GPC: DEEP GENERATIVE MODEL OF GENETIC VARIATION DATA IMPROVES IMPUTATION ACCURACY IN PRIVATE POPULATIONS

Prateek Anand[1], **Anji Liu**[2], **Meihua Dang**[3], **Boyang Fu**[4], **Xinzhu Wei**[5],
Guy Van den Broeck[1,*], **Sriram Sankararaman**[1,6,7,*,†]
[1]Department of Computer Science, University of California, Los Angeles
[2]Department of Computer Science, National University of Singapore
[3]Department of Computer Science, Stanford University
[4]Department of Biomedical Informatics, Harvard Medical School
[5]Department of Computational Biology, Cornell University
[6]Department of Human Genetics, University of California, Los Angeles
[7]Department of Computational Medicine, University of California, Los Angeles
prateek@cs.ucla.edu, anjiliu@comp.nus.edu.sg, mhdang@cs.stanford.edu,
boyang_fu@hms.harvard.edu, aprilwei@cornell.edu,
guyvdb@cs.ucla.edu, sriram@cs.ucla.edu

ABSTRACT

Artificial genomes (AGs) are increasingly used to benchmark genomic pipelines, test population genetic hypotheses, and construct reference panels for genotype imputation, while avoiding restrictions associated with sharing real genomes. However, existing approaches often struggle to jointly achieve realism, computational efficiency, and privacy preservation. 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 captures long-range dependencies among SNPs and is simple to train. We evaluate GPC across multiple ancestries in two large-scale datasets, the 1000 Genomes Project and UK Biobank. GPC matches or exceeds prior methods in generating AGs that resemble real genomes with the AGs retaining population structure underlying the training genomes. The AGs from GPC more faithfully reproduce patterns of linkage disequilibrium (LD; correlations between nearby genetic variants) across length scales. We also find that GPC consistently improves imputation accuracy by 3–33% in r^2 over the next best generative model, with gains of 13–279% for low-frequency variants (MAF <1%). For underrepresented populations, GPC improves accuracy by 12–96% over European-only reference panels. Finally, we demonstrate that GPC provides improved privacy-utility tradeoffs compared to existing approaches, enabling accurate inference when sharing real genomes is restricted.

1 INTRODUCTION

Generative models of genetic variation are central to population genomics, supporting genotype imputation (Marchini & Howie, 2010), haplotype phasing (Browning & Browning, 2011), and the generation of artificial genomes (AGs) for benchmarking and hypothesis testing (Hudson, 2002; Kelleher et al., 2016; Baumdicker et al., 2021). As sharing primary genetic data becomes increasingly restricted due to privacy constraints, accurate generative models have become essential—trained models or simulated data can be shared without exposing identifiable genomic information.

Recent deep generative models based on GANs (Yelmen et al., 2023; Szatkownik et al., 2024), VAEs (Battey et al., 2021; Geleta et al., 2023), and RBMs (Yelmen et al., 2023) can produce realistic AGs but have critical limitations: they lack tractable likelihoods for principled model comparison

[*] Joint supervision. [†] Corresponding author.

and cannot directly compute conditional probabilities needed for imputation. Training is also challenging, requiring extensive hyperparameter tuning with heuristic convergence criteria.

We introduce Genetic Probabilistic Circuits (GPC), a deep generative model based on hidden Chow-Liu trees (HCLTs) represented as probabilistic circuits (PCs). GPC captures long-range linkage disequilibrium (LD) through flexible tree structures over latent variables, while PCs enable exact likelihood computation and conditional inference in linear time. This tractability allows GPC to perform genotype imputation directly via conditional queries, rather than requiring AG generation as an intermediate step.

Across the 1000 Genomes Project and UK Biobank, we show that GPC (1) achieves higher held-out likelihoods than HMMs and other baselines, (2) generates AGs that faithfully preserve population structure and LD patterns across all length scales, (3) improves imputation accuracy, particularly for underrepresented populations, and (4) provides better privacy-utility tradeoffs than existing deep generative approaches. Specifically, GPC achieves 3–33% higher imputation r^2 than the next best generative approach, with improvements of 13–279% for low-frequency variants. In population-specific settings, direct imputation with GPC outperforms European reference panels by 12–96%.

2 Background

Population genetic simulators traditionally rely on the coalescent model (Hudson, 1983), generating genomes based on demographic history, mutation, and recombination. While highly expressive, exact inference is computationally challenging. Tractable approximations include Markovian coalescent simulators (McVean & Cardin, 2005; Kelleher et al., 2016) and the product-of-approximate-conditionals (PAC) model (Li & Stephens, 2003), which yields hidden Markov models (HMMs). HMM-based methods have been highly successful for phasing (Scheet & Stephens, 2006; Delaneau et al., 2012), imputation (Howie et al., 2012), and ancestry inference (Baran et al., 2012).

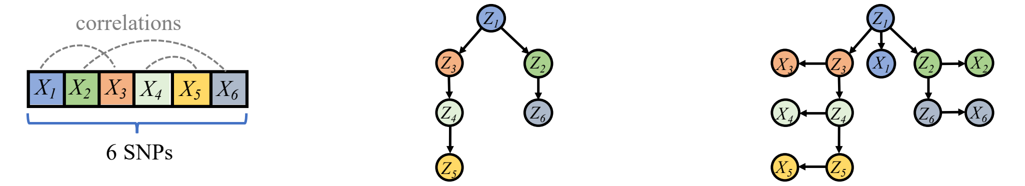
More recently, deep generative models such as GANs, VAEs, and RBMs have been applied to genetic data, offering greater expressivity than HMMs and generating AGs that resemble real genomes in PCA projections (Yelmen et al., 2023; Battey et al., 2021). However, GANs and RBMs lack tractable likelihoods, and VAEs provide only lower bounds, limiting quantitative model comparison. For imputation, these models require generating AGs to serve as reference panels for external tools (Rubinacci et al., 2020; Browning et al., 2018), rather than computing conditional probabilities directly.

Hidden Chow-Liu trees (HCLTs) (Liu & Van den Broeck, 2021) address these limitations by generalizing HMMs to allow latent variables to form arbitrary tree structures learned via the Chow-Liu algorithm (Chow & Liu, 1968), capturing long-range dependencies that chain-structured HMMs cannot. Representing HCLTs as probabilistic circuits (Vergari et al., 2020; Choi et al., 2020) enables exact marginal and conditional queries in time linear in circuit size, combining expressivity with tractability.

3 Methods Overview

GPC is based on hidden Chow-Liu trees (HCLTs) (Liu & Van den Broeck, 2021), latent variable models where each observed SNP X_n is associated with a hidden variable Z_n, and the hidden variables form a tree-structured graphical model (Figure 1). HCLTs generalize hidden Markov models (HMMs): whereas HMMs impose a fixed chain structure over hidden variables corresponding to consecutive SNPs, HCLTs learn an arbitrary tree structure via the Chow-Liu algorithm (Chow & Liu, 1968). By relaxing the chain assumption, HCLTs make fewer structural assumptions about how SNPs depend on one another, allowing the model to better capture correlations between variants — including those that are far apart in the genome. The tree structure is learned directly from data, so the model adapts to the correlation patterns present in the population being modeled rather than assuming a fixed topology.

We represent HCLTs as probabilistic circuits (PCs) (Vergari et al., 2020; Choi et al., 2020), a circuit-based formalism that enables exact computation of marginal and conditional probabilities in time linear in the number of SNPs. This tractability is what distinguishes GPC from existing deep generative approaches: rather than relying on approximate inference or requiring AGs as an intermediate step, GPC can directly compute the probability of unobserved variants conditioned on observed ones, enabling principled genotype imputation. The PC representation also enables GPU-accelerated parameter learning via Expectation-Maximization using the PyJuice package (Liu et al., 2024), making it feasible to train models with tens of millions of parameters in a few hours on a single GPU — a scale that would be computationally prohibitive for classical graphical model implementations. Sampling AGs is performed via ancestral sampling over the PC, and convergence can be monitored objectively via held-out log-likelihood. Full model and inference details are provided in Appendix A.2.



(a) Genetic data with 6 SNPs

(b) Learned CLT structure captures strong pairwise correlation between SNPs.

(c) Learned HCLT structure given the SNPs in (a).

Figure 1: **Generating HCLT structures given genetic data.** The hidden variables (Z_i) corresponding to SNPs with high pairwise correlations (X_i) are connected to each other in the HCLT graphical model.

4 Results

4.1 Data

We evaluate GPC on three datasets: 1000 Genomes Project Phase 3 (1KG) (Auton et al., 2015), UK Biobank (UKBB) (Bycroft et al., 2018), and high-coverage 1KG (Byrska-Bishop et al., 2022). The high-coverage dataset is used only for array-based imputation (see Appendix A.1 for details).

4.2 Evaluation

Baselines To benchmark our model performance in estimating density and simulating artificial genomes, we first compare it to three popular 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).

We also compare against deep learning methods that have been proposed for AG generation: (1) generative adversarial networks (WGAN) and (2) Restricted Boltzmann machines (RBM) as implemented in (Yelmen et al., 2021). For both deep learning baselines, we use the samples generated by the corresponding authors for comparison[1] in our 1KG experiments, but we also retrain their models on our own training split to ensure a fair comparison.

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 that include the top principal components and linkage disequilibrium at pairs of SNPs; (3) genotype imputation which evaluates the models' utility on an important downstream task; (4) and privacy analysis of the AGs sampled from each model.

4.3 Training Specifications

Training GPC requires minimal hyperparameter tuning: we set the number of latent states to 128 and use a small pseudocount (0.005) for smoothing. On a single NVIDIA RTX A5000 GPU, GPC trains in 2–6 hours depending on dataset size, with per-sample generation in under 5 seconds and

[1] Note that (Yelmen et al., 2021) did not do train/test splits so WGAN and RBM are actually trained on train+test.

Table 1: **Comparison of probabilistic models that support tractable likelihood computation in 1KG and UKBB data.** Averaged training and test log-likelihoods and model sizes 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**
	#params	10.00k	39.99k	163774.98k	88473.73k
UKBB	**train LL**	-1642.62	-1360.86	-554.88	**-120.10**
	test LL	-1648.03	-1362.16	-554.38	**-127.75**
	#params	9.82k	39.27k	160825.86k	88850.56k

imputation in 20 milliseconds. Unlike GANs and RBMs, convergence can be monitored via held-out log-likelihood, providing an objective stopping criterion. Full training details are in Appendix A.2.4.

4.4 Reconstructing Local Population 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 (Indep, Markov, HMM, WGAN, and RBM) 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 fully-factorized distributions, Markov chains, and HMMs 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 genomes within a single dataset or between the test dataset and an AG dataset and compute the Wasserstein distance between these pairs of distributions where 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 (Figure S3 and rows 1-2 of Table S1).

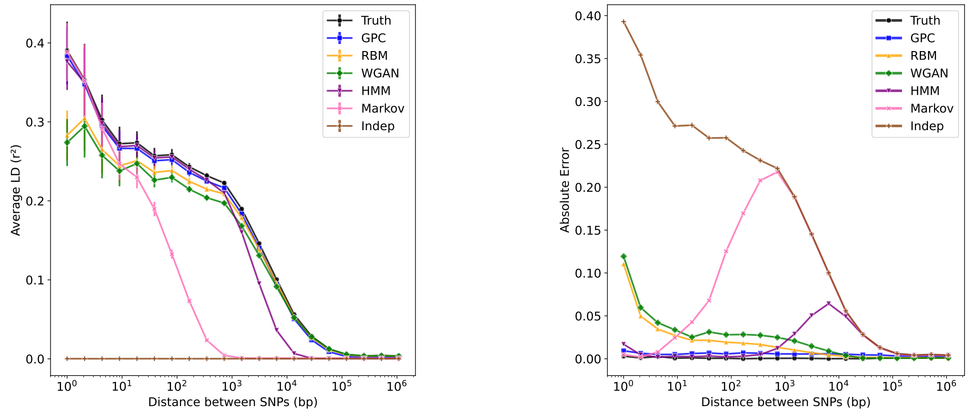
We then analyze the quality of AGs generated by all models in terms of capturing commonly-used summaries of genetic variation data. To visualize all methods in the same latent space, we merge eight datasets (training set, test set, and AGs from six methods) and apply a single PCA to the combined data. 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 (Figure S1 for 1KG and Figure S2 for UKBB).

To quantify the accuracy of these summaries, we computed the Wasserstein distance between the 2D PCA representations of the test data versus the simulated data (Table S1). Wasserstein distances between the 2D PCA representations of test data versus simulated data for the deep learning methods (WGAN, RBM, and GPC) tend to be lower than the other methods.

Since SNPs from a given genomic region tend to be correlated, 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. In both sets of plots, SNPs that were fixed (monomorphic) in at least one of the datasets (ground truth or AGs) were removed before LD computations. While WGAN and RBM tend to be accurate at longer length scales and HMM and Markov are accurate at shorter length scales, GPC is accurate across all length scales (Figure 2).

4.5 Imputation Accuracy

A key downstream application of generative models is genotype imputation. One approach is to generate AGs as reference panels for an imputation tool; we use Impute5 (Rubinacci et al., 2020) for this purpose. Alternatively, GPC can perform imputation directly via conditional probability



(a) LD decay (full range)

(b) Absolute error per bin

Figure 2: **Linkage disequilibrium decay of AGs (1KG).** LD was estimated as pairwise squared correlations r^2 between SNP genotypes of the test data and AGs. Distances were binned on a logarithmic scale, and within each bin the mean r^2 was computed (error bars denote the standard error of the mean). Truth represents the LD distribution of the training data. (a) LD decay across the full length scale of the genomic locus; (b) absolute error in mean r^2 per distance bin relative to the test data. GPC matches the true LD distribution across all SNP distances (see Figure S4 for UKBB results and Table S2 for detailed metrics).

queries over the learned model. This *direct imputation* capability is unique to GPC among the deep generative approaches we consider, and directly targets the imputation objective rather than using AG generation as an intermediate step.

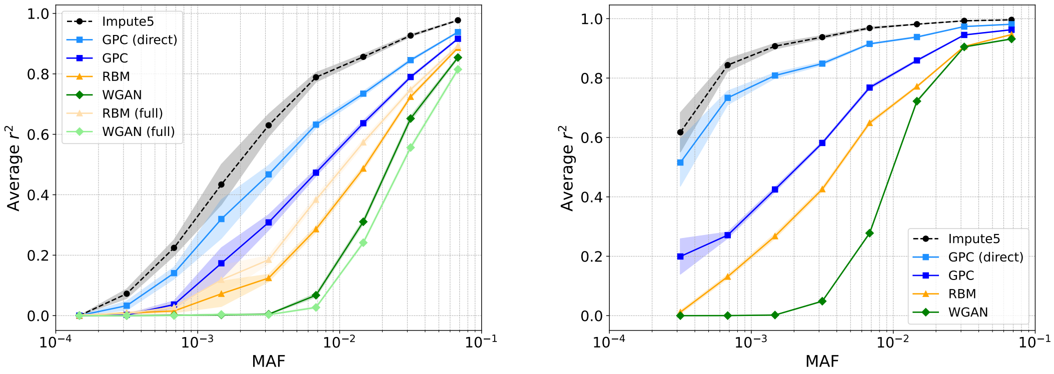
Following the same training protocol as in Section 4.4, we use 80/20 train/test splits and generate the same number of AGs per dataset. For population-specific experiments, the same 80/20 split is applied within the target population. Following the experimental protocol established in prior work (Rubinacci et al., 2020; Browning et al., 2018; Yu et al., 2022), 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 at the target SNP and the true allelic state at the target SNP (excluding monomorphic SNPs); results are aggregated into logarithmically spaced bins by minor allele frequency (MAF) with 95% confidence intervals computed from 10 bootstrapped replicates. This single-SNP imputation protocol follows the approach of Yu et al. (2022) (Sections 4.5.1 and 4.5.2). We additionally evaluate multi-SNP imputation from genotyping arrays following Rubinacci et al. (2020) and Browning et al. (2018) for which we use the high-coverage 1KG dataset (Section 4.5.3). Details on imputation using Impute5 are provided in Section A.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. 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.

4.5.1 General Imputation

We first examine the general setting, in which models are trained and tested 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 Yelmen et al. (2023).

We see that reference panels composed entirely of real genomes achieve the highest imputation accuracy across all MAF bins, providing an approximate upper bound on achievable performance (Figure 3a for 1KG and Figure 3b for UKBB). Among models capable of generating AGs, 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 MAF $< 1\%$). GPC (direct) imputation—via conditional probability queries 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, likely due to directly targeting the imputation objective rather than use of simulated AGs.



(a) 1KG

(b) UKBB

Figure 3: **General imputation.** 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. RBM/WGAN (full) show results using AGs provided by the authors of Yelmen et al. (2023) (see Table S3 for detailed metrics).

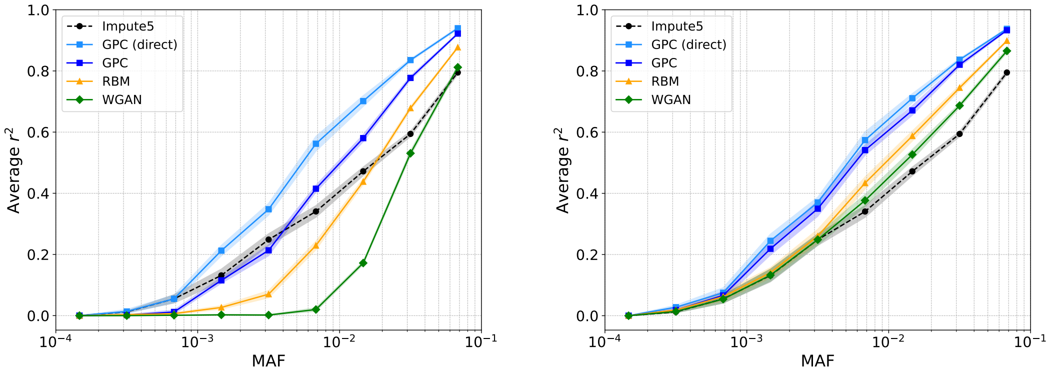
4.5.2 Population-Specific Imputation

We next consider a scenario in which reference genomes for a target population are 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. In this setting, imputation accuracy in the target population can be degraded when public European reference genomes are used. 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 (Figures S5 and S6 for 1KG and UKBB respectively). Figure 4 highlights results for the non-European target population in 1KG. 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 a reference panel of European genomes, consistent with the distributional mismatch between the reference and target populations. 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). Finally, we find that combining population-specific AGs with European genomes consistently improves accuracy across all MAF bins. Averaged across datasets and ancestries, GPC (direct) achieves a 0.011 (1.8%) improvement in r^2 over itself when including European genomes in the reference panel (0.023 (12.3%) for low-frequency variants).

A useful contrast emerges when comparing the 1KG and UKBB settings. In UKBB, the real European reference panel performs noticeably better than in 1KG, particularly at low MAF. This is a direct consequence of our sampling strategy (Section 4.1), where we intentionally subsampled equal numbers of individuals from each population. By matching the sample sizes, we ensured that differences in performance could not be attributed simply to the overwhelming availability of European data. Even under this balanced design, however, GPC (direct) consistently achieves the highest accuracy on average across all SNPs. At very low MAF, real European reference panels still have a slight advantage – likely reflecting the challenge of estimating rare variant distributions from limited population-specific data – but GPC again remains the most accurate when the European and population-specific data are combined. Overall, these results demonstrate that GPC produces high-fidelity population-specific AGs and that its imputation capability offers substantial gains across the full MAF spectrum, even in settings where European reference data are equally abundant.

Taken together, population-specific AGs from GPC can substantially enhance imputation accuracy in underrepresented groups, either by augmenting existing reference panels or through direct conditional imputation.



(a) Non-European AGs only

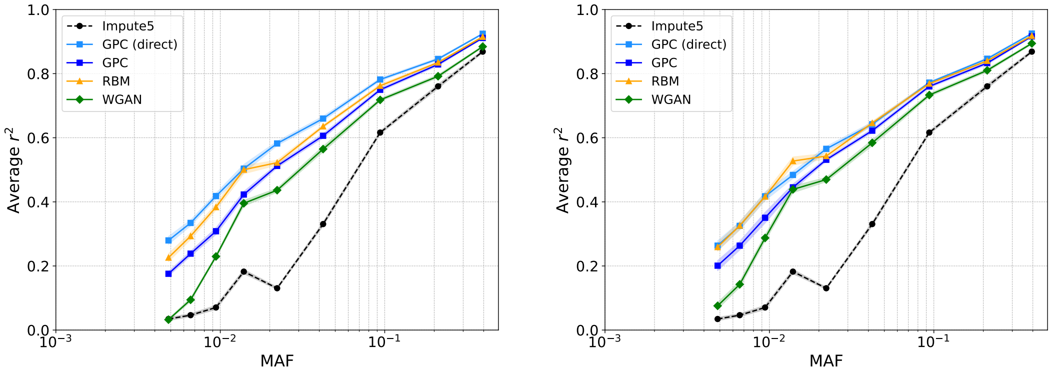
(b) Combined with European data

Figure 4: **Population-specific imputation for non-European target (1KG).** The black line shows results using Impute5 with real European data as the reference panel. (a) AG lines show results using non-European AGs alone, and the light blue line shows direct imputation with GPC trained on non-European data. (b) AG lines show results using combined reference panels (real European data plus non-European AGs), and the light blue line shows direct imputation with GPC trained on both European and non-European data (see Table S4 for detailed metrics).

4.5.3 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 A.1.3 for details). This scenario requires jointly imputing 86% of SNPs in the genomic region using the SNPs typed on the array.

GPC (direct) remains the most accurate in both the general (Figure S7) and population-specific settings (Figure S8). Figure 5 highlights results for the non-European target population. In the general setting, GPC (direct) achieves a 0.020 (3.5%) improvement in r^2 over the next best method, RBM (0.043 (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). In contrast 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 population-specific modeling may provide accuracy gains over aggregating across populations in realistic imputation scenarios.



(a) Non-European AGs only

(b) Combined with European data

Figure 5: **Array-based imputation for non-European target.** The task is to simultaneously impute 86% of SNPs ($12,551$ of $14,670$) that are absent from the HumanOmni5Exome array using high-coverage 1KG data. The black line shows results using Impute5 with real European data as the reference panel. (a) AG lines show results using non-European AGs alone, and the light blue line shows direct imputation with GPC trained on non-European data. (b) AG lines show results using combined reference panels (real European data plus non-European AGs), and the light blue line shows direct imputation with GPC trained on both European and non-European data (see Table S7 for detailed metrics).

4.6 Privacy

To evaluate the privacy of the AGs, we compute the nearest neighbor adversarial accuracy (AATS) metric introduced by Yale et al. (2020). This metric measures how well synthetic data can be distinguished from real data based on nearest-neighbor distances. 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 close to 0.5 for both components indicate an ideal balance between utility (synthetic data resembles real data) and privacy (synthetic data is not simply copying real data). We report AA_{TRUTH} and AA_{SYN} separately rather than their average, as the overall $AATS$ can mask pathological cases where poor values cancel out (see Section A.4.1 for details).

We observe that GPC obtains AA_{TRUTH} and AA_{SYN} closest to 0.5 among all deep generative models for both the 1KG and UKBB datasets (Table 2), indicating the best balance between utility and privacy. The near-zero AA_{SYN} values for RBM suggest that its synthetic samples closely mimic individual training examples rather than forming a coherent independent distribution — a sign of mode collapse and a meaningful privacy risk despite its reasonable imputation performance. Conversely, WGAN exhibits high values for both components, indicating that its synthetic distribution is too far from the real data, sacrificing utility. This ordering is consistent with what we observe in imputation accuracy, where RBM trails GPC but outperforms WGAN across most settings. However, we note that even GPC's values are not yet at the ideal 0.5, and the AA_{TRUTH} values in particular remain above 0.5, suggesting that the synthetic data is still somewhat distinguishable from real data. Stronger privacy guarantees — for instance via differentially private EM algorithms — remain an important direction for future work.

5 Discussion

We introduced GPC, a deep generative model for genetic variation that combines the expressivity of hidden Chow-Liu trees with the tractability of probabilistic circuits. Unlike GANs and RBMs,

Table 2: **Nearest neighbor adversarial accuracy** (AA_{TRUTH}, AA_{SYN}) for synthetic data generated by RBM, WGAN, and GPC, evaluated separately on training and test splits from the 1KG and UKBB datasets. Values closest to 0.5 are bolded to highlight the best trade-off between utility and privacy.

Dataset	**Metric**	RBM	WGAN	GPC
1KG	AA_{TRUTH} (Train)	0.9561	0.8103	**0.7185**
	AA_{TRUTH} (Test)	0.9928	0.7764	**0.7680**
	AA_{SYN} (Train)	0.0024	0.7356	**0.4225**
	AA_{SYN} (Test)	0.0276	0.7847	**0.5304**
UKBB	AA_{TRUTH} (Train)	0.9954	0.9674	**0.9204**
	AA_{TRUTH} (Test)	0.9962	0.9688	**0.9198**
	AA_{SYN} (Train)	0.0064	0.7768	**0.5324**
	AA_{SYN} (Test)	0.0160	0.7582	**0.4630**

GPC supports exact likelihood computation, which serves two practical purposes: it enables objective convergence monitoring during training, and it allows imputation to be performed directly via conditional probability queries rather than through the indirect route of generating AGs as reference panels. Across 1KG and UKBB, GPC accurately captures population structure and LD patterns across all length scales simultaneously — an advantage over HMMs and Markov chains, which are accurate only at short ranges, and over GANs and RBMs, which tend to be accurate only at longer ranges. Imputation performance is consistently strong, with particularly large gains for low-frequency variants and underrepresented populations where ancestry-matched reference data is scarce or restricted.

Our results reveal that the optimal imputation strategy depends on the nature of the task. When imputing single SNPs independently, combining ancestry-matched AGs with European reference data consistently improves accuracy, suggesting that increased haplotype diversity benefits inference even when there is some distributional mismatch. However, for array-based imputation where many correlated variants must be imputed jointly, population-specific models outperform combined panels, indicating that the local LD structure of the target population becomes the dominant factor. Privacy evaluation using AATS shows that GPC achieves the best utility-privacy balance among all deep generative approaches evaluated, though the AA_{TRUTH} values above 0.5 indicate that the synthetic data remains somewhat distinguishable from real data. Providing formal privacy guarantees, for instance through differentially private EM algorithms, is an important direction for future work.

Several limitations point to directions for further development. GPC currently operates on haploid genotypes within single genomic regions; extending to genome-wide scale will likely require hierarchical or tiling approaches that combine local circuit models. The model also inherits biases present in training data, and performance on underrepresented populations will ultimately depend on the quality and size of available ancestry-matched samples. Additional benchmarks on downstream tasks such as fine-mapping, polygenic risk score construction, and ancestry inference would more fully characterize the utility of GPC-generated AGs. Despite these limitations, GPC offers a tractable, easy-to-train framework that addresses a genuine gap: a generative model for genetic variation that is simultaneously expressive, computationally efficient, and capable of exact probabilistic inference, with particular benefits for underrepresented populations where existing approaches fall short.

REFERENCES

Kareem Ahmed, Zhe Zeng, Mathias Niepert, and Guy Van den Broeck. Simple: A gradient estimator for k-subset sampling. In *ICLR*, 2023.

Adam Auton, Gonçalo R. Abecasis, David M. Altshuler, Richard M. Durbin, Gonçalo R. Abecasis, David R. Bentley, Aravinda Chakravarti, Andrew G. Clark, Peter Donnelly, Evan E. Eichler, and et al. A global reference for human genetic variation. *Nature*, 526(7571):68–74, Sep 2015. doi: 10.1038/nature15393.

Yael Baran, Bogdan Pasaniuc, Sriram Sankararaman, Dara G Torgerson, Christopher Gignoux, Celeste Eng, William Rodriguez-Cintron, Rocio Chapela, Jean G Ford, Pedro C Avila, et al. Fast and accurate inference of local ancestry in latino populations. *Bioinformatics*, 28(10):1359–1367, 2012.

CJ Battey, Gabrielle C Coffing, and Andrew D Kern. Visualizing population structure with variational autoencoders. *G3*, 11(1):1–11, 2021.

Franz Baumdicker, Gertjan Bisschop, Daniel Goldstein, Graham Gower, Aaron P Ragsdale, Georgia Tsambos, Sha Zhu, Bjarki Eldon, Castedo E Ellerman, Jared G Galloway, et al. Efficient ancestry and mutation simulation with msprime 1.0. *bioRxiv*, 2021.

Brian L. Browning, Ying Zhou, and Sharon R. Browning. A one-penny imputed genome from next-generation reference panels. *The American Journal of Human Genetics*, 103(3):338–348, Sep 2018. doi: 10.1016/j.ajhg.2018.07.015.

Sharon R Browning and Brian L Browning. Haplotype phasing: existing methods and new developments. *Nature Reviews Genetics*, 12(10):703–714, 2011.

Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T. Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, and et al. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726):203–209, Oct 2018. doi: 10.1038/s41586-018-0579-z.

Marta Byrska-Bishop, Uday S. Evani, Xuefang Zhao, Anna O. Basile, Haley J. Abel, Allison A. Regier, André Corvelo, Wayne E. Clarke, Rajeeva Musunuri, Kshithija Nagulapalli, Susan Fairley, Alexi Runnels, Lara Winterkorn, Ernesto Lowy, Evan E. Eichler, Jan O. Korbel, Charles Lee, Tobias Marschall, Scott E. Devine, William T. Harvey, Weichen Zhou, Ryan E. Mills, Tobias Rausch, Sushant Kumar, Can Alkan, Fereydoun Hormozdiari, Zechen Chong, Yu Chen, Xiaofei Yang, Jiadong Lin, Mark B. Gerstein, Ye Kai, Qihui Zhu, Feyza Yilmaz, Chunlin Xiao, Paul Flicek, Soren Germer, Harrison Brand, Ira M. Hall, Michael E. Talkowski, Giuseppe Narzisi, and Michael C. Zody. High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, 185(18):3426–3440.e19, 2022. ISSN 0092-8674. doi: https://doi.org/10.1016/j.cell.2022.08.004.

YooJung Choi, Antonio Vergari, and Guy Van den Broeck. Probabilistic circuits: A unifying framework for tractable probabilistic models. oct 2020.

YooJung Choi, Meihua Dang, and Guy Van den Broeck. Group fairness by probabilistic modeling with latent fair decisions. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, Feb 2021.

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

Alvaro Correia, Robert Peharz, and Cassio P de Campos. Joints in random forests. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Meihua Dang, Antonio Vergari, and Guy Van den Broeck. Strudel: Learning structured-decomposable probabilistic circuits. In *Proceedings of the 10th International Conference on Probabilistic Graphical Models (PGM)*, sep 2020.

Adnan Darwiche. A logical approach to factoring belief networks. In *Proceedings of KR*, pp. 409–420, 2002.

Olivier Delaneau, Jonathan Marchini, and Jean-François Zagury. A linear complexity phasing method for thousands of genomes. *Nature methods*, 9(2):179–181, 2012.

Margarita Geleta, Daniel Mas Montserrat, Xavier Giro-i Nieto, and Alexander G. Ioannidis. Deep variational autoencoders for population genetics. *bioRxiv*, 2023. doi: 10.1101/2023.09.27.558320.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Bryan Howie, Christian Fuchsberger, Matthew Stephens, Jonathan Marchini, and Gonçalo R Abecasis. Fast and accurate genotype imputation in genome-wide association studies through prephasing. *Nature genetics*, 44(8):955–959, 2012.

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

Richard R Hudson. Generating samples under a wright–fisher neutral model of genetic variation. *Bioinformatics*, 18(2):337–338, 2002.

Jerome Kelleher, Alison M. Etheridge, and Gilean McVean. Efficient Coalescent Simulation and Genealogical Analysis for Large Sample Sizes. *PLoS Computational Biology*, 12(5):1–22, 2016. ISSN 15537358. doi: 10.1371/journal.pcbi.1004842.

Pasha Khosravi, YooJung Choi, Yitao Liang, Antonio Vergari, and Guy Van den Broeck. On tractable computation of expected predictions. *Advances in Neural Information Processing Systems*, 32:11169–11180, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2014.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193.

Na Li and Matthew Stephens. Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics*, 165(4):2213–2233, 2003.

Wenzhe Li, Zhe Zeng, Antonio Vergari, and Guy Van den Broeck. Tractable computation of expected kernels. In *Proceedings of the 37th Conference on Uncertainty in Aritifical Intelligence (UAI)*, jul 2021.

Anji Liu and Guy Van den Broeck. Tractable regularization of probabilistic circuits. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, dec 2021.

Anji Liu, Kareem Ahmed, and Guy Van den Broeck. Scaling tractable probabilistic circuits: A systems perspective. In *Proceedings of the 41th International Conference on Machine Learning (ICML)*, jul 2024.

Jonathan Marchini and Bryan Howie. Genotype imputation for genome-wide association studies. *Nature Reviews Genetics*, 11(7):499–511, 2010.

Gilean A T McVean and Niall J Cardin. Approximating the coalescent with recombination. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 360(1459): 1387–93, 2005. ISSN 0962-8436. doi: 10.1098/rstb.2005.1673.

Mahsan Nourani, Chiradeep Roy, Tahrima Rahman, Eric D. Ragan, Nicholas Ruozzi, and Vibhav Gogate. Don't explain without verifying veracity: An evaluation of explainable AI with video activity recognition. *CoRR*, abs/2005.02335, 2020.

Hoifung Poon and Pedro Domingos. Sum-product networks: A new deep architecture. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 689–690. IEEE, 2011.

Tahrima Rahman, Prasanna Kothalkar, and Vibhav Gogate. 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*, pp. 630–645. Springer, 2014.

Simone Rubinacci, Olivier Delaneau, and Jonathan Marchini. Genotype imputation using the positional burrows wheeler transform. *PLOS Genetics*, 16(11):1–19, 11 2020. doi: 10.1371/journal.pgen.1009049.

Paul Scheet and Matthew Stephens. A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *The American Journal of Human Genetics*, 78(4):629–644, 2006.

Nikil Roashan Selvam, Guy Van den Broeck, and YooJung Choi. Certifying fairness of probabilistic circuits. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, feb 2023.

Antoine Szatkownik, Cyril Furtlehner, Guillaume Charpiat, Burak Yelmen, and Flora Jay. Towards creating longer genetic sequences with gans: Generation in principal component space. In David A. Knowles and Sara Mostafavi (eds.), *Proceedings of the 18th Machine Learning in Computational Biology meeting*, volume 240 of *Proceedings of Machine Learning Research*, pp. 110–122. PMLR, 30 Nov–01 Dec 2024.

Antonio Vergari, YooJung Choi, Robert Peharz, and Guy Van den Broeck. Probabilistic circuits: Representations, inference, learning and applications. *AAAI Tutorial*, 2020.

Antonio Vergari, YooJung Choi, Anji Liu, Stefano Teso, and Guy Van den Broeck. A compositional atlas of tractable circuit operations for probabilistic inference. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, dec 2021.

Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P. Bennett. Generation and evaluation of privacy preserving synthetic health data. *Neurocomputing*, 416:244–255, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2019.12.136.

Burak Yelmen, Aurélien Decelle, Linda Ongaro, Davide Marnetto, Corentin Tallec, Francesco Montinaro, Cyril Furtlehner, Luca Pagani, and Flora Jay. Creating artificial human genomes using generative neural networks. *PLOS Genetics*, 17(2):1–22, 02 2021. doi: 10.1371/journal.pgen.1009303.

Burak Yelmen, Aurélien Decelle, Leila Lea Boulos, Antoine Szatkownik, Cyril Furtlehner, Guillaume Charpiat, and Flora Jay. Deep convolutional and conditional neural networks for large-scale genomic data generation. *PLOS Computational Biology*, 19(10):1–21, 10 2023. doi: 10.1371/journal.pcbi.1011584.

Ketian Yu, Sayantan Das, Jonathon LeFaive, Alan Kwong, Jacob Pleiness, Lukas Forer, Sebastian Schönherr, Christian Fuchsberger, Albert Vernon Smith, and Gonçalo Rocha Abecasis. Meta-imputation: An efficient method to combine genotype data after imputation with multiple reference panels. *The American Journal of Human Genetics*, 109(6):1007–1015, Jun 2022. doi: 10.1016/j.ajhg.2022.04.002.

Honghua Zhang, Brendan Juba, and Guy Van den Broeck. Probabilistic generating circuits. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, jul 2021.

A Appendix

A.1 Datasets

A.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 (Yelmen et al., 2023; 2021), 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 restrictive baselines and to evaluate how well it captures population and LD structure. It is also used in our single-SNP imputation experiments.

A.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. Finally, we phase the 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.

A.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.

A.2 Model and Inference Methods

A.2.1 Hidden Chow-Liu Trees

Hidden Chow-Liu trees (HCLTs) (Liu & Van den Broeck, 2021) represent a distribution over a collection of random variables (RVs) $(\mathbf{Z} = (Z_1, \ldots, Z_N), \mathbf{X} = (X_1, \ldots, X_n))$. $\mathbf{Z}$ 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 ($Z_1 \rightarrow Z_2 \rightarrow \ldots Z_N$), 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_1 and X_5 are highly correlated without being correlated with X_2, X_3, X_4.

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 ($Z_n \in \{0, \dots, L-1\}$). Figure 1 demonstrates how to construct an HCLT using genetic data. Given a dataset $\mathcal{D}$ that contains 6 SNPs (Figure 1(a)), we first invoke the Chow-Liu algorithm to generate a tree over the latent variables associated with each SNP (Figure 1(b)). The tree encodes strong variable dependencies by placing highly correlated SNPs (e.g., X_1 and X_3) 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 (Figure 1(c)). The parameters of the HCLT are those associated with the discrete conditional probability distributions $P(X_n|Z_n)$ and $P(Z_n|Z_{Pa(n)})$, where $Pa(n)$ denotes the parent of node n in the tree.

A.2.2 Probabilistic Circuits

Probabilistic Circuits (PCs) (Vergari et al., 2020; Choi et al., 2020) 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 (Goodfellow et al., 2014; Kingma & Welling, 2013). For example, the tractability of PCs helps solve problems in explainable AI (Nourani et al., 2020; Ahmed et al., 2023; Khosravi et al., 2019), algorithmic fairness (Selvam et al., 2023; Choi et al., 2021), and missing data robustness (Correia et al., 2020; Khosravi et al., 2019; Li et al., 2021).

PCs are furthermore appealing for their expressive power and suitability for density estimation. Recent advances in structure learning (Dang et al., 2020) and parameter estimations (Choi et al., 2021; Liu & Van den Broeck, 2021) allow PCs to accurately capture useful correlations in the data.

Representation PCs are an umbrella term for a wide family of tractable probabilistic models (Zhang et al., 2021), including arithmetic circuits (Darwiche, 2002), sum-product networks (Poon & Domingos, 2011), and cutset networks (Rahman et al., 2014). A PC $(\mathcal{G}, \boldsymbol{\theta})$ represents a joint probability distribution $\Pr(\mathbf{X})$ over random variables $\mathbf{X}$ through a directed acyclic graph (DAG) $\mathcal{G}$ parametrized by $\boldsymbol{\theta}$. 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 $\mathsf{ch}(n)$. Each node $n \in \mathcal{G}$ encodes a probability distribution $\Pr_n$, which is defined recursively as follows:

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

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 $(\mathcal{G}, \boldsymbol{\theta})$ 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 its encoded distribution. One can compute likelihoods by evaluating the PCs feed-forward as in Equation 1. Many common reasoning tasks such as marginal probabilities and maximum a posterior 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 (Vergari et al., 2021) 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 $(\mathcal{G}, \boldsymbol{\theta})$ is smooth if for any sum node $n \in \mathcal{G}$, its children have identical scope: $\forall c_1, c_2 \in \mathsf{ch}(n) : \phi(c_1) = \phi(c_2)$.*

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

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 (Vergari et al., 2020):

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

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

A.2.3 Parameter Estimation

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.

Fitting PCs to Data 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 (Choi et al., 2020) for more details. To transform a HCLT into an equivalent PC, we iteratively encode every conditional probability $\Pr(\mathbf{X}_n|\mathbf{X}_{Pa(n)})$ by representing each possible value of $\mathbf{X}_n$ as a sum node. Thus, the probability of $\Pr(\mathbf{X}_n = \mathbf{x}_n|\mathbf{X}_{Pa(n)} = \mathbf{x}_{Pa(n)})$ can be represented by the weight of an edge connecting $\mathbf{x}_n$ and $\mathbf{x}_{Pa(n)}$. Take the HCLT in Figure 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.

Depending on the structural constraints possessed by a PC, different parameter learning techniques can be applied. If a PC is smooth, decomposable, and deterministic (i.e., for any sum node, its children have disjoint support), its MLE parameters can be efficiently learned in closed form (Kisa et al., 2014). 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 $\mathtt{supp}(n_r)$. The context of any other node is the intersection of its support and the union of its parents' contexts:

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

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

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

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 (Koller & Friedman, 2009; Choi et al., 2021).

Concretely, given a deterministic PC $(\mathcal{G}, \boldsymbol{\theta})$ 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 (Choi et al., 2021):

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

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 (Liu et al., 2024). 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.

A.2.4 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. Using PyJuice on a single NVIDIA RTX A5000 GPU with 24GB memory, EM updates complete in under 2/10/3 seconds per epoch on the 1KG, UKBB, and high-coverage 1KG datasets, respectively. Total training time is roughly 2/6/4 hours on each dataset. Once trained, generating a single AG sample takes under 2/5/3 seconds for each dataset, and imputing all missing SNPs within a single sample takes approximately 20 milliseconds. 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) (Yale et al., 2020) 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.

In order to estimate the parameters of the simpler PGMs, INDEP and MARKOV have closed-form MLE solutions; while 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 2000–5000 epochs depending on the dataset. For MARKOV and HMM, training one model on the entire range of SNPs was too computationally intensive given the system's recursion limits. Therefore, we train 4 models, each on a continuous 25% region of the SNPs, and aggregate the results of each model to form the final output.

A.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`.

A.4 EVALUATION METRICS

A.4.1 NEAREST NEIGHBOR ADVERSARIAL ACCURACY

To evaluate the utility and privacy of the AGs, we compute the AATS metric introduced by Yale et al. (2020):

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

This formula returns an accuracy-like score between 0 and 1. T represents the true (real) data and S represents the synthetic data. Importantly, both datasets should be standardized. $d_{TS}(i)$ 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 similarly but for the synthetic data. The $AATS$ is an average of the following two terms, which we denote as AA_{TRUTH} and AA_{SYN}:

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

Each component also returns a value between 0 and 1. If the values are high, this means that the datasets are far enough from each other to be considered extremely different, or private. However, their utility would be low. On the other hand, low values indicate similarity, meaning the utility of the AGs is high but they do not preserve privacy as effectively. Ideally, we would like to balance these outcomes with both AA_{TRUTH} and AA_{SYN} being close to 0.5.

In our experiments, we do not consider the overall $AATS$ because poor values for each component can cancel each other out; consider an AA_{TRUTH} of 0 and an AA_{SYN} of 1. The former value shows that for each true sample, the nearest neighbor is a synthetic sample, indicating extreme overfitting. The latter value shows that for each synthetic sample, its nearest neighbor is another synthetic sample. While this demonstrates that the synthetic data has a meaningful structure and is internally coherent, when paired with the low AA_{TRUTH}, we realize that the synthetic data mimics the real data too well.

A.4.2 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. For each generative method, PCA was fit jointly on the real test samples and the corresponding generated samples ("coupled PCA"), and Wasserstein distances were evaluated in selected two-dimensional PCA subspaces (PC1–PC2, PC3–PC4, and PC5–PC6). Distances were computed using the entropically regularized optimal transport (Sinkhorn) algorithm as implemented in the `ot` Python package, with a regularization parameter set to $\varepsilon = 2 \times 10^{-3}$. Lower values indicate closer agreement between the distributions of real and generated samples in PCA space.

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.

Acknowledgements

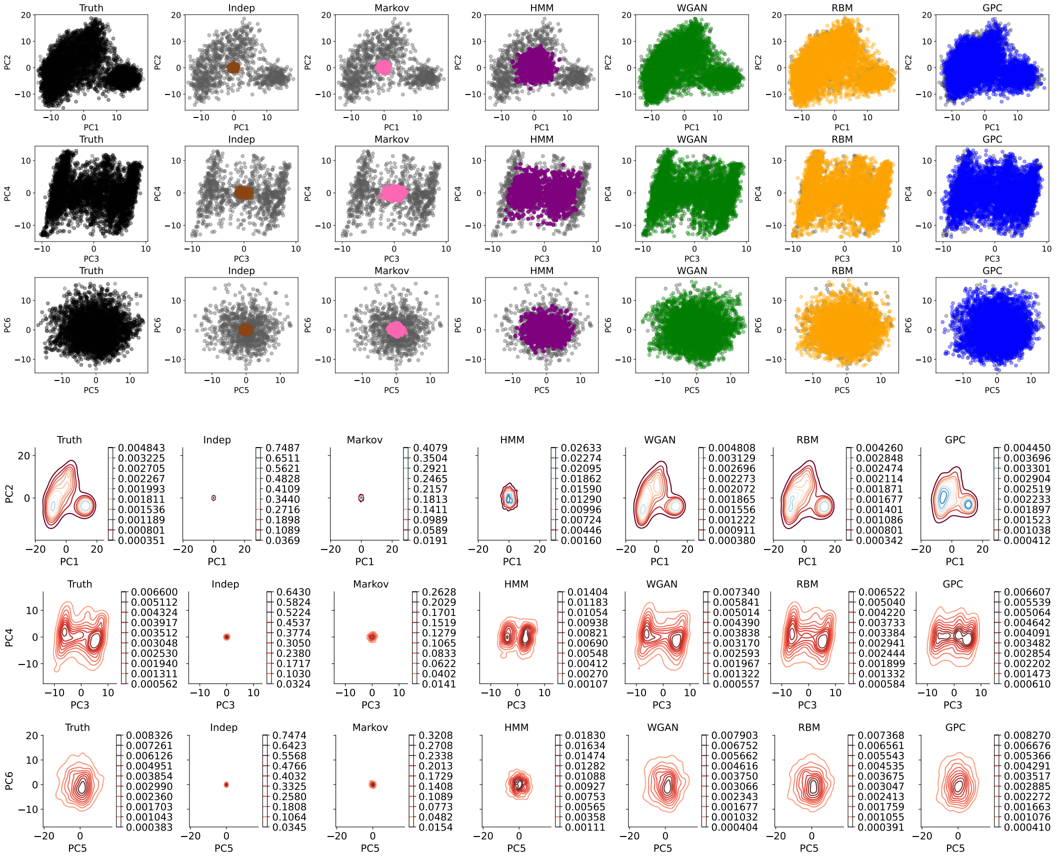
This research was conducted using the UK Biobank Resource under application 331277. We thank the participants of UK Biobank for making this work possible.

Code availability

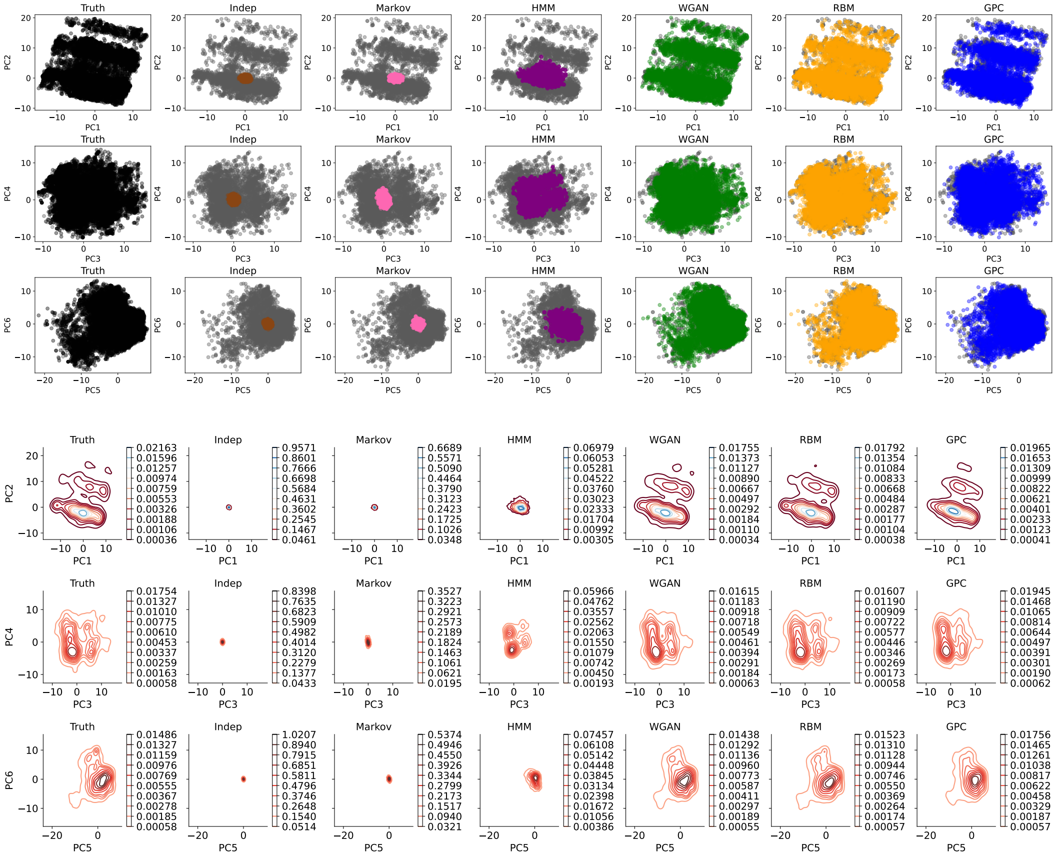
Code and experiments are available at `https://github.com/sriramlab/GPC`.

Data Availability

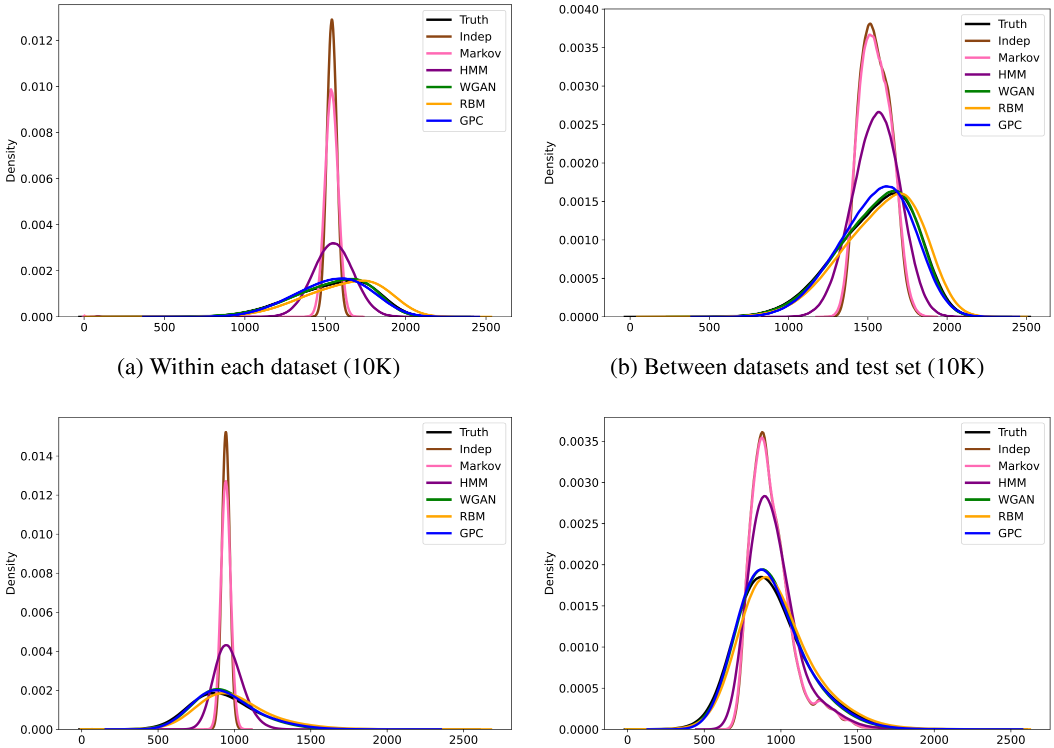
The 1000 Genomes datasets can be freely downloaded at `https://www.internationalgenome.org/data`. The UK Biobank dataset is available upon application at `https://www.ukbiobank.ac.uk/`.



Supplementary Figure S1: **Principal components analysis for models trained on the 1KG dataset**. The top half of the figure 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.



Supplementary Figure S2: **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.



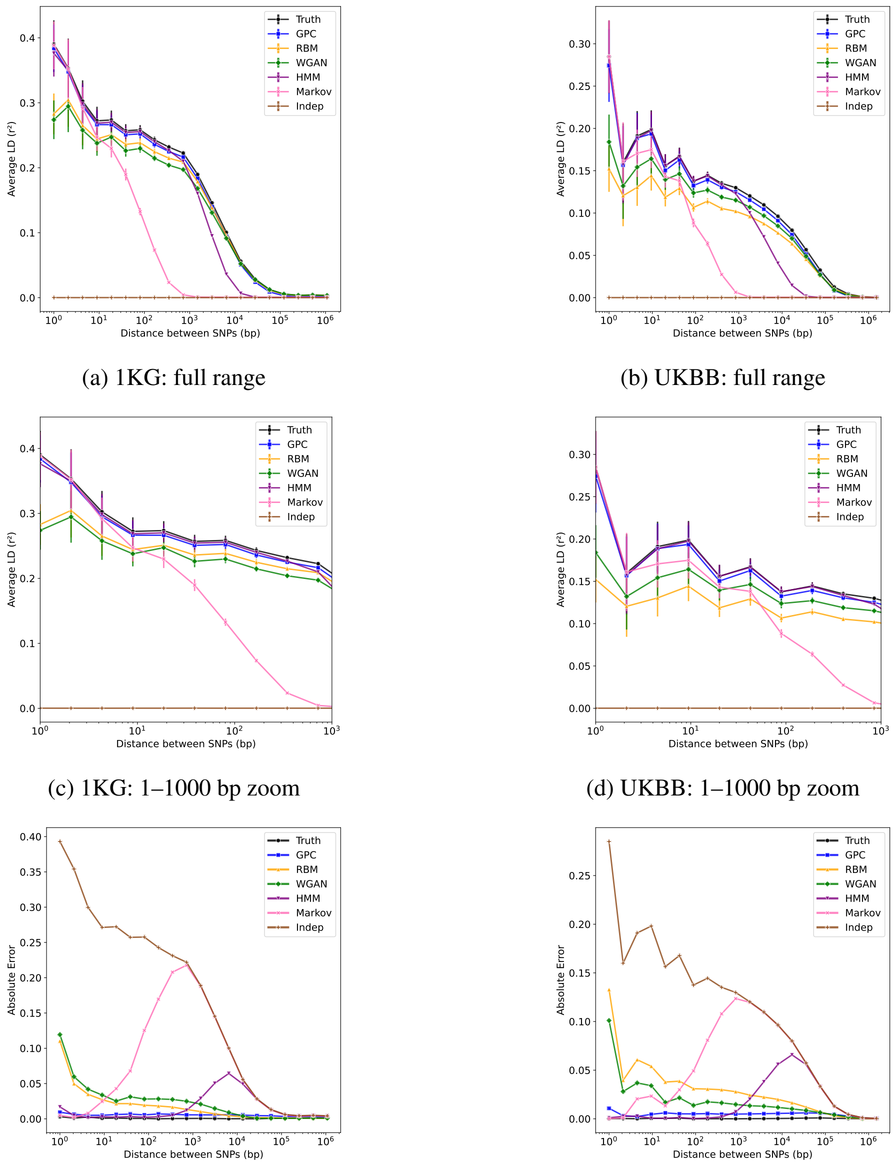
(a) Within each dataset (10K)

(b) Between datasets and test set (10K)

(c) Within each dataset (UKBB)

(d) Between datasets and test set (UKBB)

Supplementary Figure S3: **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 Table S1 for detailed metrics).



(a) 1KG: full range

(b) UKBB: full range

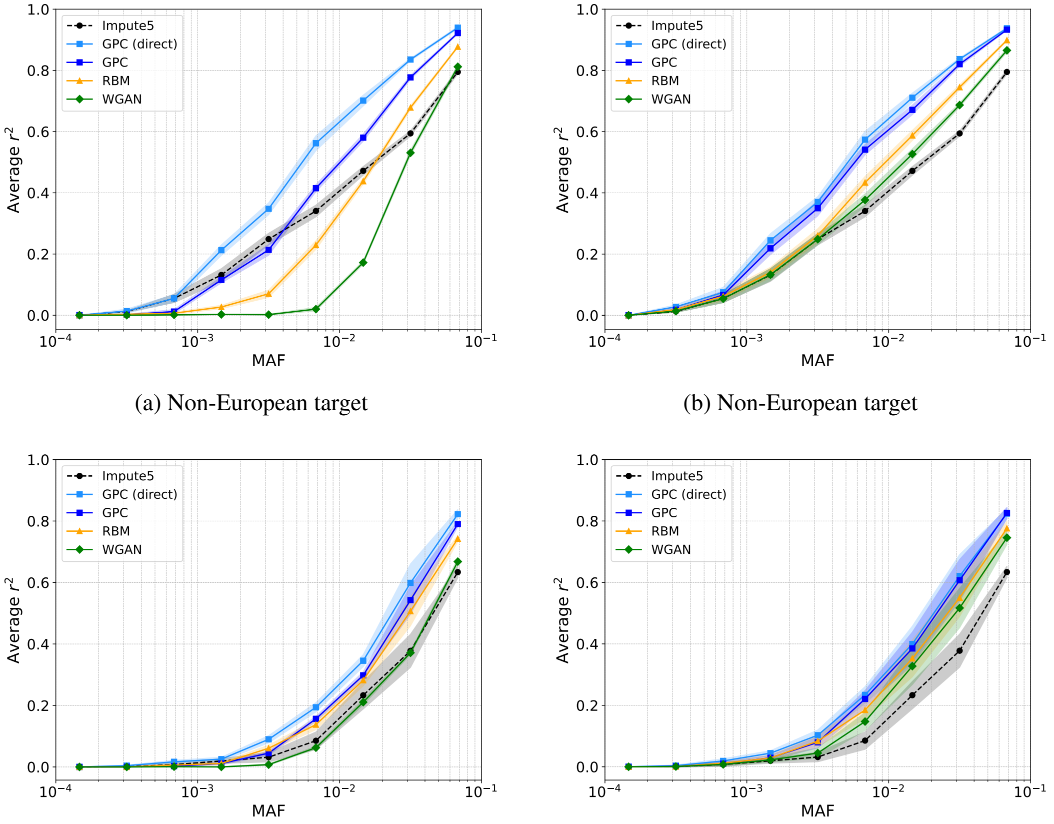
(c) 1KG: 1–1000 bp zoom

(d) UKBB: 1–1000 bp zoom

(e) 1KG: absolute error per bin

(f) UKBB: absolute error per bin

Supplementary Figure S4: **Linkage disequilibrium decay of AGs.** LD was estimated as pairwise squared correlations r^2 between SNP genotypes of the test data and AGs. Distances were binned on a logarithmic scale, and within each bin the mean r^2 was computed (error bars denote the standard error of the mean). Truth represents the LD distribution of the training data. The top row shows LD decay across the full length scale of the genomic locus; the middle row provides a zoomed-in view of LD at short length scales (1–1000 bp); the bottom row shows the absolute error in mean r^2 per distance bin relative to the test data. GPC matches the true LD distribution across all SNP distances (see Table S2 for detailed metrics).



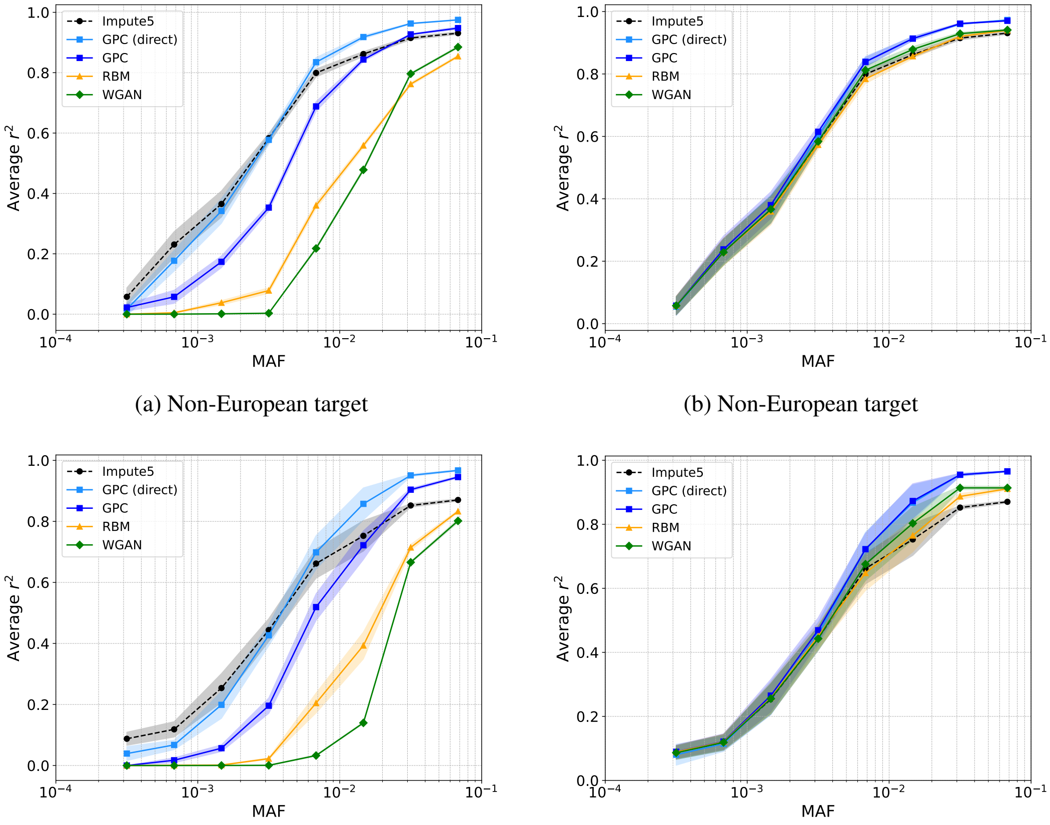
(a) Non-European target

(b) Non-European target

(c) African target

(d) African target

Supplementary Figure S5: **Population-specific imputation (1KG).** The black line shows results using Impute5 with real European data as the reference panel. 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 Table S4 for detailed metrics).



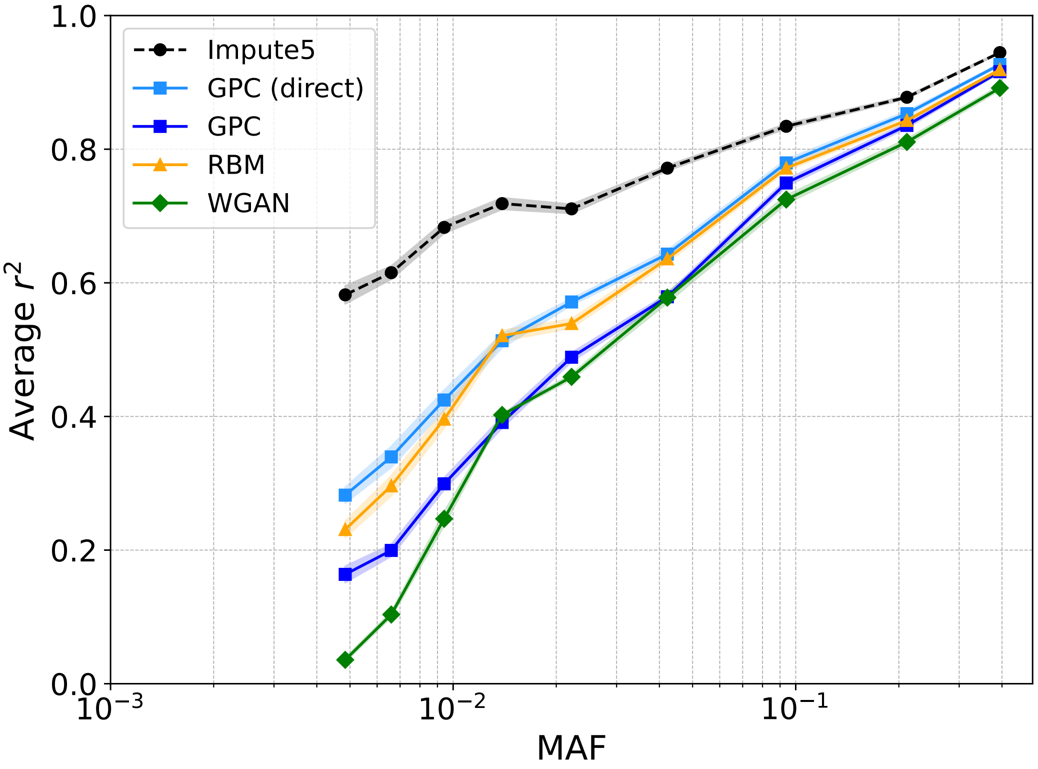
(a) Non-European target

(b) Non-European target

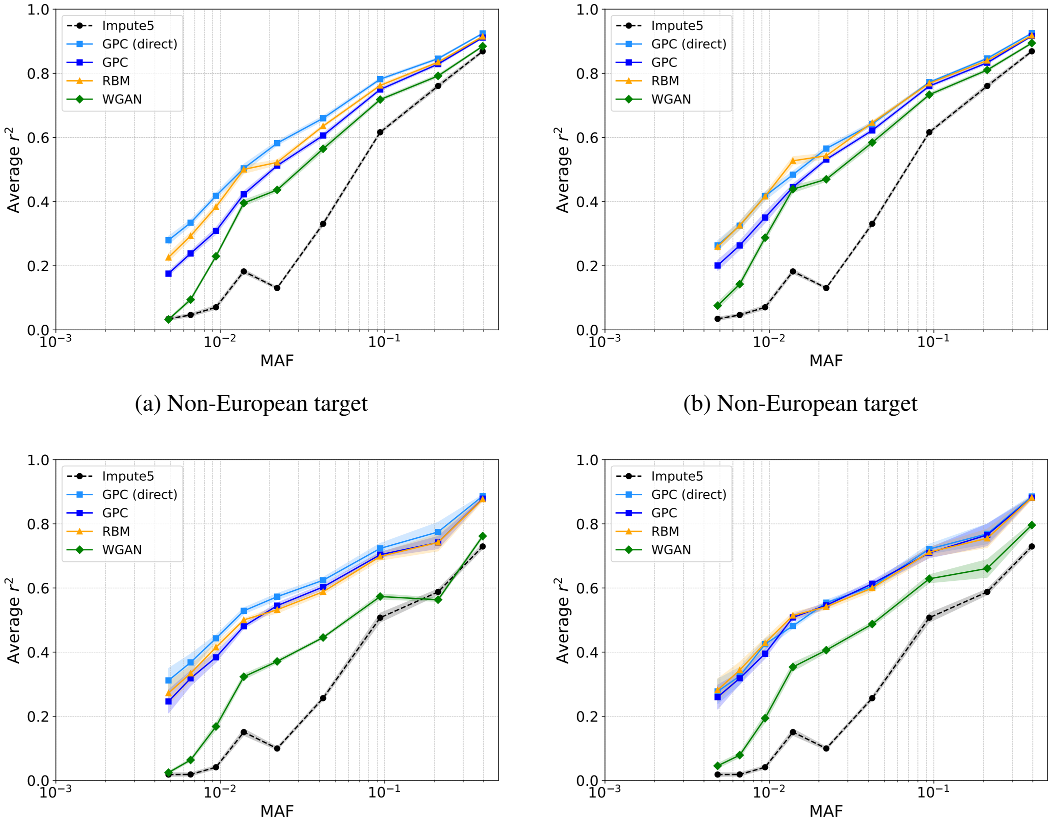
(c) African target

(d) African target

Supplementary Figure S6: **Population-specific imputation (UKBB).** The black line shows results using Impute5 with real European data as the reference panel. 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 Table S5 for detailed metrics).



Supplementary Figure S7: **Array-based imputation.** The task is to simultaneously impute 86% of SNPs (12, 551 of 14, 670) that are absent 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 Table S6 for detailed metrics).



(a) Non-European target

(b) Non-European target

(c) African target

(d) African target

Supplementary Figure S8: **Array-based imputation (population-specific).** The task is to simultaneously impute 86% of SNPs $(12,551$ of $14,670)$ that are absent from the HumanOmni5Exome array using high-coverage 1KG data. The black line shows results using Impute5 with real European data as the reference panel. 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 Table S7 for detailed metrics).

Supplementary Table S1: **Distance metrics between real and generated data. Within, Between** (Figure S3): 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.

Dataset	**Metric**	TRUTH	INDEP	MARKOV	HMM	WGAN	RBM	GPC
1KG	Within	10.66	180.23	172.70	105.88	**14.73**	61.99	23.59
	Between	5.34	128.24	125.61	86.25	**6.35**	31.50	18.73
	PCA1-2	0.0026	0.2355	0.2085	0.0348	0.0028	**0.0027**	0.0039
	PCA3-4	0.0022	0.1476	0.1148	0.0123	**0.0023**	0.0024	0.0028
	PCA5-6	0.0023	0.1011	0.0822	0.0042	0.0024	**0.0023**	0.0029
UKBB	Within	3.57	168.88	164.55	117.91	25.16	49.52	**20.86**
	Between	1.81	80.20	79.36	64.25	13.08	25.81	**11.88**
	PCA1-2	0.0019	0.0909	0.0846	0.0072	0.0022	**0.0021**	0.0024
	PCA3-4	0.0019	0.0656	0.0634	0.0244	0.0022	**0.0022**	0.0026
	PCA5-6	0.0019	0.0634	0.0542	0.0117	0.0021	**0.0020**	0.0023

Supplementary Table S2: **LD decay error metrics for Figure 2.** 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.

Dataset	**Metric**	TRUTH	INDEP	MARKOV	HMM	WGAN	RBM	GPC
1KG	MAE	0.0008	0.1676	0.0708	0.0153	0.0237	0.0182	**0.0053**
	RMSE	0.0011	0.2101	0.1031	0.0240	0.0361	0.0307	**0.0056**
UKBB	MAE	0.0003	0.1111	0.0481	0.0153	0.0186	0.0295	**0.0047**
	RMSE	0.0004	0.1336	0.0649	0.0262	0.0283	0.0414	**0.0052**

Supplementary Table S3: **Summarized metrics for Figure 3.** Mean imputation performance across 10 bootstrapped replicates for all, low-frequency (MAF $< 1\%$), and rare (MAF $< 0.1\%$) SNPs. Best generative model is bolded.

1KG

Method	Mean r^2 [95% CI]
All SNPs	
Impute5	0.649 [0.643, 0.655]
RBM (full)	0.510 [0.506, 0.514]
WGAN (full)	0.428 [0.426, 0.430]
RBM	0.494 [0.491, 0.496]
WGAN	0.445 [0.443, 0.447]
GPC	0.540 [0.536, 0.544]
GPC (direct)	**0.593 [0.589, 0.598]**
Low-freq (MAF $< 1\%$)	
Impute5	0.298 [0.286, 0.310]
RBM (full)	0.095 [0.088, 0.103]
WGAN (full)	0.004 [0.004, 0.005]
RBM	0.066 [0.060, 0.072]
WGAN	0.009 [0.009, 0.010]
GPC	0.132 [0.123, 0.140]
GPC (direct)	**0.218 [0.208, 0.228]**
Rare (MAF $< 0.1\%$)	
Impute5	0.089 [0.078, 0.099]
RBM (full)	0.012 [0.007, 0.017]
WGAN (full)	0.000 [0.000, 0.000]
RBM	0.007 [0.005, 0.009]
WGAN	0.001 [0.000, 0.001]
GPC	0.012 [0.008, 0.017]
GPC (direct)	**0.053 [0.044, 0.061]**

UKBB

Method	Mean r^2 [95% CI]
All SNPs	
Impute5	0.957 [0.954, 0.960]
RBM	0.669 [0.667, 0.670]
WGAN	0.523 [0.522, 0.524]
GPC	0.750 [0.748, 0.751]
GPC (direct)	**0.906 [0.903, 0.909]**
Low-freq (MAF $< 1\%$)	
Impute5	0.914 [0.907, 0.922]
RBM	0.381 [0.377, 0.384]
WGAN	0.087 [0.086, 0.088]
GPC	0.525 [0.522, 0.528]
GPC (direct)	**0.828 [0.822, 0.835]**
Rare (MAF $< 0.1\%$)	
Impute5	0.818 [0.796, 0.839]
RBM	0.118 [0.112, 0.123]
WGAN	0.000 [0.000, 0.000]
GPC	0.263 [0.252, 0.274]
GPC (direct)	**0.709 [0.687, 0.730]**

Supplementary Table S4: **Summarized metrics for Figure S5 (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 †.

Non-European

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5 (EUR)	0.485 [0.482, 0.488]
RBM	0.476 [0.474, 0.478]
WGAN	0.422 [0.421, 0.424]
GPC	0.522 [0.520, 0.524]
GPC (direct)†	0.563 [0.561, 0.564]†
RBM (combined)	0.528 [0.524, 0.532]
WGAN (combined)	0.507 [0.504, 0.510]
GPC (combined)	0.560 [0.555, 0.562]
GPC (direct combined)	**0.570 [0.567, 0.573]**
Low-freq (MAF $< 1\%$)	
Impute5 (EUR)	0.106 [0.101, 0.112]
RBM	0.042 [0.039, 0.046]
WGAN	0.003 [0.002, 0.004]
GPC	0.098 [0.095, 0.101]
GPC (direct)†	0.158 [0.156, 0.161]†
RBM (combined)	0.123 [0.116, 0.129]
WGAN (combined)	0.111 [0.105, 0.117]
GPC (combined)	0.160 [0.155, 0.165]
GPC (direct combined)	**0.174 [0.169, 0.178]**
Rare (MAF $< 0.1\%$)	
Impute5 (EUR)	0.021 [0.016, 0.025]
RBM	0.003 [0.001, 0.004]
WGAN	0.000 [0.000, 0.001]
GPC	0.004 [0.002, 0.006]
GPC (direct)†	0.021 [0.016, 0.026]†
RBM (combined)	0.024 [0.019, 0.029]
WGAN (combined)	0.021 [0.016, 0.025]
GPC (combined)	0.026 [0.021, 0.030]
GPC (direct combined)	**0.031 [0.025, 0.036]**

African

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5 (EUR)	0.368 [0.357, 0.379]
RBM	0.429 [0.424, 0.434]
WGAN	0.381 [0.379, 0.382]
GPC	0.442 [0.438, 0.446]
GPC (direct)†	0.462 [0.456, 0.468]†
RBM (combined)	0.446 [0.435, 0.456]
WGAN (combined)	0.423 [0.412, 0.435]
GPC (combined)	0.464 [0.454, 0.474]
GPC (direct combined)	**0.471 [0.461, 0.480]**
Low-freq (MAF $< 1\%$)	
Impute5 (EUR)	0.019 [0.013, 0.025]
RBM	0.028 [0.025, 0.030]
WGAN	0.009 [0.008, 0.009]
GPC	0.027 [0.025, 0.028]
GPC (direct)†	0.043 [0.040, 0.046]†
RBM (combined)	0.040 [0.033, 0.047]
WGAN (combined)	0.028 [0.022, 0.035]
GPC (combined)	0.043 [0.037, 0.050]
GPC (direct combined)	**0.053 [0.046, 0.059]**
Rare (MAF $< 0.1\%$)	
Impute5 (EUR)	0.003 [0.001, 0.005]
RBM	0.002 [0.001, 0.003]
WGAN	0.000 [0.000, 0.001]
GPC	0.001 [0.001, 0.001]
GPC (direct)†	0.006 [0.005, 0.008]†
RBM (combined)	0.004 [0.002, 0.006]
WGAN (combined)	0.003 [0.001, 0.005]
GPC (combined)	0.004 [0.002, 0.005]
GPC (direct combined)	**0.007 [0.005, 0.009]**

Supplementary Table S5: **Summarized metrics for Figure S6 (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 †.

Non-European

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5 (EUR)	0.733 [0.725, 0.742]
RBM	0.495 [0.493, 0.497]
WGAN	0.461 [0.460, 0.463]
GPC	0.656 [0.651, 0.661]
GPC (direct)†	0.749 [0.742, 0.757]†
RBM (combined)	0.732 [0.723, 0.740]
WGAN (combined)	0.739 [0.731, 0.748]
GPC (combined)	**0.763 [0.755, 0.771]**
GPC (direct combined)	0.759 [0.752, 0.767]
Low-freq (MAF $<1\%$)	
Impute5 (EUR)	0.506 [0.491, 0.522]
RBM	0.128 [0.123, 0.132]
WGAN	0.059 [0.058, 0.061]
GPC	0.333 [0.323, 0.343]
GPC (direct)†	0.498 [0.483, 0.513]†
RBM (combined)	0.498 [0.482, 0.514]
WGAN (combined)	0.510 [0.494, 0.526]
GPC (combined)	**0.530 [0.515, 0.546]**
GPC (direct combined)	0.522 [0.507, 0.537]
Rare (MAF $<0.1\%$)	
Impute5 (EUR)	0.211 [0.174, 0.248]
RBM	0.004 [0.003, 0.006]
WGAN	0.000 [0.000, 0.000]
GPC	0.054 [0.034, 0.073]
GPC (direct)†	0.159 [0.127, 0.190]†
RBM (combined)	0.210 [0.171, 0.248]
WGAN (combined)	0.209 [0.173, 0.245]
GPC (combined)	**0.217 [0.180, 0.255]**
GPC (direct combined)	0.210 [0.175, 0.246]

African

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5 (EUR)	0.645 [0.627, 0.663]
RBM	0.437 [0.428, 0.446]
WGAN	0.375 [0.374, 0.376]
GPC	0.582 [0.571, 0.593]
GPC (direct)†	0.679 [0.663, 0.694]†
RBM (combined)	0.659 [0.641, 0.678]
WGAN (combined)	0.668 [0.650, 0.686]
GPC (combined)	**0.702 [0.684, 0.720]**
GPC (direct combined)	0.698 [0.679, 0.716]
Low-freq (MAF $<1\%$)	
Impute5 (EUR)	0.384 [0.354, 0.414]
RBM	0.061 [0.052, 0.070]
WGAN	0.009 [0.008, 0.009]
GPC	0.207 [0.191, 0.224]
GPC (direct)†	0.364 [0.337, 0.390]†
RBM (combined)	0.383 [0.352, 0.415]
WGAN (combined)	0.388 [0.358, 0.418]
GPC (combined)	**0.410 [0.380, 0.441]**
GPC (direct combined)	0.404 [0.373, 0.434]
Rare (MAF $<0.1\%$)	
Impute5 (EUR)	0.115 [0.091, 0.139]
RBM	0.001 [0.000, 0.001]
WGAN	0.000 [0.000, 0.000]
GPC	0.015 [0.008, 0.023]
GPC (direct)†	0.064 [0.051, 0.078]†
RBM (combined)	0.117 [0.094, 0.139]
WGAN (combined)	0.115 [0.093, 0.138]
GPC (combined)	**0.118 [0.095, 0.141]**
GPC (direct combined)	0.111 [0.087, 0.135]

Supplementary Table S6: **Summarized metrics for Figure S7 (high-coverage 1KG).** Mean imputation performance across 10 bootstrapped replicates for all and low-frequency (MAF $< 1\%$) SNPs. Best generative model is bolded.

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5	0.749 [0.745, 0.753]
RBM	0.574 [0.570, 0.578]
WGAN	0.474 [0.472, 0.476]
GPC	0.515 [0.511, 0.518]
GPC (direct)	**0.594 [0.589, 0.598]**
Low-freq (MAF $< 1\%$)	
Impute5	0.626 [0.618, 0.633]
RBM	0.301 [0.290, 0.312]
WGAN	0.120 [0.116, 0.123]
GPC	0.215 [0.209, 0.222]
GPC (direct)	**0.344 [0.335, 0.352]**

Supplementary Table S7: **Summarized metrics for Figure S8 (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.

Non-European

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5 (EUR)	0.338 [0.335, 0.342]
RBM	0.565 [0.562, 0.568]
WGAN	0.462 [0.459, 0.466]
GPC	0.529 [0.527, 0.531]
GPC (direct)†	0.593 [0.590, 0.596]†
RBM (combined)	0.584 [0.580, 0.588]
WGAN (combined)	0.494 [0.489, 0.499]
GPC (combined)	0.548 [0.545, 0.552]
GPC (direct combined)	**0.584 [0.581, 0.586]**
Low-freq (MAF $<$ 1%)	
Impute5 (EUR)	0.046 [0.042, 0.049]
RBM	0.296 [0.287, 0.304]
WGAN	0.114 [0.112, 0.116]
GPC	0.237 [0.234, 0.240]
GPC (direct)†	0.339 [0.335, 0.344]†
RBM (combined)	0.329 [0.321, 0.337]
WGAN (combined)	0.162 [0.155, 0.169]
GPC (combined)	0.266 [0.256, 0.276]
GPC (direct combined)	**0.331 [0.323, 0.338]**

African

Method	**Mean r^2 [95% CI]**
All SNPs	
Impute5 (EUR)	0.268 [0.264, 0.273]
RBM	0.552 [0.543, 0.562]
WGAN	0.367 [0.363, 0.371]
GPC	0.546 [0.539, 0.552]
GPC (direct)†	0.583 [0.574, 0.591]†
RBM (combined)	**0.563 [0.553, 0.574]**
WGAN (combined)	0.407 [0.398, 0.416]
GPC (combined)	0.556 [0.548, 0.564]
GPC (direct combined)	0.562 [0.552, 0.571]
Low-freq (MAF $<$ 1%)	
Impute5 (EUR)	0.022 [0.017, 0.027]
RBM	0.339 [0.320, 0.359]
WGAN	0.082 [0.078, 0.086]
GPC	0.317 [0.299, 0.334]
GPC (direct)†	0.376 [0.355, 0.396]†
RBM (combined)	**0.350 [0.330, 0.371]**
WGAN (combined)	0.101 [0.093, 0.110]
GPC (combined)	0.325 [0.307, 0.344]
GPC (direct combined)	0.342 [0.320, 0.363]