Zawadzka, W., Szyda, J., Frąszczak, M. (2026) Inferring missing genotypes from partially observed SNP data using deep learning-based architectures, *Proceedings 13th World Congress on Genetics Applied to Livestock Production* 23763: https://doi.org/10.31274/wcgalp.23763
©2026 Zawadzka, et al. CC BY-NC-ND

Inferring missing genotypes from partially observed SNP data using deep learning-based architectures

W. Zawadzka[1*], J. Szyda[1,2], and M. Frąszczak[1]

[1]*Biostatistics Group, Wroclaw University of Environmental and Life Sciences, 51-631 Wroclaw, Poland;* [2]*Wroclaw University of Science and Technology, Wroclaw, Poland;* [*]*Corresponding author: joanna.szyda@upwr.edu.pl*

Introduction

The goal of genomic imputation is to predict missing genotypes from partially observed single nucleotide polymorphism (SNP) data. Missing genotypes commonly arise when moving from oligonucleotide arrays to whole genome sequence level; many loci are then untyped and can be inferred via imputation. Accurate imputation is crucial for downstream analyses, particularly for the detection and interpretation of rare variants. Traditional imputation tools, such as Beagle, infer missing genotypes using Hidden Markov Models that exploit local haplotype structure (Browning and Browning, 2007; Browning and Browning, 2016). These methods are widely used due to their accuracy and computational speed. Our study explores the use of deep learning (DL) for SNP imputation, investigating how different model architectures affect imputation accuracy taking the unbalanced genotypes into account. The application of DL to SNP imputation has been studied (Naito and Okada, 2024), but many works report only overall accuracy, which can be misleading under class imbalance and does not adequately reflect performance for rare variants. The motivation for using DL-based architectures is their flexibility in modeling complex patterns, which may allow improved reconstruction of rare genotypes by more effectively exploring the parameter space. We evaluate four AutoEncoder (AE) based approaches: a naïve AE using dense neural networks, a dense AE incorporating input embeddings, a convolutional neural network based AutoEncoder (CAE), and a transformer based AE utilising the BERT model. Imputation accuracy obtained using Beagle serves as a baseline for comparison.

Materials & Methods

Materials. The material consisted of a subset of Holstein bulls from the 1000 Bull Genomes Project (Hayes *et al*., 2019). A fragment of chromosome 28 spanning the first 500 SNPs (~ 66.7 kb) was used. 928 bulls were included, with 742 bulls assigned to the training and 186 bulls to the testing set. The training set was further randomly split into training and validation subsets using 80:20 ratio. Minor allele frequencies (MAFs) for the test data were calculated using PLINK2 (Chang *et al*., 2026), and rare SNPs were considered as having MAF < 0.05.
To mimic different real life complexity scenarios, multiple test datasets with missing genotypes were generated from the testing set. For each scenario, a random subset of SNP columns was masked for all individuals, with missingness ranging from 10% to 80%. DL models were trained using a missingness level of 60%, which was selected as a compromise between task difficulty and model stability. One advantage of such models is that they can be trained directly on the actual missingness pattern; fixed missingness levels here are used only for comparison purposes. Under the 60% missingness scenario applied to the test data, the corresponding training set shows a strongly imbalanced genotype distribution: genotype 0 accounts for 75.57% of observations, genotype 1 for 19.54%, and genotype 2 for 4.89%.

Autoencoder models. AE based approaches have been chosen for this comparison due to their ability to learn latent representations that capture dependencies between SNPs and to

use this information to accurately reconstruct missing genotypes. An AE consists of an encoder, which maps the input genotypes (0/1/2) into a lower dimensional latent space, and a decoder, which reconstructs the original genotype matrix from this representation. ReLU activations enable learning of complex patterns, and training minimizes weighted (using inverse class frequency) categorical cross-entropy at masked positions to focus on imputing missing genotypes.

The dense feedforward AE (Figure 1) encodes 500 SNP genotypes through two fully connected layers down to a 64-unit bottleneck, then decodes back to 500 SNPs, outputting three probabilities per SNP via SoftMax. In the DenseEmb variant, genotypes are first mapped to learnable 3D embeddings, processed with a 1D convolution and global average pooling, then passed through an encoder expanding to 1024 units and compressing to a 64-unit bottleneck.

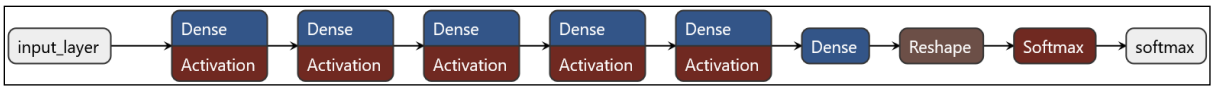

Figure 1. Architecture of deep autoencoder models.

The convolutional AE (Figure 2) encodes 500 SNPs via two 1D convolutional layers with ReLU, batch normalization, and spatial dropout, then flattens to a 16-unit dense bottleneck that captures global dependencies. The decoder mirrors this with dense and transposed convolution layers to reconstruct the SNPs. Convolutional filters use a window size of 21 and stride 3 to capture local SNP dependencies. All AE models were implemented in TensorFlow 2.20.0.

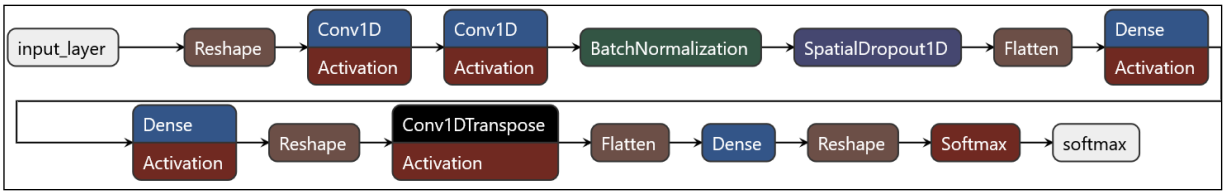

Figure 2. Architecture of convolutional autoencoder model.

BERT model. This transformer-based model was trained using the masked language modeling task (MLM), where the objective is to predict missing or masked tokens in a sequence based on their surrounding context. Genotypes were treated as words in the vocabulary, forming sequences of length 500. Implemented with BertForMaskedLM in PyTorch 2.10, training used focal loss on masked positions with inverse class-frequency weights. The architecture included 2 transformer layers, 1 attention head, a hidden size of 256, and a feedforward size of 1024.

Mask overrepresentation. Given the high global imbalance of missing genotypes, a controlled mask upsampling strategy was applied when creating masked inputs. For each training sample, SNPs were initially masked according to the same positions missing in the test set. To mitigate the extreme imbalance between genotype classes, masked SNPs with genotypes 1 and 2 were preferentially included. A multiplier controlled the relative frequency of each genotype class in the masked positions, and its optimal value 1-3 was determined via hyperparameter optimization. This approach does not constitute traditional oversampling, as no new or repeated examples are added to the mask; rather, it selectively increases the representation of less frequent genotypes in those positions.

Optimization. Model parameters were tuned with Optuna (Akiba *et al*., 2019) over 50 trials, maximizing macro F1 on the validation set. Macro F1 was chosen to account for the imbalanced genotype classes. Since the 0 genotype is overwhelmingly common, overall accuracy can be misleadingly high, whereas macro F1 ensures that performance across all three genotype classes is evaluated. Training was performed on AMD Ryzen 7 4800HS.

Results & Discussion

Performance was evaluated in three scenarios: (i) 60% missingness, matching the training condition (in-distribution); (ii) the same evaluation restricted to rare genotypes; and (iii) application to other missingness levels to test generalization. However, in realistic settings, the missingness pattern used during training is typically reflective of the target data to be imputed. The additional missingness settings are included to evaluate robustness.

Overall, in the 60% missingness scenario, the transformer-based BERT achieved the highest F1 macro score of 0.868 (Table 1). Despite having the fewest parameters, it required the longest training time of approximately 3 hours. Faster alternatives, all training in under a minute on this dataset, included dense AEs, with the version using an embedding layer before the AE performing slightly better (F1 micro 0.853 vs 0.847). The CAE (trained for approx. 8 minutes) performed worse than the other three models. Its large number of parameters did not translate into better performance, making it less efficient and less accurate. Beagle requires no training, with near real-time inference.

Table 1. All missing SNP imputation on the dataset with 60% missingness.

Model	F1 micro	F1 macro	F1 genotype 0	F1 genotype 1	F1 genotype 2	Number of parameters
BERT	**0.917**	**0.868**	**0.953**	**0.811**	**0.840**	**1779976**
DenseEmb	0.906	0.853	0.946	0.777	0.835	3166748
Dense	0.909	0.847	0.950	0.782	0.810	3133948
CAE	0.906	0.838	0.949	0.757	0.809	73035868
Beagle	0.897	0.807	0.948	0.676	0.797	-

While focusing specifically on the rare genotypes inside the missing positions, we can observe the change of the ranking (see Table 2). BERT has remained the highest scoring model (F1-macro=0.563) but the Beagle is right after it (F1-macro=0.556). The Dense AE with embedding and the CAE failed to correctly identify any genotype 2 examples. This table also highlights the limitations of relying on overall accuracy (or F1-micro), which exceed 0.96 for all models and therefore obscure the true challenge of imputing rare genotypes.

Table 2. Rare genotype (MAF < 0.05) imputation on the dataset with 60% missingness.

Model	F1 micro	F1 macro	F1 genotype 0	F1 genotype 1	F1 genotype 2
BERT	**0.954**	**0.563**	**0.976**	**0.541**	**0.173**
Beagle	0.971	0.556	0.986	0.533	0.148
Dense	0.966	0.490	0.983	0.421	0.066
DenseEmb	0.965	0.450	0.982	0.369	0
CAE	0.961	0.356	0.980	0.088	0

[1] Rare defined as MAF < 0.05.

Comparison across missingness levels. Models trained on 60% missingness performed similarly on lower missing fractions, but their performance dropped significantly on higher

fractions beyond the training range (Figure 3). Beagle remained consistent across the full range of missingness, showing the greatest adaptability.

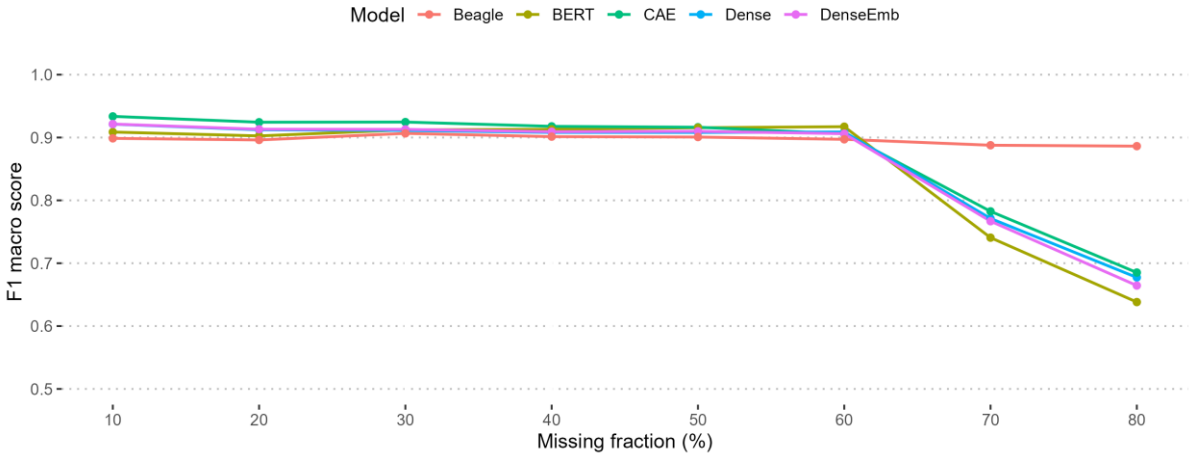

Figure 3. F1-macro comparison across models and missing fractions.

Conclusions

Deep learning models, especially transformer-based BERT, achieved high F1 scores. BERT balanced high accuracy with a small parameter count but required longer training. Dense AEs, with or without embeddings, were faster and competitive but struggled with rare genotypes. Beagle was fast and consistent across missingness levels, performing well on rare genotypes but with lower overall accuracy at 60% missingness, which corresponds to the training scenario used for the DL models. These results indicate that DL approaches perform best when the missingness structure is matched between training and application, whereas classical methods like Beagle show more stable generalization across scenarios. If longer imputation time is not a limiting factor and higher accuracy (especially of the rare variants) is the primary objective then DL-based approaches may provide an alternative to standard approaches. Accurate imputation of rare variants is critical for downstream analyses not only in humans for accurate prediction of polygenic risk scores or treatment design, but also in livestock for mate selection or accurate estimation of breeding values. Genome-wide association studies in all species also profit from accurate SNP genotype imputation that allows for processing very large phenotyped cohorts in which not all individuals are sequenced with the genome-wide resolution. However, high class imbalance between common and rare variant complicates imputation. Strategies include using imbalance-aware loss functions, controlled masking during training, and careful evaluation metrics, as standard metrics can be misleading. More experimentation is needed, particularly on larger datasets where longer linkage disequilibrium patterns could potentially improve rare genotype imputation. It would also be useful to train models on a range of missingness levels to assess true performance, since in real scenarios the missingness fraction of the test data is known.

References

Akiba T., Sano S., Yanase T., *et al*. (2019) arXiv 1907.10902. https://doi:10.1145/3292500.3330701

Browning B.L., and Browning S.R. (2007) Am. J. Hum. Genet. 81:1084-97. https://doi:10.1086/521987

Browning B.L., and Browning S.R. (2016) Am. J. Hum. Genet. 98:116-26. https://doi:10.1016/j.ajhg.2015.11.020

Chang C.C., Chow C.C., Tellier L.C.A.M., *et al*. (2015) Gigascience 4:7. https://doi:10.1186/s13742-015-0047-8

Hayes B.J., and Daetwyler H.D. (2019) Annu. Rev. Anim. Biosci. 7:89-102. https://doi:10.1146/annurev-animal-020518-115024

Naito T., and Okada Y. (2024) J. Hum. Genet. 69:481-486. https://doi:10.1038/s10038-023-01213-6.