

Deep Generative Models for Discrete Genotype Simulation

Sihan Xie^{1*}, Thierry Tribout¹, Didier Boichard¹, Blaise Hanczar^{2†}, Julien Chiquet^{3†}, Eric Barrey^{1†}

¹Université Paris-Saclay, INRAE, AgroParisTech, GABI, 78350, Jouy-en-Josas, France.

²Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France.

³Université Paris-Saclay, AgroParisTech, INRAE, UMR MIA Paris-Saclay, 91120, Palaiseau, France.

*Corresponding author(s). E-mail(s): sihan.xie@inrae.fr;

†These authors contributed equally to this work.

Abstract

Deep generative models open new avenues for simulating realistic genomic data while preserving privacy and addressing data accessibility constraints. While previous studies have primarily focused on generating gene expression or haplotype data, this study explores generating genotype data in both unconditioned and phenotype-conditioned settings, which is inherently more challenging due to the discrete nature of genotype data. In this work, we developed and evaluated commonly used generative models, including Variational Autoencoders (VAEs), Diffusion Models, and Generative Adversarial Networks (GANs), and proposed adaptation tailored to discrete genotype data. We conducted extensive experiments on large-scale datasets, including all chromosomes from cow and multiple chromosomes from human. Model performance was assessed using a well-established set of metrics drawn from both deep learning and quantitative genetics literature. Our results show that these models can effectively capture genetic patterns and preserve genotype–phenotype association. Our findings provide a comprehensive comparison of these models and offer practical guidelines for future research in genotype simulation. We have made our code publicly available at <https://github.com/SihanXXX/DiscreteGenoGen>.

Keywords: Deep Generative Models, Genotype Simulation, Quantitative Genetics

1 Introduction

The development of dense genotyping platforms and high-throughput sequencing technologies has significantly advanced genetic analysis [1, 2]. Today, genomic studies rely on large biobanks that contain vast amounts of genomic data. However, working with such datasets presents several challenges, including high sequencing costs, substantial storage requirements, privacy concerns, and access restrictions that limit data sharing. To address these issues, simulation tools and synthetic data are commonly used. Traditional statistical simulation methods are based on evolutionary models like Wright-Fisher model [3] and coalescent theory [4–6], where users need to specify evolutionary parameters or provide ancestral population. While these simulation tools [7–12] are powerful, they often simplify various aspects of population genetics, which may not fully capture the complexities of real-world datasets.

Recently, data-driven simulation methods based on deep generative models have gained attention in genomics. These approaches eliminate the need to explicitly specify genetic parameters by learning directly from data, enabling the reproduction of fine-scale genomic characteristics presented in the given population. By shifting from explicit genomic sequences to generative models, the genome-wide data remains private, while the trained models can be shared publicly without directly exposing individual-level genetic information.

Previous studies have applied generative models to various genomic modalities: [13–15] focused on gene expression data, [16, 17] focused on DNA sequence, and there is a substantial body of literature on haplotype data [18–25]. In this work, we propose a new study on genotype data, which represents genetic variation at specific positions in the genome known as Single Nucleotide Polymorphisms (SNPs). Unlike contiguous DNA sequences, which may include both coding and non-coding regions that do not directly reflect individual genetic variability, genotype data focuses on selected variant sites, making them particularly valuable for studying population-level traits and disease association. Unlike binary-valued haplotype, genotype for diploid organisms includes three possible values (0, 1, 2), representing the number of alternative alleles inherited from both parents, which introduces specific modeling challenges. Importantly, directly simulating genotype data offers several advantages over haplotype-based approaches: Haplotype-based simulation methods lack rich conditioning capabilities, whereas our method allows conditioning on phenotypic traits, enabling more flexible and application-oriented use cases. Furthermore, traditional genotype–phenotype simulation workflows via haplotypes involve a multi-step pipeline and our approach consolidates this into a single generative step, reducing complexity and avoiding additional modeling assumptions that may introduce bias. Finally, haplotype simulators are limited to genomic regions with strong linkage disequilibrium (LD) on a single chromosome, whereas our study supports genome-wide simulation, demonstrated in our experiments on cow dataset where we modeled genotypes across all 29 autosomes simultaneously, significantly extending the scale of genomic data simulation.

This paper investigates the use of deep generative models for simulating genotype data, potentially conditioned on phenotype. Specifically, we adapt models such as Variational Autoencoders (VAEs) [26], Generative Adversarial Networks (GANs) [27],

and Diffusion Models [28] to accommodate the discrete nature of genotype. Properly evaluating synthetic genotype is a critical aspect of our study, as the evaluation metrics commonly used in the Generative AI community, such as precision and recall [29], have never been applied in previous haplotype generation studies. We propose a comprehensive evaluation framework that integrates both deep learning and quantitative genetics approaches, providing a rigorous comparison of the reviewed models. Section 2 presents the models adapted for genotype data and Section 3 presents our evaluation framework. Section 4 describes the experimental setup and Section 5 presents the main results. Finally, Section 6 discusses how the proposed models can be practically implemented, along with potential challenges and future research directions.

2 Generative Models for Genotype Data

Building on recent advances in haplotype generation [19, 22–25], we adopt generative models such as Variational Autoencoders (VAEs) [26], Generative Adversarial Networks (GANs) [30], and diffusion models [28]. These models are well-suited for capturing global dependencies across all SNPs, as opposed to relying on sequential autoregressive decomposition, which may not align with the underlying biological structure. Since genotype is a discrete sequence represented as $\mathbf{x} \in \{0, 1, 2\}^n$, we propose adaptation to better handle this structure.

2.1 Variational Autoencoders

Variational Autoencoders (VAEs) [26] learn to approximate the underlying data distribution by introducing a latent variable. A VAE consists of two neural networks, parameterized by ϕ and θ : an encoder that maps the input data x to a latent representation z via the approximate posterior $q_\phi(z | x)$, and a decoder that reconstructs x from z via the likelihood $p_\theta(x | z)$. The model is trained by maximizing the Evidence Lower BOund (ELBO) on the marginal likelihood $\log p(x)$, using the reparameterization trick to enable efficient gradient-based optimization. The ELBO is given by

$$\mathcal{L}(\theta, \phi; x) = \underbrace{\mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)]}_{\text{decoder for reconstruction}} - \underbrace{D_{\text{KL}}(q_\phi(z | x) \| p(z))}_{\text{encoder for prior matching}}, \quad (1)$$

where $p(z)$ is the prior on the latent variable and D_{KL} denotes the Kullback-Leibler divergence. Optimizing this objective encourages the model to learn a meaningful, structured latent space that can be sampled to generate new, realistic data. For generation purpose, once the VAE is trained, the encoder can be discarded. New samples are generated by drawing random latent vector z from the prior and passing it through the decoder.

2.2 Diffusion Models

Diffusion models (DMs), and in particular Denoising Diffusion Probabilistic Models (DDPMs) [28], can be viewed as a Markovian hierarchical VAE where each latent x_t has the same dimension as the data x_0 and the encoder is not learned but is a fixed

Gaussian noising process. During the encoding phase, also called the forward diffusion process, we gradually add Gaussian noise to input x_0 until it becomes pure noise x_T over T steps via a Markov chain:

$$q(x_t | x_{t-1}) = \mathcal{N}(\sqrt{\alpha_t} x_{t-1}, \beta_t \mathbf{I}), \quad \alpha_t = 1 - \beta_t \quad \text{for } t = 1, \dots, T. \quad (2)$$

Because of the Markov property, the Gaussian transition, and the independence of the noise at every step, one can collapse all t steps into a single closed-form marginal:

$$q(x_t | x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) \mathbf{I}\right), \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s. \quad (3)$$

Intuitively, the hyperparameter β_t controls the amount of noise injected at step t , and $\alpha_t = 1 - \beta_t$ is the fraction of signal retained. Our goal is to undo the added noise by learning $p_\theta(x_{t-1} | x_t)$, so that starting from $x_T \sim \mathcal{N}(0, I)$ we can step-by-step recover x_0 . The true reverse $q(x_{t-1} | x_t)$ is intractable, but during training we know x_0 . Hence we can write down the exact one-step posterior as a Gaussian distribution with a closed-form mean $\mu_t(x_t, x_0)$ and variance σ_t^2 :

$$q(x_{t-1} | x_t, x_0) = \mathcal{N}\left(\underbrace{\frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\beta_t \sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_t} x_0}_{\mu_t(x_t, x_0)}, \underbrace{\beta_t \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \mathbf{I}}_{\sigma_t^2}\right). \quad (4)$$

For the reverse process, we learn to approximate the posterior in Equation 4. For the variance part, many implementations simply set $\sigma_t^2 = \beta_t$, which has a negligible loss on quality. For the mean part, since $\mu_t(x_t, x_0)$ requires the true x_0 which is unavailable at inference, the direct training objective is to predict x_0 given x_t and t . In practice, however, it is more common and empirically more stable to train a network $\epsilon_\theta(x_t, t)$ to predict the injected noise at each timestep optimized via mean-squared error loss. Then using the predicted noise $\epsilon_\theta(x_t, t)$, we first infer an estimator of x_0 , given by $\hat{x}_0 = (x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)) / \sqrt{\bar{\alpha}_t}$. Substituting \hat{x}_0 for x_0 in Equation 4 gives the familiar reverse-step update:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z, \quad z \sim \mathcal{N}(0, I). \quad (5)$$

Despite their success, DMs are not compatible with discrete data. Two main strategies have been proposed to address this limitation: (1) defining a diffusion-like process that operates in discrete space, or (2) projecting the discrete input into a continuous latent space. Additionally, DMs can be computationally demanding during inference. To address both issues, we adopt the second strategy. Various methods can be used to construct a suitable latent space. For example, [16] employed a VAE to embed DNA sequences into a continuous representation. We follow the PCA-based approach originally developed for haplotype [24, 25]. Specifically, we project genotype into a lower-dimensional PCA space [31] and train the DMs in this continuous latent space.

This single transformation yields three major benefits in one shot: it greatly reduces dimensionality and speeds up both training and inference; it transforms the discrete genotype into a continuous representation that matches the assumptions of DMs; And it allows precise reconstruction via a simple linear multiplication (Supplementary Section 1).

2.3 Generative Adversarial Networks

While VAEs and DMs learn the data distribution by explicitly maximizing likelihood, Generative Adversarial Networks (GANs) [30] adopt a fundamentally different strategy. They avoid explicit density estimation by framing generation as a two-player game: a generator G transforms a latent vector $z \sim p_z$ (typically Gaussian) into a synthetic sample $G(z)$, and a discriminator D attempts to distinguish real data from generated samples. Training proceeds by solving the minimax problem:

$$\min_G \max_D \underbrace{\mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p(z)} [\log(1 - D(G(z)))]}_{\text{binary cross-entropy loss}}. \quad (6)$$

Here, D is trained as a binary classifier to assign high probability to real sample x and low probability to generated sample $G(z)$, while G is trained to fool D by producing ever more realistic outputs.

From Equation 6, G is trained by backpropagating gradients from D through its outputs $G(z)$. This works when $G(z)$ is continuous but fails for discrete outputs, which break differentiability. In previous GAN-based haplotype generation studies [21–23], no specific treatment was proposed for this issue. G output continuous values between 0 and 1, which were passed directly to D during training. At inference, discrete values were recovered using a binarization threshold of 0.5. This binary setting can be interpreted in two ways. First, from a probabilistic perspective, G outputs the probability of class 1, and inference chooses the most likely class. Motivated by this view, we tested a probabilistic approach: using a Softmax final layer in G to predict class probabilities per SNP, and training D to distinguish these from one-hot encoded real genotype sequences. However, this method produced unsatisfactory results. The second interpretation views binarization as a quantization operation that maps continuous outputs to a discrete set. We therefore explored several threshold-based strategies for our ternary genotype data, but observed suboptimal results (Supplementary Section 2). To enable end-to-end differentiable training on discrete outputs, we instead integrate a Gumbel-Softmax [32, 33] layer into G . The Gumbel-Softmax distribution provides a continuous approximation to categorical sampling by applying a temperature-controlled softmax to perturbed logits. Concretely, for each SNP, if the final layer of G produces logits $\ell = (\ell_0, \ell_1, \ell_2)$, we then sample Gumbel noise $g_i \sim \text{Gumbel}(0, 1)$, and compute the relaxed one-hot vector

$$\tilde{p}_i = \frac{\exp((\ell_i + g_i)/\tau)}{\sum_{j=0}^2 \exp((\ell_j + g_j)/\tau)}, \quad (7)$$

where τ is a temperature parameter. As $\tau \rightarrow 0$, \tilde{p}_i becomes exactly one-hot vector. During training we anneal τ from a high initial value down toward 0 to balance

exploration and discretization. At inference we take $\arg \max_i \tilde{p}_i$ to recover a discrete value in $\{0, 1, 2\}$.

2.4 Wasserstein GANs with Gradient Penalty

GANs lack an explicit likelihood measure and can suffer from training instabilities such as mode collapse [34]. Subsequent refinements like Wasserstein GAN (WGAN) [35] was developed to address these issues. The original WGAN replaces the Jensen-Shannon divergence with the Earth-Mover (Wasserstein-1) distance by solving:

$$\min_G \max_{D \in \mathcal{D}} \mathbb{E}_{x \sim p_{\text{data}}} [D(x)] - \mathbb{E}_{z \sim p_z} [D(G(z))], \quad (8)$$

where \mathcal{D} is the set of 1-Lipschitz functions. While weight clipping was initially proposed to enforce the Lipschitz constraint, this approach is proved unstable in practice. The Wasserstein GAN with gradient penalty (WGAN-GP) [36] instead introduces a gradient penalty term that penalizes the deviation of the gradient norm from 1, leading to more stable training dynamics:

$$\min_G \max_D \underbrace{\mathbb{E}_{x \sim p_{\text{data}}} [D(x)] - \mathbb{E}_{z \sim p_z} [D(G(z))]}_{\text{Wasserstein distance between real and synthetic}} - \lambda \underbrace{\mathbb{E}_{\hat{x} \sim p_{\hat{x}}} (\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2}_{\text{Gradient penalty}}, \quad (9)$$

where $p_{\hat{x}}$ is the distribution of points interpolated between real and generated samples, and λ controls the penalty strength.

2.5 Conditional Generative Modeling

So far, we have focused on modeling the marginal distribution $p(x)$. In practice, it is more interesting to learn the conditional distribution $p(x | y)$, which provides control over the generated data through conditioning variable y . A straightforward approach is to append y to x as input during training [27]. At inference, we can sample from $p(x | y)$ by specifying a desired value of y to guide generation.

In previous work on haplotype generation [19, 20, 25], models have frequently used ancestry group as conditioning variable to reflect population structure. For genotype, a natural choice is phenotype, particularly quantitative traits, which are often associated with genetic variation. This type of conditioning enables the generation of synthetic population that mirrors specific trait distribution, making the approach well suited for phenotype driven genetics research.

3 Evaluation Metrics for Synthetic Genotype Data

Genotype lacks the intuitive visual or semantic cues of images and text. We therefore use a diverse set of metrics, offering a multi-angle assessment of synthetic genotype.

3.1 PCA and UMAP

The PCA [31] and UMAP [37] projections provide an initial visual assessment of how well the synthetic population resembles the real one. These dimensionality reduction techniques highlight global structure and potential clustering patterns, offering a qualitative sense of alignment between the two distributions. However, they do not quantitatively measure distributional similarity and should be interpreted as complementary to more rigorous evaluation metrics.

3.2 Genetic Parameters

3.2.1 Allele Frequency and Genotype Frequency

We compare allele and genotype frequencies [38] between real and synthetic cohorts as a basic sanity check. Let N be the number of individuals, for a given SNP i , let $n_{2,i}$, $n_{1,i}$, and $n_{0,i}$ denote the counts of individuals with genotype 2, 1, and 0, respectively. The allele frequency at locus i is $p_i = (2n_{2,i} + n_{1,i})/(2N)$. The genotype frequency is the proportion of each genotype class, given by $f_i(2) = n_{2,i}/N$ (homozygosity for the alternative allele), $f_i(1) = n_{1,i}/N$ (heterozygosity), $f_i(0) = n_{0,i}/N$ (homozygosity for the reference allele). A strong concordance indicates that the generative model has accurately reproduced the per-locus marginal distribution, which is a prerequisite before assessing the higher order structure.

3.2.2 Aggregated Fixation Index

The fixation index F_{ST} [39, 40] is a widely used population genetic statistic that quantifies the degree of genetic differentiation among populations. It normalizes the difference between the total heterozygosity and the average heterozygosity within populations, yielding a value between 0 (no genetic differentiation) and 1 (complete genetic differentiation). For SNP i , let $p_{\text{real},i}$ and $p_{\text{syn},i}$ denote the allele frequencies in the real and synthetic cohorts respectively, assuming both cohorts are of the same size. Thus, the combined allele frequency for the total population is $p_{T,i} = (p_{\text{real},i} + p_{\text{syn},i})/2$. Recall that for a given SNP i , the expected heterozygosity is given by $H = 1 - p^2 - (1 - p)^2$. Thus, for the two subpopulations we have $H_{\text{real},i} = 1 - p_{\text{real},i}^2 - (1 - p_{\text{real},i})^2$ and $H_{\text{syn},i} = 1 - p_{\text{syn},i}^2 - (1 - p_{\text{syn},i})^2$. The average within-subpopulation heterozygosity is $H_{S,i} = (H_{\text{real},i} + H_{\text{syn},i})/2$. The heterozygosity in the combined population is $H_{T,i} = 1 - p_{T,i}^2 - (1 - p_{T,i})^2$. The per-SNP fixation index is given by

$$F_{ST}(i) = \frac{H_{T,i} - H_{S,i}}{H_{T,i}} \quad (10)$$

Recognizing that not all SNPs are equally informative, with those exhibiting higher total heterozygosity providing greater insight into genetic diversity, we then aggregate the per-SNP fixation index into a summary metric using a weighted average:

$$F_{ST}^{\text{aggregated}} = \frac{\sum_i H_{T,i} F_{ST}(i)}{\sum_i H_{T,i}} \quad (11)$$

3.2.3 Linkage Disequilibrium and Its Decay with Physical Distance Along Chromosome

Linkage disequilibrium (LD) [41] measures the non-random association of alleles at different loci. Its decay with increasing physical distance along a chromosome reflects the effect of recombination in reshuffling genetic variation. When working with diploid genotype data, the gametic phase is often unknown, which complicates the accurate computation of LD statistics. To address this, we employ a fast estimator introduced in [42], which approximates LD between two loci without relying on the assumption of random mating or requiring iterative computation. This method is implemented in the *scikit-allel* Python library¹.

3.3 Unsupervised Metrics for Genotype Structural Similarity

3.3.1 Precision and Recall

Precision and recall, originally developed for classification, have been adapted to assess generative models [29]. In this context, precision measures the quality of the synthetic data by quantifying the fraction of generated samples that fall within the support of the real data distribution, while recall measures the diversity of the synthetic data by quantifying the fraction of real samples that fall within the support of the synthetic data distribution. The F1 score is the harmonic mean of precision and recall.

To estimate the support of a data distribution, we define, for each sample in this dataset, a threshold ϵ as the distance to its k^{th} nearest neighbor within the same set. This distance serves as the radius of a hypersphere centered on that sample, and the union of all such hyperspheres provides an estimate of the underlying manifold. Formally, let R denote the set of real samples and S the set of synthetic samples. Precision and recall are defined as follows:

$$\text{Precision} = \frac{1}{|S|} \sum_{s \in S} \mathbf{1}\left\{ \exists r \in R \text{ such that } \|s - r\| < \epsilon_r \right\}, \quad (12)$$

$$\text{Recall} = \frac{1}{|R|} \sum_{r \in R} \mathbf{1}\left\{ \exists s \in S \text{ such that } \|s - r\| < \epsilon_s \right\}. \quad (13)$$

In image-based applications, precision and recall are typically computed on high-level feature vectors extracted from pretrained VGG-16 [43] classifier. However, for genotype, no widely accepted pretrained network exists. We therefore use the original data directly for evaluation. For KNN-based manifold estimation, the L_2 distance is conventionally employed. Given that genotype is discrete, we experimented with both L_1 and L_2 distances and found no significant differences in the resulting metrics (Supplementary Section 3). We adopted L_2 distance since it's more computationally efficient. The choice of k is crucial and we selected the value of k that yielded approximately 90% precision and recall on two real datasets.

¹<https://scikit-allel.readthedocs.io/>

3.3.2 Correlation Score

To compare the moments of the real and synthetic distributions, correlation score is proposed in [13], the idea is to compute the Pearson correlation coefficient between the strictly upper-diagonal elements of the SNP-pairwise correlation matrices M_{real} and M_{syn} :

$$\rho(M_{real}, M_{syn}) = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{M_{i,j;real} - \mu(M_{real})}{\sigma(M_{real})} \times \frac{M_{i,j;syn} - \mu(M_{syn})}{\sigma(M_{syn})}, \quad (14)$$

where n is the number of SNPs, $\mu(M)$ is the mean, and $\sigma(M)$ is the standard deviation of the strictly upper-diagonal elements of M .

3.4 Supervised Metrics for Genotype–Phenotype Association

3.4.1 Genome-wide association study (GWAS)

In quantitative genetics, Genome-Wide Association Study (GWAS) [44] is a fundamental tool for identifying genetic variants associated with specific traits by examining the relationship between genotype and phenotype. In GWAS, a per-SNP regression is performed, and a two-sided t-test is used to determine whether the regression coefficient β is significantly different from 0. The corresponding p -value gives us the significance of the association. GWAS can be viewed as a feature-importance method, since each SNP's estimated effect size β and its p -value indicate how strongly that locus contributes to phenotype prediction. By comparing GWAS results obtained from synthetic population with those from real, we can directly evaluate whether our generative model has preserved key biological signals.

3.4.2 Phenotype Prediction Performance

We further evaluate synthetic genotype by its ability to predict the conditioning phenotype. Specifically, we train an XGBoost model and a multilayer perceptron (MLP) on synthetic data, then assess their performance on an independent real dataset not used during generative training. If a predictive model trained solely on synthetic data performs comparably to one trained on real data, it suggests that the synthetic population has faithfully preserved the underlying genotype–phenotype relationship.

3.5 Privacy Leakage Assessment

3.5.1 Nearest Neighbor Adversarial Accuracy (AA)

Since genotype data is highly sensitive, our synthetic data must balance utility with privacy protection. To evaluate this balance, we adopt the nearest neighbour adversarial accuracy (AA) metric proposed in [45]. This metric is conceptually similar to precision and recall. Rather than estimating the entire manifold with a full KNN approach, we use a 1NN measure to compare local neighborhood distances. The intuition is that synthetic data should be close enough to real data to preserve utility,

yet not so close as to risk privacy leakage. For each real sample, we measure whether its distance to its nearest synthetic neighbour (d_{RS}) is larger than its distance to its nearest real neighbour (d_{RR}). Likewise, for each synthetic sample, we check whether its distance to its nearest real neighbour (d_{SR}) is larger than its distance to its nearest synthetic neighbour (d_{SS}). These comparisons yield two values—one for the real dataset (AA_{real}) and one for the synthetic dataset (AA_{syn}). The overall AA score is then defined as the average of these two quantities. Formally, we have:

$$AA = \frac{1}{2} \left(\underbrace{\frac{1}{N} \sum_{i=1}^N \mathbf{1}(d_{RS}(i) > d_{RR}(i))}_{AA_{real}} + \underbrace{\frac{1}{N} \sum_{i=1}^N \mathbf{1}(d_{SR}(i) > d_{SS}(i))}_{AA_{syn}} \right). \quad (15)$$

Same as in the calculation of precision and recall, we use L_2 distance. An AA value near 0 indicates overfitting, while an AA value near 1 suggests underfitting. Ideally, an AA value around 0.5 reflects a good tradeoff between utility and privacy.

4 Experimental Setting

The following section outlines our experimental setup, including the datasets, model architectures, hyperparameter choices, synthetic data simulation and metric computation process. A schematic overview is provided in Figure 1.

4.1 Datasets

Since the frequency distribution and correlation between SNPs can vary across population groups, techniques developed using data from one group may not generalize well to others. Therefore, we used two large-scale genomic datasets from different species: a private Holstein cow cohort and the human dataset from UK Biobank² [46]. Genotype calls for diploid organisms are encoded as 0 for homozygous reference, 1 for heterozygous, and 2 for homozygous alternative alleles.

Cow: Our cow dataset comprises 93,484 Holstein individuals genotyped at 50,161 SNPs across all 29 pairs of autosomes. The selected phenotype is fat content (FC), a milk production trait that reflects the proportion of fat in milk. Fat is a key component in dairy products and influences the taste, texture, and richness of milk, making it an important criterion in milk pricing. FC has relatively high heritability, estimated at approximately 0.50. For selection purpose, FC was analyzed with a mixed model that accounts for various fixed environmental effects, the permanent environmental effect of the cow, and the breeding value (Supplementary Section 4). The so-called Yield Deviations (YD) are therefore by-products of the French Holstein Single Step genomic evaluation [47, 48]. The YD of FC for a cow is the mean of its phenotypes that have been adjusted for all non-genetic effects estimated in the genetic evaluation, and serves as the conditioning phenotype in our study. We assessed model performance on two individual chromosomes: Chromosome 5 (2,238 SNPs), where the MGST1 gene [49]

²<https://www.ukbiobank.ac.uk/>

is located, and Chromosome 14 (1,771 SNPs), where the DGAT1 gene [50] is located. In our experiments, these chromosomes exhibited the strongest GWAS signals for the selected trait. We also evaluated the models using the full concatenated genotype across all chromosomes.

Human: The UK Biobank provides genotype and phenotype data for 488,377 participants, including 805,426 variants comprising both SNPs and INDELS [51], across the 22 autosomes, sex chromosomes, and the mitochondrial chromosome. We used sex and height as conditioning phenotypes, as height is a highly heritable and polygenic trait [52]. Following the pipeline proposed in [53] to assemble our study subsets, we performed quality control using PLINK 1.9³, including checks for sex discordance, individual and SNP missingness, minor-allele-frequency filtering, Hardy–Weinberg equilibrium testing, and LD-based tag SNP selection. Missing genotypes were imputed using Beagle 5.4⁴. To recover biologically relevant height loci, we incorporated annotations from Ensembl⁵ and extracted 3,493 SNPs associated with height identified in previous studies. In a final cohort of 291,023 individuals, we constructed 4 genotype datasets: the 3,493 height-associated variants; Chromosome 6 (12,283 SNPs) where a QTL was detected by GWAS; Chromosome 12 (9,780 SNPs) where IGF-1 gene [54] is located; a combined set of 42,409 SNPs from Chromosomes 3, 6, 12, and 17.

For VAE, GAN, and WGAN models, the input genotype sequences were first transformed using one-hot encoding. For DM, we applied PCA and retained the number of principal components that captured 90% of the total variance in each dataset. Across all experiments, 70% of the samples were used for training, 15% for validation, and 15% for testing.

4.2 Models and Training

All the models were based on fully connected layers. Since genotype sequence has no inherent spatial or temporal structure, we avoided convolutional or recurrent layers. Instead, each model was composed of a sequence of dense layers, with layer widths heuristically scaled based on the data dimension. To improve training stability and gradient flow, we added residual connections [55]. For VAE, the encoder and decoder shared a symmetric architecture. For GAN and WGAN, we used the identical generator and discriminator architectures. For WGAN, the gradient penalty coefficient λ was set to 10 and we performed 5 discriminator updates for every generator update. For DM, we experimented with multiple noise-addition schedules and found that a linear β schedule yields the best performance.

We performed a grid search over the network architecture and the key training hyperparameters. We provide a full and detailed description of the model architectures and hyperparameter choices for all four models applied to the full cow chromosome dataset (Supplementary Section 5). To determine when to stop training, we monitored the F1 score since it balances precision and recall. Training was terminated once this score no longer improved.

³<https://www.cog-genomics.org/plink/1.9/>

⁴<https://faculty.washington.edu/browning/beagle/beagle.html>

⁵<https://www.ensembl.org/index.html>

4.3 Inference and Evaluation

We generated synthetic population under two scenarios. In the unconditional setting, the only required input was latent noise sampled from the Gaussian prior used during training. In the conditional setting, phenotype values were additionally sampled from the training set and provided as conditioning inputs. All metrics, except for the phenotype-prediction metric, were computed on the validation set. For the phenotype-prediction metric, we selected the best model using the validation set and reported its performance on the test set. All metrics were averaged over 5 independent runs and 10,000 synthetic samples were generated per run. For metrics that require hyperparameter tuning, we suggest selecting the values that deliver satisfactory performance on two real datasets.

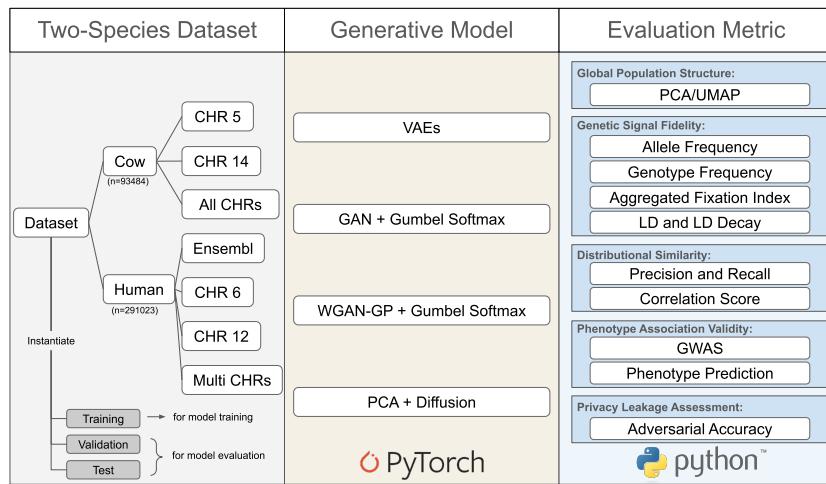


Fig. 1: Schema of our generative modeling task. We used data from two species and constructed sub-datasets at different scales, ranging from single-chromosome to multi-chromosome settings. For the human dataset, the Multi CHRs setting included chromosomes 3, 6, 12, and 17. Four generative models were implemented: VAE, GAN with Gumbel-Softmax, WGAN-GP with Gumbel-Softmax, and PCA combined with diffusion. These models were evaluated across different aspects relevant to performance.

5 Results

5.1 Do Generative Models Capture the Statistical and Genetic Structure of the Real Population?

Global Distribution Resemblance: A preliminary UMAP visualization of real and synthetic cow populations (Figure 2) shows that all models except GAN can approximate the overall data distribution. VAE and DM, both likelihood-based generative models, capture the global central structure well, with synthetic clusters centered around the real data. DM appears slightly better than VAE, as the latter shows more dispersion at the edges. WGAN performs best in this setting, effectively covering both the central structure and the broader population heterogeneity. To further investigate, we compared the first 32 principal components of WGAN-generated and real data and observed near-perfect alignment (Figure 2e), suggesting strong distributional fidelity.

Genetic Parameters and Linkage Structure Comparison: Figure 3 shows the comparison of allele and genotype frequencies between real and synthetic populations. Among all models, WGAN clearly outperforms the others, achieving nearly perfect correlation score between real and synthetic frequency distributions. While the other models can partially capture the frequency profiles, we observed a consistent pattern of deviation: the frequency plots exhibit a sigmoid-like distortion. This phenomenon reflects a known Matthew effect [56, 57], where the model tends to overestimate high-frequency variants and underestimate rare ones, amplifying existing disparities in the data. For VAE, this phenomenon may be linked to its likelihood-based objective, which encourages prioritizing frequent patterns to maximize likelihood. DM performs better in this regard, possibly due to its hierarchical noise removal mechanism. GAN, which is known to suffer from mode collapse [34], exhibits this effect more severely. WGAN appears to be the only model that successfully mitigates this bias and accurately preserves the full frequency spectrum. In terms of LD, shown in Figure 4, all models except GAN manage to reproduce the original LD block structure and show a similar decay pattern with increasing distance. However, both VAE and WGAN tend to underestimate the strength of LD, while DM most closely matches the LD structure observed in the real population.

Quantitative Evaluation Metrics: Table 1 summarizes the results across all quantitative metrics. For relatively small datasets (e.g., single chromosome in cow dataset with around a thousand SNPs), VAE, WGAN, and DM perform well across most metrics. In contrast, GAN suffers from mode collapse, leading to a recall score close to 0. For larger-scale datasets (e.g., full chromosomes in cow and multiple chromosomes in human), WGAN consistently outperforms the other models. This is particularly evident for the recall metric: while all models tend to achieve high precision, WGAN is the only model that significantly improves recall. This aligns with the UMAP visualization in Figure 2, which shows WGAN covering the full data distribution more effectively. Overall, WGAN achieves the best results across most metrics, although DM occasionally surpasses it in correlation score on human datasets.

Factors Affecting the Complexity of Generative Modeling: Table 1 suggests that the difficulty of generative modeling is related to the input dimensionality: higher dimensions generally make learning more challenging, and we indeed observed

this trend. However, we also observed that in the human dataset, CHR 6 has a higher input dimension than CHR 12, yet it is easier for models to learn. This indicates that input dimension alone does not fully explain the complexity. Upon further analysis, we found that SNP dependency also plays an important role. When SNPs exhibit stronger dependency, generative models can more easily capture the underlying distribution (Supplementary Section 6). Comparing across datasets, models consistently perform better on cow dataset than on human dataset. This may be attributed to differences in genetic structure: the cow dataset exhibits stronger LD, implying higher SNP dependency, while the human dataset shows greater genetic variability, which increases learning complexity.

On the Robustness of Evaluation Metrics: When assessing the robustness of the evaluation metrics, we found that $F_{ST}^{\text{aggregated}}$, F1 and correlation score are highly correlated: good performance in one metric typically results in good performance across the others. A clear trade-off exists between precision and recall: models can achieve high precision by capturing only the core of the real distribution, whereas recall reflects how well the model covers the full diversity. A more detailed examination of the AA score reveals an important nuance, consistent with findings reported in [23]: extreme scenarios may yield a favorable global AA score while masking poor generative behavior. Ideally, both AA_{real} and AA_{syn} should be close to 0.50. However, when applying DM to human dataset, we observed a global AA score of 0.50 resulting from an imbalanced case where $AA_{\text{real}} \approx 0$ and $AA_{\text{syn}} \approx 1$. This means that real samples are closer to synthetic ones than to other real samples, while synthetic samples are only close to each other and fail to reflect the diversity of the real distribution. This is also consistent with the observation that precision is close to 1 while recall is near 0, indicating that the model captures only a subset of the true distribution (Supplementary Section 7).

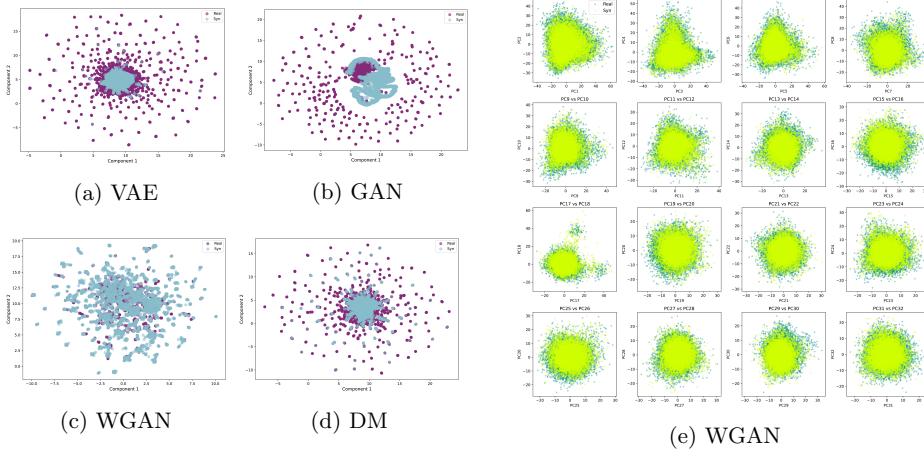


Fig. 2: PCA and UMAP Visualizations of Real and Model-Generated Synthetic Populations on All Chromosomes of Cow Dataset. (a), (b), (c), (d): UMAP of real data and synthetic genotype generated by VAE, GAN, WGAN, and DM, respectively. (e): First 32 principal components of real and WGAN-generated synthetic genotype, explaining approximately 12% of the total variance.

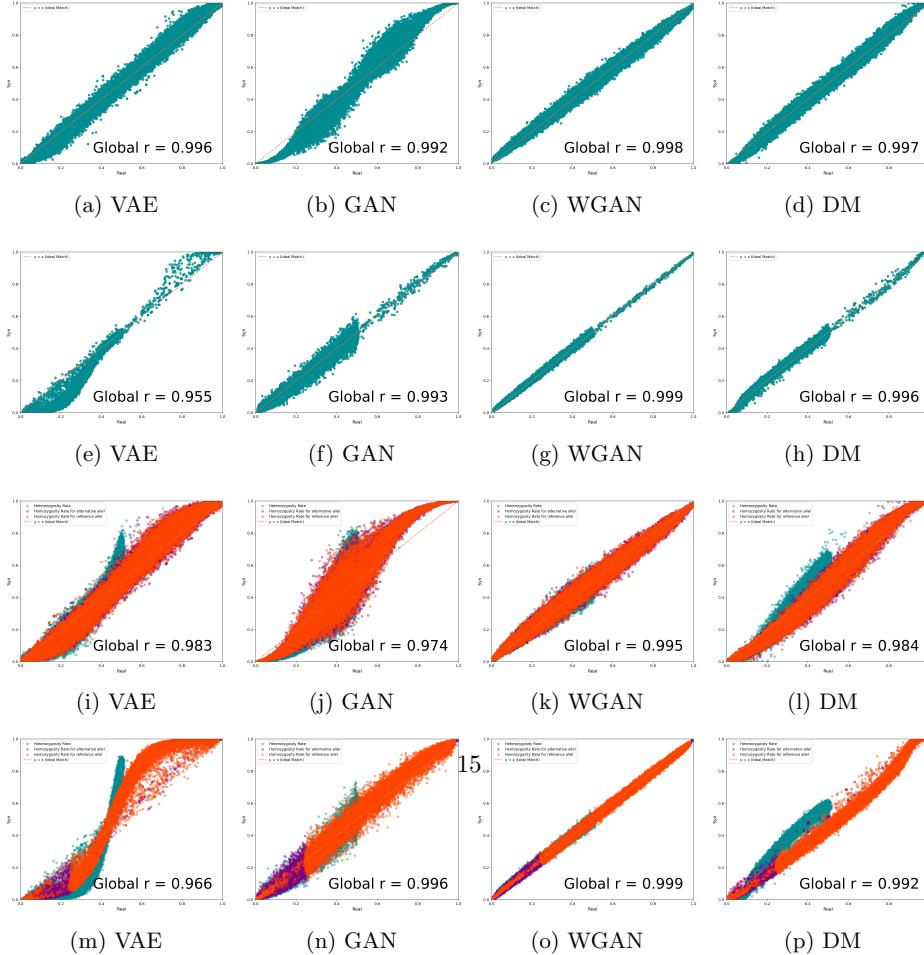


Fig. 3: Comparison of Genetic Parameters Between Real and Synthetic Populations Across All Chromosomes of the Cow Dataset and Multiple Chromosomes of the Human Dataset. (a) to (d): Allele frequency comparison between real and synthetic genotype in cow dataset. (e) to (h): Allele frequency comparison between real and synthetic genotype in human dataset. (i) to (l): Genotype frequency comparison between real and synthetic genotype in cow dataset. (m) to (p): Genotype frequency comparison between real and synthetic genotype in human dataset.

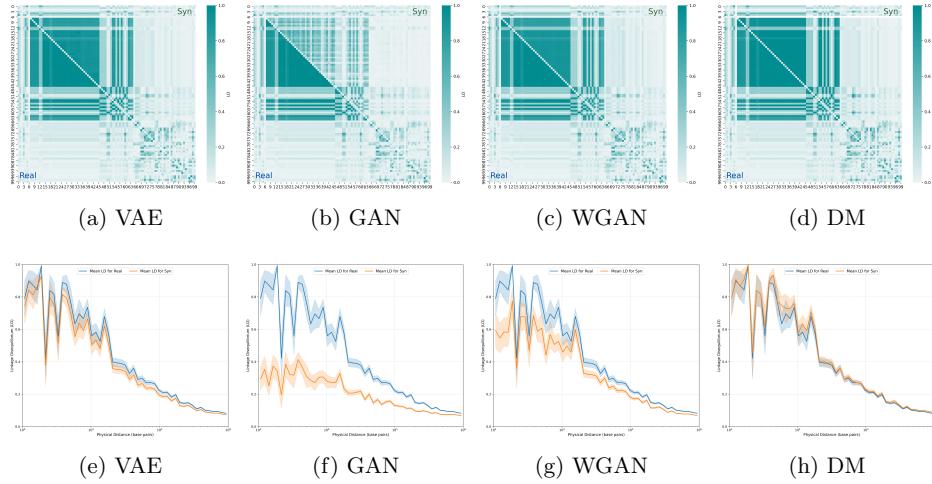


Fig. 4: Comparison of Linkage Disequilibrium Between Real and Synthetic Populations on Chromosome 14 of Cow Dataset. (a) to (d): LD block heatmaps, where the upper diagonal represents pairwise LD in synthetic population and the lower diagonal represents pairwise LD in real population. (e) to (h): LD decay with respect to physical distance in real and synthetic populations.

5.2 Do Generative Models Preserve Genotype–Phenotype Association?

In the previous unconditional setting, WGAN and DM demonstrated superior performance, especially for large-dimensional datasets. We then evaluated their performance under the conditional setting by using phenotype as conditioning variable to investigate whether the models could also capture the genotype-phenotype association. Figure 5 presents the results of GWAS analysis on the full set of chromosomes in cow dataset, comparing real and synthetic populations. Both WGAN and DM are able to recover the 3 main quantitative trait locus (QTL) regions. When examining the regression coefficients β in GWAS, we observed that the WGAN-generated synthetic population shows a higher correlation with the real population’s β values compared to the DM-generated population.

Table 2 summarizes the predictive performance of machine learning and deep learning models on synthetic genotype to predict the conditioning phenotype, compared to the results from the real population. Similarly to the previous unconditional setting, for relatively small datasets (e.g., single chromosome in cow dataset), both models achieve comparable performance to that obtained with real datasets. However, for more complex datasets, WGAN-generated synthetic genotype appears to better preserve the complex genotype-phenotype relationship, especially when using MLP as the prediction model. This aligns with WGAN’s ability to improve the recall metric and

Table 1: Quantitative performance indicators for all generative models on Cow and Human datasets.

Dataset	Chromosome	Model	$F_{ST}^{\text{aggregated}}$	Precision (%) ↑	Recall (%) ↑	$F1$ (%) ↑	Corr(%) ↑	AA
Cow	CHR 14 (1771 SNPs)	VAE	1.81e-4 ± 7e-6	99.06 ± 0.04	99.70 ± 0.04	99.38 ± 0.02	96.87 ± 0.04	0.63 ± 2e-3
		GAN	1.88e-4 ± 4e-6	80.88 ± 0.23	57.97 ± 0.45	67.53 ± 0.24	72.60 ± 0.13	0.99 ± 3e-4
		WGAN	1.19e-4 ± 2e-6	99.64 ± 0.04	99.88 ± 0.02	99.76 ± 0.02	98.65 ± 0.02	0.55 ± 3e-3
		DM	3.07e-4 ± 6e-6	99.92 ± 0.01	99.13 ± 0.03	99.52 ± 0.02	98.53 ± 0.01	0.63 ± 2e-3
Cow	CHR 5 (2238 SNPs)	VAE	4.05e-4 ± 5e-6	99.87 ± 0.02	99.51 ± 0.03	99.69 ± 0.02	97.21 ± 0.04	0.68 ± 2e-3
		GAN	3.99e-3 ± 4e-5	88.40 ± 0.21	0.01 ± 0.00	0.01 ± 0.01	55.40 ± 0.05	1.00 ± 7e-5
		WGAN	1.22e-4 ± 4e-6	98.98 ± 0.07	99.86 ± 0.03	99.42 ± 0.04	98.74 ± 0.01	0.63 ± 2e-3
		DM	3.10e-4 ± 4e-6	99.86 ± 0.02	99.26 ± 0.07	99.56 ± 0.04	98.17 ± 0.01	0.65 ± 3e-3
All CHRs (50161 SNPs)		VAE	1.80e-3 ± 1e-5	99.99 ± 0.01	11.65 ± 1.03	20.85 ± 1.65	73.03 ± 0.11	0.96 ± 1e-3
		GAN	5.58e-3 ± 1e-5	100 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.52 ± 0.01	0.98 ± 2e-3
		WGAN	6.21e-4 ± 5e-6	92.00 ± 0.16	99.93 ± 0.01	95.80 ± 0.09	83.32 ± 0.06	0.74 ± 7e-3
		DM	1.10e-3 ± 1e-6	100 ± 0.00	40.59 ± 0.63	57.74 ± 0.64	76.56 ± 0.10	0.94 ± 1e-3
Ensembl (3493 SNPs)		VAE	2.88e-2 ± 3e-5	100 ± 0.00	0.29 ± 0.24	0.57 ± 0.44	39.74 ± 1.35	0.50 ± 1e-5
		GAN	5.00e-3 ± 9e-6	99.98 ± 0.01	0.00 ± 0.00	0.00 ± 0.00	34.03 ± 0.09	0.52 ± 5e-4
		WGAN	1.31e-4 ± 2e-6	71.84 ± 0.11	97.86 ± 0.11	82.85 ± 0.11	83.74 ± 0.03	0.76 ± 1e-2
		DM	1.53e-3 ± 5e-6	100 ± 0.00	13.96 ± 0.07	24.49 ± 0.11	61.73 ± 0.72	0.50 ± 3e-4
Human	CHR 6 (12283 SNPs)	VAE	6.08e-3 ± 3e-5	99.99 ± 0.01	0.05 ± 0.07	0.10 ± 0.13	64.93 ± 0.03	0.50 ± 6e-5
		GAN	1.62e-3 ± 6e-6	99.02 ± 0.12	0.17 ± 0.06	0.34 ± 0.11	20.51 ± 0.46	0.52 ± 4e-4
		WGAN	2.24e-4 ± 1e-6	57.76 ± 0.33	97.83 ± 0.11	72.63 ± 0.23	53.97 ± 0.10	0.73 ± 2e-2
		DM	9.54e-4 ± 4e-6	100 ± 0.00	1.20 ± 0.05	2.36 ± 0.09	54.65 ± 0.77	0.50 ± 9e-5
Human	CHR 12 (9780 SNPs)	VAE	1.40e-2 ± 5e-5	99.99 ± 0.01	0.04 ± 0.01	0.08 ± 0.03	26.91 ± 0.21	0.50 ± 6e-4
		GAN	9.20e-4 ± 6e-6	21.15 ± 0.25	0.95 ± 0.09	1.82 ± 0.17	8.30 ± 0.08	0.99 ± 6e-3
		WGAN	1.13e-4 ± 1e-6	55.28 ± 0.72	75.19 ± 0.53	63.71 ± 0.57	40.16 ± 0.19	0.55 ± 4e-3
		DM	9.23e-4 ± 3e-6	100 ± 0.00	1.20 ± 0.03	2.38 ± 0.06	40.35 ± 0.30	0.50 ± 1e-4
Human	Multi CHRs (42409 SNPs)	VAE	1.77e-2 ± 5e-5	100 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	5.57 ± 0.07	0.50 ± 2e-3
		GAN	1.46e-3 ± 5e-6	100 ± 0.01	0.65 ± 0.03	1.30 ± 0.05	5.10 ± 0.17	0.51 ± 7e-5
		WGAN	1.63e-4 ± 1e-6	45.80 ± 0.42	64.35 ± 1.28	53.50 ± 0.46	19.58 ± 0.50	0.52 ± 1e-2
		DM	9.56e-4 ± 3e-6	100 ± 0.00	1.00 ± 0.01	1.98 ± 0.02	20.04 ± 0.23	0.50 ± 8e-5

Note: Scientific notation is used, for example, ae-b denotes $a \times 10^{-b}$, where e represents base-10 exponentiation.

more fully capture the distribution of real data. Overall, these results suggest that WGAN is able to generate synthetic population with genotype–phenotype association that closely mirror those observed in real data, as reflected by its consistently strong predictive performance across datasets and predictive model types.

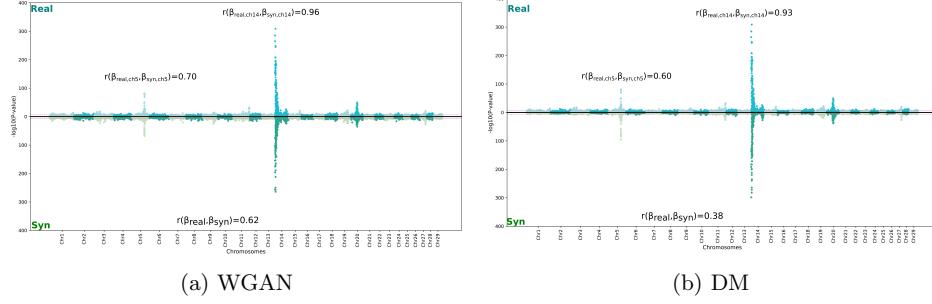


Fig. 5: GWAS Comparison of Real and Synthetic Populations Across All Chromosomes of Cow Dataset. (a) Compared with WGAN-generated genotype. (b) Compared with DM-generated genotype.

6 Discussion and Conclusion

The primary objective of this study was to investigate the effectiveness of widely used deep generative models for simulating genotype. We proposed specific adaptation for VAE, GAN, WGAN and diffusion models to better handle the discrete nature of genotype representation. Our experiments revealed that no single model performs best across all evaluation metrics and datasets. Each dataset exhibits distinct genetic properties, and we found that model performance is influenced by both the dimensionality of genotype sequence and the degree of SNP dependence. For relatively small and simple datasets (e.g., with a few thousand SNPs), we recommend using VAE due to its computational efficiency, training stability, and minimal hyperparameter tuning. For larger and more complex datasets with higher genetic diversity, WGAN-based model consistently outperforms the other models, particularly in capturing the overall distribution and the genotype–phenotype association.

We also proposed a comprehensive evaluation framework that combines multiple metrics to assess synthetic genotype quality from different angles. Since each metric captures a specific aspect, using them in combination provides a more complete evaluation. We found that not all previously developed metrics are robust. The AA score can produce misleading results in certain edge cases. Other metrics, such as the correlation score, are reliable but computationally intensive. Among all the metrics, recall stands out as a particularly valuable supervision signal during training, as it is more difficult to optimize and indicative of a model’s ability to capture diversity. We recommend using PCA, $F_{ST}^{\text{aggregated}}$, precision, and recall during training to decide when to stop and only compute the more costly metrics afterward.

Table 2: Comparison of phenotype-prediction performance using real and synthetic genotype data.

Dataset		XGBoost		MLP	
		MSE ↓	r ↑	MSE ↓	r ↑
Cow	CHR 14	Real	0.60 ± 0.0054	0.65 ± 0.0034	0.67 ± 0.0088
		WGAN	0.62 ± 0.0050	0.63 ± 0.0012	0.72 ± 0.0323
		DM	0.61 ± 0.0056	0.64 ± 0.0023	0.59 ± 0.0041
	CHR 5	Real	0.95 ± 0.0004	0.23 ± 0.0031	1.12 ± 0.0096
		WGAN	0.95 ± 0.0014	0.23 ± 0.0021	1.14 ± 0.0168
		DM	0.96 ± 0.0019	0.21 ± 0.0045	1.14 ± 0.0320
	All CHRs	Real	0.46 ± 0.0053	0.76 ± 0.0023	0.40 ± 0.0135
		WGAN	0.52 ± 0.0069	0.72 ± 0.0027	0.49 ± 0.0261
		DM	0.53 ± 0.0145	0.71 ± 0.0084	0.47 ± 0.0248
Human	Ensembl	Real	40.87 ± 0.1936	0.72 ± 0.0016	68.33 ± 6.1033
		WGAN	43.44 ± 0.1392	0.70 ± 0.0011	69.87 ± 9.3012
		DM	43.28 ± 0.6887	0.70 ± 0.0047	114.62 ± 31.7599
	CHR 6	Real	42.81 ± 0.0791	0.71 ± 0.0006	88.53 ± 4.7244
		WGAN	43.30 ± 0.1146	0.70 ± 0.0010	94.62 ± 5.2922
		DM	44.65 ± 0.9555	0.69 ± 0.0079	218.95 ± 27.5067
	CHR 12	Real	42.96 ± 0.0912	0.70 ± 0.0007	94.33 ± 5.6021
		WGAN	43.41 ± 0.0684	0.70 ± 0.0006	96.52 ± 6.2831
		DM	45.15 ± 1.4170	0.69 ± 0.0123	237.82 ± 14.8840
	Multi CHRs	Real	42.63 ± 0.1547	0.71 ± 0.0013	92.91 ± 3.2856
		WGAN	43.44 ± 0.0800	0.70 ± 0.0006	96.51 ± 1.4907
		DM	44.65 ± 0.9555	0.69 ± 0.0079	149.87 ± 5.7670

Our results align with previous studies on haplotype generation that suggest generative models can accurately capture the genetic structure. Moreover, to our knowledge, this is the first work to show that conditioning on phenotype allows generative models to produce synthetic population that preserve genotype–phenotype association. The resulting synthetic population can be effectively used in downstream applications, such as GWAS, highlighting the potential of generative models to support genetics research.

Several future research directions can be envisioned. In this study, we focused on models that learn the joint distribution of the entire genotype sequence, favoring a biologically grounded approach over sequential modeling. However, recent advances inspired by natural language processing, such as transformer-based models applied to DNA sequence [58–60], could also be adapted for genotype and merit further investigation. Another potential direction is the development of post-training refinement algorithms to improve the quality of generated sequences [61]. On the data side, future

work could aim to better model additional features of genotype data, such as rare variants, population heterogeneity, and multi-phenotype conditioning. Incorporating modules that explicitly capture genotype–phenotype interaction could further enhance biological relevance. Lastly, exploring frugal learning strategies would be valuable, given the high dimensionality of genotype data and the computational demands of generative models.

Supplementary Data. Supplementary data is available online.

Data Availability Statement. We provide code for model training, evaluation metrics, and experiments (<https://github.com/SihanXXX/DiscreteGenoGen>). We also provide trained generative models for cows, allowing others to use them directly to generate synthetic genotype. This supports the main motivation of our research, which is to enable genotype data sharing in a compact and privacy-preserving way. Access to the UK Biobank requires a separate application, which can be submitted at: <https://www.ukbiobank.ac.uk/enable-your-research/apply-for-access>.

Ethics Statement. This study demonstrates that generative models can effectively capture genetic structure and reproduce genotype–phenotype association within the given population dataset. However, all validations were conducted in a numerical and computational framework. No experimental validation has been conducted to confirm the biological relevance of the synthetic population. For instance, if a SNP appears to be associated with a trait in the synthetic population, this merely reflects statistical pattern learned from the original data and should not be interpreted as a new biological discovery. All data handling procedures in this project comply with the General Data Protection Regulation (GDPR).

Acknowledgements. This work was supported by the INRAE DigitBio Metaprogram. We thank Jocelyn De-Goér-De-Herve for managing the GPU infrastructure and GPT4 for spelling and grammar checks. This study makes use of data from the UKBiobank under application number 96326.

Author Contributions Statement. SX curated the human dataset, implemented the methods and wrote the paper. TT and DB curated the cow dataset. EB, JC, and BH supervised the project and reviewed the manuscript. All the authors read and approved the manuscript.

References

- [1] Reuter, J., Spacek, D.V., Snyder, M.: High-throughput sequencing technologies. *Molecular Cell* **58**(4), 586–597 (2015)
- [2] Churko, J., Mantalas, G., Snyder, M., *et al.*: Overview of high throughput sequencing technologies to elucidate molecular pathways in cardiovascular diseases. *Circulation research* **112**, 1613–23 (2013)
- [3] Gravel, S.: Population genetics models of local ancestry. *Genetics* **191**(2), 607–619 (2012)

- [4] Kingman, J.F.C.: The coalescent. Stochastic Processes and their Applications **13**(3), 235–248 (1982)
- [5] Hudson, R.R.: Gene genealogies and the coalescent process. Oxford surveys in evolutionary biology **7**(1), 44 (1990)
- [6] Kelleher, J., Etheridge, A.M., McVean, G.: Efficient coalescent simulation and genealogical analysis for large sample sizes. PLOS Computational Biology **12**(5), 1–22 (2016)
- [7] Hudson, R.R.: Generating samples under a Wright–Fisher neutral model of genetic variation. Bioinformatics **18**(2), 337–338 (2002)
- [8] Teshima, K.M., Innan, H.: mbs: modifying hudson’s ms software to generate samples of dna sequences with a biallelic site under selection. BMC Bioinformatics **10**(1), 166 (2009)
- [9] Baumdicker, F., Bisschop, G., Goldstein, D., *et al.*: Efficient ancestry and mutation simulation with msprime 1.0. Genetics **220**(3), 229 (2021)
- [10] Haller, B.C., Messer, P.W.: Slim 3: Forward genetic simulations beyond the Wright–Fisher model. Molecular Biology and Evolution **36**(3), 632–637 (2019)
- [11] Ewing, G., Hermisson, J.: Msms: a coalescent simulation program including recombination, demographic structure and selection at a single locus. Bioinformatics **26**(16), 2064–2065 (2010)
- [12] Peng, B., Kimmel, M.: simupop: a forward-time population genetics simulation environment. Bioinformatics **21**(18), 3686–3687 (2005)
- [13] Viñas, R., Andrés-Terré, H., Liò, P., *et al.*: Adversarial generation of gene expression data. Bioinformatics **38**(3), 730–737 (2021)
- [14] Lacan, A., Sebag, M., Hanczar, B.: Gan-based data augmentation for transcriptomics: survey and comparative assessment. Bioinformatics **39**(Supplement_1), 111–120 (2023)
- [15] Lacan, A., André, R., Sebag, M., *et al.*: In silico generation of gene expression profiles using diffusion models. bioRxiv (2024)
- [16] Li, Z., Ni, Y., Huygelen, T.A.B., *et al.*: Latent Diffusion Model for DNA Sequence Generation (2023)
- [17] Brix, G., Durrant, M.G., Ku, J., *et al.*: Genome modeling and design across all domains of life with evo 2. bioRxiv (2025)
- [18] Perera, M., Montserrat, D.M., Barrabés, M., *et al.*: Generative moment matching networks for genotype simulation. bioRxiv (2022)

- [19] Geleta, M., Montserrat, D.M., Giro-i-Nieto, X., et al.: Deep variational autoencoders for population genetics. bioRxiv (2023)
- [20] Montserrat, D.M., Bustamante, C., Ioannidis, A.: Class-conditional vae-gan for local-ancestry simulation. arXiv preprint arXiv:1911.13220 (2019)
- [21] Nußberger, J., Boesel, F., Lenz, S., et al.: Synthetic observations from deep generative models and binary omics data with limited sample size. Briefings in Bioinformatics **22**(4), 226 (2020)
- [22] Yelmen, B., Decelle, A., Ongaro, L., et al.: Creating artificial human genomes using generative neural networks. PLoS Genetics **17**(2), 1009303 (2021)
- [23] Yelmen, B., Decelle, A., Boulos, L.L., et al.: Deep convolutional and conditional neural networks for large-scale genomic data generation. PLOS Computational Biology **19**(10), 1011584 (2023)
- [24] Szatkownik, A., Furtlechner, C., Charpiat, G., et al.: Latent generative modeling of long genetic sequences with gans. bioRxiv, 2024–08 (2024)
- [25] Szatkownik, A., Planche, L., Demeulle, M., et al.: Diffusion-based artificial genomes and their usefulness for local ancestry inference. bioRxiv, 2024–10 (2024)
- [26] Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Foundations and Trends in Machine Learning **12**(4), 307–392 (2019)
- [27] Mirza, M., Osindero, S.: Conditional generative adversarial nets. CoRR (2014)
- [28] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual (2020)
- [29] Kynkääniemi, T., Karras, T., Laine, S., et al.: Improved precision and recall metric for assessing generative models. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
- [30] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
- [31] Wold, S., Esbensen, K., Geladi, P.: Principal component analysis. Chemometrics and Intelligent Laboratory Systems **2**(1), 37–52 (1987). Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists
- [32] Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings (2017)

- [33] Kusner, M.J., Hernández-Lobato, J.M.: Gans for sequences of discrete elements with the gumbel-softmax distribution. arXiv preprint arXiv:1611.04051 (2016)
- [34] Bau, D., Zhu, J., Wulff, J., *et al.*: Seeing what a GAN cannot generate. In: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, pp. 4501–4510 (2019)
- [35] Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 214–223 (2017)
- [36] Gulrajani, I., Ahmed, F., Arjovsky, M., *et al.*: Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
- [37] McInnes, L., Healy, J., Saul, N., *et al.*: Umap: Uniform manifold approximation and projection. *The Journal of Open Source Software* **3**(29), 861 (2018)
- [38] Laland, K.N., Uller, T., Feldman, M.W., *et al.*: The extended evolutionary synthesis: its structure, assumptions and predictions. *Proceedings of the Royal Society B: Biological Sciences* **282**(1813), 20151019 (2015)
- [39] Wright, S.: The genetical structure of populations. *Annals of Eugenics* **15**(1), 323–354 (1949)
- [40] Weir, B.S., Cockerham, C.C.: Estimating f-statistics for the analysis of population structure. *Evolution* **38**(6), 1358–1370 (1984)
- [41] Slatkin, M.: Linkage disequilibrium: Understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics* **9**(6), 477–485 (2008)
- [42] Rogers, A.R., Huff, C.: Linkage disequilibrium between loci with unknown phase. *Genetics* **182**(3), 839–844 (2009)
- [43] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
- [44] Uffelmann, E., Huang, Q.Q., Munung, N.S., *et al.*: Genome-wide association studies. *Nature Reviews Methods Primers* **1**(1), 59 (2021)
- [45] Yale, A., Dash, S., Dutta, R., *et al.*: Generation and Evaluation of Privacy Preserving Synthetic Health Data. *Neurocomputing* **416**, 244–255 (2020)
- [46] Sudlow, C., Gallacher, J., Allen, N., *et al.*: Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS Medicine* **12**(3), 1001779 (2015)

- [47] Tribout, T., Ducrocq, V., Boichard, D.: Hssgbup: a Single-Step SNP BLUP genomic evaluation software adapted to large livestock populations. In: Proceedings of the 6th International Conference of Quantitative Genetics, pp. 2–12 (2020)
- [48] Fernando, R.L., Cheng, H., Golden, B.L., *et al.*: Computational strategies for alternative single-step bayesian regression models with large numbers of genotyped and non-genotyped animals. *Genetics Selection Evolution* **48**(1), 96 (2016)
- [49] Littlejohn, M., Tiplady, K., Fink, T., *et al.*: Sequence-based association analysis reveals an mgst1 eqtl with pleiotropic effects on bovine milk composition. *Scientific Reports* **6** (2016)
- [50] Winter, A., Krämer, W., Werner, F., *et al.*: Association of a lysine-232/alanine polymorphism in a bovine gene encoding acyl-coa:diacylglycerol acyltransferase (dgat1) with variation at a quantitative trait locus for milk fat content. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 9300–5 (2002)
- [51] Mullaney, J.M., Mills, R.E., Pittard, W.S., *et al.*: Small insertions and deletions (indels) in human genomes. *Human Molecular Genetics* **19**(R2), 131–136 (2010)
- [52] Lango Allen, H., Estrada, K., Lettre, G., *et al.*: Hundreds of variants clustered in genomic loci and biological pathways affect human height. *Nature* **467**(7317), 832–838 (2010)
- [53] Anderson, C.A., Pettersson, F.H., Clarke, G.M., *et al.*: Data quality control in genetic case-control association studies. *Nature Protocols* **5**(9), 1564–1573 (2010)
- [54] Tahimic, C., Wang, Y., Bikle, D.: Anabolic effects of igf-1 signaling on the skeleton. *Frontiers in Endocrinology* **4**, 6 (2013)
- [55] He, K., Zhang, X., Ren, S., *et al.*: Deep Residual Learning for Image Recognition. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR '16, pp. 770–778 (2016)
- [56] Gao, C., Huang, K., Chen, J., *et al.*: Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 238–248 (2023)
- [57] Ganev, G., Oprisanu, B., Cristofaro, E.D.: Robin Hood and Matthew Effects: Differential Privacy Has Disparate Impact on Synthetic Data (2022)
- [58] Ji, Y., Zhou, Z., Liu, H., *et al.*: Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*

37(15), 2112–2120 (2021)

- [59] Zhang, D., Zhang, W., Zhao, Y., et al.: Dnagpt: A generalized pre-trained tool for multiple dna sequence analysis tasks. bioRxiv (2024)
- [60] Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., et al.: Nucleotide transformer: building and evaluating robust foundation models for human genomics. Nature Methods **22**, 287–297 (2025)
- [61] Li, Z., Ni, Y., Xia, G., et al.: Absorb & escape: Overcoming single model limitations in generating heterogeneous genomic sequences. In: Advances in Neural Information Processing Systems, vol. 37, pp. 21949–21978 (2024)

Supplementary Materials

1 Principal Component Analysis (PCA) Results on Genotype Datasets

Dataset	Number of PCs	MSE	SNP Matching Accuracy
Cow CHR 14 (1,771 SNPs)	155	0.0290	97.1089%
Cow CHR 5 (2,238 SNPs)	206	0.0281	97.1995%
Cow All CHRs (50,161 SNPs)*	4,819	0.0277	97.2411%
Human Ensembl (3,493 SNPs)	2,026	0.0339	96.7983%
Human CHR 6 (12,283 SNPs)	5,149	0.0143	98.5745%
Human CHR 12 (9,780 SNPs)	4,438	0.0143	98.5726%
Human Multi CHRs (42,409 SNPs)*	18,769	0.0144	98.5688%

Table 1: Principal Component Analysis (PCA) performed on each dataset with 90% of the variance retained from the original data. The continuous values are then reconstructed to discrete values (0, 1, or 2) by assigning the closest discrete value, for example, $(-\infty, 0.5) \rightarrow 0$, $[0.5, 1.5] \rightarrow 1$, and $(1.5, +\infty) \rightarrow 2$.

* We apply PCA to each chromosome individually, and then concatenate the results.

2 Failed thresholding strategy when attempting to map generated continuous values to 0/1/2

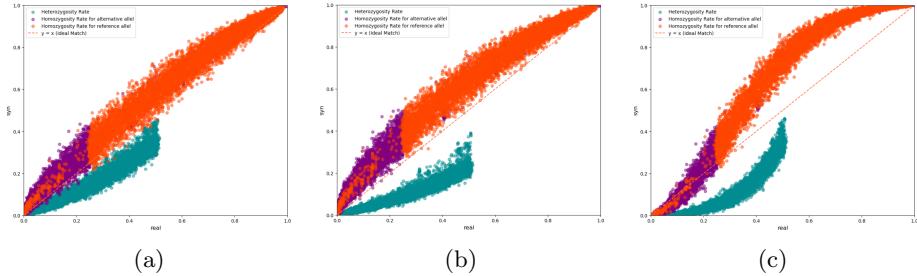


Figure 1: Comparison of genotype frequencies under different thresholding strategies used to map continuous values to discrete set 0, 1, 2. (a) Thresholds at $[0, \frac{1}{2}, \frac{3}{2}, 2]$, (b) Thresholds at $[0, \frac{2}{3}, \frac{4}{3}, 2]$, and (c) Dynamic thresholding for each SNP based on Hardy–Weinberg equilibrium and allele frequency.

Additionally, it is possible to construct a synthetic population with exactly the same genotype frequencies as the original population. This can be achieved by first ranking all the continuous outputs and then mapping them to genotype categories (0/1/2) according to the frequency proportion observed in the original data. While this approach ensures perfect alignment in genotype frequency, it tends to degrade performance on other metrics. In particular, the recall metric typically drops to 0, indicating poor diversity in the generated samples.

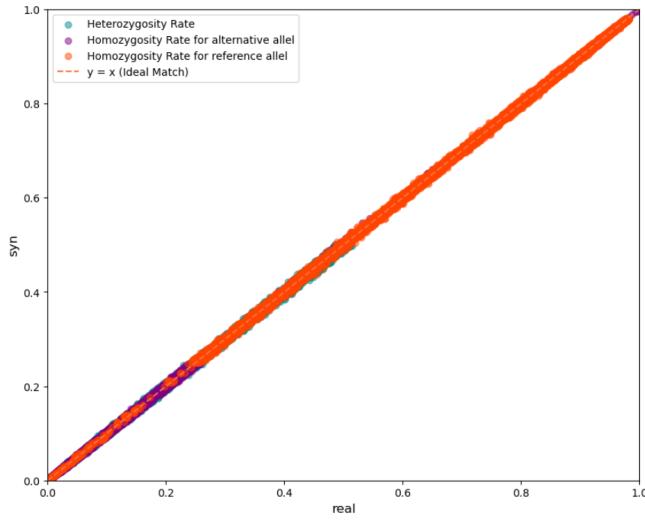


Figure 2: Exact genotype frequency matching between real and synthetic populations using a thresholding strategy based on genotype frequency.

3 Precision and Recall with Different Distance Metrics

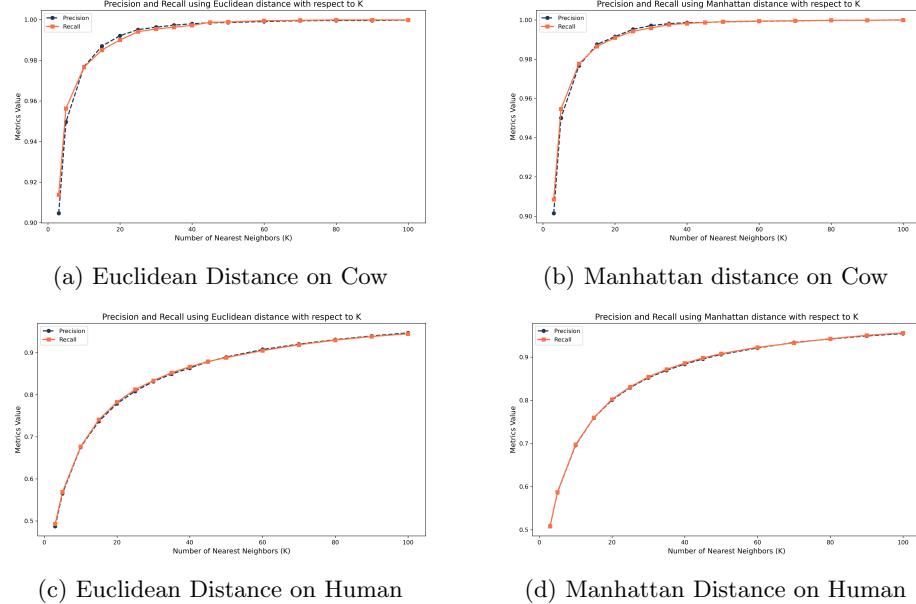


Figure 3: Evolution of Precision and Recall with respect to k using Euclidean and Manhattan Distances on Cow and Human Datasets.

4 Method for Calculating Yield Deviation of Fat Content

Fat content is measured by mid-infrared spectrometry on individual milk samples collected during monthly test days. For selection purpose, the trait analyzed is the average value over the entire lactation, with records repeated across lactations. To account for this, we use a multiplicative mixed model that adjusts for heterogeneous phenotypic variances within herd-year and region-year classes and accommodates multiple records per cow. The model accounts for fixed environmental effect, random permanent environmental effect, random genetic effect, and residual effect.

Let i index cows, j index repeated records for each cow, and h index the level corresponding to variance heterogeneity. The j^{th} record for cow i , associated with heterogeneity level h , is denoted by $y_{i,j,h}$. Formally, it is modeled as:

$$\begin{aligned}
 y_{i,j,h} = & (\text{effect}_{\text{herd} \times \text{year}} \\
 & + \text{effect}_{\text{age at calving} \times \text{region} \times \text{year}} \\
 & + \text{effect}_{\text{calving year} \times \text{calving month} \times \text{region} \times \text{year}} \quad \text{fixed env effect} \\
 & + \text{effect}_{\text{dry period length} \times \text{region} \times \text{year}} \\
 & + b_i \quad \text{permanent env effect} \\
 & + a_i \quad \text{genetic effect} \\
 & + e_i) \quad \text{residual effect} \\
 & \times e^{\gamma_h/2} \quad \text{heterogeneous variance adjustment factor}
 \end{aligned}$$

where we have:

- **fixed env effect**: fixed effect associated with different combinations of environmental factors.
- **permanent env effect**: random effect accounting for the permanent environmental effect specific to cow i .
- **genetic effect**: additive genetic effect of cow i such that $\text{Var}(a) = \mathbf{G}\sigma_a^2$, where \mathbf{G} is the genomic relationship matrix and σ_a^2 is the additive genetic variance.
- **residual effect**: random residual error term for cow i such that $\text{Var}(e) = \mathbf{I}\sigma_e^2$, where σ_e^2 is the residual variance.
- **heterogeneous variance adjustment factor**: γ_h is the parameter modeling heterogeneity in phenotypic variance, decided by [herd \times year] and [region \times year] [**heterogeneous variance**]. The factor $e^{\gamma_h/2}$ is applied to standardize variance across environments.

In our study, the estimated additive genetic variance σ_a^2 is 8.84, the variance of the permanent environmental effect is 3.54, and the residual variance σ_e^2 is 5.3.

Therefore, the heritability h^2 of the fat content trait is $8.84/(8.84+3.54+5.3) = 0.5$.

We first adjust each performance record j of cow i to account for heterogeneous variance, fixed environmental effects, and the permanent environmental effect. The adjusted record, denoted $y_{i,j}^{\text{adjusted}}$, is given by

$$y_{i,j}^{\text{adjusted}} = y_{i,j,h} \times e^{-\gamma_h/2} - \text{fixed env effect} - \text{permanent env effect}$$

Finally, the yield deviation YD_i for cow i is computed as a weighted average of all its adjusted records, given by

$$YD_i = \frac{\sum_{j=1}^3 w_{i,j} \times y_{i,j}^{\text{adjusted}}}{\sum_{j=1}^3 w_{i,j}},$$

where $w_{i,j}$ reflects the amount of information contained in the yield deviation, accounting for the number of elementary records per cow, the size of the environmental effect groups, the repeatability, and the heritability of the trait.

5 Neural Network Architecture and Hyperparameter Selection for All Chromosomes of the Cow Dataset

5.1 VAE

Model Architecture

- **Input dimension:** 150,483 features, corresponding to one-hot encoded genotypes for 50,161 SNPs.
- **Encoder:**
 - Fully connected layers: $150,483 \rightarrow 4,096 \rightarrow 2,048 \rightarrow 1,024$
 - One residual block at 1,024 units (2 linear layers with skip connection)
 - Linear projection to 512, followed by two separate linear heads to infer:
 - * Mean vector $\mu \in \mathbf{R}^{256}$
 - * Log-variance vector $\log \sigma^2 \in \mathbf{R}^{256}$
 - All layers are followed by Batch Normalization to stabilize training and LeakyReLU activation (slope = 0.05) to avoid neuron inactivation.
- **Latent space:** 256-dimensional latent vector
- **Decoder:**

- Fully connected layers: $256 \rightarrow 512 \rightarrow 1,024$
- One residual block at 1,024 units (same structure as encoder)
- Continuation: $1,024 \rightarrow 2,048 \rightarrow 4,096 \rightarrow 150,483$ (The reverse one-hot decoding is handled during the loss computation, not as an explicit output layer)
- All hidden layers include Batch Normalization and LeakyReLU (slope = 0.05)

Training Hyperparameters

- **Batch size:** 2,048
- **Optimizer:** Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$
- **Learning rate:** 5×10^{-4}

5.2 GAN

Model Architecture

• Generator

- **Input:** 256-dimensional latent vector
- **Initial transformation:**
 - * Linear layer: $256 \rightarrow 1,045$
 - * Batch Normalization + LeakyReLU
- **Residual blocks:** Two blocks with intermediate expansion:
 - * Block 1: $1,045 \rightarrow 1,045$ (residual) $\rightarrow 2,090$
 - * Block 2: $2,090 \rightarrow 2,090$ (residual) $\rightarrow 4,180$
 - * Each residual block consists of two fully connected layers with skip connection, Batch Normalization, and LeakyReLU
- **Final projection:**
 - * Linear: $4,180 \rightarrow 150,483$ (matches the one-hot encoded genotype size)
 - * Reshape to [batch size, 50161, 3]
 - * Apply Gumbel-Softmax to obtain discrete-like output while maintaining differentiability

• Discriminator

- **Input:** 150,483-dimensional vector
- **Initial transformation:**
 - * Linear layer: $150,483 \rightarrow 4,180$
 - * LeakyReLU activation

- **Residual blocks:** Two blocks with progressive contraction:
 - * Block 1: $4,180 \rightarrow 4,180$ (residual) $\rightarrow 2,090$
 - * Block 2: $2,090 \rightarrow 2,090$ (residual) $\rightarrow 1,045$
 - * Each residual block includes two linear layers and LeakyReLU activations
- **Final layer:**
 - * Linear: $1,045 \rightarrow 1$
 - * Sigmoid activation for binary classification.
- **Activation functions:** All hidden layers use LeakyReLU (negative slope = 0.05)
- **Normalization:** Batch Normalization is used in the generator but omitted in the discriminator

Training Hyperparameters

- **Batch size:** 128
- **Optimizer:** Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$ for both generator and discriminator
- **Learning rate:** 1×10^{-4} for both generator and discriminator
- **Gumbel-Softmax temperature:** Linearly annealed from 1.0 to 0.1 during training

5.3 WGAN

Here we present the conditional version, where phenotype information is provided during training. For the unconditional setting, it suffices to remove the phenotype-related tensors and adjust the tensor shapes accordingly.

Model Architecture

- **Generator:**
 - **Input:** A 260-dimensional vector composed of:
 - * Latent noise vector z of dimension 256
 - * Phenotype conditioning vector, repeated and concatenated of dimension 4
 - **Initial Layer:**
 - * Linear: $260 \rightarrow 1,045$
 - * Batch Normalization + LeakyReLU
 - **Residual Blocks:** Two sequential residual blocks:

- * **Block 1:**
 - ResNetBlock: A phenotype vector of dimension 4 is first injected, resulting in an input dimension of 1,049. The block consists of two linear layers with Batch Normalization and a skip connection, maintaining the dimensionality at 1,049.
 - Linear: $1,049 \rightarrow 2,090 + \text{LeakyReLU}$
- * **Block 2:**
 - ResNetBlock: Similarly, a phenotype vector of dimension 4 is injected, resulting in a dimension of 2,094. The block maintains this dimensionality through two linear layers with Batch Normalization and a skip connection.
 - Linear: $2,094 \rightarrow 4,180 + \text{LeakyReLU}$
- **Final projection:**
 - * Linear: inject phenotype vector of dimension 4 $\rightarrow 4,184 \rightarrow 150,483$
 - * Reshape to [batch size, 50161, 3]
 - * Apply Gumbel-Softmax
- **Critic:**
 - **Input:** A flattened genotype sequence of dimension 150,483 is concatenated with a phenotype vector of dimension 4, resulting in a total input dimension of 150,487.
 - **Initial Layer:**
 - * Linear: $150,487 \rightarrow 4,180$
 - * LeakyReLU
 - **Residual Blocks:**
 - * **Block 1:**
 - ResNetBlock: inject phenotype vector of dimension 4 $\rightarrow 4,184 \rightarrow 4,184$
 - Linear: $4,184 \rightarrow 2,090 + \text{LeakyReLU}$
 - * **Block 2:**
 - ResNetBlock: inject phenotype vector of dimension 4 $\rightarrow 2,094 \rightarrow 2,094$
 - Linear: $2,094 \rightarrow 1,045 + \text{LeakyReLU}$
 - **Final output:**
 - * Linear: $1,045 \rightarrow 1$ (critic score)
- **Activation functions:** All hidden layers use LeakyReLU (negative slope = 0.05)
- **Normalization:** Batch Normalization is used in the generator but omitted in the critic

Training Hyperparameters

- **Batch size:** 128
- **Optimizer:** Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.9$ for both generator and critic
- **Learning rate:** 1×10^{-4} for both generator and critic
- **Critic updates per generator update:** 5
- **Gradient penalty coefficient (λ):** 10
- **Gumbel-Softmax temperature:** Linearly annealed from 1.0 to 0.1 during training

5.4 DM

Similarly, we provide the conditional version here. To switch to the unconditional setting, simply remove all phenotype-related tensors and adjust the tensor shapes accordingly.

Model Architecture

- **Input structure:**
 - The input tensor is a concatenation of:
 - * Noisy data vector $x \in \mathbf{R}^{4819}$ (corresponding to the PCA-latent genotype data)
 - * Time embedding $t_{\text{emb}} \in \mathbf{R}^{256}$
 - * Phenotype embedding $pheno_{\text{emb}} \in \mathbf{R}^{64}$
 - Total input dimension: 5,139
- **Time embedding:**
 - Uses a sinusoidal positional encoding, similar to that in the original DDPM implementation.
 - Embedding dimension is set to 256
- **Phenotype embedding:**
 - A continuous label (scalar) is projected to a higher-dimensional space using a linear layer
 - Embedding dimension is set to 64
- **Noise predictor architecture:**
 - **fc1:** Input layer mapping from 5,139 \rightarrow 8,192

- After the first hidden layer, time and phenotype embeddings are re-injected ($8192 + 256 + 64 = 8512$)
- **fc2:** $8,512 \rightarrow 8,192$
- **fc3:** $8,192 \rightarrow 6,144$
- **out:** Final output layer from $6,144 \rightarrow 4,819$
- **Residual connection:**
 - * **res:** A skip connection projects the input via a linear layer: $5,139 \rightarrow 4,819$
 - * The final output is computed as **out** + **res**
- **Normalization:** Layer Normalization is applied after each internal fully connected layer
- **Activation:** ReLU is used after each normalization layer

Training Hyperparameters

- **Batch size:** 4,086
- **Diffusion process:**
 - Number of diffusion steps: 1,500
 - β schedule: linear
- **Time Sampling Strategy:** Antithetic sampling
- **Optimizer:**
 - Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$
 - Learning rate: 3×10^{-4}
 - Learning rate scheduler: Cosine Annealing
 - Minimum learning rate: 3×10^{-6}
 - Warm-up:
 - * Strategy: Linear warm-up
 - * Period: first 1,000 steps

6 Impact of SNP Dependence on the Difficulty of Generative Modeling

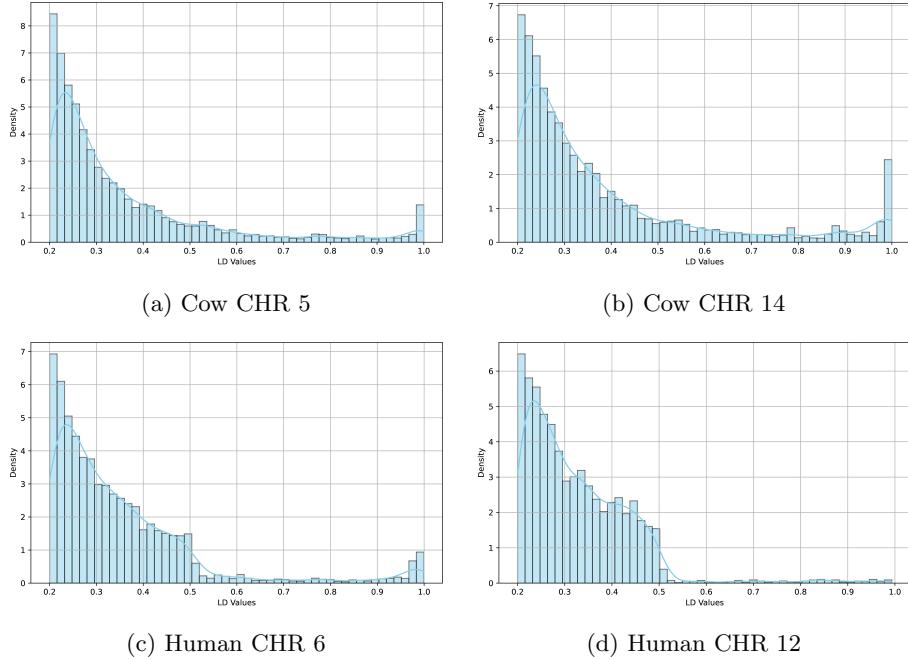


Figure 4: Distribution of LD values across different datasets, focusing on values ≥ 0.20 . In Cow dataset (a and b), SNPs exhibit stronger correlations compared to the Human dataset (c and d). In Human dataset, Chromosome 6 (c) contains more high-LD SNP pairs than Chromosome 12 (d). This makes it easier for generative models to learn Chromosome 6, despite its larger dimension compared to Chromosome 12.

7 Limitation of AA Score as a Robust Metric

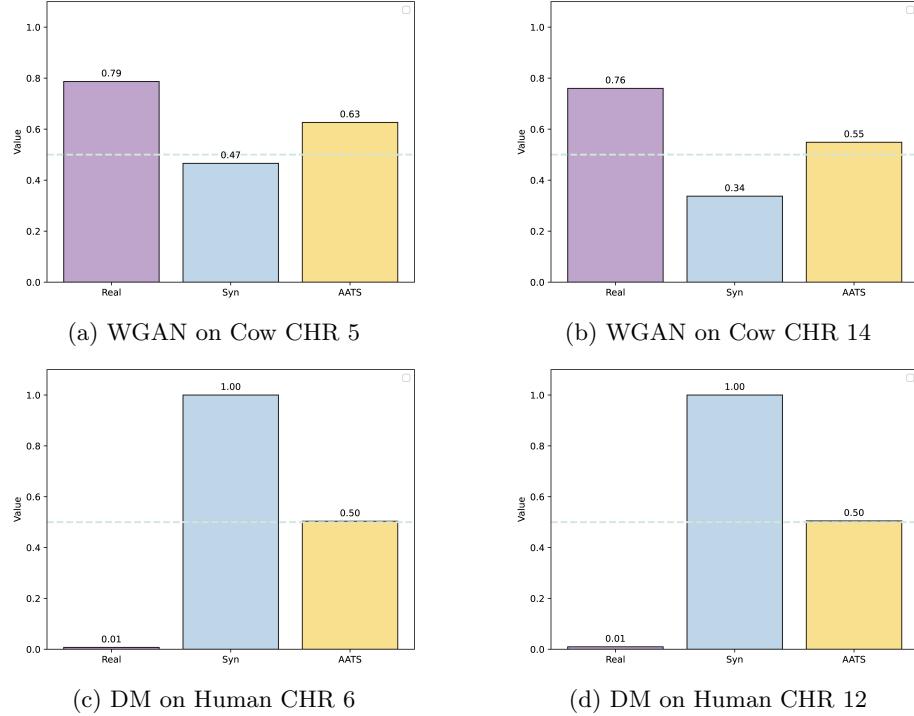


Figure 5: A detailed investigation of the AA score: (a) and (b) show cases where the AA score serves as a good evaluation metric, with both AA_{real} and AA_{syn} yielding good scores, resulting in an AA score around 0.50. (c) and (d) depict an anomalous scenario where $AA_{real} \approx 0$ and $AA_{syn} \approx 1$, yet the AA score remains around 0.50, highlighting a limitation of using AA as a metric.

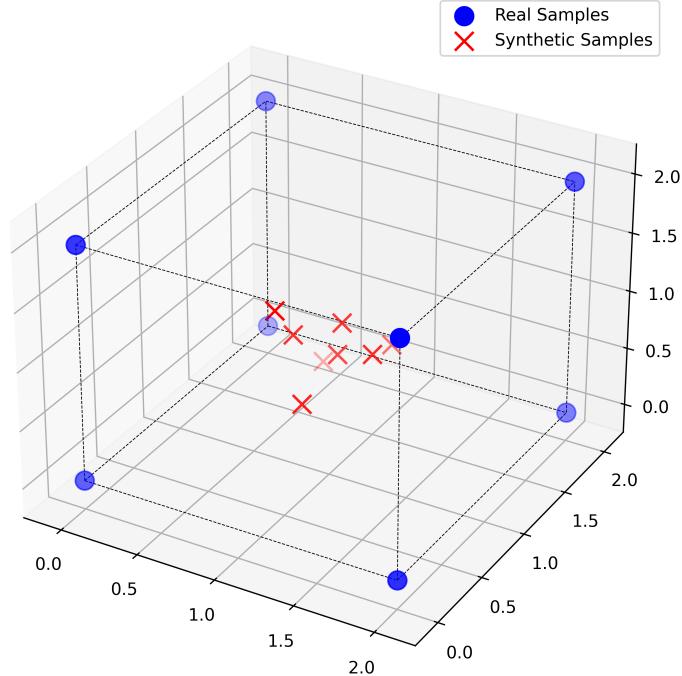


Figure 6: Geometric illustration of a scenario where $AA_{real} = 0$ and $AA_{syn} = 1$: Eight real samples are positioned at the corners of a cube, with each pair of real samples' distance equal to 2 ($d_{RR} = 2$). Eight synthetic samples are tightly clustered at the center of the cube [1, 1, 1]. In this setup, each real sample is closer to a synthetic sample than to any other real samples ($d_{RS} < \sqrt{3} < d_{RR}$), yielding $AA_{real} = 0$. Conversely, each synthetic sample is closest to another synthetic sample in the cluster, yielding $AA_{syn} = 1$. If we relate this scenario to the precision and recall metrics with $\forall k$, we obtain a precision of 1 because every synthetic sample falls within the support of a real sample. However, the recall is 0, as none of the real samples fall within the support of any synthetic sample.