairs and genomic positions, whereas the annotated distances average the local comparisons defined above. Smoothing is used only for visualization. The TMRCA axis is linear up to 2 and logarithmic thereafter, while density is expressed per unit linear TMRCA.

Agreement in clade probabilities is measured by

$$\mathrm{RMSE}_{\mathrm{clade}}(A,B) = \left[\frac{1}{L|\mathcal{U}|}\int_0^L \sum_{c\in\mathcal{U}}\left(q^A_c(x) - q^B_c(x)\right)^2\, dx\right]^{1/2}. \tag{75}$$

This comparison uses the same fixed clade universe as the Brier score and measures agreement between methods without using true-clade membership. ARGInfer and SINGER serve as reference samplers; agreement with their marginal summaries does not establish equality of the full ARG posterior.

For ARGFlow sampling diagnostics, we additionally compute target-to-policy log-density ratios

$$\delta_k = \log\widetilde{R}(G_k) - \log P_\theta(\tau_k), \qquad \log P_\theta(\tau_k) = \sum_t \log P_\theta(a_t^{(k)} \mid s_t^{(k)}). \tag{76}$$

We summarize their dispersion to diagnose variation in the target-to-policy density ratio. This diagnostic does not alter the equal-weight posterior summaries above.

E Additional Posterior Diagnostics

We examine the relationship between the learned sampling density and the unnormalized target density, complementing the marginal posterior comparisons in the main text. Unlike Figure 2, which compares posterior-mean coalescence times with simulated truth, this analysis tests relative densities of complete ancestral histories. We use the first $1{,}000$ temperature-one policy draws per dataset, selected in generation order, for this diagnostic. The stored chronological actions retain the discrete choices and continuous waiting times needed to evaluate the policy density. Likelihoods recomputed from completed histories agree with the incremental likelihoods to within 10^{-7} log units. For a complete chronological trajectory τ producing an ARG G, define

$$u(G) = \log p(Y \mid G) + \log p(G), \qquad v_\theta(\tau) = \sum_{t=0}^{T-1}\log P_\theta(a_t \mid s_t). \tag{77}$$

Under the chronological-history representation, exact reward-proportional sampling implies $v_\theta(\tau) = u(G) - \log Z$. We therefore compare these quantities using a slope-one reference line with a fitted intercept,

$$\widehat{b} = \frac{1}{S}\sum_{k=1}^{S}\left[v_\theta(\tau_k) - u(G_k)\right]. \tag{78}$$

The residual

$$e_k = v_\theta(\tau_k) - u(G_k) - \widehat{b} \tag{79}$$

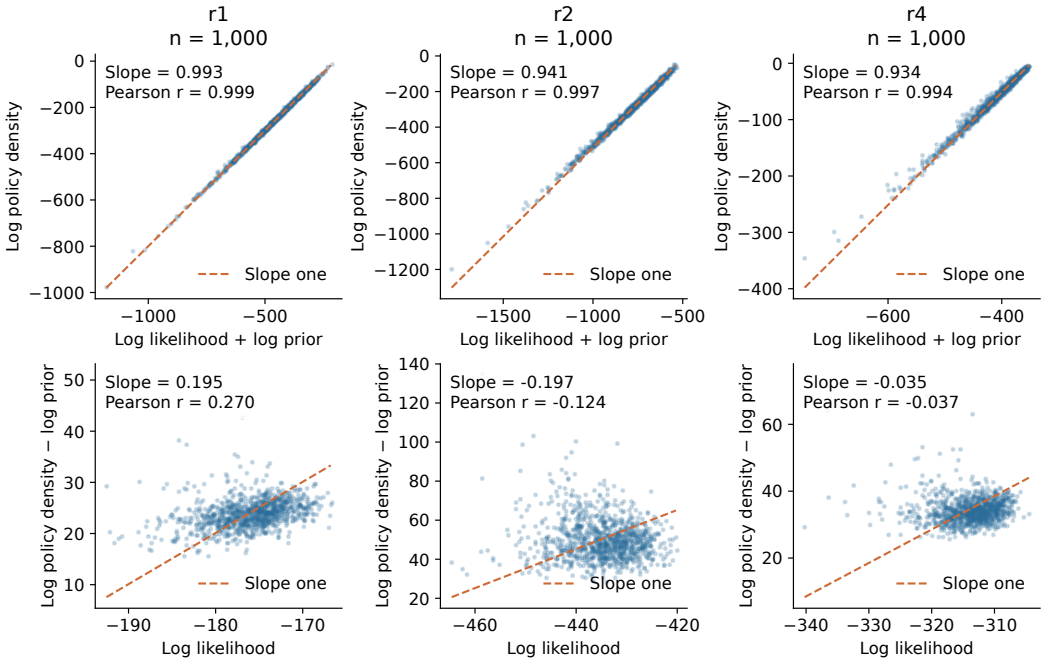


Figure 4: Full-history density agreement on 1,000 policy draws per dataset. Upper panels compare policy log density with log likelihood plus log prior; lower panels remove the shared prior contribution. Dashed lines have slope one and a fitted intercept shared between the two views within each dataset. Annotations give ordinary least-squares slopes and Pearson correlations. High raw correlations coexist with substantial departures from the prior-relative slope-one relationship.

Table 6: Raw density-fit diagnostics on 1,000 policy draws per dataset. Slopes have ideal value one. Each row summarizes draws from one trained policy, rather than independent training replicates.

Dataset	Slope	Pearson r
r1	0.993	0.999
r2	0.941	0.997
r4	0.934	0.994

measures deviation from a common normalization across sampled histories. We summarize this deviation by its root mean squared value and report the fitted regression slope and Pearson correlation as complementary diagnostics. A high correlation alone does not establish the required slope-one relationship.

To examine the contribution of the sequence data separately from the prior, we also compare

$$v_\theta(\tau) - \log p(G) \quad \text{against} \quad \log p(Y \mid G). \tag{80}$$

This representation preserves the slope-one condition while removing the shared prior contribution from the two axes. The raw and prior-relative views are evaluated on the same histories, with a single fitted normalization offset within each dataset. The analysis characterizes density agreement on the evaluated samples; it does not establish coverage of histories rarely visited by the policy. Subtracting the prior from both axes leaves every slope-one residual unchanged. Consequently, the residual RMSE is identical in the two views and equals the population standard deviation of $u(G) - v_\theta(\tau)$. We report this quantity once per dataset.

Figure 4 and Table 6 show raw correlations above 0.99 in all three settings. However, prior-relative slopes are far from one, and correlations are weakly positive for r1 and negative for r2 and r4. The raw plots therefore do not establish that the likelihood-induced change from the prior has been learned accurately. Residual RMSE is smallest for r1 and largest for r2, with appreciable full-history density mismatch in every setting. Agreement of marginal summaries in the main text should be interpreted alongside these remaining discrepancies.

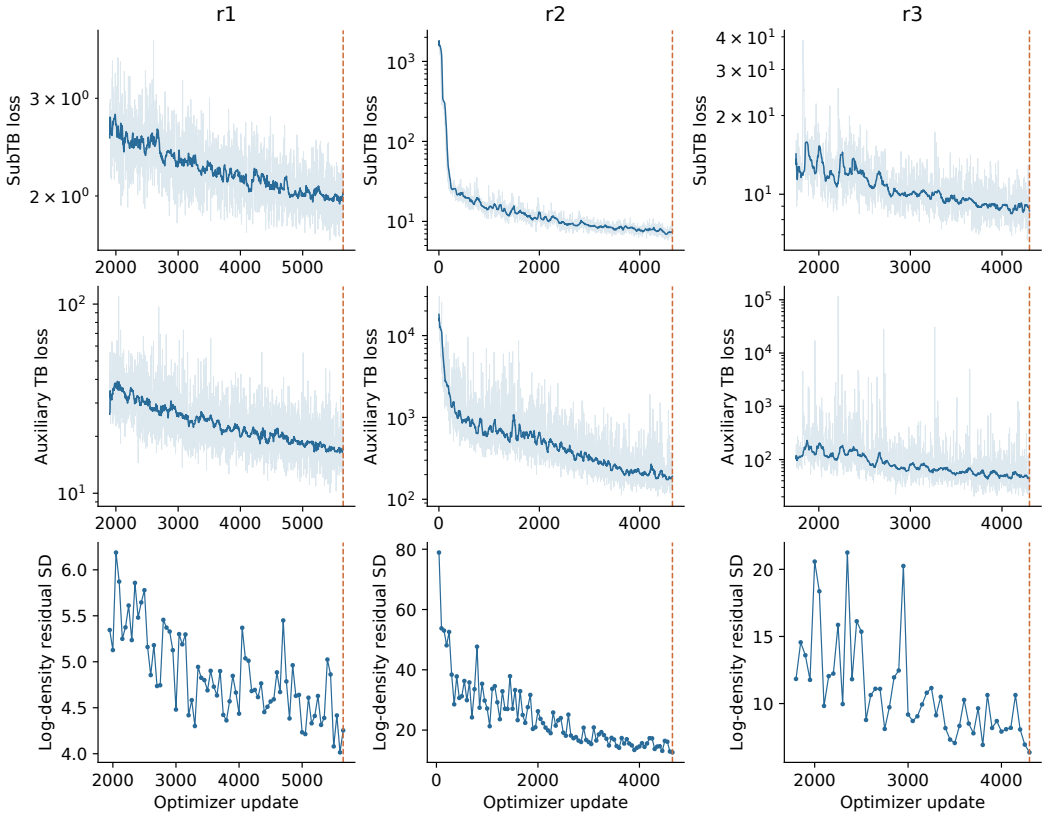


Figure 5: Training progress up to the checkpoints used in the main results. The first two rows show raw losses faintly and trailing 50-update medians as solid curves, using logarithmic vertical axes. The bottom row shows unsmoothed density-residual standard deviations on 256 policy draws per evaluation. Dashed vertical lines mark the selected checkpoints. The horizontal axes retain the recorded optimizer-update numbering.

Training progress. Figure 5 shows the SubTB and auxiliary trajectory-balance losses from the corresponding training runs, together with the standard deviation of the target-to-policy log-density ratio on 256 freshly generated policy histories at each logged evaluation. The evaluation seed is fixed at 100007; these measurements assess progress within a run, not variation across independent training seeds. Asynchronous evaluations are positioned at the evaluated checkpoint's update, rather than the later update at which their results were logged.

Both training losses generally decline over the displayed intervals. Evaluation residual standard deviation decreases from 5.35 to 4.25 between updates 1950 and 5650 for r1, from 78.91 to 12.60 between updates 50 and 4650 for r2, and from 11.83 to 6.36 between updates 1800 and 4300 for r4. The evaluation curves remain nonmonotonic, particularly for r4. Optimization improves density agreement overall, but lower loss does not imply exact proportional sampling.

Dependence on history complexity. The full-history density mismatch has a systematic association with history complexity: centered log-density residuals increase with recombination-event count, particularly for r2 and r4. Figure 6 plots the centered log-density residual against the number of recombination events, using the same dataset-wide intercept as Figure 4. The correlations are 0.174, 0.631, and 0.481 for r1, r2, and r4, respectively. Positive residuals indicate larger policy-to-target density ratios relative to the fitted common offset. This association is measured on policy-sampled histories and does not by itself establish a bias in the marginal recombination-count distribution or isolate recombination decisions as the source of error.

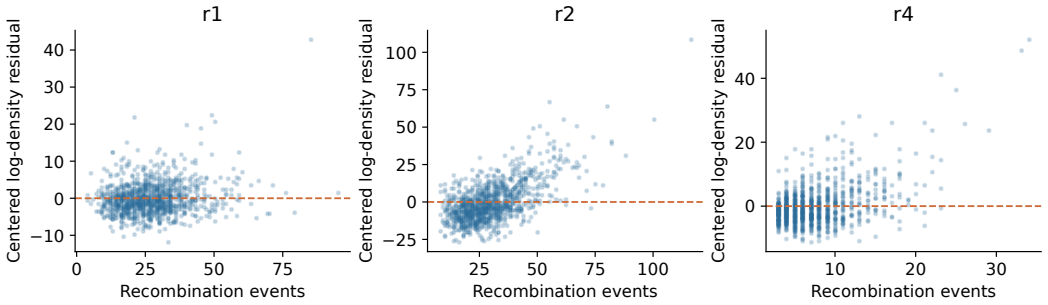


Figure 6: Full-history density residuals versus recombination-event count on the same 1,000 draws per dataset as Figure 4. Each point is one complete history. The zero line uses a single fitted normalization offset per dataset; residuals are not recentered within recombination-count groups. Residuals tend to increase with recombination count, particularly for r2 and r4, indicating structured density error among policy-sampled histories.

F Reproducibility Commands

The implementation is provided in the supplementary archive `code_supplement.zip`. Run commands from the extracted repository root after installing its dependencies. Simulated datasets and prepared baseline inputs are located in `paper/datasets/`, training configurations in `paper/config/`, and manuscript evaluation scripts in `paper/scripts/paper_datasets/`. Supplied trained models belong under `paper/checkpoints/`. Sampling and baseline inference require new output directories. The commands below use a separate `paper/outputs/reproduction/` directory to preserve supplied results.

```
REPO=/path/to/ARG-Optimise
cd "$REPO"
python -m pip install -r requirements.txt
DATA="$REPO/paper/datasets"
SCRIPTS="$REPO/validation/scripts"
RUNS="$REPO/paper/outputs/reproduction"
```

F.1 ARGFlow Training and Posterior Sampling

All three training configurations use `resume_checkpoint: null`, flow-head base learning rate 10^{-3}, and replay fraction 0.15. The following commands train a new model for each dataset. Training is stochastic, so fresh runs need not reproduce the supplied models exactly.

```
for d in r1 r2 r4; do
  python train.py --config "$REPO/paper/config/$d.yaml"
done
```

Posterior samples can be generated from a supplied or newly trained model. The commands below use the supplied models and generate 1,000 temperature-one ARGs per dataset in batches of at most 64. The sampling script writes each ensemble under `<output-dir>/<dataset>/<checkpoint-name>/`.

```
CHECKPOINTS="$REPO/paper/checkpoints"
CKPT_R1="$CHECKPOINTS/r1/checkpoint.pt"
CKPT_R2="$CHECKPOINTS/r2/checkpoint.pt"
CKPT_R4="$CHECKPOINTS/r4/checkpoint.pt"

for d in r1 r2 r4; do
  case "$d" in
    r1) checkpoint="$CKPT_R1" ;;
    r2) checkpoint="$CKPT_R2" ;;
    r4) checkpoint="$CKPT_R4" ;;
  esac
  python "$SCRIPTS/sample_final_checkpoints.py" \
    --dataset "$d" --checkpoint "$checkpoint" \
    --num-args 1000 --batch-size 64 --seed 20260925 \
    --device cuda --output-dir "$RUNS/argflow"
done
```

The generated manifests record sample identities, settings, and model and sample hashes, allowing the evaluated inputs to be traced and checked.

F.2 ARGINFER

Prepared nucleotide haplotypes, ancestral alleles, and SNP positions are included in `paper/datasets/<dataset>/arginfer_inputs/`. For newly simulated datasets without prepared inputs, run:

```
python "$SCRIPTS/prepare_arginfer_inputs.py" --datasets "$DATA"
```

The converter refuses to overwrite existing input directories. The baseline launcher runs 1,200,001 MCMC iterations, discards the first 200,000, and retains every 1,000th iteration, yielding 1,000 ARGs at iterations 201,000 through 1,200,000. The unnumbered scratch ARG is excluded from evaluation.

```
for d in r1 r2 r4; do
  OUTPUT_ROOT="$RUNS/ARGInfer" \
    bash paper/scripts/run_arginfer.sh "$d"
done
```

F.3 SINGER

We use SINGER 0.1.9-beta. Set `SINGER_DIR` to its installation directory containing `singer_master` and `convert_to_tskit`. The launcher requests 1100 saved ARGs with thinning interval 500, polarization probability 0.99, and seed 7. Conversion retains indices 100–1099, yielding 1,000 tree sequences per dataset after burn-in.

```
export SINGER_DIR=/path/to/singer-0.1.9-beta-linux-x86_64
for d in r1 r2 r4; do
  OUTPUT_ROOT="$RUNS/SINGER" \
    bash paper/scripts/run_singer.sh "$d"
done
```

F.4 EVALUATION

The evaluation scripts use 1,000 equally weighted draws per method and dataset. When an archive contains more draws, they select the first 1,000 in generation order for ARGFlow and saved-iteration order after burn-in for the baselines. The configuration sets `expected_samples: 1000` and `sample_selection: first`. Insufficient samples cause validation to fail. Evaluation outputs record the selected indices and input hashes.

To evaluate the supplied sample archives at their configured paths, run:

```
python paper/scripts/paper_datasets/evaluate.py
python paper/scripts/paper_datasets/figure_2.py
python paper/scripts/paper_datasets/figure_3.py
```

To evaluate the newly generated samples from the commands above, provide all three output roots:

```
for script in evaluate figure_2 figure_3; do
  python "paper/scripts/paper_datasets/$script.py" \
    --argflow-dir "$RUNS/argflow" \
    --singer-dir "$RUNS/SINGER" \
    --arginfer-dir "$RUNS/ARGInfer"
done
```

The scripts validate inputs before computing the reconstruction metrics and posterior comparisons. Results are written to `paper/outputs/evaluation/`, `paper/outputs/figure_2/`, and `paper/outputs/figure_3/`, respectively.