

RESEARCH ARTICLE

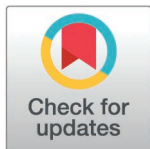
GPC: An expressive and tractable deep generative model for genetic variation data

Prateek Anand^{1*}, Anji Liu², Meihua Dang³, Boyang Fu⁴, Xinzhu Wei⁵,
Guy Van den Broeck^{1†}, Sriram Sankararaman^{1,6,7†*}

1 Department of Computer Science, University of California, Los Angeles, Los Angeles, California, United States of America, **2** Department of Computer Science, National University of Singapore, Singapore, Singapore, **3** Department of Computer Science, Stanford University, Stanford, California, United States of America, **4** Department of Biomedical Informatics, Harvard Medical School, Boston, Massachusetts, United States of America, **5** Department of Computational Biology, Cornell University, Ithaca, New York, United States of America, **6** Department of Human Genetics, University of California, Los Angeles, California, United States of America, **7** Department of Computational Medicine, University of California, Los Angeles, California, United States of America

† These authors are joint supervision authors on this work.

* prateek@cs.ucla.edu (PA); sriram@cs.ucla.edu (SS)



OPEN ACCESS

Citation: Anand P, Liu A, Dang M, Fu B, Wei X, Van den Broeck G, et al. (2026) GPC: An expressive and tractable deep generative model for genetic variation data. PLoS Genet 22(10): e1012321. <https://doi.org/10.1371/journal.pgen.1012321>

Editor: Yun Li, University of North Carolina at Chapel Hill, UNITED STATES OF AMERICA

Received: May 4, 2026

Accepted: September 17, 2026

Published: October 6, 2026

Peer Review History: PLOS recognizes the benefits of transparency in the peer review process; therefore, we enable the publication of all of the content of peer review and author responses alongside final, published articles. The editorial history of this article is available here: <https://doi.org/10.1371/journal.pgen.1012321>

Copyright: © 2026 Anand et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution,

Abstract

Generative models play an important role in population genetics, being used to generate artificial genomes (AGs) that help benchmark methods, test evolutionary hypotheses, and construct reference panels for imputation while working around data-sharing restrictions. Existing generative models of genetic variation, however, struggle to faithfully express dependencies in the data while retaining tractability and preserving privacy. We introduce Genetic Probabilistic Circuits (GPC), a deep generative model for genetic variation data based on hidden Chow-Liu trees represented as probabilistic circuits. GPC generalizes traditional hidden Markov models by allowing arbitrary tree structures over latent variables, enabling it to capture long-range dependencies among SNPs. GPC is tractable, supporting exact computation of marginal and conditional probabilities that enables both AG generation and direct genotype imputation (avoiding the need for simulating AGs). We show that GPC generates AGs that accurately reproduce population structure and linkage disequilibrium patterns across a range of length scales. Compared to other deep generative approaches, GPC consistently improves imputation accuracy, with particularly strong gains for low-frequency variants and in populations that are not well-represented in public reference panels. Finally, we show that GPC better preserves privacy of the training data, thereby providing a practical framework for AG generation in settings with limited data access.

and reproduction in any medium, provided the original author and source are credited.

Data availability statement: Code and experiments are available at github.com/sriramlab/GPC. The 1000 Genomes datasets can be freely downloaded at internationalgenome.org/ data. The UK Biobank dataset is available upon application at ukbiobank.ac.uk.

Funding: The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. PA and SS are supported in part by NIH R35 GM153406 and NSF CAREER 1943497. AL is supported in part by National University of Singapore under its Start-up Grant SUG 251RES250. XW is supported in part by NIH R35 GM150579 and NSF IIS 2435801. GVdB is supported in part by the DARPA ANSR, CODORD, and SAFRON programs under awards FA8750 23 2 0004, HR00112590089, and HR00112530141, NSF IIS 1943641, and gifts from Adobe Research, Cisco Research, Qualcomm, and Amazon. Approved for public release; distribution is unlimited.

Competing interests: The authors have declared that no competing interests exist.

Author summary

Human genetic data are essential for studying disease, but they are also deeply personal, and privacy rules often prevent researchers from sharing them. One way around this is to create artificial genomes: synthetic data that reproduce the statistical patterns of a real population without belonging to any real person. To be useful, artificial genomes have to capture how variants across the genome relate to one another, and they must not reveal information about the people whose data was used to build them. Existing methods struggle to do both at once. We describe a model that learns which positions in the genome most strongly influence each other and connects them directly, instead of assuming that only neighboring positions matter. The model is built so that its probabilities can be calculated exactly, which lets it fill in a person's missing genetic information directly rather than by first simulating artificial data. We find that this improves the accuracy of filling in missing variants, especially rare ones and those in populations that are poorly represented in public reference collections, while revealing less about the individuals used for training.

1 Introduction

Generative models of genetic variation play a central role in population genomics [1,2]. By modeling dependencies across individuals and variants, these models support core analyses such as genotype imputation [3], haplotype phasing [4], and ancestry inference [5]. They also form the foundation of simulators that generate artificial genomes (AGs) [6–10], which are widely used to test evolutionary hypotheses, estimate population genetic parameters, validate empirical results, and benchmark new methods. As the sharing of primary genetic data becomes increasingly restricted due to privacy and consent constraints, accurate simulation tools have become essential for reproducibility and equity. Trained generative models, or data simulated from them, can in principle be shared without requiring access to identifiable genomic data.

Classical population genetic simulators rely on the coalescent model [11], which generates genomes based on demographic history together with mutation and recombination rates. The coalescent with recombination [12] connects observed genetic variation to latent genealogies that change along the genome [13] and is highly expressive, but exact inference under this model is computationally challenging, motivating tractable approximations. One class approximates the coalescent as a Markovian process along the genome [14,15], forming the basis of widely used simulators such as *msprime* [9] and its extensions [10]. A second class dispenses with explicit genealogical modeling and instead directly approximates the data distribution. A prominent example is the product-of-approximate-conditionals (PAC) model [16], which naturally yields a hidden Markov model (HMM) [17]. HMM-based models have proven highly successful in applications including haplotype phasing [18,19], genotype imputation [3,20], and ancestry inference [21,22].

Motivated by the success of deep learning, recent work has explored deep generative models as an alternative approach to represent genetic variation [23–25]. Models based on generative adversarial networks (GANs) [25–28], variational autoencoders (VAEs) [23,29,30], restricted Boltzmann machines (RBMs) [25], and more recently diffusion models [31] can be more expressive than HMMs and often produce AGs that visually resemble real genomes in principal component analysis (PCA) or allele frequency summaries. However, these models have critical limitations for genomic applications. First, GANs do not define a probability distribution over the data, making likelihood-based inference impossible. RBMs define a distribution but require computing an intractable partition function over exponentially many configurations, preventing exact likelihood evaluation. VAEs define a probability distribution but the marginal likelihood requires integrating over a continuous latent space with a nonlinear decoder, which is intractable; only a lower bound (the ELBO) is accessible in practice. Diffusion models can in principle compute exact likelihoods via probability flow ordinary differential equations, but scale poorly to SNP data: existing approaches [31] require lossy dimensionality reduction (e.g., per-gene PCA) as a preprocessing step before the model operates [31], and have not been evaluated on imputation tasks. Second, none of these approaches support efficient conditional probability estimation. This limitation makes model comparison difficult and limits their use in downstream tasks such as genotype imputation, which require computing the probability of unobserved variants conditioned on observed variants. A common workaround involves generating AGs from these models, which then serve as a reference panel for imputation methods such as Impute5, Beagle, or Minimac [32–34], though this approach can introduce additional noise in the imputation process. Third, training these models is challenging due to complex optimization objectives and large hyperparameter spaces, and convergence must be assessed through subjective visual inspection rather than principled likelihood monitoring. These difficulties collectively highlight the need for deep generative models that are both expressive and tractable.

We address this need by introducing GPC (Genetic Probabilistic Circuits), a class of deep generative models that fits genetic variation data with high accuracy while supporting efficient and exact inference. To model the distribution over single nucleotide polymorphisms (SNPs) in an individual, we propose a latent variable model in which each SNP is associated with a hidden variable and the hidden variables form a tree-structured graphical model. This model, termed the hidden Chow-Liu tree (HCLT) [35], generalizes the HMM. Classical HMMs impose a chain structure where hidden variables corresponding to consecutive SNPs are adjacent in the tree. HCLTs relax this constraint by allowing arbitrary tree structures, which enables the model to capture long-range dependencies, or linkage disequilibrium (LD), across the genome [36]. This flexible structure allows HCLTs to more faithfully represent the distribution of genetic variation. While HCLTs are more expressive than HMMs, naively training and performing inference in HCLTs at the scale required for modern genomic datasets would be computationally prohibitive. Our approach represents HCLTs as probabilistic circuits (PCs) [37,38], a circuit-based framework that permits tractable inference tasks such as exact likelihood and conditional probability computation. This representation allows us to leverage stochastic gradient-based learning and GPU acceleration to train HCLTs with millions of parameters on large genomic datasets.

GPC, like GANs and RBMs, can generate AGs for use as reference panels with standard imputation tools. Unlike these methods, GPC can also perform imputation directly through efficient computation of conditional probabilities, without the need for generating AGs as an intermediate step. We show that this ability further improves imputation accuracy.

We first compare GPC to baselines on genomic regions chosen from the 1000 Genomes Project (1KG) and UK Biobank (UKBB), demonstrating that GPC achieves higher held-out log-likelihoods than simpler probabilistic models and generates AGs that accurately capture population structure and linkage disequilibrium patterns across all distance ranges. We then evaluate GPC extensively for genotype imputation. Across both datasets, GPC yields significant improvements in imputation accuracy compared to other deep generative models, with substantial improvements in settings where reference genomes from the target population are not publicly available. In such population-specific settings, GPC consistently outperforms other deep generative methods, with particularly strong gains for low-frequency variants. Finally, we find that AGs produced by GPC offer improved privacy properties compared to previous deep generative models. Overall,

our results indicate that the increased expressivity and tractability of GPC lead to more accurate and privacy-preserving models of human genetic variation.

2 Results

2.1 Methods overview

We present GPC, a deep generative model for genetic variation data. GPC is based on hidden Chow-Liu trees (HCLTs) [35], a class of latent variable models that generalize hidden Markov models (HMMs), represented as probabilistic circuits (PCs) [37,38] for efficient inference and learning.

Hidden Chow-Liu trees HCLTs are tree-structured graphical models that represent the probability distribution over observed random variables, $\mathbf{X} = (X_1, \dots, X_N)$. In the context of genetic variation across N SNPs, each observed X_n represents the genotype at SNP n (e.g., $X_n \in \{0, 1\}$ for haploid genomes). HCLTs rely on the following key designs:

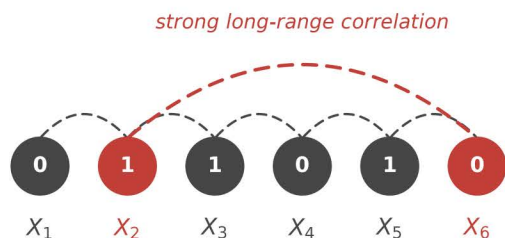
- **Hidden Variable Mapping:** For every observed variable X_n , the model introduces a corresponding hidden variable Z_n (which takes one of L discrete values). Together, these form hidden variables $\mathbf{Z} = (Z_1, \dots, Z_N)$.
- **Chow-Liu Tree Architecture:** Each observed variable X_n connects solely to its hidden counterpart Z_n . The hidden variables, however, connect to each other to form a tree structure. This tree is learned using the Chow-Liu algorithm [39] to capture the strongest pairwise correlations between variables (Fig 1).
- **Parameterization:** The entire HCLT is defined by just two sets of conditional probability distributions: the emissions $P(X_n|Z_n)$ and the tree transitions $P(Z_n|Z_{\text{Pa}(n)})$, where $\text{Pa}(n)$ denotes the parent of node n in the hidden variable tree.

The critical distinction from HMMs is structural. Classical HMMs impose a chain structure where hidden variables corresponding to consecutive SNPs are adjacent in the tree ($Z_1 \rightarrow Z_2 \rightarrow \dots \rightarrow Z_N$). This chain topology can only propagate information between distant SNPs through all intermediate variables. HCLTs relax this constraint by allowing arbitrary tree structures, so that SNPs with strong long-range correlations (such as X_2 and X_6 being highly correlated without being correlated with intermediate SNPs) can be placed close together in the tree (Fig 1). This flexible structure allows HCLTs to more faithfully capture linkage disequilibrium patterns at all length scales. Fig 1 contrasts the two topologies on the same example. Under the chain (Fig 1b), Z_2 and Z_6 are four edges apart, so the dependence between X_2 and X_6 has no direct edge and must be carried through Z_3 , Z_4 and Z_5 . Each of these latent variables has a limited number of states, so some of the correlation is lost at every step. In the learned Chow-Liu tree (Fig 1c) the same pair is adjacent, and the dependence is represented by a single edge.

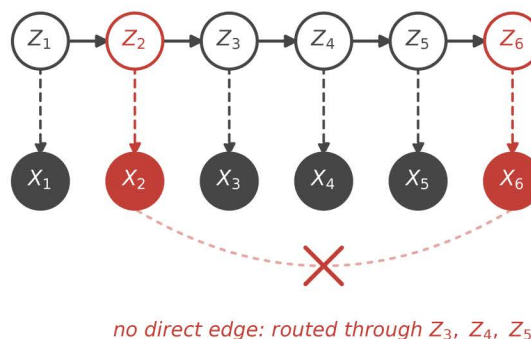
Probabilistic circuits and tractable inference While HCLTs are more expressive than HMMs, naively training and performing inference in HCLTs at the scale required for modern genomic datasets would be computationally prohibitive. Our key insight is to represent HCLTs as probabilistic circuits (PCs) [37,38]. A PC represents a joint probability distribution through a directed acyclic graph (DAG) consisting of input, sum, and product nodes. Product nodes define factorized distributions over their inputs, while sum nodes represent mixtures weighted by learnable parameters. Under structural constraints known as smoothness and decomposability, arbitrary marginal queries can be computed in time linear in the size of the circuit. This tractability is what enables GPC to perform three key operations efficiently: (1) computing likelihoods on held-out data for principled model comparison, (2) generating artificial genomes via ancestral sampling, and (3) performing direct genotype imputation through exact conditional probability computation. By representing HCLTs as equivalent PCs and training them using the PC package PyJuice [40], we can extensively parallelize Expectation-Maximization (EM) updates on GPUs, enabling efficient training even for models with over 88 million parameters (Section 4.2).

Generating AGs from GPC is performed via ancestral sampling through the circuit in time linear in the circuit size. For genotype imputation, GPC computes conditional probabilities $p(X_{\text{missing}}|X_{\text{observed}})$ directly as ratios of marginal queries, both of which can be evaluated via a single feedforward pass through the circuit. This direct imputation capability, unique

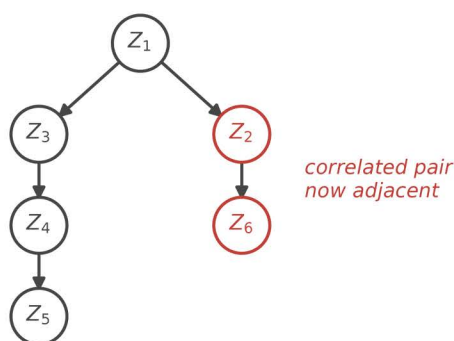
(a) Genotype data



(b) HMM: chain over latent variables



(c) Learned Chow-Liu tree



(d) HCLT: tree over latent variables

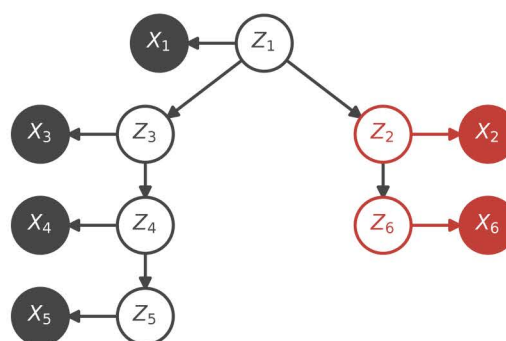


Fig 1. Generating HCLT structures given genetic data. (a) **Genotype data:** An example dataset of 6 SNPs ($X_1, \dots, X_6$), where dashed lines indicate SNPs with strong correlations. The highlighted pair X_2 and X_6 is strongly correlated despite being distant in genomic order. (b) **HMM:** The chain over latent variables imposed by a hidden Markov model, with each Z_n emitting its own X_n . Because Z_2 and Z_6 are not adjacent, the dependence between X_2 and X_6 has no direct edge: it must be routed through Z_3, Z_4 and Z_5 , and is weakened at each intervening latent variable, since each has a finite number of states. (c) **Learned Chow-Liu tree:** A Chow-Liu Tree learned from the SNP data. The tree structure places latent variables with stronger dependencies closer together, capturing the dominant correlation structure among the SNPs; the same correlated pair is now adjacent, so the dependence is carried by a single edge. (d) **HCLT:** The Hidden CLT constructed from the CLT, in which each latent variable Z_i is additionally connected to its corresponding observed SNP X_i as a child node. The resulting structure respects the dependencies among SNPs while providing a principled generative model over the observed data. The topology of a Chow-Liu tree learned from genetic data is shown in Fig 5.

<https://doi.org/10.1371/journal.pgen.1012321.g001>

to GPC among the deep generative approaches we consider, eliminates the need for intermediate AG generation and, as we show, yields consistently higher accuracy (see Section 4.2.2 for full details on sampling and conditional inference).

2.2 Data

Our experiments evaluate different aspects of GPC in terms of its ability to generate realistic AGs, its ability to perform genotype imputation, and its privacy properties. We perform the evaluations on three datasets with distinct properties: 1000 Genomes Project Phase 3 (1KG) [41], UK Biobank (UKBB) [42], and high-coverage 1KG (high-coverage 1KG) [43], with the high-coverage 1KG dataset only being used for the array-based imputation experiment (see Section 4.1 for details).

2.3 Evaluation

Baselines We compare GPC against two categories of baselines. The first category consists of deep generative models that have been previously proposed for AG generation: (1) generative adversarial networks (WGAN) and (2) restricted

Boltzmann machines (RBM) as implemented in [25]. We focus our comparison on these two methods as they are the most established deep generative approaches specifically designed and benchmarked for the local genetic structure and imputation tasks we evaluate here. More recent approaches including VAEs [29,30] and diffusion models [31] have been proposed for related but distinct tasks, and do not support efficient, tractable probabilistic inference; we leave systematic comparison with these methods to future work. For both WGAN and RBM baselines, we use the samples generated by the corresponding authors for comparison in our 1KG experiments (note that [25] did not perform train/test splits so WGAN and RBM are actually trained on train+test), but we also retrain their models on our own training split to ensure a fair comparison. In imputation experiments, we additionally compare against Impute5 [32], a state-of-the-art imputation tool widely used in genetics. Impute5 serves a dual role: it is used with real reference genomes to establish performance benchmarks, and it is used with AG reference panels generated by each method to evaluate their utility for imputation.

The second category consists of simpler probabilistic graphical models (PGMs) that support tractable likelihood computation: fully-factorized distributions (INDEP), Markov chain models of order 1 (MARKOV), and non-homogeneous hidden Markov models (HMM). These serve as baselines for density estimation and AG quality comparisons. The HMM is built within the same circuit framework as GPC and differs from it only in the latent graph, a chain rather than a learned tree, so every comparison between the two isolates the contribution of the topology. Full specifications are given in Section 4.4.

Evaluation criteria We evaluate these models using the following metrics: (1) log-likelihood on test data to assess the capability of each model as a density estimator; (2) summaries of AGs sampled from each model, including the top principal components and linkage disequilibrium at pairs of SNPs; (3) genotype imputation, which evaluates the models' utility on an important downstream task (Fig 2); and (4) privacy analysis of the AGs sampled from each model.

2.4 GPC trains efficiently with objective convergence criteria

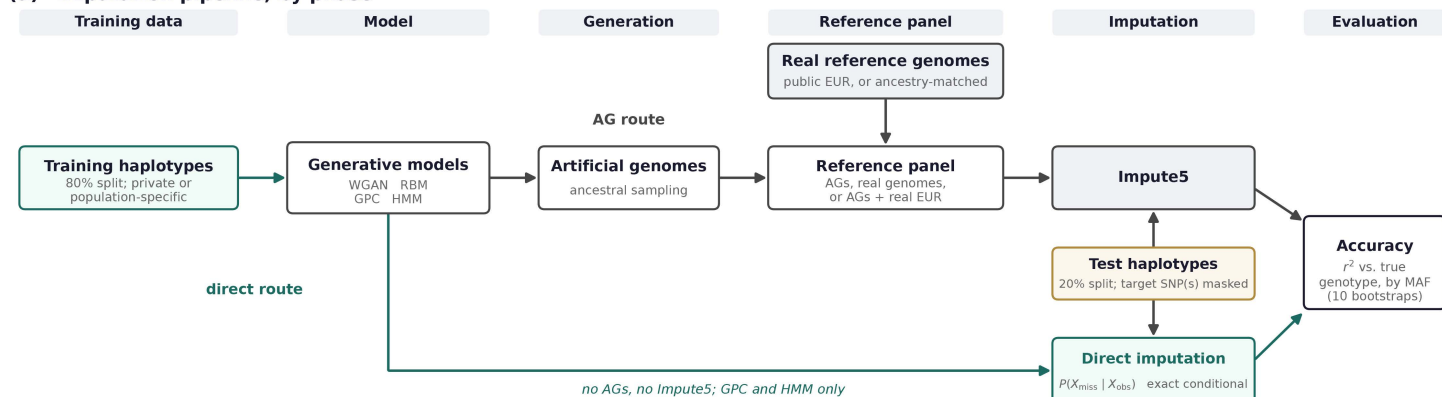
Training GPC requires minimal hyperparameter tuning. We set the number of latent states to 128 (the maximum feasible given memory constraints) and use a small pseudocount (0.005) for smoothing. Models were trained for 2,000–5,000 epochs depending on dataset size. We use the PC package PyJuice [40] to train GPC efficiently on GPUs, achieving EM updates in under 2, 10, and 3 seconds per epoch on the 1KG, UKBB, and high-coverage 1KG datasets, respectively. On a single NVIDIA RTX A5000 GPU with 24GB available memory, GPC trains in roughly 2, 6, and 4 hours on each dataset, respectively. Once the model is loaded, generating a single sample can be done in under 2, 5, and 3 seconds for each dataset, respectively. Imputing the high-coverage 1KG dataset for the entire held-out set (12,551 SNPs across 1,002 haplotypes) takes about one second in a single conditional query; Impute5 with a real reference panel completes the same task in about four seconds. Inference is inexpensive by either route, and the cost that separates the methods is training rather than imputation. A systematic comparison of training time and memory against RBM and WGAN, and the resulting working range of GPC, is given in Section 4.5. A key advantage over other deep generative approaches is that convergence can be monitored via held-out log-likelihood, providing an objective stopping criterion rather than subjective visual inspection. Full training details for GPC and all baseline methods are provided in Section 4.4.

2.5 GPC accurately reconstructs local genetic structure

We first compared the ability of different generative models to represent genetic variation in the 1KG and UKBB datasets. We simulate AGs with GPC and all five baselines (WGAN, RBM, HMM, INDEP, and MARKOV) for comparison. For both datasets, we use an 80% train / 20% test split performed at the individual level before haplotype separation; we generate 5,008 AGs from each model for 1KG and 10,000 AGs for UKBB.

GPC learns more accurate probabilistic models than HMM, MARKOV, and INDEP as measured by their log likelihood on the test dataset that was not used for model fitting (Table 1). Note that we do not compare with WGAN and RBM since they do not support tractable exact likelihood computation. We additionally evaluated the quality of AGs based on whether they preserve distances across pairs of haploid genomes. To do this, we compute the pairwise differences of haploid

(a) Imputation pipeline, by phase



(b) The three settings differ only in the data splits and which SNPs are masked

Setting	Training data	Test data	SNPs imputed
General	80% of a cosmopolitan panel (1KG, UKBB)	20% held-out, same ancestries	one SNP at a time, all others observed
Population-specific	80% of the target population (African / non-European), optionally + real EUR	20% held-out of the target population	one SNP at a time, all others observed
Array-based	80% of high-coverage 1KG	20% held-out	12,551 non-array SNPs jointly, from 2,119 typed

Fig 2. Experimental design for the imputation evaluations. (a) Training haplotypes are used to fit each generative model. Along the *AG route*, the fitted model is sampled to produce artificial genomes, which form an Impute5 reference panel, either the AGs alone or the AGs merged with real European genomes, and Impute5 then imputes the masked SNPs of the held-out test haplotypes. Along the *direct route*, the conditional $P(X_{\text{miss}} | X_{\text{obs}})$ is computed exactly from the model itself, with no artificial genomes and no Impute5; this route is available only to models that support tractable conditional inference. A separate benchmark supplies real reference genomes to the same Impute5 step, which isolates the contribution of Impute5 from that of the generative model. Both routes end at the same accuracy computation, the squared Pearson correlation r^2 between imputed dosages and the true genotypes, binned by minor allele frequency with confidence intervals from 10 bootstrap replicates. (b) The three imputation settings share this pipeline and differ only in the training split, the test split, and which SNPs are masked: one SNP at a time for the general and population-specific settings, and the non-array SNPs jointly for the array-based setting.

<https://doi.org/10.1371/journal.pgen.1012321.g002>

Table 1. Log-likelihoods of probabilistic models that support tractable likelihood computation in 1KG and UKBB datasets. Averaged training and test log-likelihoods for Indep, Markov HMM, and GPC. The bold values highlight the best averaged log-likelihoods.

Dataset	Category	Indep	Markov	HMM	GPC
1KG	train LL	-2386.81	-1806.33	-591.08	-202.51
	test LL	-2404.51	-1819.96	-599.88	-265.06
UKBB	train LL	-1642.62	-1360.86	-554.88	-120.10
	test LL	-1648.03	-1362.16	-554.38	-127.75

<https://doi.org/10.1371/journal.pgen.1012321.t001>

genomes within a single dataset or between the test dataset and an AG dataset and compute the Wasserstein distance between these pairs of distributions; a lower Wasserstein distance indicates that the AGs tend to be more similar to real genomes in the test dataset. WGAN, RBM, and GPC all capture the distribution well with GPC having second-lowest distance behind WGAN on 1KG and having the lowest distance on UKBB (S1 Fig and rows 1–2 of S1 Table).

We then analyze the quality of AGs generated by all models in terms of capturing commonly-used summaries of genetic variation data. To visualize all methods in the same latent space, we merge eight datasets (training set, test set, and AGs from six methods) and apply a single PCA to the combined data. Inspecting the top six principal components

(PCs) of genomes in the test set and the AGs, we find that AGs generated from GPC qualitatively capture the dominant structure in this dataset as do the deep learning methods (WGAN and RBM), unlike INDEP, MARKOV, and HMM (Fig 3 for 1KG and S2 Fig for UKBB). To quantify the accuracy of these summaries, we compute the Wasserstein distance between the 2D PCA representations of the test genomes versus the AGs (S1 Table) and find that distances for the deep learning methods (WGAN, RBM, and GPC) tend to be lower than simpler models.

We also examined patterns of linkage disequilibrium (LD) to assess how the pairwise short and long-range correlations of SNPs can be captured by AGs. While WGAN and RBM tend to be accurate at longer length scales and HMM

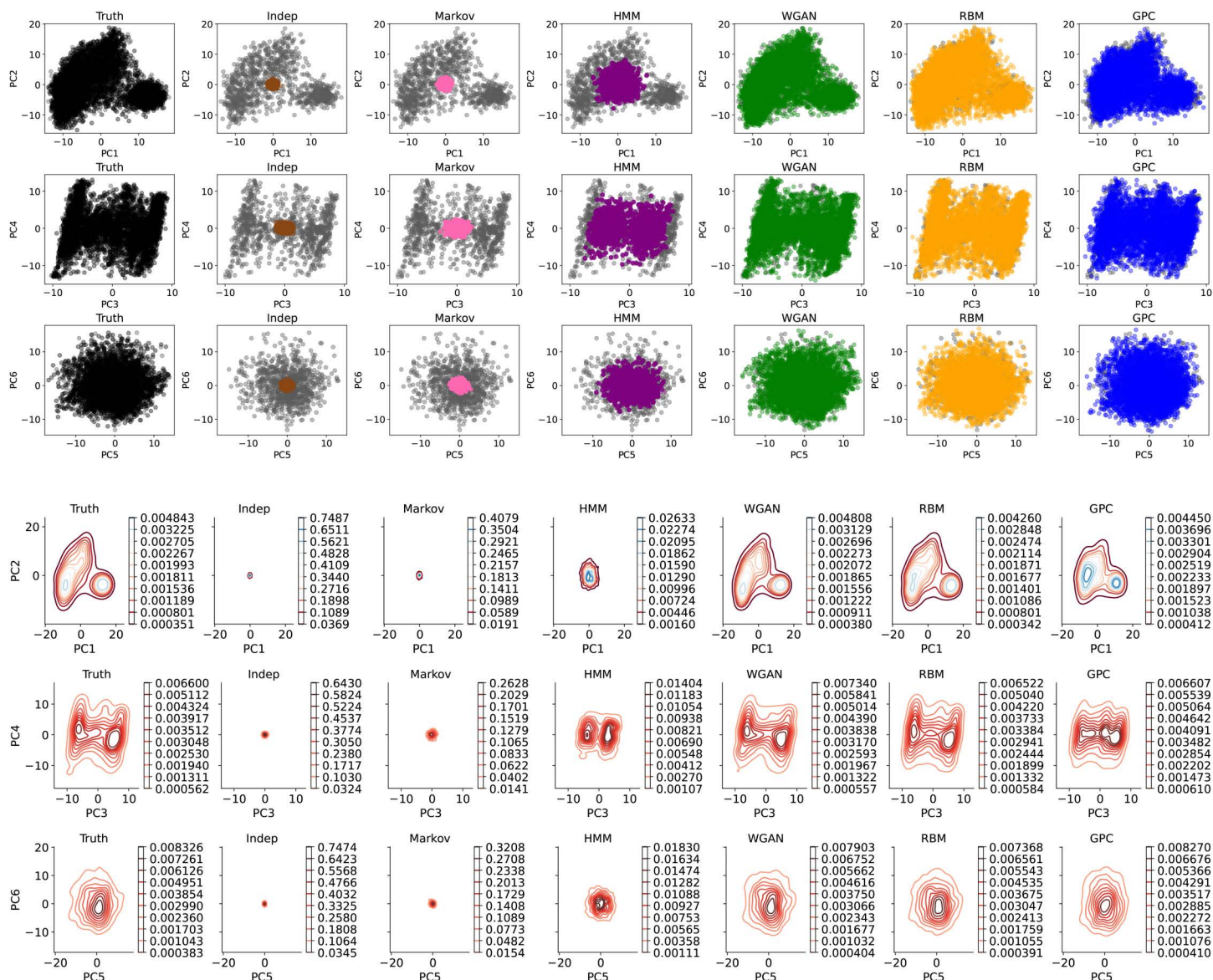


Fig 3. Principal components analysis for AGs generated by models trained on the 1KG dataset. (Top half) The top six principal components (PCs) of the held-out test set of genomes (gray) vs. AGs generated by each of INDEP (brown), MARKOV (pink), HMM (purple), WGAN (green), RBM (orange), and GPC (blue). The leftmost plot shows the PCs of genomes from the training set as "perfectly" generated data (black). (Bottom half) A density map of the PCs for each dataset. (see S2 Table for detailed metrics).

<https://doi.org/10.1371/journal.pgen.1012321.g003>

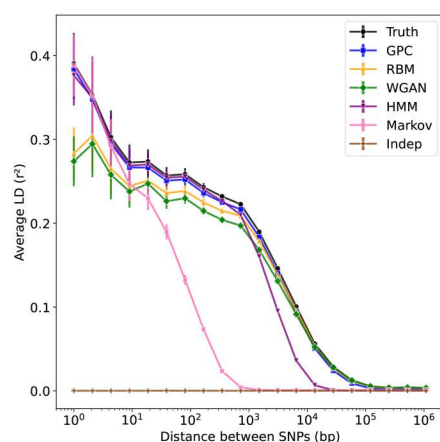
and MARKOV are accurate at shorter length scales, GPC is accurate across all length scales (Fig 4). Finally, we examined the topology of the CLT learned on the 1KG dataset to gain intuition into why GPC captures LD at all scales. In contrast to the chain-structured HMM in which every node has degree at most 2 and every edge connects adjacent SNPs along the genome, the learned CLT has 4,599 leaf nodes, 397 nodes with degree ≥ 5 , and a maximum degree of 161, with 18.3% of edges connecting SNPs more than 1,000 positions apart in sequential index (Fig 5; S3 and S4 Figs). The non-chain tree topology allows the model to place strongly correlated SNPs close together in the tree regardless of their genomic positions, enabling direct propagation of information across long ranges without passing through all intermediate variables.

2.6 GPC improves imputation accuracy across diverse scenarios

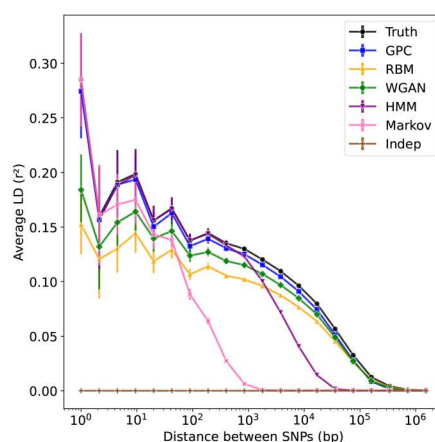
A key downstream application of generative models and AGs is genotype imputation. To assess the quality of artificial genomes (AGs) produced by GPC and competing generative models, we evaluate their utility in several imputation settings across the datasets described in Section 2.2. One approach is to generate AGs from these models, which then serve as reference panels for Impute5 [32] and other standard imputation tools. Alternatively, as described in Section 2.1, GPC can perform *direct imputation* by leveraging efficient conditional probability computation over the learned model, without the need to generate AGs. Fig 2 summarizes the full experimental design: panel (a) traces both routes from training haplotypes to the final r^2 , and panel (b) states how the three imputation settings differ. The two routes measure different things, and we distinguish them throughout. The AG route evaluates a *pipeline* (generative model, then AG reference panel, then Impute5), so its accuracy reflects all three components, not the generative model alone. The direct route evaluates the generative model on its own, since the conditional $P(X_{\text{miss}} | X_{\text{obs}})$ is computed exactly from the model with no artificial genomes and no external imputation tool, and is available only to models that support tractable conditional inference. Direct evidence of model quality is therefore given by the held-out log-likelihoods of Table 1, the population-structure and LD comparisons of Figs 3 and 4, the privacy analysis of Table 2, and the direct imputation lines in the imputation figures; the AG panel lines instead quantify the downstream utility of the artificial genomes. The real-reference-panel Impute5 line bounds the contribution of the Impute5 component itself, since it applies the same tool to real reference genomes.

Following the same training protocol as in Section 2.5, we use 80/20 train/test splits and generate the same number of AGs per dataset. For population-specific experiments, the same 80/20 split is applied within the target population. Following the experimental protocol established in prior work [32,33,44], we evaluate imputation accuracy at each SNP by removing the selected SNP from the test haplotypes while keeping all other SNPs observed. Imputation is then performed conditional on the remaining observed SNPs in the test haplotypes. We calculate the squared Pearson correlation (r^2) between the imputed posterior probabilities of carrying allele 1 and the true allelic state at the target SNP (excluding monomorphic SNPs), aggregating results into bins by minor allele frequency (MAF) with 95% confidence intervals computed from 10 bootstrapped replicates. This imputation protocol follows the approach of Yu et al. [44] (Sections 2.6.1 and 2.6.2). We additionally evaluate imputation from genotyping arrays following Rubinacci et al. [32] and Browning et al. [33] for which we use the high-coverage 1KG dataset (Section 2.6.3) (additional details in Section 4.3).

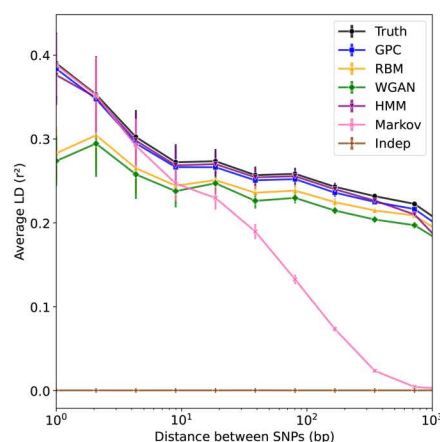
We consider two broad scenarios for imputation. In the first (general) scenario, imputation is applied to genotypes from a cosmopolitan sample (test set) consisting of individuals from multiple ancestries with AGs generated by different models that were also trained on a cosmopolitan panel of matched ancestries consisting of a distinct set of individuals (training set). This setting reflects common practice in large-scale genetic studies where diverse reference panels are used to impute missing genotypes across heterogeneous cohorts. The second (population-specific) scenario considers a setting in which imputation is to be performed in a population for which reference genomes are limited or restricted. Because the target population might be under-represented in public reference panels, imputation accuracy in these populations might be adversely impacted. This scenario is particularly relevant for underrepresented populations in genomic research, where privacy constraints or limited sequencing resources may restrict the availability of ancestry-matched reference data. In a



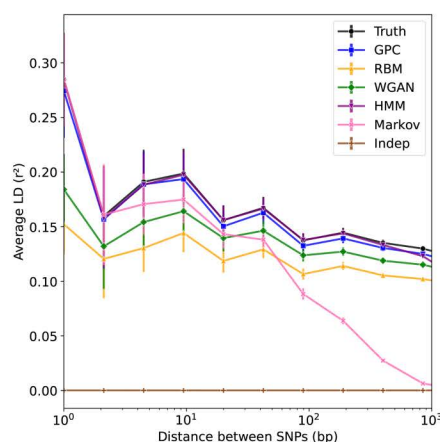
(a) 1KG: full range



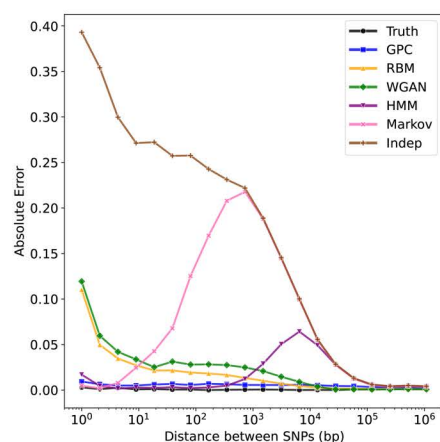
(b) UKBB: full range



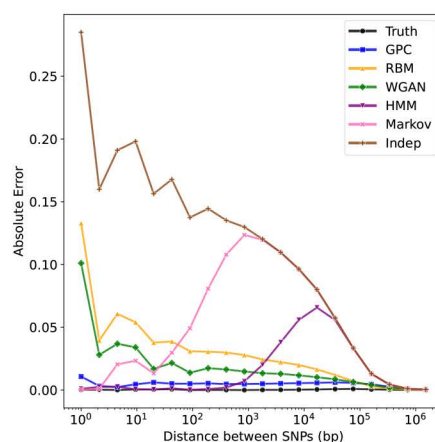
(c) 1KG: 1–1000 bp zoom



(d) UKBB: 1–1000 bp zoom



(e) 1KG: absolute error per bin



(f) UKBB: absolute error per bin

Fig 4. Patterns of linkage disequilibrium (LD) in the AGs generated by models. We compare LD between pairs of SNPs in the AGs generated by each of the models trained on the 1KG (left) and the UKBB (right) datasets respectively. We plot mean squared correlation r^2 within bins of physical distance (error bars denote the standard error of the mean); truth represents the LD distribution of the training data. We plot the decay of LD across the full length scale of the locus (top row); at shorter length scales (1–1000 bp; middle row); and the absolute error in mean r^2 per bin relative to the test data

(bottom row) (see [S3 Table](#) for detailed metrics). (a) 1KG: full range. (b) UKBB: full range. (c) 1KG: 1–1000 bp zoom. (d) UKBB: 1–1000 bp zoom. (e) 1KG: absolute error per bin. (f) UKBB: absolute error per bin.

<https://doi.org/10.1371/journal.pgen.1012321.g004>

realistic setting, a researcher may only have access to publicly available European reference genomes while population-specific data remains private. GPC and other generative approaches can learn from such private data and either generate AGs or perform direct imputation without exposing the underlying genomes. For the population-specific analyses, we designate a target population which is limited or restricted in access. We consider two sets of target populations: all individuals of non-European ancestry and all individuals of African ancestry. In each of these settings, we also explore how the accuracy of imputation is impacted when AGs are combined with real genomes (possibly with different genetic ancestry characteristics than the target population) that might be available.

2.6.1 General imputation. We first examine the general setting, in which models are trained on genomes from diverse ancestries and are then used to impute into target genomes drawn from similar ancestries. To mimic this scenario, we train and test models on random splits of the 1KG and UKBB datasets. For 1KG, we also include a comparison to 5,008 AGs based on fitting WGAN and RBM to the same genomic region that were made available by the authors of [25].

We see that reference panels composed entirely of real genomes used with Impute5 achieve the highest imputation accuracy across MAF bins, providing an approximate upper bound on achievable performance ([Fig 6a](#) for 1KG and [Fig 6b](#) for UKBB). Among the generative models, GPC obtains the highest imputation accuracy across MAF bins. Averaged across both datasets, GPC (direct) achieves a 0.168 (27.5%) improvement in r^2 over the next best method, RBM (0.230 (174%) for low-frequency variants with $\text{MAF} < 1\%$). GPC (direct) imputation, via conditional probability computation rather than simulated AGs, tends to be the more accurate version, achieving a 0.105 (15.5%) improvement in r^2 (0.195 (61.5%) for low-frequency variants) over GPC with AG-based imputation, likely due to directly targeting the imputation objective rather than relying on simulated AGs. To assess the robustness of these findings, we repeated the UKBB general imputation experiment on five additional LD blocks across distinct chromosomes (see [Section 4.1.2](#) for details) to find the same qualitative pattern holds across all regions: GPC (direct) consistently improves over RBM (the next best method) across the range of MAF ([Fig 6c](#); additional per-chromosome results in [S5 Fig](#)).

2.6.2 Population-specific imputation. We next consider imputation in populations for which reference genomes are limited or restricted. Because large public datasets are predominantly of European ancestry, we consider scenarios where the target population consists of individuals of either non-European or African ancestry. “Non-European (pooled)” denotes a heterogeneous group rather than a single population: for 1KG it comprises the African, admixed American, East Asian and South Asian superpopulations, and for UKBB the corresponding self-reported groups. Ancestry is taken from the IGS sample table for 1KG and from self-reported ethnic background (field 21000) for UKBB. Per-group counts for both datasets are given in [S11 Table](#). In this setting, imputation accuracy in the target population can be degraded when public European reference genomes are the only data available to the imputation method. AGs specific to the target population generated by GPC can mitigate this challenge by learning from an ancestry-matched set of private reference genomes. We also evaluate combined reference panels formed by augmenting real European data with population-specific AGs.

Analogous to the general imputation experiments, GPC (direct) remains the most accurate across MAF bins ([Figs 7](#) and [S6](#) for 1KG and UKBB respectively). Averaged across datasets and ancestries, GPC (direct) achieves a 0.154 (33%) improvement in r^2 over the next best method, RBM (0.202 (279%) for low-frequency variants). Unlike in the general imputation setting, GPC, when used for direct imputation, tends to be more accurate than using Impute5 with a reference panel of European genomes, likely due to the distributional mismatch between the reference and target populations. This holds consistently across the 1KG dataset. In the UKBB dataset, GPC (direct) outperforms Impute5 for common variants

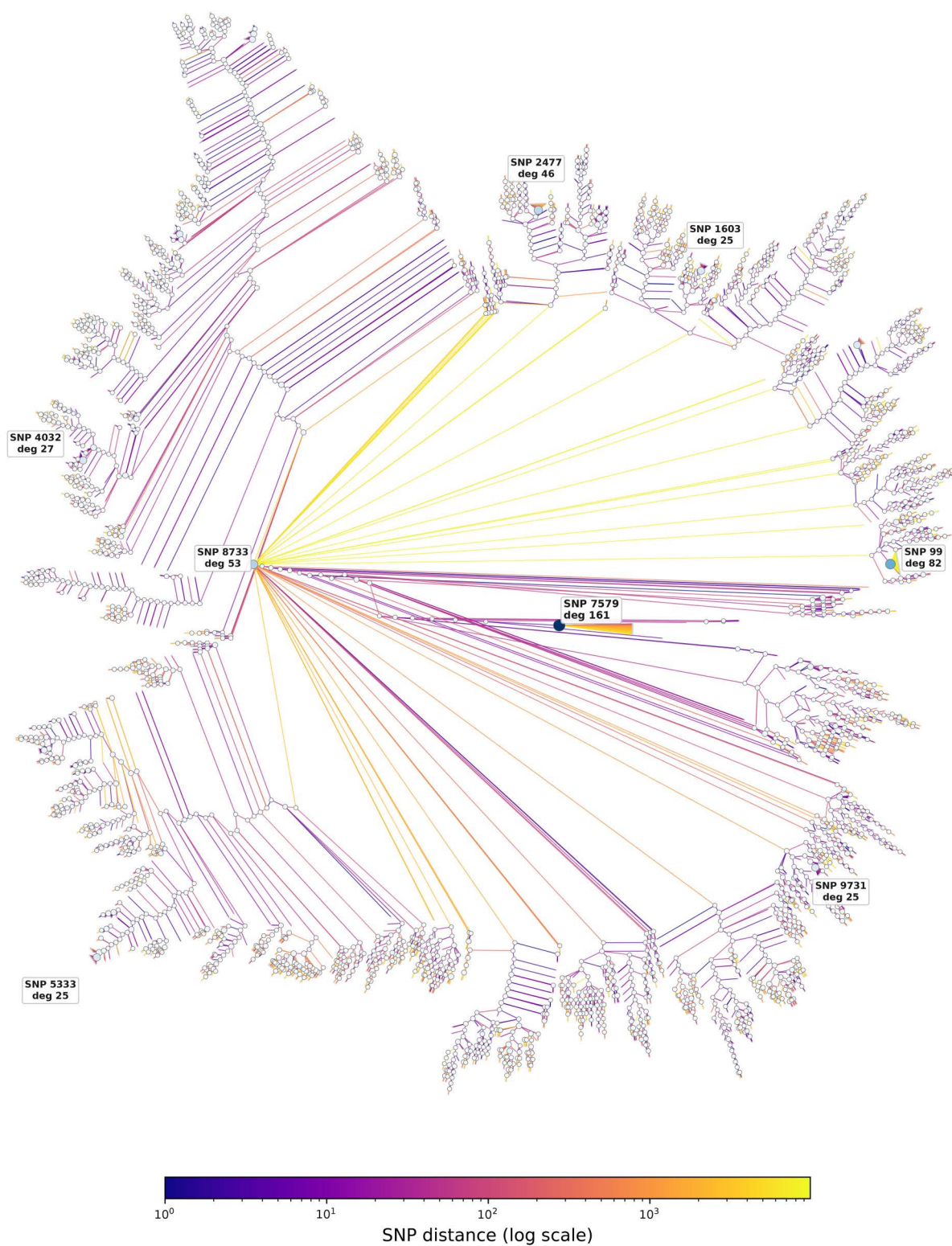


Fig 5. Compact radial visualization of the Chow-Liu Tree (CLT) learned from the 1KG dataset. Starting from the full 10,000-SNP tree (S4 Fig), all degree-2 nodes forming unbranched chains are contracted into single edges between their endpoints, retaining only branching points and leaves. Edge color encodes genomic distance $|i-j|$ on a log scale (purple: nearby; yellow: distant). Node size scales with degree; the top 8 highest-degree nodes are

labeled. Unlike an HMM, which forms a single unbranched chain, the CLT exhibits rich branching structure with long-range edges connecting SNPs thousands of positions apart, capturing distal LD directly rather than routing it through all intermediate variables.

<https://doi.org/10.1371/journal.pgen.1012321.g005>

Table 2. Nearest neighbor adversarial accuracy (AA_{TRUTH} , AA_{SYN}) for AGs generated by RBM, WGAN, and GPC, evaluated on training and test splits from 1KG and UKBB. Distances are standardized Euclidean. This upweights rare variants relative to common ones, exposing failures in rare-variant modeling. Values closest to 0.5 are bolded.

Dataset	Metric	HMM	RBM	WGAN	GPC
1KG	AA_{TRUTH} (Train)	0.9996	0.9561	0.8103	0.7185
	AA_{TRUTH} (Test)	0.9996	0.9916	0.8183	0.8199
	AA_{SYN} (Train)	0.8874	0.0024	0.7356	0.4225
	AA_{SYN} (Test)	0.9026	0.0212	0.7680	0.4912
UKBB	AA_{TRUTH} (Train)	1.0000	0.9934	0.9664	0.9096
	AA_{TRUTH} (Test)	0.9996	0.9970	0.9692	0.9240
	AA_{SYN} (Train)	0.9256	0.0094	0.7828	0.5380
	AA_{SYN} (Test)	0.9242	0.0114	0.7730	0.4686

<https://doi.org/10.1371/journal.pgen.1012321.t002>

but is slightly less accurate for rare variants. We attribute this to our balanced sampling strategy (Section 4.1.2), in which the European reference panel available to Impute5 and the population-specific training data for the generative models are matched in sample size. Under this design, a large European reference panel (even one mismatched in ancestry) can partially compensate for distributional mismatch at rare variants, where population-specific sample sizes are most limiting. Averaged across datasets and ancestries, GPC (direct) achieves a 0.056 (12.3%) improvement in r^2 over Impute5 using European genomes (0.012 (42.1%) for low-frequency variants).

We also examined the accuracy of the models in admixed individuals, which have substantial long-range LD, by evaluating the imputation accuracy in the general 1KG experiment by restricting to the subset of admixed individuals within the held-out set. The trends are the same: every method is less accurate than on the full test set, as expected, but the ordering is unchanged and GPC (direct) remains the most accurate generative model (S7 Fig and S10 Table, which also describe how the subset is defined).

Taken together, these results demonstrate that population-specific AGs from GPC can substantially enhance imputation accuracy in underrepresented groups, either by augmenting existing reference panels or through direct conditional imputation. The magnitude of improvement depends on several interacting factors: the degree of genetic distance between the target population and the available public reference panel, the sample size of the private population-specific data, and the frequency of the variants being imputed. When public reference data is both ancestrally mismatched and matched in size to the private data, as in our balanced UKBB design, gains over a large European-only panel are most pronounced for common variants, while rare variant performance remains sensitive to total sample size regardless of ancestry. These findings suggest that in practice, combining ancestry-matched private data with available public genome (even those from a different ancestry) will generally yield the best imputation accuracy, and that the relative value of each component will depend on the specific target population and variant frequency spectrum.

2.6.3 Array-based imputation. We also evaluate GPC in a more realistic scenario of imputation from variants genotyped on a commonly-used SNP array. In this experiment, we attempted to impute SNPs genotyped in the high-coverage 1KG dataset from variants that were typed on the HumanOmni5Exome array (see Section 4.1.3 for details).

GPC (direct) remains the most accurate in both the general (Fig 8) and population-specific settings (Fig 9). In the general setting, GPC (direct) achieves a 0.020 (3.5%) improvement in r^2 over the next best method, RBM (0.043

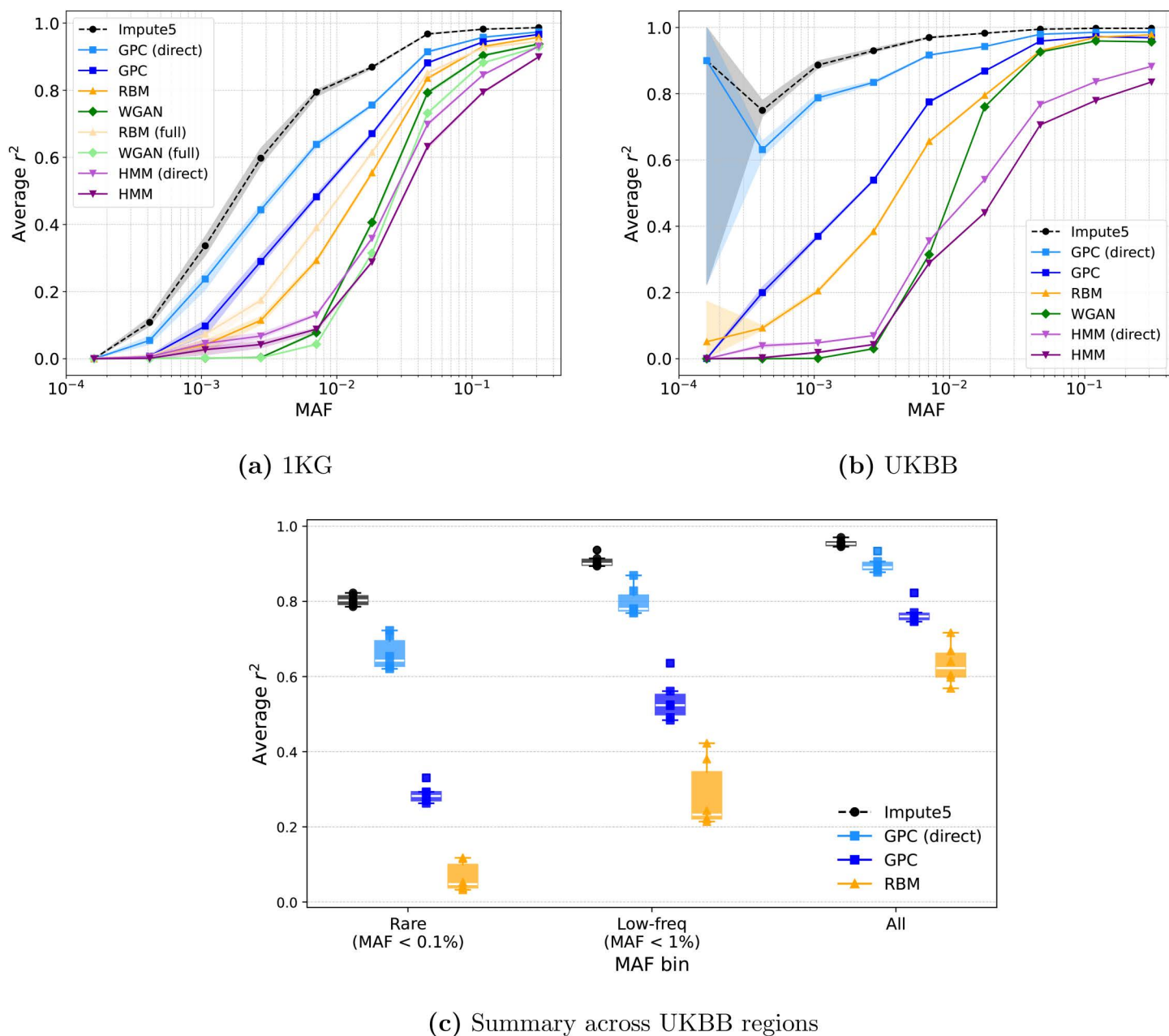
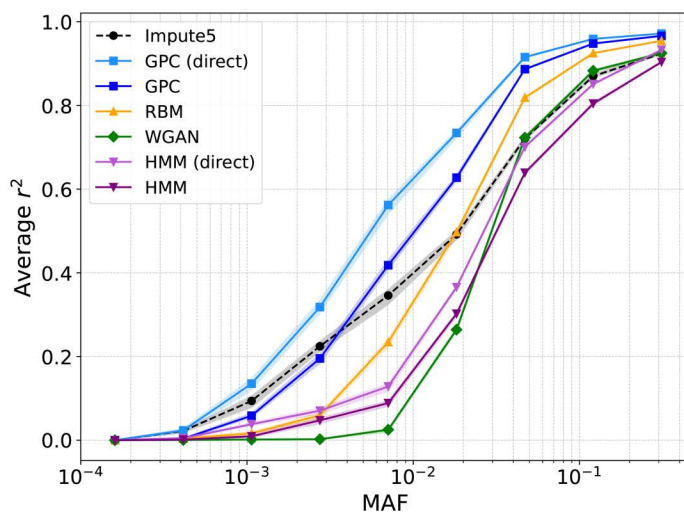
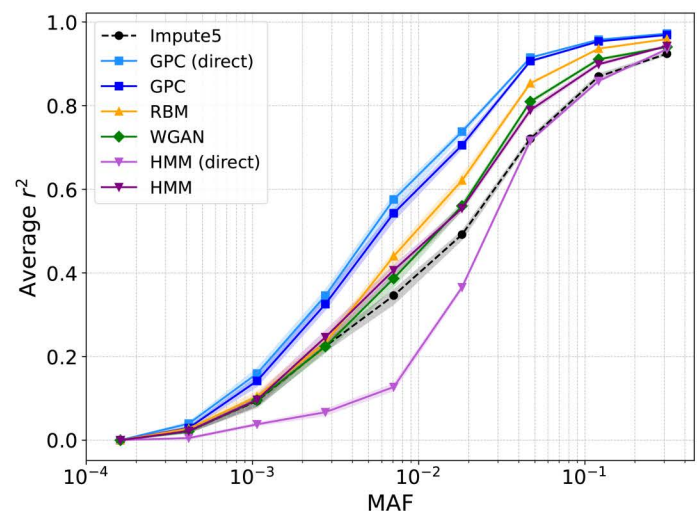


Fig 6. General imputation. The light blue lines denote direct imputation using GPC while the other lines show the results of imputation using Impute5 with AG reference panels in the 1KG (left) and the UKBB (right) datasets respectively. We also plot the results of imputation using real reference genomes (black dotted lines) while RBM/WGAN (full) show results using AGs provided by the authors of [25] in their analysis of the same locus in the 1KG dataset (see S4 Table for detailed metrics). (c) summarizes UKBB imputation performance for the best-performing methods (excludes WGAN) by MAF bin, averaged across chr22 (the main UKBB region shown in (b)) and five additional chromosomes (chr2, 11, 12, 14, 18; see S5 Fig), demonstrating that results are robust across genomic regions. (a) 1KG. (b) UKBB. (c) Summary across UKBB regions.

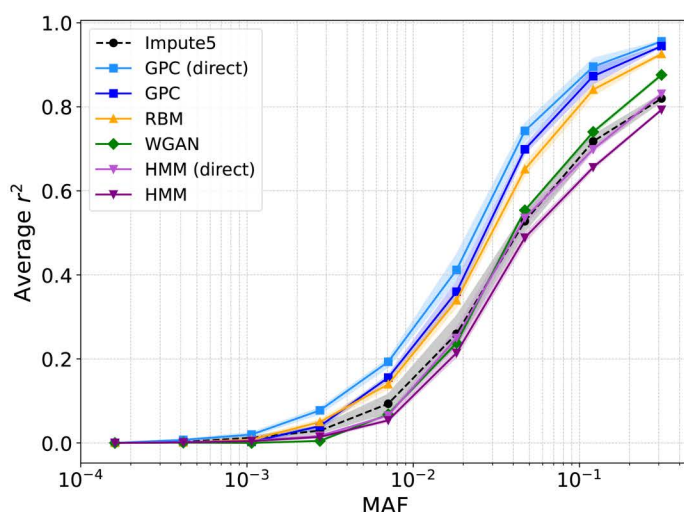
<https://doi.org/10.1371/journal.pgen.1012321.g006>



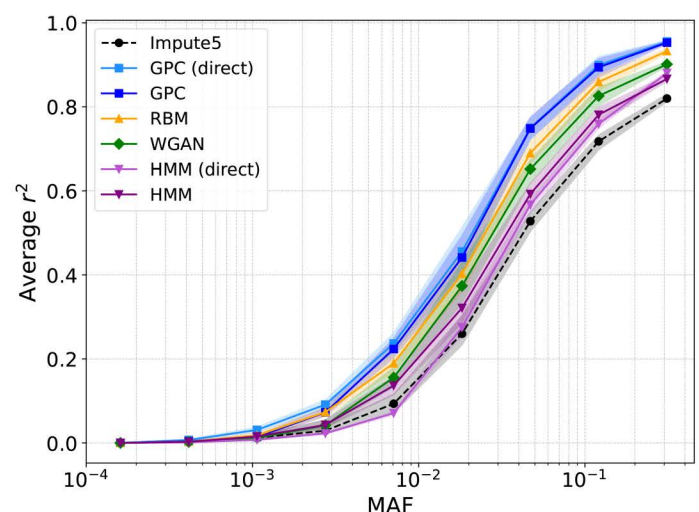
(a) Non-European target



(b) Non-European target



(c) African target



(d) African target

Fig 7. Population-specific imputation in the 1KG dataset. On the left, the AG lines show imputation results using population-specific AGs (where the target genomes correspond to non-European populations in part a and African populations in part c) while the light blue line shows direct imputation with GPC trained on population-specific genomes. On the right, the AG lines show results using combined reference panels (real European genomes plus population-specific AGs) while the light blue line shows direct imputation with GPC trained on both European and population-specific genomes (where the target genomes correspond to non-European populations in part b and African populations in part d). The black line shows results using Impute5 with real European genomes as the reference panel (see [S6 Table](#) for detailed metrics). (a) Non-European target. (b) Non-European target. (c) African target. (d) African target.

<https://doi.org/10.1371/journal.pgen.1012321.g007>

(14.3%) for low-frequency variants). In the population-specific setting, averaged across both ancestries, GPC (direct) achieves a 0.030 (5.3%) improvement in r^2 over RBM (0.040 (12.7%) for low-frequency variants). Similar to the previous population-specific setting, GPC (direct) outperforms Impute5 with the European-only reference panel, achieving a 0.285 (96.5%) improvement in r^2 averaged across both ancestries (0.324 (1,123%) for low-frequency variants). In contrast

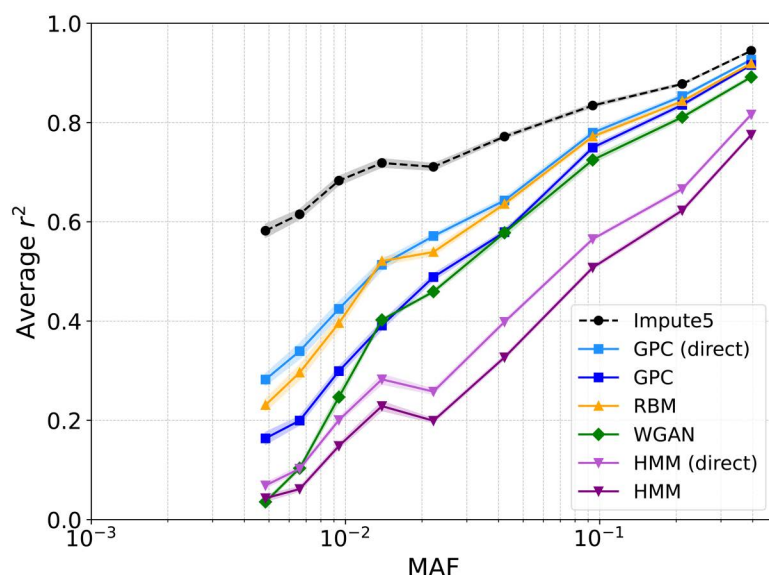


Fig 8. Array-based imputation. This task aims to impute SNPs from the HumanOmni5Exome array using high-coverage 1KG data. The black and light blue lines show results using Impute5 with real training data as the reference panel and direct imputation using GPC, respectively. The other lines show results using Impute5 with AG reference panels (see [S8 Table](#) for detailed metrics).

<https://doi.org/10.1371/journal.pgen.1012321.g008>

to our previous imputation experiments, here the best performance is achieved by GPC (direct) trained solely on the population-specific data, rather than a combined population-specific plus European dataset. Thus, the optimal imputation strategy is likely determined by a combination of genetic ancestries, sample size, and SNP sets, and needs further investigation. These results suggest that in realistic imputation scenarios, population-specific modeling may provide accuracy gains over approaches that aggregate across populations. Evaluating imputation accuracy restricting to the admixed subset of the held-out individuals shows the same trends as in the general setting: accuracy drops for every method, the ordering is unchanged, and GPC (direct) remains the most accurate generative model ([S7 Fig](#) and [S10 Table](#)).

2.7 GPC balances considerations of privacy and utility

To evaluate the privacy of the AGs, we compute the nearest neighbor adversarial accuracy (AATS) metric introduced by Yale et al. [45]. The metric decomposes into two components: AA_{TRUTH} , which measures whether real samples are closer to other real samples than to synthetic ones, and AA_{SYN} , which measures the analogous property for synthetic samples. Values near 0.5 for both components indicate an ideal balance between utility and privacy. We report AA_{TRUTH} and AA_{SYN} separately rather than their average, as the overall AATS can mask pathological cases where poor values in one component cancel those in the other. We evaluate on both training and held-out test sets and standardize genotypes prior to distance computation as this upweights the impact of rare variants which tend to be more challenging to model (see Section 4.6.1 for details).

Under this evaluation, GPC achieves AA_{SYN} closest to 0.5 across all splits and both datasets, and AA_{TRUTH} closest to 0.5 in all but one case (1KG test, where WGAN is marginally closer: 0.818 vs. 0.820), indicating the best overall balance between utility and privacy ([Table 2](#)). The near-zero AA_{SYN} values for RBM paired with near-unity AA_{TRUTH} reveal a distinct failure mode: real samples still find other real samples as their nearest neighbors (real-to-real distances are smaller than real-to-synthetic), but each synthetic sample is closer to some individual real genome than to any other synthetic sample. Rather than forming a coherent distribution, RBM produces samples that float near individual training haplotypes without capturing broader structure, a privacy risk because each synthetic genome can potentially be traced back to a specific

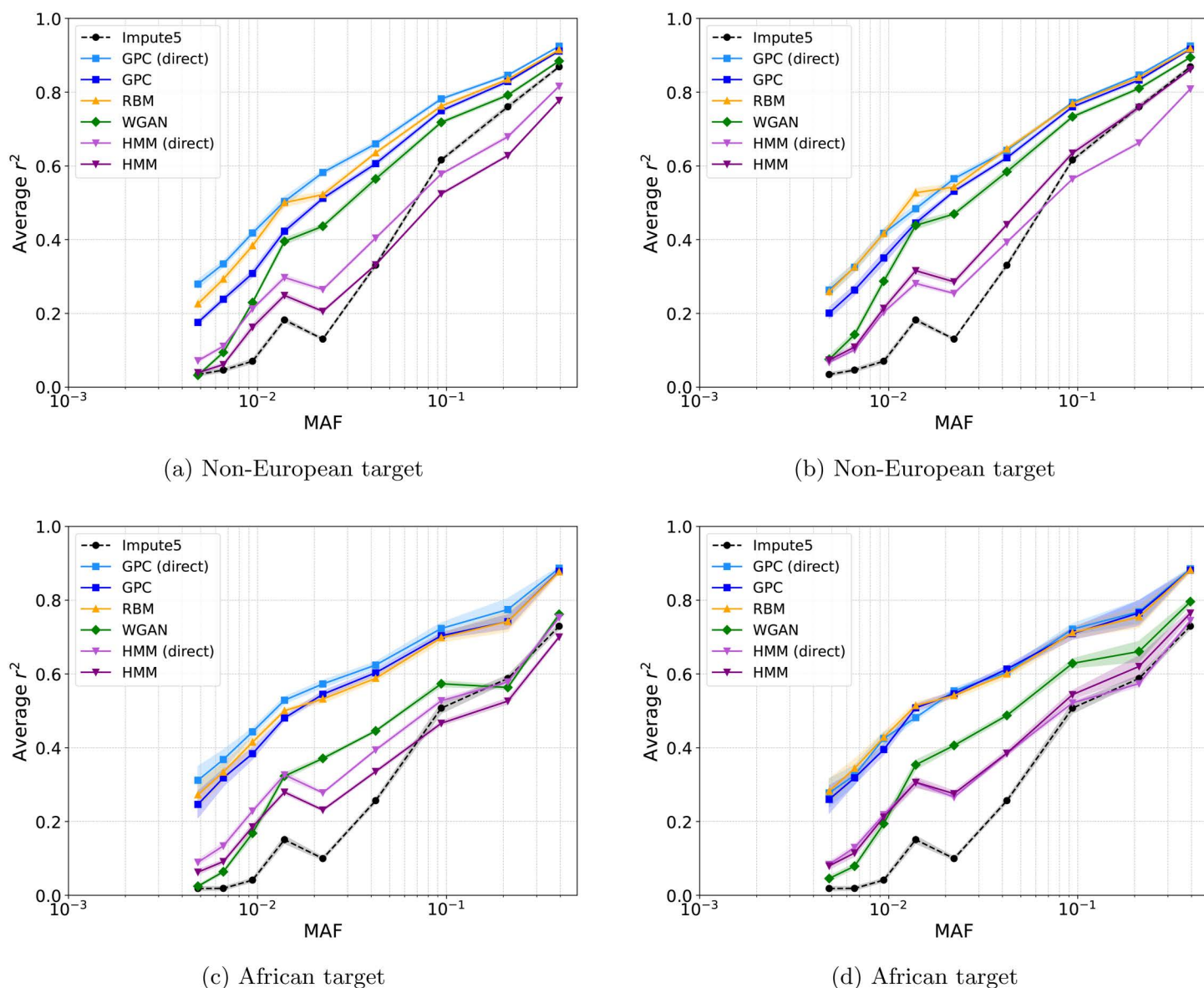


Fig 9. Array-based imputation (population-specific). The black line shows results using Impute5 with real European reference genomes. On the left (parts a and c), the AG lines show results using population-specific AGs alone, and the light blue line shows direct imputation with GPC trained on population-specific data. On the right (parts b and d), the AG lines show results using combined reference panels (real European data plus population-specific AGs), and the light blue line shows direct imputation with GPC trained on both European and population-specific data (see [S9 Table](#) for detailed metrics). (a) Non-European target. (b) Non-European target. (c) African target. (d) African target.

<https://doi.org/10.1371/journal.pgen.1012321.g009>

training individual. Conversely, WGAN exhibits uniformly high values for both components, indicating that its synthetic distribution occupies a region largely disjoint from the real data, sacrificing utility. The chain-structured HMM shows the same limitations as WGAN in its most extreme form: AA_{TRUTH} is 1.0000 on both datasets, meaning no real haplotype ever has an artificial genome as its nearest neighbor, with AA_{SYN} likewise near unity. Its artificial genomes are separable from real ones in both directions, which is maximal privacy obtained by failing to model the data rather than by any favorable trade-off. Together with RBM at the opposite extreme, the two components identify two distinct ways of scoring well on a single

averaged AATS while being unusable. We note that even for GPC, AA_{TRUTH} remains noticeably above 0.5, suggesting that the AGs are still somewhat distinguishable from real data.

3 Discussion

In this work, we introduced GPC, a deep generative model for genetic variation based on hidden Chow-Liu trees (HCLTs) and probabilistic circuits (PCs). Unlike prior deep generative models (e.g., GANs, VAEs, and RBMs), GPC combines tractable probabilistic inference with an expressive latent variable structure, enabling principled convergence monitoring via held-out log-likelihood, accurate reconstruction of population structure and linkage disequilibrium patterns across all distance scales, and direct genotype imputation through exact conditional probability computation. In population-specific settings, GPC (direct) consistently outperforms other deep generative methods as well as Impute5 using publicly available European reference genomes, with particularly strong gains for low-frequency variants that are disproportionately population-specific. We also found that GPC AGs provide improved privacy properties compared to RBMs and WGANs, though we emphasize that demonstrating improved performance on a single metric does not guarantee complete privacy preservation. Other attack models, such as membership inference attacks [46,47] or attribute inference attacks [48], may still pose risks.

Our results also suggest reasons why WGAN and RBM are less suited to this data. For RBM, AA_{SYN} near zero together with AA_{TRUTH} near one (Table 2) indicates that each synthetic haplotype lies closer to some individual real haplotype than to any other synthetic one, so the samples cluster around training points rather than covering the distribution. This is both a privacy risk and a failure to represent the full range of variation. For WGAN, uniformly high values on both components indicate the opposite behavior, artificial genomes that occupy a region largely disjoint from the real data. Both reproduce long-range LD reasonably but degrade at short range (Fig 4), whereas GPC is accurate at all scales. We believe two differences contribute to this gap. First, neither model supplies direct supervision on the dependence between particular pairs of SNPs: the RBM couples all SNPs through a single fully connected hidden layer that carries no notion of genomic position, and the WGAN critic returns a single scalar score per haplotype. GPC instead learns its tree from the pairwise mutual information of the data itself, so the dependence structure is fitted directly to the observed patterns of LD. Second, neither model admits a tractable likelihood, so convergence cannot be assessed objectively; the authors of the original works describe having to try multiple combinations of architectures and hyperparameter values, with some runs failing to converge, and assess convergence by visual overlap in principal component space. GPC is fit by EM and monitored by held-out log-likelihood, which supplies an objective stopping criterion.

Additionally, recent work has extended the space of deep generative approaches for genetic data to include VAEs [29,30], GANs operating in reduced-dimensional spaces [26,28], and diffusion models [31], as well as genome-wide WGAN and CRBM variants [25]. Our comparisons are restricted to dense local genomic regions, and we did not benchmark against the genome-wide variants. Modeling a local region and modeling a whole genome are different problems: the genome-wide models operate at a much lower SNP density and are typically built on genotyping-array frameworks, which are ascertained for common variation, whereas the regions studied here are sequence-based and substantially denser. The genome-wide WGAN and CRBM variants of Yelmen et al. [25] in particular are the same model families we benchmark against, scaled up by tiling and convolution rather than redesigned. We also note that GPC cannot currently be run at genome-wide scale either, a limitation we return to below. As noted previously, none of these approaches support efficient exact conditional inference: GANs do not define a tractable likelihood, RBMs have an intractable partition function that prevents exact likelihood evaluation, VAEs provide only a lower bound, and while diffusion models can compute exact likelihoods via probability flow ODEs, exact conditional inference requires either approximations or a full multi-step denoising pass per query. They can therefore only be evaluated for imputation through the artificial-genome route, which our results indicate is the weaker of the two.

Several limitations of GPC remain. First, GPC is currently applied to single genomic regions. Memory limitations during construction of the Chow-Liu tree confines it to local regions, as quantified in Section 4.5, so reaching genome-wide scale

will require hierarchical or distributed approaches; a straightforward alternative that treats LD blocks as independent would ignore LD across blocks. Second, while our new results suggest GPC maintains its relative improvements in imputation accuracy for admixed populations, a more detailed understanding of the effectiveness of generative models in this setting is needed: the impact of training on cosmopolitan panels relative to panels from an appropriately matched admixed population, whether AGs generated in these populations realistically capture patterns of population structure, LD and local ancestry, and how these inferences are affected by demographic history. The extent of long-range LD in admixed populations also makes the LD-block strategy above less effective there. Third, our current implementation of GPC is restricted to haploid genomes to enable direct comparisons with prior approaches; extending GPC to generate diploid genotype data would increase the applicability of the model. Fourth, like all generative approaches, GPC inherits biases from the training data, and ensuring fairness across diverse cohorts remains an open challenge.

Each of these limitations points to a direction for further work. Extending GPC to genome-wide scale, whether by hierarchical composition or by distributed training, would make it directly comparable with the genome-wide WGAN and CRBM variants and with the newer VAE- and diffusion-based approaches across imputation and other downstream tasks, and would allow robustness to be assessed across varying LD structures and recombination rates. The privacy results reported here are empirical, and fitting GPC with differentially private EM algorithms [49] would allow formal guarantees instead. Finally, benchmarks such as rare variant association studies, fine-mapping, or polygenic risk scores would give a more complete picture of the utility of artificial genomes and of direct imputation in downstream analyses.

4 Materials and methods

4.1 Datasets

4.1.1 1000 Genomes Project Phase 3 (1KG). We first use data from the 1000 Genomes Project (Phase 3), consisting of 2,504 unrelated diploid genomes spanning diverse ancestries. Because the genomes are phased, we split each into its two constituent haplotypes, yielding 5,008 haplotypes in total. To avoid any possibility of data leakage, all train/test splits are performed at the diploid-individual level, before haplotype separation.

Following prior work [24,25], we analyze a contiguous 10K-SNP region on chromosome 15 (15:27,379,578 – 15:29,625,035). This dataset is used primarily to compare GPC to baselines and to evaluate how well it captures population and LD structure. It is also used in our single-SNP imputation experiments.

4.1.2 UK Biobank (UKBB). To evaluate performance in a larger, secondary cohort, we analyze genotypes from the UK Biobank (UKBB). We analyzed high-quality imputed SNPs (with a hard call threshold of 0.2 and an INFO score ≥ 0.8) with MAF $\geq 0.1\%$. We apply standard quality control by retaining SNPs that are under Hardy-Weinberg equilibrium ($p < 10^{-6}$) and are confidently imputed in more than 99% of the individuals. We retain individuals with no kinship to other individuals (UKBB field 22021) and exclude those with missing ancestry information. Following these procedures, we select an LD block on chromosome 22 (22:29,456,546 – 22:32,665,772). This yields 337,862 individuals and 9,820 SNPs. To assess robustness across genomic regions, we additionally select five LD blocks on separate chromosomes: chromosome 2 (2:118,367,466 – 2:121,303,783; 9,227 SNPs), chromosome 11 (11:101,331,121 – 11:103,959,636; 9,507 SNPs), chromosome 12 (12:85,990,426 – 12:89,682,122; 9,823 SNPs), chromosome 14 (14:72,889,615 – 14:76,444,767; 9,248 SNPs), and chromosome 18 (18:51,554,175 – 18:55,213,838; 9,415 SNPs). These additional regions are used for the general imputation replication experiment. We phased all data using Beagle 5.5.

To ensure computational feasibility for all methods while maintaining statistical power, we randomly select 5,000 individuals each from the European (EUR), non-European (Non-EUR), and African (AFR) ancestry groups, yielding 10,000 haplotypes per group after phasing. Due to partial overlap between the Non-EUR and AFR subsets, the combined imputation dataset includes 26,924 total haplotypes. Similar to 1KG, we use this dataset to evaluate population/LD structure and single-SNP imputation. The balanced sampling strategy also allows us to assess whether equal amounts of European and non-European data can compensate for distributional mismatch when imputing into non-European test populations.

4.1.3 High-coverage 1KG. To evaluate GPC in a realistic imputation scenario, we consider the high-coverage whole-genome sequencing release of 1KG, using the same genomic region as in the earlier 1KG analysis, mapped to build 38 coordinates (15:27,134,431 – 15:29,332,831). We restrict the analysis to the same set of 2,504 unrelated individuals. This region contains 56,766 variants. After selecting biallelic SNPs and removing those with minor allele count (MAC) ≤ 20 , 14,670 SNPs remain.

Among these, 2,119 SNPs are present on the HumanOmni5Exome-4v1-2_A genotyping array. The remaining 12,551 SNPs (86%) are imputed simultaneously, providing a setting that closely mirrors real imputation pipelines. This dataset is used for our array-based imputation experiments, both overall and stratified by ancestry.

4.2 Model and inference methods

4.2.1 Hidden Chow-Liu trees. Hidden Chow-Liu trees (HCLTs) [35] represent a distribution over a collection of random variables (RVs) ($\mathbf{Z} = (Z_1, \dots, Z_N)$, $\mathbf{X} = (X_1, \dots, X_N)$). $\mathbf{Z}$ denotes hidden or latent RVs while $\mathbf{X}$ denotes observed RVs. The joint distribution is described by a graphical model ($\mathcal{G}$) in which the nodes in the graph represent the RVs and the lack of edges among the nodes represents conditional independence assumptions. In HCLTs, we have edges from each hidden variable to its corresponding observed random variable ($Z_n \rightarrow X_n$) while the edges among the hidden variables form a tree. When the graph over the hidden variables is a chain ($Z_1 \rightarrow Z_2 \rightarrow \dots \rightarrow Z_N$), we obtain a hidden Markov model (HMM). By permitting tree-structured graphs, HCLTs generalize HMMs and can provide a better representation of the data to capture long-range dependence, e.g., RV X_2 and X_6 are highly correlated without being correlated with X_3 , X_4 , X_5 (Fig 1).

For genetic variation data over N single nucleotide polymorphisms (SNPs), each X_n denotes the genotype value at SNP $n \in \{1, \dots, N\}$ ($X_n \in \{0, 1\}$ when we model haploid genomes). Each Z_n is a discrete RV that can take one of L values ($Z_n \in \{0, \dots, L-1\}$). Fig 1 demonstrates how to construct an HCLT using genetic data. Given a dataset $\mathcal{D}$ that contains 6 SNPs (Fig 1(a)), we first invoke the Chow-Liu algorithm to generate a tree over the latent variables associated with each SNP (Fig 1(b)). The Chow-Liu algorithm computes pairwise mutual information between all variable pairs and finds the maximum-weight spanning tree, which can be done efficiently even for large SNP panels since it only requires computing pairwise statistics and solving a minimum spanning tree problem. The tree encodes strong variable dependencies by placing highly correlated SNPs (e.g., X_1 and X_2) closer in the generated tree. Finally, the HCLT is constructed by adding an edge from every latent variable Z_i to its corresponding observed variable X_i (Fig 1(c)). The parameters of the HCLT are those associated with the discrete conditional probability distributions $P(X_n|Z_n)$ and $P(Z_n|Z_{\text{Pa}(n)})$, where $\text{Pa}(n)$ denotes the parent of node n in the tree.

4.2.2 Probabilistic circuits. Probabilistic Circuits (PCs) [37,38] are a class of probabilistic models that support tractable probabilistic inference. These capabilities have allowed PCs to perform various probabilistic reasoning tasks that are out of reach for most deep generative models [50,51], including problems in explainable AI [52–54], algorithmic fairness [55,56], and missing data robustness [54,57,58]. PCs are furthermore appealing for their expressive power and suitability for density estimation, with recent advances in structure learning [59] and parameter estimation [35,56] allowing PCs to accurately capture useful correlations in the data.

Representation PCs are an umbrella term for a wide family of tractable probabilistic models [60], including arithmetic circuits [61], sum-product networks [62], and cutset networks [63]. A PC ($\mathcal{G}, \theta$) represents a joint probability distribution $\Pr(\mathbf{X})$ over random variables $\mathbf{X}$ through a directed acyclic graph (DAG) $\mathcal{G}$ parametrized by θ . The DAG $\mathcal{G}$ consists of three types of nodes: *input*, *sum*, and *product*. Each leaf node is an input node; each inner node n (i.e., sum or product) receives inputs from its children $\text{ch}(n)$. Each node $n \in \mathcal{G}$ encodes a probability distribution $\Pr_n$, which is defined recursively as follows:

$$\Pr_n(\mathbf{x}) = \begin{cases} f_n(\mathbf{x}) & \text{if } n \text{ is an input node,} \\ \prod_{c \in \text{ch}(n)} \Pr_c(\mathbf{x}) & \text{if } n \text{ is a product node,} \\ \sum_{c \in \text{ch}(n)} \theta_{n,c} \Pr_c(\mathbf{x}) & \text{if } n \text{ is a sum node,} \end{cases} \quad (1)$$

where $f_n(\mathbf{x})$ is a univariate input distribution (e.g., Binomial, Gaussian), and $\theta_{n,c}$ denotes the parameter that corresponds to edge (n,c) . Intuitively, a product node defines a factorized distribution over its inputs, and a sum node represents a mixture over its input distributions weighted by θ . Finally, the probability distribution of a PC is defined as the distribution represented by its root node. The size of a PC $(\mathcal{G}, \theta)$ is defined as the number of parameterized edges in its DAG $\mathcal{G}$.

Inference In contrast to many other generative models, PCs support efficient reasoning over their encoded distribution. One can compute likelihoods by evaluating the PC feed-forward as in Equation 1. Many common reasoning tasks such as marginal probabilities and maximum a posteriori probability (MAP) are also supported by PCs. To guarantee the efficiency for computing these queries, the DAG of the PC should satisfy certain structural constraints. Please refer to [64] for a more detailed summary of various inference scenarios for PCs.

To support linear-time computation (with respect to the size of the PC) of arbitrary marginal queries, PCs need to satisfy two structural properties: smoothness and decomposability. Both are properties of the scope $\phi(n)$ of PC units n , that is, the collection of variables defined by all its input nodes.

Definition 1 (smoothness). A PC $(\mathcal{G}, \theta)$ is smooth if for any sum node $n \in \mathcal{G}$, its children have identical scope: $\forall c_1, c_2 \in \text{ch}(n) : \phi(c_1) = \phi(c_2)$.

Definition 2 (decomposability). A PC $(\mathcal{G}, \theta)$ is decomposable if for any product node $n \in \mathcal{G}$, its children have disjoint scopes: $\forall c_1, c_2 \in \text{ch}(n), c_1 \neq c_2 : \phi(c_1) \cap \phi(c_2) = \emptyset$.

Given a smooth and decomposable PC, querying an arbitrary marginal probability boils down to a feedforward evaluation of its DAG, thus the computation time is linear with respect to the size of the PC.

Sampling and conditional inference Sampling from a PC is performed via ancestral sampling. Starting from the root node, we traverse sum nodes by sampling a child proportionally to the edge weights $\theta_{n,c}$, and traverse product nodes by recursively sampling from all children (since they define independent factorizations over disjoint variable sets). At input nodes, we sample from the corresponding univariate distributions. This procedure generates a complete assignment to all variables in time linear in the circuit size.

For genotype imputation, we compute conditional probabilities $p(X_{\text{missing}}|X_{\text{observed}})$. Smooth and decomposable PCs support exact marginalization over any subset of variables: to marginalize out a variable X_i , we simply replace its input distribution with the constant 1 (summing over all possible values). Conditional queries are then computed as ratios of two marginal queries [37]:

$$p(X_{\text{missing}} = x | X_{\text{observed}} = e) = \frac{p(X_{\text{missing}} = x, X_{\text{observed}} = e)}{p(X_{\text{observed}} = e)}. \quad (2)$$

Both the numerator and denominator can be computed via a single feedforward pass through the circuit, making conditional inference efficient.

4.2.3 HCLTs as probabilistic circuits. HCLTs can be represented as smooth and decomposable PCs, meaning they support efficient (i.e., linear in the size of the PC) computation of marginal queries and likelihoods. When considering both the observed and hidden variables, HCLTs are also deterministic (i.e., for any sum node, its children have disjoint support), meaning that the MAP instance $\arg \max_{\mathbf{x} \in \mathbf{X}, \mathbf{z} \in \mathbf{Z}} \Pr_n(\mathbf{x}, \mathbf{z})$ can be evaluated in linear time. However, HCLTs are not deterministic with respect to the observed variables $\mathbf{X}$ alone, which has implications for parameter estimation as described below.

4.2.4 Parameter estimation. Any probabilistic graphical model (PGM) can be transformed into a PC that encodes the same probability distribution. We demonstrate the high-level idea of this transformation, and refer interested readers to [38] for more details. To transform an HCLT into an equivalent PC, we iteratively encode every conditional probability $\Pr(\mathbf{X}_n | \mathbf{X}_{\text{Pa}(n)})$ by representing each possible value of $\mathbf{X}_n$ as a sum node. Thus, the probability of $\Pr(\mathbf{X}_n = \mathbf{x}_n | \mathbf{X}_{\text{Pa}(n)} = \mathbf{x}_{\text{Pa}(n)})$ can be represented by the weight of an edge connecting $\mathbf{x}_n$ and $\mathbf{x}_{\text{Pa}(n)}$. Take the HCLT in Fig 1 as an example. The

conditional probabilities are encoded into a single PC in a bottom-up manner: we first encode $\Pr(X_5|Z_5)$ and then followed by $\Pr(X_4|Z_4)$ and $\Pr(Z_5|Z_4)$, and so on.

If a PC is smooth, decomposable, and deterministic, its maximum-likelihood estimation (MLE) parameters can be efficiently learned in closed form [65]. To formalize the MLE parameters, we define the context γ_n of any node n as follows. The context of the root node n_r is its support $\text{supp}(n_r)$. The context of any other node is the intersection of its support and the union of its parents' contexts:

$$\gamma_n := \bigcup_{c \in \text{pa}(n)} \gamma_c \cap \text{supp}(n).$$

For any sum node n and its child c , the associated MLE parameter $\theta_{n,c}^*$ on a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ is

$$\theta_{n,c} = F_{\mathcal{D}}(n, c) / \sum_{c \in \text{ch}(n)} F_{\mathcal{D}}(n, c), \quad \text{where } F_{\mathcal{D}}(n, c) := \sum_{i=1}^N \mathbb{1}[\mathbf{x}_i \in \gamma_n \cap \gamma_c]. \quad (3)$$

The quantity $F_{\mathcal{D}}(n, c)$ is called the *circuit flow* of edge (n, c) . Intuitively, circuit flows count the number of samples in $\mathcal{D}$ that “activate” an edge.

However, since HCLTs are not deterministic with respect to the observed variables $\mathbf{X}$, MLE does not have a closed-form expression, and we instead resort to Expectation-Maximization (EM), where in the E step we compute the *expected circuit flow* given incomplete data, and in the M step we estimate the closed-form MLE parameters given those expected flows [56,66].

Concretely, given a deterministic PC $(\mathcal{G}, \theta)$ with root node r and an incomplete dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, the parameters for the next EM iteration are given by [56]:

$$\theta_{n,c}^{(\text{new})} = \text{EF}_{\mathcal{D}, \theta}(n, c) / \sum_{c \in \text{ch}(n)} \text{EF}_{\mathcal{D}, \theta}(n, c), \quad \text{where } \text{EF}_{\mathcal{D}, \theta}(n, c) := \sum_{i=1}^N \mathbb{E}_{\mathbf{z} \sim \text{Pr}_{\theta}(\cdot | \mathbf{x}_i)} [\mathbb{1}[\mathbf{z}\mathbf{x}_i \in \gamma_n \cap \gamma_c]] \quad (4)$$

defines an expected version of circuit flow for edge (n, c) given samples with missing values in $\mathcal{D}$.

Using their equivalent PC representations, HCLTs can be trained efficiently using the PC package PyJuice [40]. By representing graphical models (e.g., HCLTs) as PCs, we take advantage of the structure of the model to extensively parallelize the computation required by EM updates (Equation 4). Further, we develop specialized GPU kernels to significantly speed up the EM algorithm. As a result, despite the model having over 88 million parameters, a single EM epoch on the 1KG dataset can be done in less than 2 seconds.

4.3 Impute5 specifications

Haploid imputation is specified in Impute5, with all other SNPs in the region serving as buffer (observed context). Further, we provide the corresponding fine-scale human recombination map for the chromosome being imputed. These are linked in the Impute5 documentation: <https://github.com/odelaneau/shapeit4/tree/master/maps>.

4.4 Training specifications

Training GPC is straightforward, requiring minimal hyperparameter tuning. We set the number of latent states (for each hidden variable in the HCLT) to 128 for both datasets, the maximum feasible given memory constraints. We observe that changing the pseudocount parameter (used to smooth probability estimates) does not significantly impact the training procedure and can be left at a small default value (0.005).

Based on initial experiments with small validation sets, we find that training for 2,000–5,000 epochs works well for these datasets, though longer training is possible without overfitting (with minimal performance gains). For the 1KG data, we trained GPC for 5,000 epochs. Due to the larger sample size (and longer training time per epoch) in the UKBB data, we stopped at 2,000 epochs. Although we do not tune the pseudocount, this would be the only parameter to consider adjusting for further optimization. Additionally, since we use full-batch EM to learn the circuit parameters, we do not need to tune other standard hyperparameters such as learning rate or batch size.

A key advantage of GPC over other deep generative approaches is that we can probabilistically determine convergence by monitoring held-out log-likelihood, which provides an objective, quantitative stopping criterion. This stands in contrast to methods that lack tractable likelihoods, where convergence must be assessed through slower, less consistent visual inspection methods that are more prone to human error.

The baseline RBM and WGAN methods require substantially more time and experimentation to tune, primarily due to their larger hyperparameter spaces and inability to calculate likelihoods or determine convergence probabilistically. For RBMs, the partition function is intractable due to an exponentially large number of configurations over the visible and hidden nodes. Therefore, we must rely on indirect metrics such as Nearest Neighbor Adversarial Accuracy (AATS) [45] and visual overlap in principal component space. Similarly, WGANs do not define a probability distribution over the data, preventing direct evaluation of sample likelihoods. Here too, we must rely on visual overlap in the principal component space to assess convergence, a subjective and time-consuming process.

We utilize the training code and notebooks provided by the authors on GitHub to tune both methods. Although we initially struggled to achieve optimal performance on the 1KG African data subset, we resolved this issue after consulting with the authors. For the RBM, we modified several parameters (e.g., higher learning rate, fewer epochs, fewer Gibbs sampling steps). For the WGAN, we increased the batch size. Otherwise, all other parameters were kept at their default values. When training on the UKBB data, we retained the default parameters; similar to the GPC case, we used fewer epochs due to longer training time but still achieved good overlap in the principal component space based on visual inspection.

The simpler PGM baselines, INDEP and MARKOV, model the observed SNPs directly without latent variables. INDEP treats each SNP as an independent Bernoulli random variable, estimating $P(X_n = 1)$ separately for each site via its sample mean (calculated by a single pass over the training data). MARKOV is a first-order Markov chain: it models each SNP conditioned only on the immediately preceding one in genomic order, $P(X_n | X_{n-1})$, and is likewise fit in closed form via maximum likelihood. Both models therefore require no iterative training. For HMM, we use the EM algorithm with random initialization. Similar to GPC, we set the number of latent states for HMM at 128 and train for 2,000–5,000 epochs depending on the dataset. Using these settings, the held-out log-likelihood converged effectively. We tried higher values of latent states for HMM but these all resulted in worse log-likelihoods and slower training overall. The HMM baseline is therefore not an external implementation but an ablation of GPC itself: it is compiled with the same circuit library, uses the same number of latent states ($L = 128$), the same pseudocount (0.005), the same full-batch EM procedure and the same epoch budget, and is trained on identical data. The only difference is the latent graph, a chain rather than the learned Chow-Liu tree. Every comparison between GPC and HMM in this work is thus a controlled measurement of what the tree topology contributes, with all other components held fixed.

4.5 Computational cost and the working range of GPC

To characterize computational cost we measured training time and peak GPU memory for GPC, RBM and WGAN on identical data and a single GPU, varying the number of training haplotypes and the number of SNPs (S8 Fig). Each method was run with the default parameters of its published training script. The haplotype counts 500, 1,000, 2,000 and 4,006 double up to the full training set, and each is a nested random subsample so that the trend with N is not confounded by which haplotypes were drawn. The SNP counts 2,047, 4,095, 8,191 and 12,287 are set by WGAN, whose generator emits only lengths of the form $\text{latent size} \times 2^{12} - 1$ and cannot train on fewer than 8,191 SNPs; the two smaller sizes are

therefore zero-padded up to 8,191, as its published pipeline does, which is why its cost is flat below that length. At the largest grid point, $N=4,006$ haplotypes and $M=12,287$ SNPs, GPC trains in 3.3 hours against 22.8 for WGAN and 35.3 for RBM, and this ordering holds at every point of the grid. These figures are for a subset of the high-coverage region, so they run slightly below the roughly four hours quoted in Section 2.4 for a model over all 14,670 of its SNPs. The advantage comes from the number of epochs required rather than the cost of each one: a GPC epoch is more expensive than an RBM epoch (2.35 versus 1.27 seconds), but GPC converges in 5,000 epochs where RBM uses 100,000 and WGAN 2,001, the defaults of their published training scripts. That difference is itself a consequence of how convergence is assessed: GPC exposes a held-out likelihood, so training stops when it flattens, whereas RBM and WGAN provide no such quantity and their epoch counts must be fixed in advance or settled after the fact by inspecting principal-component overlap by eye. Peak memory follows the same pattern in reverse, with RBM lowest at 1.3 GB, GPC at 4.7 GB and WGAN highest at 12.8 GB, the last being largely independent of dataset size because it is set by a fixed architecture.

Training itself is inexpensive: the largest model reported here uses about 5 GB of GPU memory, so a single modern card is sufficient, and neither the number of latent states nor the epoch budget constrains the region size. What sets the limit is host memory during construction of the Chow–Liu tree, which grows with the square of the number of SNPs. In practice this places the working range of GPC at roughly 10,000–15,000 SNPs on a standard workstation, which is the scale of the regions used throughout this work; reaching substantially larger regions requires proportionally more memory. This ceiling reflects how the tree is currently constructed rather than a property of the model: the step builds an explicit graph over all pairs of SNPs that is discarded once the tree is found, and taking an equivalent tree directly from the pairwise statistics uses roughly seven times less memory. We have confirmed on real data that the two routes return the same tree; extrapolating the same quadratic scaling would place the range on the order of 40,000 SNPs, though we have not built a model at that size.

4.6 Evaluation metrics

4.6.1 Nearest neighbor adversarial accuracy. To evaluate the utility and privacy of the AGs, we compute the AATS metric introduced by Yale et al. [45]:

$$\mathcal{AATS} = \frac{1}{2} \left(\frac{1}{n} \sum_{i=1}^n \mathbf{1}(d_{TS}(i) > d_{TT}(i)) + \frac{1}{n} \sum_{i=1}^n \mathbf{1}(d_{ST}(i) > d_{SS}(i)) \right) \quad (5)$$

T represents the true (real) data and S represents the synthetic data. $d_{TS}(i)$ is the distance between sample i in the true data and its nearest neighbor in the synthetic data, while $d_{TT}(i)$ is the distance between sample i in the true data and its nearest neighbor in the true data (not including itself). $d_{ST}(i)$ and $d_{SS}(i)$ are defined analogously for the synthetic data. Following Yale et al. [45], we use Euclidean distance. The $\mathcal{AATS}$ is an average of two components, which we denote AA_{TRUTH} and AA_{SYN} :

$$\underbrace{\frac{1}{n} \sum_{i=1}^n \mathbf{1}(d_{TS}(i) > d_{TT}(i))}_{AA_{\text{TRUTH}}} \quad \underbrace{\frac{1}{n} \sum_{i=1}^n \mathbf{1}(d_{ST}(i) > d_{SS}(i))}_{AA_{\text{SYN}}} \quad (6)$$

Each component returns a value between 0 and 1, with values near 0.5 indicating the ideal balance between utility and privacy. We do not report the overall $\mathcal{AATS}$ because poor values in each component can cancel each other out: for instance, an AA_{TRUTH} of 0 paired with an AA_{SYN} of 1 would average to 0.5, masking both extreme memorization and a collapsed synthetic distribution.

Experimental setup Using a 50/50 train/test split, we construct four datasets of equal size n : a real training set R_{tr} , a real test set R_{te} , and two disjoint halves of the generated synthetic data, A_1 and A_2 . We compute the AATS components

twice: pairing (R_{tr}, A_1) for train metrics and (R_{te}, A_2) for test metrics. For 1KG, $n=2,504$, matching each real split exactly. For UKBB, the real dataset has 13,462 samples per split but we generated only 10,000 synthetic samples total, so we set $n=5,000$: A_1 and A_2 are the first and second halves of the 10,000 generated samples, and 5,000 haplotypes are drawn without replacement from each real split using a fixed random seed.

Standardization For each SNP, we compute the mean μ_j and standard deviation σ_j from R_{tr} only, and apply the transformation $x \mapsto (x - \mu_j)/\sigma_j$ to all four datasets (R_{tr} , R_{te} , A_1 , A_2) before computing any distances. Monomorphic sites ($\sigma_j = 0$) are left unchanged. Fitting the scaler on R_{tr} and applying it uniformly ensures that no information from the test set or the synthetic data influences the scaling. This choice is motivated by the fact that standardization reweights each SNP's contribution to Euclidean distance by $1/\sigma_j \propto 1/\sqrt{p_j(1-p_j)}$, amplifying the contribution of rare variants (low MAF p_j) and making the metric sensitive to failures in rare-variant modeling that would otherwise go undetected.

4.6.2 Wasserstein distance calculation. To quantify differences between real and generated data, we used Wasserstein distances in two complementary settings. For comparisons in PCA space, we reported a two-dimensional Wasserstein distance computed between the empirical distributions of real (test set) and generated individuals. Distances were computed using the entropically regularized optimal transport (Sinkhorn) algorithm as implemented in the `ot` Python package, with a regularization parameter set to $\varepsilon = 2 \times 10^{-3}$, and evaluated in selected two-dimensional PCA subspaces (PC1–PC2, PC3–PC4, and PC5–PC6). Lower values indicate closer agreement between the distributions of real and generated samples in PCA space. We report these distances in two different bases, because they answer different questions. In the *coupled* basis (S1 Table), PCA is fit separately for each generative method on that method's artificial genomes together with the real test samples. This is a valid within-method diagnostic of whether a method's samples resemble the test samples, but each method is then scored in a different basis, with different retained axes and different scaling, so the resulting distances are not directly comparable across methods. In the *common* basis (S2 Table), a single PCA is fit once on the union of all eight datasets (the training set, the test set and the six sets of artificial genomes) and every dataset is projected into that same space before the distances are computed. Because all methods then share one set of axes and one scaling, these values are comparable across methods. The common basis is also the space in which the PCA figures are drawn (Figs 3 and S2), so the reported distances correspond directly to the separation visible in those plots.

To assess similarity at the haplotype level, we computed one-dimensional Wasserstein distances between distributions of pairwise haplotypic distances. For each pair of haplotypes, we computed the Manhattan (cityblock) distance, which for haploid genomes equals the number of differing SNP positions, *i.e.*, Hamming distance. We define two comparison metrics, both using the distribution of pairwise distances within the real test set (real–real) as the reference. The *within* metric compares internal diversity: it measures the Wasserstein distance between the distribution of all pairwise distances among AGs (generated–generated; $N_G(N_G - 1)/2$ values for N_G AGs) and the distribution among real test haplotypes (real–real; $N_R(N_R - 1)/2$ values for N_R real genomes). This quantifies whether generated samples exhibit similar diversity to real samples. The *between* metric compares the distribution of cross-distances between generated and real test haplotypes (generated–real; $N_G \times N_R$ values) to the same real–real reference distribution. This quantifies whether generated haplotypes are as close to real haplotypes as real haplotypes are to each other. One-dimensional Wasserstein distances were computed using `scipy.stats.wasserstein_distance`, which handles distributions with different numbers of samples by treating them as empirical distributions.

Supporting information

S1 Fig. Distribution of haplotypic pairwise Euclidean distances. (a–b) show results for the 1KG dataset; (c–d) for the UKBB dataset. (a,c) visualize distances *within* each dataset (*i.e.*, intra-group variation). (b,d) show distances *between* the AG-generated samples and the test set (*i.e.*, realism and mode collapse). Each panel compares across different generative models (see S1 Table for detailed metrics).

(PDF)

S2 Fig. Principal components analysis for models trained on the UKBB dataset. The top half shows a scatter plot of the top six principal components of the test set (gray) vs. AGs generated via Indep (brown), Markov (pink), HMM (purple), WGAN (green), RBM (orange), and GPC (blue). The left plot considers the train set as “perfectly” generated data (black). The bottom half of the figure shows a density map of the principal components for each dataset. All deep generative models are able to capture the population structure with good accuracy (see [S2 Table](#) for detailed metrics).
(PDF)

S3 Fig. Comparison of the structures of Chow–Liu Tree (CLT) and a Hidden Markov Model (HMM) on the 1KG dataset. (a) Degree distribution of CLT nodes versus HMM nodes. In an HMM chain over N SNPs, exactly 2 nodes have degree 1 and all remaining $N-2$ nodes have degree 2. The CLT instead exhibits a highly heterogeneous degree distribution, with 4,599 leaf nodes, 2,962 degree-2 nodes, and 397 nodes with degree ≥ 5 , including one node (SNP 7579) with maximum degree 161. (b) Distribution of sequential edge spans ($|i-j|$ for each connected pair of SNPs), plotted on a log-scale y -axis. An HMM consists entirely of adjacent-SNP edges (dashed red line at distance 1), whereas the CLT spans distances up to 9,832 SNP positions, with 18.3% of edges connecting SNPs more than 1,000 positions apart. (c) Summary statistics comparing CLT and HMM topologies. Together, these panels demonstrate that the Chow–Liu algorithm learns a tree structure fundamentally distinct from a chain, directly linking SNPs with strong mutual information regardless of their genomic proximity and thereby encoding both local and long-range LD structure.
(PDF)

S4 Fig. Radial visualization of the Chow–Liu Tree (CLT) learned from the 1KG dataset. Each of the 10,000 nodes represents one SNP; edges connect pairs of SNPs whose latent variables are adjacent in the tree. Edge color encodes the sequential SNP distance $|i-j|$ on a log scale, ranging from deep purple (nearby SNPs, $|i-j| \approx 1$) to yellow (SNPs thousands of positions apart, $|i-j| \sim 10^3$). High-degree nodes (degree ≥ 3) are shown as filled circles scaled by degree; leaf and degree-2 nodes appear as small gray circles. The top 8 highest-degree nodes are labeled with their SNP index and degree. In contrast to an HMM, which would appear as a single unbranched chain, the CLT exhibits a highly branching structure. For the highest-degree node SNP 7579 (degree 161), the edges leaving it are not visually apparent as individual lines because the vast majority of its 161 connected SNPs are themselves leaves with no further subtrees extending from them; they appear simply as the small gray circles immediately surrounding it. This contrasts with SNP 8733 (degree 53), whose connected nodes have large subtrees branching out from them, producing the visually prominent clusters that extend outward from that region of the tree. The long orange and yellow edges encode direct connections between SNPs thousands of genomic positions apart, capturing long-range LD that a chain topology cannot represent without propagating information through all intermediate variables.
(PDF)

S5 Fig. General imputation across additional UKBB regions. The black and light blue lines denote imputation results using Impute5 with real reference genomes and direct imputation using GPC, respectively. The other lines show imputation results using Impute5 with AG reference panels. r^2 is shown as a function of MAF for each of the five additional chromosomes (chr2, 11, 12, 14, 18) separately (see [S5 Table](#) for detailed metrics).
(PDF)

S6 Fig. Population-specific imputation in the UKBB dataset. On the left, the AG lines show results using population-specific AGs alone while the light blue line shows direct imputation with GPC trained on population-specific genomes (where the target genomes correspond to non-European populations in part a and African populations in part c). On the right, the AG lines show results using combined reference panels (real European data plus population-specific AGs), and the light blue line shows direct imputation with GPC trained on both European and population-specific genomes (where

the target genomes correspond to non-European populations in part b and African populations in part d). The black line shows results using Impute5 with real European data as the reference panel (see [S7 Table](#) for detailed metrics).

(PDF)

S7 Fig. Imputation accuracy in admixed individuals. As Fig 6a and Fig 8, but with the target set restricted to the 212 admixed haplotypes (106 individuals) of the held-out split: the four American populations (CLM, MXL, PEL, PUR) and the two recently admixed African-ancestry populations (ASW, ACB). No model is retrained; these are the cosmopolitan models of the general 80/20 experiment, scored on a subset of their own test set. Every method is less accurate than on the full test set, as expected, but the ordering is unchanged and GPC (direct) remains the most accurate generative model in both settings. Detailed values are in [S10 Table](#).

(PDF)

S8 Fig. Training cost and peak memory against dataset size. Measured on one NVIDIA RTX A5000 with no other process on the device, each method at the default parameters of its published training script. Left column varies haplotypes N , right column varies SNPs M ; rows are total training time, per-epoch cost, and peak GPU memory. Total time is per-epoch cost times each method's epoch count (5,000 for GPC, 100,000 for RBM, 2,001 for WGAN), which is why GPC is fastest overall despite a more expensive epoch than RBM. WGAN cannot train on fewer than 8,191 SNPs, since its critic halves the sequence twelve times and a shorter input leaves a single position at the innermost layer; the two smaller sizes are zero-padded up to 8,191 as its published pipeline does, so its cost is flat below that.

(PDF)

S1 Table. Distance metrics between real and generated data. Within, Between (S1 Fig): Within denotes the Wasserstein distance between the distributions of pairwise haplotypic distances computed within a single dataset (e.g., generated-generated) and the corresponding distribution computed within the real test set (real-real). Between denotes the Wasserstein distance between the distribution of pairwise haplotypic distances computed between generated and real test individuals (generated-real) and the distribution computed within the real test set (real-real). In both cases, the real test set serves as the reference distribution. **PCA1–2, PCA3–4, PCA5–6:** Wasserstein 2D distances between the PCA representations of real (test set) versus generated individuals. Truth represents the training set. For each PCA distance, one PCA is fit on the method AGs and test data. Bolded values indicate the best among all compared models. Because a separate PCA is fit for each method, these PCA distances are a within-method diagnostic and are not directly comparable across methods; [S2 Table](#) reports the same distances in a single common basis shared by all methods.

(PDF)

S2 Table. PCA distances in a common basis. Wasserstein 2D distances between the PCA representations of real (test set) versus generated individuals, computed after fitting a single PCA once on the union of all eight datasets (the training set, the test set and the six sets of artificial genomes) and projecting every dataset into that same space. This is the space shown in Figs 3 and S2. Unlike [S1 Table](#), where a separate PCA is fit for each method, these values share one set of axes and one scaling and are therefore directly comparable across methods. Truth represents the training set. Bolded values indicate the best among all compared models. The chain-structured HMM is the setting most affected by the change of basis (PCA1–2: 0.0348 $\rightarrow$ 0.0840 on 1KG and 0.0072 $\rightarrow$ 0.0390 on UKBB), because a per-method fit is free to orient its axes around that method's artificial genomes.

(PDF)

S3 Table. LD decay error metrics for Fig 4. Mean absolute error (MAE) and root mean squared error (RMSE) are computed as differences between the mean values of each bin on the LD decay curves across genomic distance. Lower values indicate better preservation of LD structure; best-performing values are bolded.

(PDF)

S4 Table. Summarized metrics for Fig 6. Mean imputation performance across 10 bootstrapped replicates for all, low-frequency (MAF <1%), and rare (MAF <0.1%) SNPs. Best generative model is bolded.

(PDF)

S5 Table. Summarized metrics for S5 Fig (additional UKBB regions). Mean imputation performance across 10 bootstrapped replicates for all, low-frequency (MAF <1%), and rare (MAF <0.1%) SNPs on five additional LD blocks in the UKBB dataset. Best generative model is bolded.

(PDF)

S6 Table. Summarized metrics for Fig 7 (1KG). Mean imputation performance across 10 bootstrapped replicates for all, low-frequency (MAF <1%), and rare (MAF <0.1%) SNPs. Best generative model using combined data is bolded; best using population-specific data only is marked with †.

(PDF)

S7 Table. Summarized metrics for S6 Fig (UKBB). Mean imputation performance across 10 bootstrapped replicates for all, low-frequency (MAF <1%), and rare (MAF <0.1%) SNPs. Best generative model using combined data is bolded; best using population-specific data only is marked with †.

(PDF)

S8 Table. Summarized metrics for Fig 8 (high-coverage 1KG). Mean imputation performance across 10 bootstrapped replicates for all and low-frequency (MAF <1%) SNPs. Best generative model is bolded.

(PDF)

S9 Table. Summarized metrics for Fig 9 (high-coverage 1KG). Mean imputation performance across 10 bootstrapped replicates for all and low-frequency (MAF <1%) SNPs. Best generative model using combined data is bolded; best using population-specific data only is marked with †. In this experiment, GPC (direct)† performs better than models using combined data.

(PDF)

S10 Table. Imputation accuracy restricted to admixed individuals. Mean r^2 [95% CI] over 10 bootstrap replicates, for the 212 haplotypes (106 individuals) of the held-out test split whose 1KG population is admixed: the four American populations (CLM, MXL, PEL, PUR) together with the two recently admixed African-ancestry populations (ASW, ACB). No model is retrained. These are the cosmopolitan models of the general 80/20 experiment, trained on the full training split rather than on population-specific data as in Section 2.6.2, and the models, reference panels and protocol are exactly those of that experiment; only the set of target haplotypes changes. The 1KG and high-coverage 1KG experiments use the same individuals and the same 80/20 split, so the subset is identical in both panels. Best generative model is bolded. Per-population counts are given in S11 Table.

(PDF)

S11 Table. Ancestry composition of the imputation datasets. Counts are numbers of individuals. For 1KG, ancestry is the superpopulation code from the IGSR sample table, restricted to the 2,504 unrelated individuals, with the constituent population codes listed for each superpopulation. One individual carries two superpopulation labels and is shown as its own row; it is included in the non-European group, which is every individual not labeled European. The African target population is the AFR subset. For UKBB, ancestry is self-reported ethnic background (field 21000), and 5,000 individuals were sampled from each of the European, African and non-European groups. The non-European group includes individuals of African ancestry by design, so it overlaps the African group by 1,538 individuals and the three groups together contain 13,462 distinct individuals rather than 15,000.

(PDF)

Acknowledgments

This research was conducted using the UK Biobank Resource under application 331277. We thank the participants of UK Biobank for making this work possible. We also thank the authors of [25], Burak Yelmen and Aurélien Decelle, for their assistance with training and parameter selection for the WGAN and RBM models.

Author contributions

Conceptualization: Xinzhu Wei, Guy Van den Broeck, Sriram Sankararaman.

Data curation: Prateek Anand, Sriram Sankararaman.

Formal analysis: Prateek Anand.

Funding acquisition: Guy Van den Broeck, Sriram Sankararaman.

Investigation: Prateek Anand.

Methodology: Prateek Anand, Anji Liu, Meihua Dang.

Resources: Meihua Dang, Boyang Fu, Xinzhu Wei.

Software: Prateek Anand, Anji Liu.

Supervision: Guy Van den Broeck, Sriram Sankararaman.

Validation: Meihua Dang, Boyang Fu, Xinzhu Wei.

Visualization: Prateek Anand.

Writing – original draft: Prateek Anand.

Writing – review & editing: Anji Liu, Boyang Fu, Guy Van den Broeck, Sriram Sankararaman.

References

1. Nordborg M. Coalescent theory. Handbook of Statistical Genomics: Two Volume Set. 2019. p. 145–30.
2. Wakeley J. Developments in coalescent theory from single loci to chromosomes. *Theor Popul Biol.* 2020;133:56–64. <https://doi.org/10.1016/j.tpb.2020.02.002> PMID: 32145225
3. Marchini J, Howie B. Genotype imputation for genome-wide association studies. *Nat Rev Genet.* 2010;11(7):499–511. <https://doi.org/10.1038/nrg2796> PMID: 20517342
4. Browning SR, Browning BL. Haplotype phasing: existing methods and new developments. *Nat Rev Genet.* 2011;12(10):703–14. <https://doi.org/10.1038/nrg3054> PMID: 21921926
5. Mas Montserrat D, Bustamante C, Ioannidis A. Class-conditional vae-gan for local-ancestry simulation. *arXiv preprint.* 2019;arXiv:1911.13220.
6. Hudson RR. Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics.* 2002;18(2):337–8. <https://doi.org/10.1093/bioinformatics/18.2.337> PMID: 11847089
7. Hellenthal G, Stephens M. msHOT: modifying Hudson's ms simulator to incorporate crossover and gene conversion hotspots. *Bioinformatics.* 2007;23(4):520–1. <https://doi.org/10.1093/bioinformatics/btl622> PMID: 17150995
8. Excoffier L, Foll M. fastsimcoal: a continuous-time coalescent simulator of genomic diversity under arbitrarily complex evolutionary scenarios. *Bioinformatics.* 2011;27(9):1332–4. <https://doi.org/10.1093/bioinformatics/btr124> PMID: 21398675
9. Kelleher J, Etheridge AM, McVean G. Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. *PLoS Comput Biol.* 2016;12(5):e1004842. <https://doi.org/10.1371/journal.pcbi.1004842> PMID: 27145223
10. Baumdicker F, Bisschop G, Goldstein D, Gower G, Ragsdale AP, Tsambos G, et al. Efficient ancestry and mutation simulation with msprime 1.0. *bioRxiv.* 2021.
11. Hudson RR. Properties of a neutral allele model with intragenic recombination. *Theor Popul Biol.* 1983;23(2):183–201. [https://doi.org/10.1016/0040-5809\(83\)90013-8](https://doi.org/10.1016/0040-5809(83)90013-8) PMID: 6612631
12. Griffiths RC, Marjoram P, Griffiths and Paul Marjoram. An Ancestral Recombination Graph. In Donnelly P, Tavaré S, editors. *Progress in Population Genetics and Human Evolution, IMA Volumes in Mathematics and its Applications.* Vol. 87. Springer; 1997. p. 257–70.
13. Wiuf C, Hein J. Recombination as a point process along sequences. *Theor Popul Biol.* 1999;55(3):248–59. <https://doi.org/10.1006/tpbi.1998.1403> PMID: 10366550

14. McVean GAT, Cardin NJ. Approximating the coalescent with recombination. *Philos Trans R Soc Lond B Biol Sci*. 2005;360(1459):1387–93. <https://doi.org/10.1098/rstb.2005.1673> PMID: 16048782
15. Marjoram P, Wall JD. Fast “coalescent” simulation. *BMC Genet*. 2006;7:16. <https://doi.org/10.1186/1471-2156-7-16> PMID: 16539698
16. Li N, Stephens M. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*. 2003;165(4):2213–33. <https://doi.org/10.1093/genetics/165.4.2213> PMID: 14704198
17. Rabiner L, Juang B. An introduction to hidden Markov models. *IEEE ASSP Mag*. 1986;3(1):4–16. <https://doi.org/10.1109/massp.1986.1165342>
18. Scheet P, Stephens M. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am J Hum Genet*. 2006;78(4):629–44. <https://doi.org/10.1086/502802> PMID: 16532393
19. Delaneau O, Marchini J, Zagury J-F. A linear complexity phasing method for thousands of genomes. *Nat Methods*. 2011;9(2):179–81. <https://doi.org/10.1038/nmeth.1785> PMID: 22138821
20. Howie B, Fuchsberger C, Stephens M, Marchini J, Abecasis GR. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. *Nat Genet*. 2012;44(8):955–9. <https://doi.org/10.1038/ng.2354> PMID: 22820512
21. Baran Y, Pasaniuc B, Sankararaman S, Torgerson DG, Gignoux C, Eng C, et al. Fast and accurate inference of local ancestry in latino populations. *Bioinformatics*. 2012;28(10):1359–67.
22. Price AL, Tandon A, Patterson N, Barnes KC, Rafaels N, Ruczinski I, et al. Sensitive detection of chromosomal segments of distinct ancestry in admixed populations. *PLoS Genet*. 2009;5(6):e1000519. <https://doi.org/10.1371/journal.pgen.1000519> PMID: 19543370
23. Battey CJ, Coffing GC, Kern AD. Visualizing population structure with variational autoencoders. *G3 (Bethesda)*. 2021;11(1):jkaa036. <https://doi.org/10.1093/g3journal/jkaa036> PMID: 33561250
24. Yelmen B, Decelle A, Ongaro L, Marnetto D, Tallec C, Montinaro F, et al. Creating artificial human genomes using generative neural networks. *PLoS Genet*. 2021;17(2):e1009303. <https://doi.org/10.1371/journal.pgen.1009303> PMID: 33539374
25. Yelmen B, Decelle A, Boulos LL, Szatkownik A, Furtlehner C, Charpiat G, et al. Deep convolutional and conditional neural networks for large-scale genomic data generation. *PLoS Comput Biol*. 2023;19(10):e1011584. <https://doi.org/10.1371/journal.pcbi.1011584> PMID: 37903158
26. Szatkownik A, Furtlehner CF, Charpiat G, Yelmen B, Jay F. Towards creating longer genetic sequences with gans: Generation in principal component space. In Knowles DA, Mostafavi S, editors. *Proceedings of the 18th Machine Learning in Computational Biology meeting, volume 240 of Proceedings of Machine Learning Research*. 2024. p. 110–22.
27. Riley R, Mathieson I, Mathieson S. Interpreting generative adversarial networks to infer natural selection from genetic data. *Genetics*. 2024;226(4):iyae024. <https://doi.org/10.1093/genetics/iyae024> PMID: 38386895
28. Shaked A, Gronau I. Genome-AC-GAN: Enhancing synthetic genotype generation through auxiliary classification. *bioRxiv*. 2024.
29. Geleta M, Montserrat DM, Giro-I-Nieto X, Ioannidis AG. Autoencoders for genomic variation analysis. *Genome Res*. 2026;36(2):348–60. <https://doi.org/10.1101/gr.280086.124> PMID: 41558827
30. Burnard C, Mancheron A, Ritchie W. Generating realistic artificial human genomes using adversarial autoencoders. *NAR Genom Bioinform*. 2025;7(3):lqaf101. <https://doi.org/10.1093/nargab/lqaf101> PMID: 40708851
31. Kenneweg P, Dandinasivara R, Luo X, Hammer B, Schönhuth A. Generating synthetic genotypes using diffusion models. *Bioinformatics*. 2025;41(1):i484–92. <https://doi.org/10.1093/bioinformatics/btaf209> PMID: 40662828
32. Rubinacci S, Delaneau O, Marchini J. Genotype imputation using the Positional Burrows Wheeler Transform. *PLoS Genet*. 2020;16(11):e1009049. <https://doi.org/10.1371/journal.pgen.1009049> PMID: 33196638
33. Browning BL, Zhou Y, Browning SR. A One-Penny Imputed Genome from Next-Generation Reference Panels. *Am J Hum Genet*. 2018;103(3):338–48. <https://doi.org/10.1016/j.ajhg.2018.07.015> PMID: 30100085
34. Fuchsberger C, Abecasis GR, Hinds DA. Minimac2: faster genotype imputation. *Bioinformatics*. 2014;31(5):782–4.
35. Liu A, Van den Broeck G. Tractable regularization of probabilistic circuits. In: *Advances in Neural Information Processing Systems 35 (NeurIPS)*. 2021.
36. Pritchard JK, Przeworski M. Linkage disequilibrium in humans: models and data. *Am J Hum Genet*. 2001;69(1):1–14. <https://doi.org/10.1086/321275> PMID: 11410837
37. Vergari A, Choi YJ, Peharz R, Van den Broeck G. Probabilistic circuits: representations, inference, learning and applications. In: *AAAI Tutorial*. 2020.
38. Choi YJ, Vergari A, Van den Broeck G. Probabilistic circuits: A unifying framework for tractable probabilistic models. 2020.
39. Chow C, Liu C. Approximating discrete probability distributions with dependence trees. *IEEE Trans Inform Theory*. 1968;14(3):462–7. <https://doi.org/10.1109/tit.1968.1054142>
40. Liu A, Ahmed K, Van den Broeck G. Scaling tractable probabilistic circuits: A systems perspective. In: *Proceedings of the 41th International Conference on Machine Learning (ICML)*. 2024.
41. Auton A, Brooks LD, Durbin RM, Garrison EP, Kang HM, et al, 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*. 2015;526(7571):68–74. <https://doi.org/10.1038/nature15393> PMID: 26432245

42. Bycroft C, Freeman C, Petkova D, Band G, Elliott LT, Sharp K, et al. The UK Biobank resource with deep phenotyping and genomic data. *Nature*. 2018;562(7726):203–9. <https://doi.org/10.1038/s41586-018-0579-z> PMID: 30305743
43. Byrsk-Bishop M, Evani US, Zhao X, Basile AO, Abel HJ, Regier AA, et al. High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*. 2022;185(18):3426–3440.e19.
44. Yu K, Das S, LeFaive J, Kwong A, Pleiness J, Forer L, et al. Meta-imputation: An efficient method to combine genotype data after imputation with multiple reference panels. *Am J Hum Genet*. 2022;109(6):1007–15.
45. Yale A, Dash S, Dutta R, Guyon I, Pavao A, Bennett KP. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*. 2020;416:244–55. <https://doi.org/10.1016/j.neucom.2019.12.136>
46. Shokri R, Stronati M, Song C, Shmatikov V. Membership Inference Attacks Against Machine Learning Models. In: 2017 IEEE Symposium on Security and Privacy (SP). IEEE; 2017. p. 3–18.
47. Homer N, Szelinger S, Redman M, Duggan D, Tembe W, Muehling J, et al. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS Genetics*. 2008;4(8):e1000167.
48. Fredrikson M, Jha S, Ristenpart T. Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security. 2015. p. 1322–33.
49. Park M, Foulds J, Chaudhuri K, Welling M. Dp-em: Differentially private expectation maximization. 2016.
50. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. *Adv Neural Inform Process Syst*. 2014;2672–80.
51. Kingma DP, Welling M. Auto-encoding variational bayes. *arXiv preprint*. 2013;arxiv:1312.6114.
52. Nourani M, Roy C, Rahman T, Ragan ED, Ruozzi N, Gogate V. Don't explain without verifying veracity: An evaluation of explainable AI with video activity recognition. *CoRR*. 2020;abs/2005.02335.
53. Ahmed K, Zeng Z, Niepert M, Van den Broeck G. Simple: A gradient estimator for k-subset sampling. In: ICLR. 2023.
54. Khosravi P, Choi Y, Liang Y, Vergari A, Van den Broeck G. On tractable computation of expected predictions. *Adv Neural Inform Process Syst*. 2019;32:11169–80.
55. Selvam NR, Van den Broeck G, Choi Y. Certifying Fairness of Probabilistic Circuits. *AAAI*. 2023;37(10):12278–86. <https://doi.org/10.1609/aaai.v37i10.26447>
56. Choi Y, Dang M, Van den Broeck G. Group Fairness by Probabilistic Modeling with Latent Fair Decisions. *AAAI*. 2021;35(13):12051–9. <https://doi.org/10.1609/aaai.v35i13.17431>
57. Correia A, Peharz R, de Campos CP. Joints in random forests. In: Advances in Neural Information Processing Systems 33 (NeurIPS). 2020.
58. Li W, Zeng Z, Vergari A, Van den Broeck G. Tractable computation of expected kernels. In: Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI). 2021.
59. Dang M, Vergari A, Van den Broeck G. Strudel: Learning structured-decomposable probabilistic circuits. In: Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM). 2020.
60. Zhang H, Juba B, Van den Broeck G. Probabilistic generating circuits. In: Proceedings of the 38th International Conference on Machine Learning (ICML). 2021.
61. Darwiche A. A logical approach to factoring belief networks. In: Proceedings of KR. 2002. p. 409–20.
62. Poon H, Domingos P. Sum-product networks: A new deep architecture. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). 2011. p. 689–90.
63. Rahman T, Kothalkar P, Gogate V. Cutset Networks: A Simple, Tractable, and Scalable Approach for Improving the Accuracy of Chow-Liu Trees. In: Joint European conference on machine learning and knowledge discovery in databases. Springer; 2014. p. 630–45.
64. Vergari A, Choi YJ, Liu A, Teso S, Van den Broeck G. A compositional atlas of tractable circuit operations for probabilistic inference. In: Advances in Neural Information Processing Systems 35 (NeurIPS). 2021.
65. Doga K, Van den Broeck G, Choi A, Darwiche A. Probabilistic sentential decision diagrams. In: Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning. 2014.
66. Daphne Koller and Nir Friedman. Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning. The MIT Press; 2009.