Split-Transformer Impute (STI): A Transformer Framework for Genotype Imputation

Mohammad Erfan Mowlaei[1], Chong Li[1], Oveis Jamialahmadi[2], Raquel Dias[3], Junjie Chen[4], Benyamin Jamialahmadi[5], Timothy Richard Rebbeck[6,7], Vincenzo Carnevale[8,9], Sudhir Kumar[1,8,10], Xinghua Shi[1,8*]

[1*]Computer & Information Sciences, Temple University, 1925 N. 12th Street, Philadelphia, 19122, PA, USA.
[2]Department of Molecular and Clinical Medicine, Institute of Medicine, Sahlgrenska Academy, Wallenberg Laboratory, University of Gothenburg, Gothenburg, Sweden.
[3]Department of Microbiology and Cell Science, University of Florida, 1355 Museum Dr, Gainesville, 32603, FL, USA.
[4]Computer Science and Technology, Harbin Institute of Technology, Shenzhen University Town, Shenzhen, 518055, Guangdong, China.
[5]David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, N2L 3G1, ON, CA.
[6]Division of Population Sciences, Dana-Farber Cancer Institute, 450 Brookline Ave, Boston, 02215, MA, USA.
[7]Department of Epidemiology, Harvard T. H. Chan School of Public Health, 677 Huntington Ave, Boston, 02115, MA, USA.
[8]Institute for Genomics and Evolutionary Medicine, Temple University, 1925 N. 12th Street, Philadelphia, 19122, PA, USA.
[9]Institute for Computational Molecular Science, Temple University, 1925 N. 12th Street, Philadelphia, 19122, PA, USA.
[10]Department of Biology, Temple University, 1925 N. 12th Street, Philadelphia, 19122, PA, USA.

*Corresponding author(s). E-mail(s): mindyshi@temple.edu;
Contributing authors: mohammad.erfan.mowlaei@temple.edu; chong.li0001@temple.edu; oveis.jamialahmadi@wlab.gu.se; raquel.dias@ufl.edu; junjiechen@hit.edu.cn; B2jamial@uwaterloo.ca; timothy_rebbeck@dfci.harvard.edu; vincenzo.carnevale@temple.edu; s.kumar@temple.edu;

Abstract

Motivation: Despite recent advances in sequencing technologies, genome-scale datasets continue to have missing bases and genomic segments. Such incomplete datasets can undermine downstream analyses, such as disease risk prediction and association studies. Consequently, the imputation of missing information is a common pre-processing step for which many methodologies have been developed. However, the imputation of genotypes of certain genomic regions and variants, including large structural variants, remains a challenging problem.

Results: Here, we present a transformer-based deep learning framework, called a split-transformer impute (STI) model, for accurate genome-scale genotype imputation. Empowered by the attention-based transformer model, STI can be trained for any collection of genomes automatically using self-supervision. STI handles multi-allelic genotypes naturally, unlike other models that need special treatments. STI models automatically learned genome-wide patterns of linkage disequilibrium (LD), evidenced by much higher imputation accuracy in high LD regions. Also, STI models trained through sporadic masking for self-supervision performed well in imputing systematically missing information. Our imputation results on the human 1000 Genomes Project show that STI can achieve high imputation accuracy, comparable to the state-of-the-art genotype imputation methods, with the additional capability to impute multi-allelic structural variants and other types of genetic variants. Moreover, STI showed excellent performance without needing any special presuppositions about the patterns in the underlying data when applied to a collection of yeast genomes, pointing to easy adaptability and application of STI to impute missing genotypes in any species.

Keywords: Genotype, Structural variation, Imputation, Deep learning, Transformer

1 Introduction

Genetic and genomic studies, such as linkage analysis, genome-wide association study (GWAS), and polygenic risk score (PRS) estimation, enable us to dissect the genetic architecture of complex traits and diseases [1]. In recent years, whole-genome sequencing (WGS) platforms and techniques have substantially improved and become increasingly cost-effective, resulting in the accumulation of large collections of genotypes and deeper insights into the genetic architecture of various traits and diseases.

Although the resolution of genotyping has steadily improved over time, genotype data still contain many missing values and untyped loci [2]. The missing data may decrease statistical power in disease association studies and causal variant discovery [3–5]. Causes of missing genotypes include the difficulty in sequencing rare alleles [6–8], failure of experimental assays, genotype calling errors, and differences in densities and properties of genotyping platforms [3]. As such, genotype missingness, as depicted in Figure 1, can be classified into two distinct categories: sporadic missingness, where for each site/segment, some values could be absent, and systematic missingness, in which some genomic loci or segments are not genotyped. These challenges in handling missing data are further compounded when considering different types of genetic variation in addition to Single Nucleotide Variants (SNVs). Compared with SNVs, Structural Variations (SVs) pose greater challenges in genotype calling and imputation due to their increased complexity, limitations of current sequencing technologies, extensive allelic diversity, and their variable frequencies within populations [9, 10]. Moreover, SVs can have a more significant impact on genetic diseases than SNVs, so their accurate imputation can lead to enhancements in disease association studies [11].

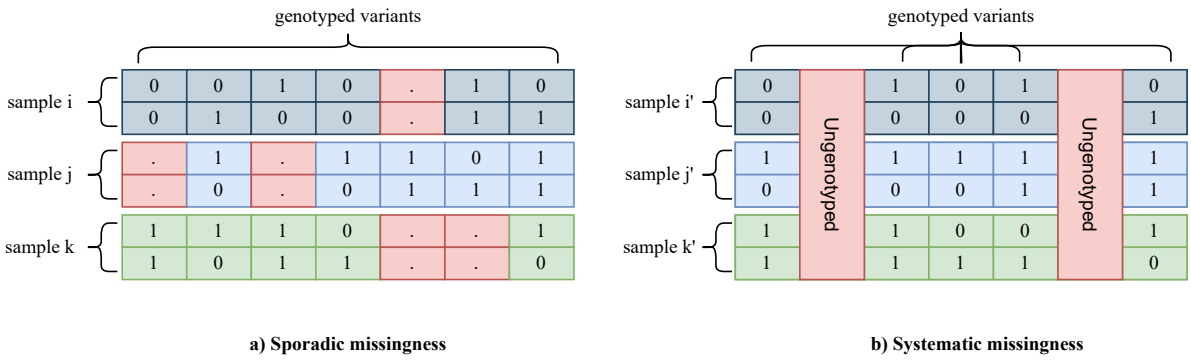

Fig. 1 Missing genotype classification. *a. Sporadic missingness:* This category is commonly associated with methods in genotype calling and assay failures. *b. Systematic missingness:* Differences in sequencing resolution are common causes of this type of missingness.

Consequently, there is a common need for reliable imputation of genotypes using computational methods. Imputation is the process of inferring missing values in the data based on the information already present in the dataset, such as the density and distribution of bases and structural variants within and among sequences in the dataset. Imputation of missing data in genomics needs specialized methods because genomic information is inherently different from data in many other domains, such as vision or natural language processing. Curse-of-dimensionality, linear and non-linear correlations among the variants [12], and shared segments of sequences due to common descent are among the unique characteristics of genotype data. Furthermore, given the multifaceted nature of genotype imputation, it is well-recognized that no single method can serve as a universal solution. Consequently, it is a common practice in the field to utilize multiple imputation tools for a particular study.

Widely-used imputation methods often require a reference panel of genome sequences to impute missing information in sequences, assuming that the missing information comes from the same ancestry patterns as those in the reference panel [13]. These methods utilize Hidden Markov Models (HMMs), graphical models, and haplotype-cluster algorithms to impute missing values [14]. For example, Minimac4 [15], the most recent version of MACH [16], uses an HMM. For each individual, Minimac4 updates the phase iteratively in both directions based on haplotypes in the reference panel and neighboring loci in the individual. It splits sequences into overlapping chunks in order to reduce memory consumption and make the model scalable. Similarly, Shapeit5 [17], IMPUTE2 [18] and BEAGLE [19] also employ HMMs to perform imputation. The Haplotype-clustering algorithm is utilized in the fastPHASE [20] in order to cluster haplotypes in an SNV-wise manner and impute missing values per locus. GLIMPSE2 [21] uses a HMM in order to genotype low-coverage whole-genome sequencing (WGS) data.

Deep Learning (DL) methods have been recently introduced for genomic imputations. Sparse Convolutional Denoising autoencoder (SCDA) is used in [14] to impute missing data in the Human Leukocyte Antigen (HLA) region on chromosome 6 and yeast [22] genotypes. In [3] an improvement in SCDA training is proposed to improve the performance. Similarly, [23] used autoencoders on identified linkage disequilibrium (LD) blocks as well as focal loss to improve the performance. RNN-IMP [24] utilizes recurrent neural networks (RNNs) and augments the samples using recombination and mutation in order to impute systematic missingness in genotype data. GRUD [25] utilizes RNNs in an adversarial training schema. DEEP*HLA [26] uses a convolutional neural network to perform imputation on pre-phased genotypes at the gene level. Inspired by DEEP*HLA, HLARIMNT [27] uses transformers for the same task.

Though the overall performance of existing imputation methods for genomic data is generally good, most of them cannot directly handle multi-allelic variants. Also, their performances have not been evaluated for imputing SVs to the best of our knowledge. Also, the training of DL models [3, 14, 23, 26] is generally slow, and they need many more samples to perform as well as methods based on HMMs, such as Minimac4 [15] and Beagle5.4 [19]. Although RNN-IMP and GRUD [24, 25] addressed this performance disparity, these methods require retraining when the variant sets in the target are different from those in the training set. They are also not designed to handle sporadic missingness (Figure 1.a). Finally, the rest of the existing DL methods rely on convolutional neural networks (CNNs), which excel at exploiting local patterns but do not exhibit a robust mechanism to effectively capture pairwise correlations among local and distant markers simultaneously, such as the presence of LD blocks in genotypes.

An effective solution to this is the attention mechanism in the transformer architecture, capable of capturing local and distal interactions in genomes[28]. The attention mechanism in DL mimics visual attention to focus on specific parts of pictures [29, 30] by calculating importance scores among genomic loci. Therefore, attention can capture global interactions amongst markers. Transformers utilize multi-head attention to capture intricate and multi-level interactions among the variants. AlphaFold2 [31] and ESMFold [32] are successful examples of transformers in biological sequence analysis.

In this article, we present a novel genotype imputation model, STI, based on the attention mechanisms in a transformer framework. Our model utilizes attention to capture patterns among SNVs and SVs in the genome collections analyzed. We found STI to achieve high imputation accuracy at a modest memory consumption cost, achieved by dividing the data into chunks (following [15]) that enables efficient application of STI to long sequences. Furthermore, STI needs to be trained only once, unlike other DL models, following that the imputation times in STI are faster than classical methods (Table 11 in the Supplement). In brief, our study makes the following key contributions.

- We propose a DL transformer framework termed STI, designed to specifically address the genotype imputation problem.
- STI imputation does not need a standard reference panel, which makes it more generally applicable to various data formats.
- STI excels at SV imputation, where the variants harbor a higher degree of complexity while achieving comparable performance to competing imputation models for SNV imputation.
- We analyze the effect of different masking rates on building better imputation models and explain the reasons for the STI improvements.

2 Results

2.1 Overview of the study

In this section, first, we present the results of our empirical study to find the optimal masking percentage (masking rate, MaskR) for STI training in order to eliminate the need for building imputation models for specific missing rate (MissR) in the target dataset, which is often required by some other machine learning approaches [3, 14]. After that, we present results for sporadic missingness imputation on the yeast dataset, SVs in human chromosome 22, and extensive SVs dataset (see Subsection 4.1 in Methods for the dataset details). We benchmarked STI's performance against classical imputation methods (Beagle5.4 and Minimac 4.1.4), deep learning models (SCDA, AE, DEEP*HLA), and a variant of STI that uses no embedding (STI-NE). The details of STI architecture and aforementioned methods are discussed in Subsections 4.3 and 4.4, respectively.

2.2 Optimal masking rate analysis

For this analysis, we used the HLA region on chromosome 6 from the human 1000 Genomes Project and performed a 3-fold cross-validation on the data. The aim was to examine the relationship of MaskR for the training set with varying MissR in target sets. The results are presented in Figure 2 in which the results on the left/right column belong to the validation/test set, respectively.

Figures 2 *a & b* show that the performances of STI models trained using MaskR of 0.5 and 0.7 were high for imputations in which MissRs were up to 0.5. Therefore, a single STI model, trained with a MaskR of 0.5, could be used for a variety of research datasets as long as the MissR is less than 0.5. However, STI models trained with MaskR > 0.5 is needed for reliable imputations when the target datasets have more than 50 percent missing variants. Therefore, we recommend STI models trained with a MaskR of 0.5 for imputing sporadically missing variants and a higher MaskR (up to 0.8, as indicated by our empirical studies data) for other datasets.

Figures 2 *a & b* also show that when the target MissR is sufficiently low, the performance gap of the imputation models is not discernible. The performance gap becomes evident with a MissR of 0.2 or higher. The underlying cause of this observation is that when the MissR is extremely low, a sufficient number of variants in LD with the target variant are readily available, making predictions less challenging for all the models. Conversely, a large MissRs means that the amount of information from LD blocks diminishes, presenting a greater challenge to the imputation model.

Figures 2 *c & d* show that, generally, STI models trained with lower MaskR will produce poor performance for imputing missing SNVs located in regions with high LD. For instance, variants in regions with LD $= 0.01$ have the lowest accuracy for all the masking percentages. Additionally, these results indicate that it is easier to predict missing data in high LD regions compared to low LD regions, which aligns well with biological expectations that low LD regions do not benefit from additional information (LD) available for better imputation of high LD regions. These trends suggest that the use of a low MaskR prevents the model from learning LD patterns, resulting in a worse performance. In other words, the model training needs to effectively disturb the LD blocks (and other latent patterns among variants) to capture direct and indirect correlations and haplotypes. Consequently, MaskR of 0.5 and higher provides robust results across a large range of target MissR values.

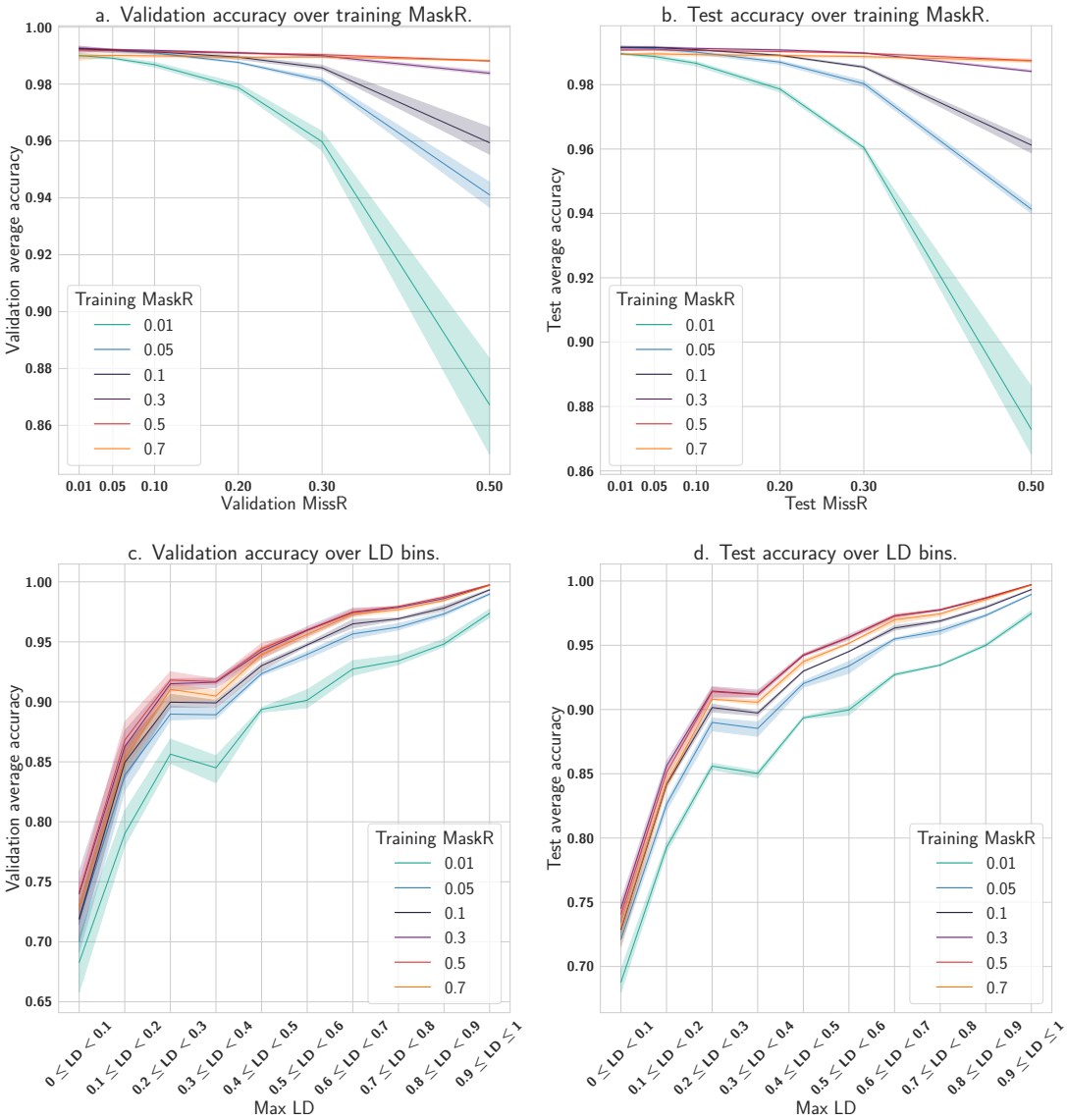

Fig. 2 Average accuracy over 3-fold cross-validation for validation and test sets in the HLA dataset using different MaskR values during training. *a.* and *b.* A breakdown of average accuracy for various MissRs of validation/test set when the model is trained using different MaskR values. The patterns show that a model trained using a higher MaskR is more robust across different target MissRs. *c.* and *d.* Average accuracy for validation/test sets over 3 folds and MissRs of 0.01, 0.05, 0.1, 0.2, 0.3, and 0.5 calculated for various LD bins. The trend suggests that a higher MaskR increases the performance across LD bins, which could be attributed to the impact of MaskR on STI to learn LD patterns comprehensively. When MaskR is low, STI imputations do not benefit from the LD patterns present and, thus, STI does not learn the majority of pairwise correlations (LD) among the variants. Consequently, STI is not able to infer the missing value using all possible information in the respective LD block of the target variant.

2.3 The relative performance of STI for sporadic missingness

For each dataset in this experiment, we performed a 3-fold cross-validation where missing values were introduced using fixed random seeds to ensure reproducibility of results across experiments and methods. The missing values were distributed randomly according to one of three strategies: uniformly, based on Minor Allele Frequency (MAF), or based on LD. These methods were chosen to ensure that missing values are representative of the data distribution in different biological aspects. Further details on these procedures can be found in the methods section. In all of the experiments, missing positions in the test sets were the same for all the methods.

The overall results for the yeast and chromosome 22 datasets are presented in Table 1. The numerical values in this table indicate the average of the metric values on the test sets in a 3-fold cross-validation.

We used maximum LD bins and/or MAF bins (Figure 4 b, and c) to distribute missing positions in the datasets extracted from the human 1000 Genomes Project. If bins had too few positions (e.g., at a 0.01 MissR on chromosome 22 datasets), we excluded this MissR for the experiments related to these datasets. We used a consistent approach to introduce missing values in chromosomes 6, 10, 16, and 22 based on LD distributions and a single test MissR of 0.2. In this experiment, we focused on comparing Minimac, Beagle, and STI, because they were identified as the top performers from classical and DL methods, respectively, in prior experiments. We employed 3-fold cross-validation for both methods, training and imputing each chromosome separately. R^2 was calculated for each variant, and the results were averaged over fold, chromosome, and SV type. Figure 3 presents the experimental results for the extensive structural variation datasets where the top plot shows the improvement that STI provides compared to the best of other methods for each SV type, and is calculated as follows:

$$Improvement(\%) = \frac{R^2_{STI} - R^2_{Best}}{R^2_{Best}} * 100$$

Yeast dataset: Missing positions in samples were selected randomly, as the LD analysis showed that the maximum LD for all the SNVs was high in the $[0.8, 1.0]$ range. As mentioned, Minimac4.1.4 and Beagle5.4 cannot be used to impute variants of the yeast dataset due to the lack of a reference panel. However, STI could be applied and outperformed other methods, achieving a minimum average imputation accuracy of 99.86%. Overall, all the applicable models performed well on the yeast dataset, which we attribute to the presence of high LD among SNVs in this dataset.

Deletions in chromosome 22: For this dataset, we introduced missing positions proportional to the maximum-LD/MAF distribution Figure 4.b. Overall, STI emerged as the best or the second-best model for imputation across all the metrics. STI was more accurate than others for LD/MAF missingness distribution schema. Furthermore, SCDA+ demonstrates a substantial performance advantage over AE in terms of IQS and R^2 in the majority of the cases. **Table 8** in the Supplement shows the accuracy trends for different maximum LD values for this dataset when missing values are distributed proportional to variant density in maximum LD bins. Minimac4.1.4 and Beagle5.4 were less accurate for SNVs with lower maximum LD compared to AE, SCDA+, STI-NE, and STI. Since HMMs and graphical models rely on conditional probabilities, we suggest that they would perform relatively weak due to a low correlation between the events (states).

All SVs in chromosome 22: Similar to the previous dataset, missing positions were distributed among SVs based on maximum-LD/MAF (Figure 4.c). Despite having a reference panel, Minimac4.1.4 cannot be used for some missing variants for this dataset because it can only handle bi-allelic events. Furthermore, IQS is not well-defined for multi-allelic events.

Table 1 shows that STI outperforms all other methods on average accuracy and F1-score. STI performance in terms of R^2 is much better than the competing methods at high MissRs. R^2 considers the correlation among genotypes encoded as categorical values. As such, depending on the difference in encoded values for the predicted and the ground truth genotypes, the penalty can be severe. For example, if 0|0, 0|1, and 1|1 are encoded as 0, 1, and 2 in genotypes and the ground truth for a given genotype is 0|0, the model is punished moderately(severely) for predicting 0|1(1|1). Additionally, SCDA+ outperforms AE in most comparisons, indicating the effectiveness of our proposed training procedure.

Extensive structural variation datasets: In this experiment, we focus on R^2 between the predicted and ground truth genotypes as R^2 was the most discriminating metric for comparing the performance in imputing SVs. For estimating R^2, predictions are converted into categorical values, e.g., 0|0, 0|1, and 1|1 are encoded as 0, 1, and 2. Any discrepancy between the model's prediction and the ground truth leads to a substantial penalty on the correlation, enabling us to see differences more clearly. We found STI consistently outperforms Beagle5.4 and Minimac4 across various SV types, often by a noticeable margin. The underlying cause of this observation is the lack of high LD in this dataset (Figure 4) and fundamental differences between HMMs and the Transformer model. In HMMs, information propagation between two distant variants occurs sequentially through intermediate sites. However, this mechanism falters when the LD block is sparse, leading to reduced performance. In contrast, STI employs a direct variant-to-variant attention mechanism within each chunk without needing to model an intermediate site, which effectively mitigates the limitations posed by a weak LD. Furthermore, the multi-head attention mechanism equips STI to discern higher-order and complex patterns among variants, which appear to be

Table 1 Experimental results for imputing sporadic missingness averaged over 3-fold cross-validation using different MissRs. The numerical values in parentheses show the standard deviation. N/A values indicate that the model could not impute that specific dataset. For all the metrics, the higher the value, the better the models are performing in imputation. In these experiments, accuracy and f1-score calculations distinguish heterozygous alternative alleles by encoding them to distinct categorical values; however, for IQS and R^2 the encoding values remain identical. Bold values indicate the top result in each row. Beagle5.4 generally performs the best in terms of R^2 and IQS for bi-allelic variants, but STI outshines other methods in imputing all SVs (and multi-allelic variants). *Note:* H- in the first column indicates that the dataset is a human genome dataset.

			Method						
Dataset	Metric	MissR	AE	SCDA+	DEEP*HLA	Beagle	Minimac	STI-NE	STI
Yeast	Accuracy	0.01	99.8(3.9e-03)	99.8(7.2e-03)	99.8(6.1e-03)	N/A	N/A	99.8(6.9e-02)	**99.9(7.9e-03)**
		0.05	99.8(2.4e-03)	99.8(6.3e-04)	99.8(3.7e-03)	N/A	N/A	99.8(3.0e-02)	**99.9(5.6e-03)**
		0.10	99.8(4.8e-03)	99.8(3.2e-03)	99.8(2.9e-03)	N/A	N/A	99.8(2.4e-02)	**99.9(7.7e-03)**
		0.20	99.8(1.6e-03)	99.8(1.7e-03)	99.8(3.1e-03)	N/A	N/A	99.8(3.0e-02)	**99.9(4.9e-03)**
	F1-score	0.01	0.998(3.61e-05)	0.998(6.93e-05)	0.998(6.24e-05)	N/A	N/A	0.998(6.93e-04)	**0.999(8.08e-05)**
		0.05	0.998(2.52e-05)	0.998(5.77e-06)	0.998(3.51e-05)	N/A	N/A	0.998(3.02e-04)	**0.999(5.29e-05)**
		0.10	0.998(5.03e-05)	0.998(3.21e-05)	0.998(3.06e-05)	N/A	N/A	0.998(2.43e-04)	**0.999(7.94e-05)**
		0.20	0.998(1.53e-05)	0.998(1.73e-05)	0.998(3.21e-05)	N/A	N/A	0.998(3.04e-04)	**0.999(5.03e-05)**
	IQS	0.01	1.000(1.09e-05)	1.000(1.80e-05)	**1.000(6.52e-06)**	N/A	N/A	0.999(1.81e-03)	1.000(4.27e-05)
		0.05	**1.000(9.82e-06)**	1.000(1.56e-05)	1.000(1.19e-05)	N/A	N/A	0.999(8.91e-04)	1.000(3.76e-05)
		0.10	**1.000(1.87e-05)**	1.000(2.00e-05)	1.000(1.95e-05)	N/A	N/A	0.999(6.94e-04)	1.000(3.60e-05)
		0.20	**0.999(1.73e-05)**	0.999(2.01e-05)	0.999(2.67e-05)	N/A	N/A	0.999(7.64e-04)	0.999(3.11e-05)
	R^2	0.01	1.000(2.18e-05)	1.000(3.61e-05)	**1.000(1.31e-05)**	N/A	N/A	0.999(3.49e-03)	1.000(8.53e-05)
		0.05	**1.000(1.97e-05)**	1.000(3.12e-05)	1.000(2.37e-05)	N/A	N/A	1.000(1.72e-03)	1.000(7.50e-05)
		0.10	**1.000(3.74e-05)**	1.000(3.98e-05)	1.000(3.85e-05)	N/A	N/A	0.999(1.33e-03)	1.000(7.19e-05)
		0.20	**1.000(3.34e-05)**	1.000(3.90e-05)	1.000(5.21e-05)	N/A	N/A	0.999(1.47e-03)	1.000(6.23e-05)
H-Chr22 Del(LD)	Accuracy	0.05	96.9(1.0e+00)	97.0(1.1e+00)	92.5(5.8e+00)	96.1(8.9e-02)	96.2(6.1e-02)	97.8(9.7e-02)	**97.9(1.2e-01)**
		0.10	96.7(8.8e-01)	97.3(3.3e-01)	93.7(3.5e+00)	96.0(7.4e-02)	96.2(8.4e-02)	97.5(4.8e-02)	**97.6(5.2e-02)**
		0.20	96.2(6.8e-01)	97.1(6.7e-02)	94.3(2.6e+00)	96.1(4.5e-02)	96.3(5.5e-02)	97.2(3.2e-02)	**97.3(4.1e-02)**
	F1-score	0.05	0.960(2.03e-02)	0.968(1.00e-02)	0.922(2.46e-02)	0.952(8.30e-04)	0.955(1.09e-03)	0.976(1.45e-03)	**0.977(2.09e-03)**
		0.10	0.959(1.83e-02)	0.970(3.87e-03)	0.930(1.20e-02)	0.952(7.34e-04)	0.955(8.38e-04)	0.972(1.07e-03)	**0.973(1.62e-03)**
		0.20	0.947(1.61e-02)	0.966(1.28e-03)	0.934(7.46e-03)	0.952(7.28e-04)	0.955(4.53e-04)	0.967(5.21e-04)	**0.968(1.18e-03)**
	IQS	0.05	0.621(2.13e-01)	0.758(8.92e-02)	0.364(2.61e-02)	**0.975(6.57e-03)**	0.794(1.43e-02)	0.942(1.21e-02)	0.971(8.85e-03)
		0.10	0.671(2.52e-01)	0.782(5.67e-02)	0.354(4.44e-02)	**0.947(7.82e-03)**	0.776(1.41e-02)	0.919(8.20e-03)	0.946(7.80e-03)
		0.20	0.477(2.01e-01)	0.787(2.33e-02)	0.338(7.16e-02)	0.891(2.21e-02)	0.739(2.24e-02)	0.874(1.08e-02)	**0.895(1.54e-02)**
	R^2	0.05	0.254(2.21e-01)	0.415(5.02e-02)	0.079(1.37e-01)	**0.601(1.88e-02)**	0.418(1.19e-02)	0.567(2.73e-02)	0.596(1.85e-02)
		0.10	0.294(2.55e-01)	0.417(4.21e-02)	0.068(1.18e-01)	**0.564(1.29e-02)**	0.392(5.63e-03)	0.539(2.42e-02)	0.563(1.82e-02)
		0.20	0.100(1.74e-01)	0.391(1.66e-02)	0.049(8.45e-02)	0.495(1.77e-02)	0.347(1.65e-02)	0.486(1.91e-02)	**0.501(1.66e-02)**
H-Chr22 Del(MAF)	Accuracy	0.05	96.4(4.3e-02)	96.2(7.8e-01)	83.0(8.4e+00)	96.5(1.1e-01)	96.1(1.0e-01)	96.8(7.1e-02)	**96.9(1.1e-01)**
		0.10	96.6(8.4e-02)	96.7(3.0e-01)	90.0(3.2e+00)	96.5(1.1e-01)	96.1(1.5e-01)	96.8(1.4e-01)	**96.9(1.4e-01)**
		0.20	96.0(1.2e-01)	96.3(1.3e-01)	92.8(1.3e+00)	96.0(4.4e-02)	95.6(1.0e-01)	96.3(9.4e-02)	**96.4(9.9e-02)**
	F1-score	0.05	0.959(1.81e-03)	0.956(4.07e-03)	0.879(4.60e-02)	0.961(2.00e-03)	0.959(2.24e-03)	**0.961(5.88e-04)**	0.961(1.37e-03)
		0.10	0.959(3.41e-03)	0.958(2.06e-03)	0.918(1.53e-02)	0.960(1.01e-03)	0.960(1.70e-03)	0.960(1.55e-03)	**0.961(1.63e-03)**
		0.20	0.947(6.69e-03)	0.953(1.30e-03)	0.926(5.39e-03)	0.954(8.44e-04)	**0.955(1.39e-03)**	0.954(6.46e-04)	0.955(1.61e-03)
	IQS	0.05	0.865(1.62e-02)	0.736(1.28e-01)	0.476(1.78e-01)	**0.977(5.70e-03)**	0.603(2.36e-02)	0.960(2.30e-02)	0.961(5.66e-03)
		0.10	0.742(9.40e-02)	0.778(9.54e-02)	0.424(1.15e-01)	**0.952(9.15e-03)**	0.589(1.86e-02)	0.939(1.87e-02)	0.946(1.08e-02)
		0.20	0.501(2.26e-01)	0.789(2.07e-02)	0.374(4.93e-02)	**0.903(1.03e-02)**	0.563(1.80e-02)	0.887(1.51e-02)	0.901(1.26e-02)
	R^2	0.05	0.492(3.47e-02)	0.408(5.34e-02)	0.327(1.12e-01)	**0.602(2.41e-02)**	0.213(4.61e-03)	0.584(3.51e-02)	0.594(2.03e-02)
		0.10	0.363(7.72e-02)	0.421(3.01e-02)	0.291(7.68e-02)	**0.570(2.46e-02)**	0.199(4.32e-03)	0.557(2.82e-02)	0.568(2.28e-02)
		0.20	0.127(2.05e-01)	0.398(7.29e-03)	0.226(3.59e-02)	0.508(2.11e-02)	0.174(4.69e-03)	0.495(2.04e-02)	**0.508(2.04e-02)**
H-Chr22(LD)	Accuracy	0.05	95.1(8.7e-02)	94.2(2.2e+00)	95.3(1.5e-01)	95.3(9.8e-02)	N/A	95.6(1.2e-01)	**95.6(2.3e-01)**
		0.10	95.2(8.4e-02)	95.1(6.8e-01)	95.3(9.9e-02)	95.4(4.3e-02)	N/A	**95.6(5.6e-02)**	95.6(1.1e-01)
		0.20	95.2(2.1e-01)	95.5(1.6e-01)	95.3(7.4e-02)	95.4(7.2e-02)	N/A	95.6(2.5e-02)	**95.7(8.3e-02)**
	F1-score	0.05	0.935(6.57e-03)	0.936(1.36e-02)	0.937(3.33e-03)	0.943(3.18e-03)	N/A	0.944(1.86e-03)	**0.947(3.04e-03)**
		0.10	0.942(3.53e-04)	0.942(6.06e-03)	0.936(1.69e-03)	0.944(1.37e-04)	N/A	0.945(6.47e-04)	**0.947(8.47e-04)**
		0.20	0.935(8.38e-03)	0.943(2.66e-03)	0.935(1.16e-03)	0.943(8.28e-04)	N/A	0.945(9.05e-04)	**0.947(6.35e-05)**
	R^2	0.05	0.335(2.93e-01)	0.394(2.24e-01)	0.518(3.10e-02)	0.443(2.25e-02)	N/A	0.623(2.04e-02)	**0.627(1.84e-02)**
		0.10	0.455(1.73e-02)	0.415(1.80e-01)	0.496(2.87e-02)	0.418(2.11e-02)	N/A	0.585(1.95e-02)	**0.588(1.82e-02)**
		0.20	0.219(1.90e-01)	0.430(4.60e-02)	0.442(2.57e-02)	0.374(2.28e-02)	N/A	0.518(2.07e-02)	**0.523(2.06e-02)**
H-Chr22(MAF)	Accuracy	0.05	95.5(1.4e-01)	95.7(8.4e-02)	66.9(3.1e+01)	95.5(1.0e-01)	N/A	95.8(1.2e-02)	**96.0(9.4e-02)**
		0.10	94.8(2.7e-02)	95.1(3.2e-02)	82.8(1.5e+01)	94.9(1.3e-02)	N/A	95.1(5.6e-02)	**95.3(2.2e-02)**
		0.20	95.2(8.3e-02)	95.5(3.5e-02)	94.3(4.9e-01)	95.2(1.9e-02)	N/A	95.4(2.4e-02)	**95.6(1.6e-02)**
	F1-score	0.05	0.943(1.85e-03)	0.948(9.52e-04)	0.749(2.22e-01)	0.947(8.21e-04)	N/A	0.946(6.31e-04)	**0.950(7.80e-04)**
		0.10	0.937(4.41e-04)	0.942(6.32e-04)	0.864(8.25e-02)	0.940(7.34e-04)	N/A	0.939(7.37e-04)	**0.943(3.44e-04)**
		0.20	0.940(2.93e-03)	0.946(2.88e-04)	0.931(7.38e-03)	0.943(2.23e-04)	N/A	0.943(4.38e-04)	**0.947(7.96e-04)**
	R^2	0.05	0.460(8.40e-02)	0.510(1.17e-01)	0.068(6.14e-02)	0.446(2.22e-02)	N/A	0.612(3.58e-02)	**0.630(2.24e-02)**
		0.10	0.458(2.65e-02)	0.537(1.77e-02)	0.129(1.36e-01)	0.429(1.82e-02)	N/A	0.586(3.08e-02)	**0.600(2.13e-02)**
		0.20	0.287(9.88e-02)	0.483(1.42e-02)	0.211(1.80e-01)	0.383(2.73e-02)	N/A	0.521(2.81e-02)	**0.532(2.70e-02)**

crucial for better imputations in the absence of strong LD patterns. These capabilities highlight STI's superiority in managing SV imputation challenges where traditional HMM-based approaches may be suboptimal. This is particularly the case for duplications (DUP) and insertions (INS) where STI is able to attain a very high R^2 value. This observation matches our expectations since these two types of SVs are relatively challenging in genotype calling as well [33].

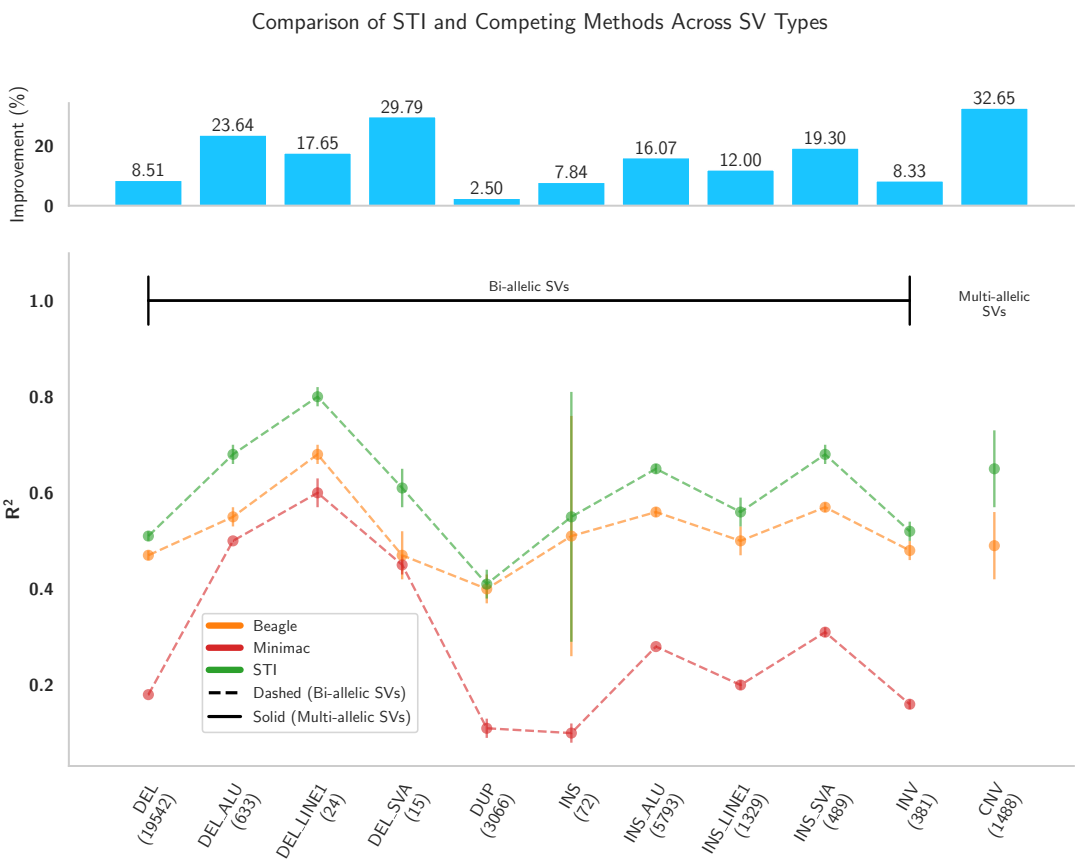

Fig. 3 Comparison of Beagle and STI across SV types. Average R^2 of ground-truth genotypes in the test sets and respective predictions over 3-fold cross-validations on chromosomes 6, 10, 16, and 20. The experiments are performed on each chromosome separately, and the results are averaged over chromosomes and folds. Vertical lines indicate standard deviations. The improvement plot shows R^2 score difference between STI and the best of other methods, normalized by the best R^2 scores for each SV type.

3 Discussion

More accurate genotype imputations will improve the performance of downstream functional and biomedical genomic studies. Scientists frequently need to employ multiple tools, adapted based on the degree of missingness and types of variants missing, within individual pipelines to carry out imputations. To address this problem, we have presented STI, a masked DL framework, which appears to be one of the first uses of transformer architecture. While STI is currently limited in few ways, we believe that it is a step towards developing a unified approach for successfully imputing missing values for a range of datasets, from small to large amount of missingness, as well as SNVs and SVs. We explored STI's performance for a range of masking rates (training) and missing rates (application), which revealed that a single STI model, trained with a masking rate of 0.5, could be applied for imputing SNVs and SVs. STI's performance in imputing SNVs and SVs was comparable to many other methods and approaches for SNVs and SVs found in low and high LD regions. That is, STI is capable of effectively capturing short and long-range correlations among SNVs/SVs.

STI also performed well in imputing values that were missing systematically (Figure 1.b and Supplementary Information; Ablation and Experimental Results sections). Furthermore, STI offers two additional advantages. First, it can be applied directly to resequencing datasets from any species because, unlike HMMs based on Li and Stephens model [34], a transformer model does not need hard-coded or parametric assumptions about underlying characteristics of the genomic data, such as mutation rate and density of LD blocks, and captures inherent patterns automatically. Secondly, an STI model needs to be trained once to impute sporadic and systematic missingness rapidly and accurately. Therefore, we expect the STI framework to spur the next generation of approaches to advance generalized and efficient

imputing, which could served by online imputation servers because of the low computational burden of imputation after training.

Also, STI can be extended by integrating secure and privacy-preserving mechanisms through homomorphic encryption and Tensorflow-compatible libraries such as tf-encrypted [35]) to provide secure and privacy-preserving genotype imputation [36, 37], which is imperative yet under-explored in current tools.

Currently, training an STI model is resource- and time-demanding. One of our development plans for STI is to address these issues. Moreover, STI is a transformer model, and transformers are known to require a large number of samples to achieve optimal performance. Therefore, we expect STI's performance to greatly improve even for SNVs and SVs in low LD regions as the number of samples is increased due to the release of resources such as those from TopMed and Gnomad4.0.

4 Methods

In this section, we first introduce the datasets we used in this study and discuss their characteristics. This is followed by the architectural design of STI and the procedure for model training.

4.1 Data

We used five datasets from two sequencing projects [10, 22] to fine-tune and benchmark STI against the baselines. The 1000 Genomes project datasets are pre-phased using Shapeit2. Thus, the phasing information of the test sets is propagated to the training sets, but the process is identical for all the methods so it will not bias the results in favor of any imputation method. Scikit-allel package [38] is employed to compute LD and MAF for the datasets. In the sporadic missingness experiment, we use 3-fold cross-validation to assess the performance of the methods. In a 3-fold cross-validation, each dataset is separated into three distinct partitions where there is no sample overlap. Each time, one of the partitions is used as the test set while the remaining two partitions are used for the training process. For the DL models, a validation set is selected from the training samples for early stopping. To ensure that the same training/validation/test set is used across different methods, we used fixed random seeds for splitting the data into folds and introducing missingness into the test sets.

We used three schemes to introduce sporadic missingness: random selection, MAF-distributed selection, and LD-distributed selection. For the latter two, we computed MAF and LD on the whole data, as depicted in Figure 4, and selected the missing SNVs/SVs for the test set evaluation proportional to those. That is, if 10% of the variants have an MAF in the range of [0.2, 0.3), we selected 10% of the missing values from these specific variants. This approach ensures that the missing data are imputed based on the distribution of MAF or LD in the data, providing a representative imputation strategy. Regardless of the scheme, we used fixed random seeds per sample to decide the missing genotypes. Therefore, the missing genotypes across the test samples are not identical. However, the coordinates of missing values are identical across the competing methods. The characteristics of the datasets we use in our experiments are as follows:

4.1.1 HLA dataset

This dataset contains human leukocyte antigen genotypes, covering a 3 Mbp region at chromosome 6p21.31 and sitting at a major histocompatibility complex (MHC) region. HLA region regulates the immune system in humans [39]. It is highly polymorphic and heterogeneous among individuals; i.e., it harbors various alleles, enabling the adaptive immune system to be fine-tuned [40]. In this study, we used the genotypes of this region, obtained from phase 3 of the 1000 Genomes Project [10], which contained 7161 unique genetic variants for 2504 individuals from five super-populations across the world: American (AMR), East Asian (EAS), European (EUR), South Asian (SAS), and African (AFR). The majority of SNVs in this dataset exhibit maximum LD values in the range of $[0.9, 1.0]$. We used this dataset for our masking study and fine-tuning the hyper-parameters of DEEP*HLA, SCDA, AE, and STI.

4.1.2 Yeast dataset

The second dataset is the comprehensively assayed yeast dataset [22], representing a simple genetic background and high correlation among genotypes. This dataset contains 4390 genotyped profiles for

28,220 genetic variants. The samples were obtained by sequencing crosses between two strains of yeast, namely an isolate from a vineyard (RM) and a popular laboratory strain (BY). In the original dataset, the data is encoded as -1/1 for BY/RM, which are mapped to 0/1 in our code, respectively, before one-hot encoding.

4.1.3 Chromosome 22 datasets

We used structural variation data from the 1000 Genomes Project in two settings. In the first, we only selected deletions (DEL), excluding ALU/SVA/LINE1 deletions, among all SVs. This resulted in 573 positions harboring bi-allelic events in the dataset. In the second, a total of 848 SVs including, but not limited to deletions, insertions, duplications, inversions (INV), and copy number variations (CNV) in chromosome 22 are selected. As shown in Figure 4 b & c, the majority of SVs in chromosome 22 exhibit a low LD, rendering these datasets challenging for imputation compared to SNVs. According to Figure 4.d, deletions cover a wide range of LD among them and other SVs, making them a good target for a separate bi-allelic dataset.

4.1.4 Extensive structural variation datasets

In the concluding experiment, we undertook a thorough investigation of SV imputation using the human 1000 Genomes Project, selecting 4187, 3126, 2062, and 1569 SVs located in chromosomes 6, 10, 16, and 20, respectively. This selection strategy was informed by the aim to encompass chromosomes of different lengths, providing a representative cross-section of the genome. This diverse chromosome selection allows for a broader understanding of the genomic distribution and characteristics of SVs, facilitating a more nuanced analysis of their presence and impact across different regions of the human genome. These SVs include deletions, duplications, insertions, inversions, and copy number variations. Among these SVs, 469 of them are multi-allelic (CNVs). For each model, we train on and impute each chromosome separately, and take the average of the results over folds, chromosomes, and SV type. The distributions of SVs in these chromosomes in terms of MAF and LD are presented in Figures 4 e, f, g, and h, indicating low LD and diverse MAF in general for the mentioned SV datasets.

4.2 Training procedure

In previous studies, the training data is masked using different rates to match the test set, e.g., [3, 14]. In our experiments, we observed improved performance of the model with 50% dynamic and random masking of the variants in the training data. So we trained the DL model once and reused it multiple times. Notably, this masking is similar to the masking performed in modern large language models. The benefit of such a technique in genomic data imputation is the notable reduction in the inference (imputation) times when compared to the fastest traditional methods. Consequently, a DL model trained in this manner becomes particularly advantageous for deployment on imputation servers, where re-training needs to be avoided for quick and efficient processing.

Another improvement we achieved was by representing phased diploids into haploids, followed by one-hot encoding. That is, instead of feeding (one-hot encoded) phased diploids to the models, we fed them haploids. This idea is proposed in [26], but there is no discussion about the merits of this procedure. We surmised that predicting haploids would be easier because mutations in paternal and maternal haploids are independent of each other. In the output, diploid genotypes were reconstructed by combining corresponding haploids together.

4.3 Split-Transformer Impute architecture

Split-Transformer Impute is an extended transformer model [28] specifically tailored for genotype imputation. STI models do not require any additional information provided by a reference panel, except for the genotypes and their relative positions. This makes STI adaptable to any genotype data and allows it to be applied to a wider range of datasets with less effort and fewer preparations. Moreover, although here we focus on sporadic missingness, once STI is trained on a dataset, it can predict both sporadic missingness and systematic missingness in genotype data as long as the target variants are a subset of the training variants. An overview of STI is presented in Figure 5. We implemented STI and the rest of the DL models using Tensorflow framework [41] in Python. In order to train the models, we used tensor

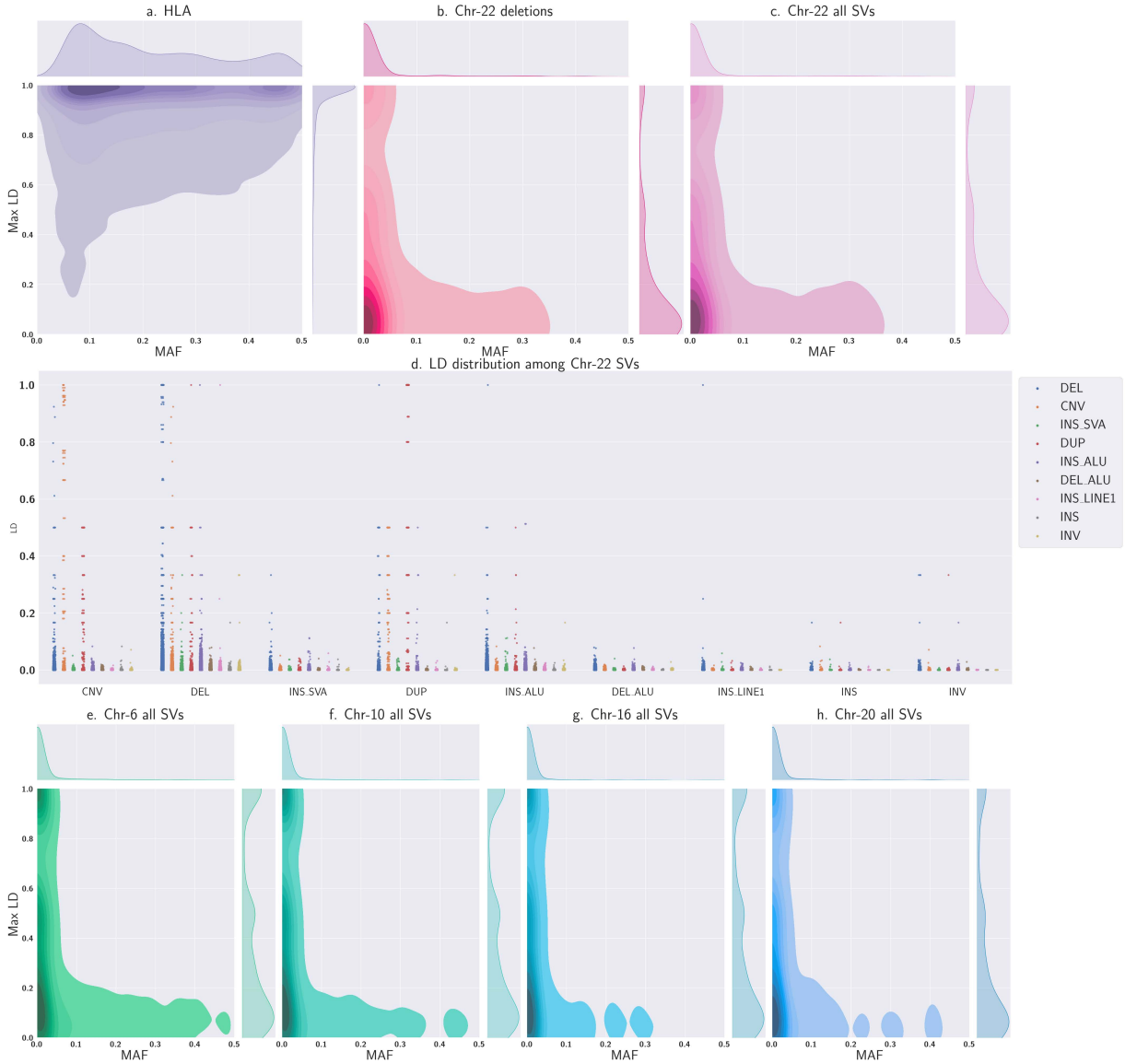

Fig. 4 MAF and LD distributions of benchmark datasets from 1000 Genomes Project. MAF and maximum LD distributions are presented using kernel density estimation plots for SNVs/SVs in *a. HLA region on chromosome 6*, *b. deletions in chromosome 22*, *c. SVs in chromosome 22*, *e. SVs in chromosome 6*, *f. SVs in chromosome 10*, *g. SVs in chromosome 16*, and *h. SVs in chromosome 20*. Overall, SVs exhibit a low LD value, posing a significant challenge to imputation methods. Plot *d. LD among different SV types in chromosome 22* shows that structural events are commonly correlated with deletions. Furthermore, deletion, copy number variation, and duplication events appear in different ranges of LD, while the rest of the events are limited to $LD \leq 0.1$. Lastly, the majority of correlated SVs to deletions are of the same event, making deletions a good separate dataset for our experiment.

processing units (TPU) provided by the Google Colaboratory platform, but a GPU implementation of STI is available as well. A learning rate scheduler and early stopping are employed in order to reduce the loss and training duration.

4.3.1 Cat-Embedding

One important part of STI is categorical embedding (Figure 5.b), termed as Cat-Embedding, which enables it to learn embedding representation per allele in each position. For the imputation task, we consider missing values as another allele that is equivalent to special tokens in natural language processing. The corresponding vector for each allele is added to the respective positional variant embedding vector

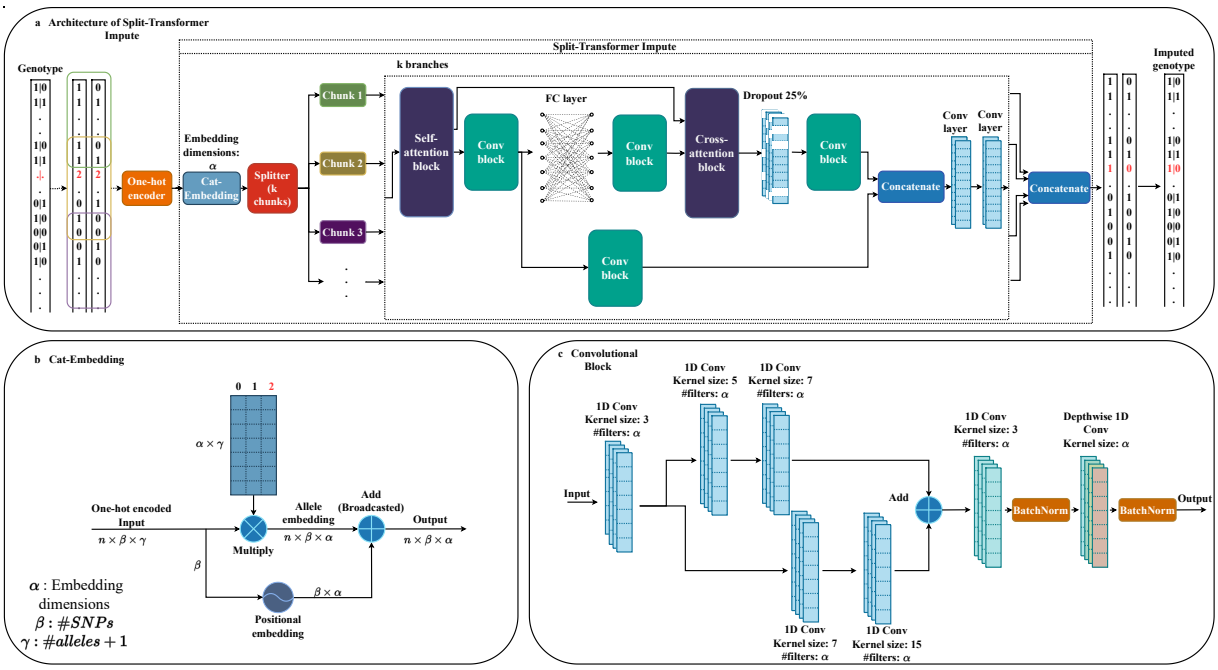

Fig. 5 Split-Transformer Impute architecture. *a. Overall pipeline of the proposed framework:* the data is separated into paternal and maternal haplotypes in the case of diplotypes, and it remains the same for haplotypes. While the figure shows phased genotypes, STI can handle unphased data as well (though the performance degrades). Next, the data is one-hot encoded and fed into our Cat-Embedding layer, followed by splitting the data vertically into k chunks. The chunks have overlap in order to capture information for the SNVs residing around the chunks' edges. Each branch passes through a unique set of attention, convolution, and fully connected layers. In the self-attention block, the flanking variants that come from the neighboring chunks are removed after applying multi-head attention. Finally, the results of all branches are assembled to generate the final sequence. *b. Workflow of proposed Cat-Embedding:* we consider a unique vector space for each unique categorical value in each SNV/feature. To save computational resources, instead of pre-allocating these vectors, we use the addition of positional embedding and categorical value embeddings in order to generate unique embedding vectors for each categorical value in each SNV/feature. We consider a missing (or masked) value as another categorical value (allele) in our model. Here, 2 (highlighted in red) represents the missing value. *c. Convolution blocks:* two parallel convolutional branches with varying kernel sizes are used in our convolution blocks. These multi-scale convolutional blocks allow STI to capture information at multiple spatial scales in the input data, similar to the pattern-matching idea used in classical computer vision methods using convolution. Given the variable sizes of LD blocks, multi-scaled convolution is expected to excel at capturing LD patterns compared to single-scaled convolutions.

to generate the final embedding. The idea is similar to a natural language processing embedding layer that accepts word indices, except that Cat-Embedding accepts one-hot encoded data.

4.3.2 Splitting

While the multi-headed attention in a transformer offers significant advantages, a major drawback is quadratic memory cost for computations that becomes important in genomic analysis, since the number of variants in a sample is normally in the thousands. In genotypes, the majority of interactions are local [42]. Therefore, it is of great importance to limit the scope of attention to save computational resources. To do so, we split the variants into chunks (vertical partitioning). The chunk size and overlap size are employed in a comparable manner in Minimac4.1.4 and analogous software applications. In order to prevent loss of imputation accuracy at chunk borders, we include flanking variants from neighboring chunks and discard them after applying self-attention to get the original variants in the chunk. Though the average LD block size in the dataset can be used to decide the size of overlap, we do not use LD blocks directly to decide the chunk size in the current version.

Each chunk passes through a dedicated branch inside the model, leading to increased imputation quality. Ideally, having a vast number of samples allows training a single model with attention across the whole genome. However, when the number of samples is not enough, the model is left with untrained parameters, resulting in poor performance. Hence, chunking regulates the number of parameters. In a

vanilla transformer, the cost of computing global attention is quadratic with respect to the number of SNVs (m^2); however, the amount is lowered to $(m/w) \times (w+o)^2 = mw$ in STI, considering that the overlaps of chunks are negligible compared to the chunk size. For instance, for $m = 10^4$ and a chunk size of 10^3, STI uses 10 times less memory for attention computations compared to a vanilla transformer.

4.3.3 Attention

The attention blocks are implemented similarly to those of other transformers, such as self-attention blocks in Vision Transformer (ViT) [43]. There is a difference between the first and second attention blocks in the branches. The first block is a self-attention block, meaning that the query, key, and value of the attention layer are the same. The output of multi-head attention in Tensorflow has the same dimensions as the query. By excluding the neighboring variants of a chunk from the query and only including them in the key and value, we involve them in the attention mechanism and, at the same time, shrink the output of a chunk to the target size (chunk size without counting flanking/overlap variants) after applying multi-headed attention. In the second block, the query is the output of the previous layer, while the key and value are the outputs of the first self-attention block. This skip connection considerably affects the overall performance of the model.

4.3.4 Convolutional block

Convolutional blocks, as illustrated in Figure 5.c, are also crucial components of STI. Through empirical studies, we found that using two parallel convolutional branches with varying kernel sizes, similar to the Inception module [44], is the best trade-off between accuracy gain and increase in a number of model parameters, compared to using a single branch or more than two branches. Furthermore, a Depth-wise convolutional layer at the end of the block helps STI extract local information without mixing channel information and substantially improves imputation accuracy.

4.3.5 Output formation

Finally, the outputs of all branches are concatenated to form the output, that is, either maternal or paternal haplotype in the case of 1000 Genomes Project datasets or the genotypes in the case of yeast. For the former, by assembling maternal and paternal haplotypes, we obtain imputed genotypes, and the latter needs no further post-processing. Since genetic variations in parents are independent, directly encoding and imputing the genotypes in diploid life forms results in lower imputation accuracy compared to imputing their haplotypes. Hence we undergo extra steps to separate diplotypes into haplotypes in pre-processing, and combining respective predicted haplotypes into diplotypes in post-processing for the human 1000 Genomes Project dataset.

4.3.6 Loss function

For the loss function, we used a combination of Kullback–Leibler divergence (D_{KL}) and categorical cross-entropy (CCE), similar to the loss function of variational autoencoder [45], as follows:

$$Loss(y, \hat{y}) = (\theta)CCE(y, \hat{y}) + (1-\theta)D(y\|\hat{y}), \tag{1}$$

where θ is the weight parameter. The first term, representing categorical cross-entropy, and the second term, representing Kullback–Leibler divergence loss, are calculated as follows:

$$CCE(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \log\left(p(y_{ij})\right) \tag{2}$$

$$D_{KL}(y\|\hat{y}) = \sum_{i=1}^{N} p(y_i)\frac{p(y_i)}{p(\hat{y}_i)} \tag{3}$$

We set θ to 0.5, meaning that STI minimizes Equations 2, 3 equally. CCE captures reconstruction error between the input and the output, while D_{KL} measures asymmetric distance, with y as the base, between their probability distributions. In our experiments, omitting any of these losses resulted in reduced model performance. Theoretically, KL-divergence and cross-entropy are related and using both might not seem

to contribute to the performance of the model. However, adding KL-divergence to the loss term helps the model to retain the probability/dosage distribution of alleles per variant. In other words, while cross-entropy focuses on predicting the correct genotype, KL-divergence acts as a regularization factor and penalizes the model whenever the shape of the predicted probability distribution (allele probability/-dosage) shows divergence from the ground truth. Moreover, the mathematical relation of D_{KL} and CCE can be summarized as follows:

$$D_{KL}(y\|\hat{y}) = CCE(y, \hat{y}) - CCE(y) \tag{4}$$

where $CCE(y)$ is the entropy of the ground truth. According to Equation 4 minimizing $D_{KL}(y\|\hat{y})$ is equivalent to minimizing $CCE(y, \hat{y})$ under the condition that the entropy of the ground truth remains constant. However, in deep learning models, data is typically processed in mini-batches. This means that the entropy of each mini-batch may not accurately represent the entropy of the entire ground truth. As a result, Equation 4 does not hold for the DL models in general.

4.4 Baseline models

In order to benchmark our model, we compare STI to state-of-the-art imputation models capable of imputing sporadic missingness: SCDA [14], AE [3], DEEP*HLA [26], Beagle5.4 [19], and Minimac4.1.4 [15]. In [14], experimental results indicate that SCDA outperforms classical ML models for genotype imputation. Hence, we do not include classical ML models in our benchmarking analyses. It is worth noting that DEEP*HLA is not originally designed for genotype imputation in general, but we modified and fine-tuned it to work for this problem. Additionally, in order to assess the contribution of Cat-Embedding, we replaced it with a convolution layer in STI, named the resulting model STI-NE, fine-tuned it, and applied it to the benchmark datasets. Lastly, we trained SCDA, in addition to DEEP*HLA and STI, using our proposed pre-processing and training procedure, and compared it to AE. Since AE and original SCDA are the same and only differ in the training process (which results in AE outperforming SCDA), we believe that this comparison can demonstrate the effectiveness of our proposed pre-processing and training procedure.

For SCDA and AE, hyper-parameter tuning information on the yeast dataset is present in the original papers. For SCDA, AE, DEEP*HLA, and STI, we conducted a grid search for optimal hyper-parameters on the HLA dataset using validation sets in a 3-fold cross-validation. We assessed the impact of these hyper-parameters on the performance of the models within the HLA dataset and applied these findings to select suitable hyper-parameters for the yeast dataset in the case of DEEP*HLA and STI, and for the SV dataset across all four mentioned methods. The upper limit for the hyper-parameters was the resource limit of Google Colaboratory using Nvidia Titan IV GPU with 16 GB of RAM size for AE, and roughly the same limitation for TPU RAM size. Minimac4.1.4 and Beagle5.4 do not require fine tuning for the experiments we run.

4.5 Experimental settings

The input to all DL models is one-hot encoded. While STI can handle diploids, we found that the best performance was achieved when the inputs of the DL models were haplotypes, an analysis inspired by [26]. Therefore, for the HLA dataset and chromosome 22 datasets, we separated each diplotype into maternal and paternal haplotypes, fed them into the model, and reconstituted the resulting predictions for DEEP*HLA [26], SCDA [14], and STI. We continue using diplotypes as inputs for AE [3] since it is an improved version of SCDA in which the training process was modified, and we wanted to keep it intact. By doing so, we also compare the improvement in AE to our implementation of SCDA, called SCDA+, in which we use proposed pre-processing in conjunction with the changes to the training process as a contribution. The yeast dataset contains haplotypes, so there is no need for the aforementioned extra steps.

In this study, to evaluate the imputation power of the models, multiple evaluation metrics are used including imputation accuracy, imputation quality score (IQS) [46], weighted $F1$-score, and correlation between imputed and real genotypes in terms of R^2 [47]. Accuracy and weighted $F1$-score are calculated only for positions with missing genotypes and for these metrics, heterozygous genotypes are encoded differently; i.e., *0|1* and *1|0* are encoded to two different categorical values. IQS adjusts the chance concordance between predicted and the ground truth SNVs and is defined for bi-allelic events. Therefore,

IQS cannot be calculated for any SV in chromosome 22. R^2 is the squared Pearson correlation coefficient between the imputed genotypes and the true genotypes at a specific locus. The definition of these metrics is provided in the Metrics section of the Supplement.

Data availability

All data used in this study are publicly available. The yeast dataset can be found as the *Supplementary Data 5* at https://www.nature.com/articles/ncomms9712, the rest of datasets are extracted from the 1000 Genomes Project phase 3 dataset available at http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/. Instructions on how to prepare the data for Missing variants experiment can be found in https://github.com/kanamekojima/rnnimp.

Code availability

The source code of STI is publicly available on GitHub (https://github.com/shilab/STI).

Acknowledgements. This work is partially supported by the US National Science Foundation (DBI 1750632) and the National Institutes of Health (GM-0126567-03). This research includes calculations carried out on HPC resources supported in part by the National Science Foundation through major research instrumentation grant number 1625061 and by the US Army Research Laboratory under contract number W911NF-16-2-0189. We appreciate the suggestion provided by Dr. Francisco McGee in designing the model, leading to improvements in performance. Additionally, we would like to thank Dr. Kaname Kojima for helping us obtain the data for the Missing variant experiment and providing experimental results for RNN-IMP (the experiment in the supplement) and Emily Thyrum for proofreading the manuscript.

5 Contributions

M.E.M. developed the method with the help of J.C., B.J., V.C, and X.S. and M.E.M. implemented the code. C.L. and O.J. prepared the datasets. M.E.M. and R.D. performed the experiments. M.E.M, S.K., C.L., O.J., and X.S. conducted the data analysis. M.E.M, C.L., S.K., T.R.R., V.C, and X.S. wrote the manuscript. All the authors read and approved the submitted manuscript.

Ethics declarations

Not applicable.

Competing interests. The authors declare that they have no competing interests.

Declarations

Not applicable.

References

[1] Lewis, C.M., Vassos, E.: Polygenic risk scores: from research tools to clinical instruments. Genome medicine **12**(1), 1–11 (2020)

[2] Torkamaneh, D., Belzile, F.: Accurate imputation of untyped variants from deep sequencing data. Deep Sequencing Data Analysis, 271–281 (2021)

[3] Song, M., Greenbaum, J., Luttrell IV, J., Zhou, W., Wu, C., Luo, Z., Qiu, C., Zhao, L.J., Su, K.-J., Tian, Q., et al.: An autoencoder-based deep learning method for genotype imputation. Frontiers in Artificial Intelligence **5** (2022)

[4] Das, S., Abecasis, G.R., Browning, B.L.: Genotype imputation from large reference panels. Annu Rev Genomics Hum Genet **19**(1), 73–96 (2018)

[5] Graffelman, J., Nelson, S., Gogarten, S., Weir, B.: Exact inference for hardy-weinberg proportions with missing genotypes: Single and multiple imputation. G3: Genes, Genomes, Genetics **5**(11), 2365–2373 (2015)

[6] Wigginton, J.E., Cutler, D.J., Abecasis, G.R.: A note on exact tests of hardy-weinberg equilibrium. The American Journal of Human Genetics **76**(5), 887–893 (2005)

[7] Pei, Y.-F., Li, J., Zhang, L., Papasian, C.J., Deng, H.-W.: Analyses and comparison of accuracy of different genotype imputation methods. PloS one **3**(10), 3551 (2008)

[8] Auer, P.L., Wang, G., Project, N.E.S., Leal, S.M.: Testing for rare variant associations in the presence of missing data. Genetic epidemiology **37**(6), 529–538 (2013)

[9] Chiang, C., Scott, A.J., Davis, J.R., Tsang, E.K., Li, X., Kim, Y., Hadzic, T., Damani, F.N., Ganel, L., Consortium, G., *et al.*: The impact of structural variation on human gene expression. Nature genetics **49**(5), 692–699 (2017)

[10] Consortium, .G.P., *et al.*: A global reference for human genetic variation. Nature **526**(7571), 68 (2015)

[11] Liu, Z., Roberts, R., Mercer, T.R., Xu, J., Sedlazeck, F.J., Tong, W.: Towards accurate and reliable resolution of structural variants for clinical diagnosis. Genome biology **23**(1), 68 (2022)

[12] Bartlett, J.W., Seaman, S.R., White, I.R., Carpenter, J.R., Initiative*, A.D.N.: Multiple imputation of covariates by fully conditional specification: accommodating the substantive model. Statistical methods in medical research **24**(4), 462–487 (2015)

[13] Song, M., Greenbaum, J., Luttrell IV, J., Zhou, W., Wu, C., Shen, H., Gong, P., Zhang, C., Deng, H.-W.: A review of integrative imputation for multi-omics datasets. Frontiers in genetics **11**, 570255 (2020)

[14] Chen, J., Shi, X.: Sparse convolutional denoising autoencoders for genotype imputation. Genes **10**(9), 652 (2019)

[15] Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A.E., Kwong, A., Vrieze, S.I., Chew, E.Y., Levy, S., McGue, M., *et al.*: Next-generation genotype imputation service and methods. Nature genetics **48**(10), 1284–1287 (2016)

[16] Li, Y., Willer, C.J., Ding, J., Scheet, P., Abecasis, G.R.: Mach: using sequence and genotype data to estimate haplotypes and unobserved genotypes. Genetic epidemiology **34**(8), 816–834 (2010)

[17] Hofmeister, R.J., Ribeiro, D.M., Rubinacci, S., Delaneau, O.: Accurate rare variant phasing of whole-genome and whole-exome sequencing data in the uk biobank. Nature Genetics **55**(7), 1243–1249 (2023)

[18] Howie, B.N., Donnelly, P., Marchini, J.: A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS genetics **5**(6), 1000529 (2009)

[19] Browning, B.L., Zhou, Y., Browning, S.R.: A one-penny imputed genome from next-generation reference panels. The American Journal of Human Genetics **103**(3), 338–348 (2018)

[20] Scheet, P., Stephens, M.: 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)

[21] Rubinacci, S., Hofmeister, R.J., Mota, B., Delaneau, O.: Imputation of low-coverage sequencing data from 150,119 uk biobank genomes. Nature Genetics **55**(7), 1088–1090 (2023)

[22] Bloom, J.S., Kotenko, I., Sadhu, M.J., Treusch, S., Albert, F.W., Kruglyak, L.: Genetic interactions contribute less than additive effects to quantitative trait variation in yeast. Nature communications **6**(1), 1–6 (2015)

[23] Dias, R., Evans, D., Chen, S.-F., Chen, K.-Y., Loguercio, S., Chan, L., Torkamani, A.: Rapid, reference-free human genotype imputation with denoising autoencoders. Elife **11**, 75600 (2022)

[24] Kojima, K., Tadaka, S., Katsuoka, F., Tamiya, G., Yamamoto, M., Kinoshita, K.: A genotype imputation method for de-identified haplotype reference information by using recurrent neural network. PLoS Computational Biology **16**(10), 1008207 (2020)

[25] Chi Duong, V., Minh Vu, G., Khac Nguyen, T., Tran The Nguyen, H., Luong Pham, T., S. Vo, N., Hong Hoang, T.: A rapid and reference-free imputation method for low-cost genotyping platforms. Scientific Reports **13**(1), 23083 (2023)

[26] Naito, T., Suzuki, K., Hirata, J., Kamatani, Y., Matsuda, K., Toda, T., Okada, Y.: A deep learning method for hla imputation and trans-ethnic mhc fine-mapping of type 1 diabetes. Nature communications **12**(1), 1–14 (2021)

[27] Tanaka, K., Kato, K., Nonaka, N., Seita, J.: Efficient hla imputation from sequential snps data by transformer. arXiv preprint arXiv:2211.06430 (2022)

[28] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[29] Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Annual review of neuroscience **18**(1), 193–222 (1995)

[30] Cho, K., Courville, A., Bengio, Y.: Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia **17**(11), 1875–1886 (2015)

[31] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., *et al.*: Highly accurate protein structure prediction with alphafold. Nature **596**(7873), 583–589 (2021)

[32] Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., *et al.*: Evolutionary-scale prediction of atomic-level protein structure with a language model. Science **379**(6637), 1123–1130 (2023)

[33] Sudmant, P.H., Rausch, T., Gardner, E.J., Handsaker, R.E., Abyzov, A., Huddleston, J., Zhang, Y., Ye, K., Jun, G., Hsi-Yang Fritz, M., *et al.*: An integrated map of structural variation in 2,504 human genomes. Nature **526**(7571), 75–81 (2015)

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

[35] Tf-Encrypted: TF-encrypted/TF-encrypted: A framework for encrypted machine learning in tensorflow. https://github.com/tf-encrypted/tf-encrypted

[36] Kuo, T.-T., Jiang, X., Tang, H., Wang, X., Harmanci, A., Kim, M., Post, K., Bu, D., Bath, T., Kim, J., *et al.*: The evolving privacy and security concerns for genomic data analysis and sharing as observed from the idash competition. Journal of the American Medical Informatics Association **29**(12), 2182–2190 (2022)

[37] Kim, M., Harmanci, A.O., Bossuat, J.-P., Carpov, S., Cheon, J.H., Chillotti, I., Cho, W., Froelicher, D., Gama, N., Georgieva, M., *et al.*: Ultrafast homomorphic encryption models enable secure outsourcing of genotype imputation. Cell systems **12**(11), 1108–1120 (2021)

[38] Miles, A., bot, R., M., Ralph, P., Harding, N., Pisupati, R., Rae, S., Millar, T.: Cggh/scikit-allel: V1.3.3. https://doi.org/10.5281/zenodo.4759368 . https://doi.org/10.5281/zenodo.4759368

[39] Hillert, J.: Human leukocyte antigen studies in multiple sclerosis. Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society **36**(S1), 15–17 (1994)

[40] Terasaki, P.I., Cai, J.: Human leukocyte antigen antibodies and chronic rejection: from association to causation. Transplantation **86**(3), 377–383 (2008)

[41] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)

[42] Weir, B.: Linkage disequilibrium and association mapping. Annual review of genomics and human genetics **9**(1), 129–142 (2008)

[43] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

[44] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)

[45] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[46] Lin, P., Hartz, S.M., Zhang, Z., Saccone, S.F., Wang, J., Tischfield, J.A., Edenberg, H.J., Kramer, J.R., M. Goate, A., Bierut, L.J., *et al.*: A new statistic to evaluate imputation reliability. PloS one **5**(3), 9697 (2010)

[47] Deng, T., Zhang, P., Garrick, D., Gao, H., Wang, L., Zhao, F.: Comparison of genotype imputation for snp array and low-coverage whole-genome sequencing data. Frontiers in genetics **12**, 704118 (2022)