Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI-24)
Special Track on AI for Good

An Embarrassingly Simple Approach to Enhance Transformer Performance in Genomic Selection for Crop Breeding

Renqi Chen[1], **Wenwei Han**[1,2,3], **Haohao Zhang**[4], **Haoyang Su**[1], **Zhefan Wang**[1], **Xiaolei Liu**[5], **Hao Jiang**[2,3], **Wanli Ouyang**[1] and **Nanqing Dong**[1]
[1]Shanghai Artificial Intelligence Laboratory
[2]Institute of Computing Technology, Chinese Academy of Sciences
[3]School of Computer Science, University of Chinese Academy of Sciences
[4]School of Computer Science and Artificial Intelligence, Wuhan University of Technology
[5]School of Animal Science and Technology, Huazhong Agricultural University
{chenrenqi, hanwenwei, dongnanqing}@pjlab.org.cn

Abstract

Genomic selection (GS), as a critical crop breeding strategy, plays a key role in enhancing food production and addressing the global hunger crisis. The predominant approaches in GS currently revolve around employing statistical methods for prediction. However, statistical methods often come with two main limitations: strong statistical priors and linear assumptions. A recent trend is to capture the non-linear relationships between markers by deep learning. However, as crop datasets are commonly long sequences with limited samples, the robustness of deep learning models, especially Transformers, remains a challenge. In this work, to unleash the unexplored potential of attention mechanism for the task of interest, we propose a simple yet effective Transformer-based framework that enables end-to-end training of the whole sequence. Via experiments on *rice3k* and *wheat3k* datasets, we show that, with simple tricks such as k-mer tokenization and random masking, Transformer can achieve overall superior performance against seminal methods on GS tasks of interest.

1 Introduction

Grain production security serves as the cornerstone of human existence, exerting a pivotal role in shaping the health, stability, and prosperity of global society. Addressing global hunger not only aligns with the United Nations Sustainable Development Goals (SDGs) [United Nations, 2023a] of Zero Hunger, but also adheres to the Leaving No One Behind Principle (LNOB) [United Nations, 2023b]. However, according to the report by the Food and Agriculture Organization of the United Nations in 2023 [United Nations, 2023c], around 735 million people worldwide were suffering from hunger. Therefore, utilizing technological advancements to boost grain production is vital for eliminating hunger by 2030. Crop breeding stands as a fundamental approach to enhancing grain production. Continuously improving crop varieties to increase yield and resilience effectively boosts grain production, thereby meeting the escalating demand for food.

Method	Assumption		Training	
	Stats	Linear	End2End	GPU
GBLUP [VanRaden, 2008]	○	○		
BayesA [Meuwissen *et al.*, 2001]	○	○		
DLGWAS [Manor and Segal, 2013]	○	○		✓
DNNGP [Wang *et al.*, 2023]		○		✓
Ours			✓	✓

Table 1: Comparison with the state-of-the-art methods for genotype-to-phenotype prediction task. "Stats" and "Linear" denote whether the statistical prior and linear assumption is required for the method, respectively. "End2End" and "GPU" denote whether the model can be trained in an end-to-end fashion and supported by GPU. "○" and "✓" denote conformity.

Genomic selection (GS), proposed by [Meuwissen *et al.*, 2001], is regarded as a promising breeding paradigm [Boichard *et al.*, 2012; García-Ruiz *et al.*, 2016]. GS usually involves predicting the phenotypes of polygenic traits in plants or crops by using high-density markers covering the entire genome. By implementing early screening of candidate populations, the genetic process can be accelerated, reducing generation intervals and significantly shortening breeding cycles. Unlike phenotype selection [Siepielski *et al.*, 2013; Kingsolver and Pfennig, 2007] and marker-assisted selection [Ribaut and Hoisington, 1998; Xu and Crouch, 2008], GS can utilize single nucleotide polymorphism (SNP) data obtained from organisms. SNP sequences are *loci* sequences with two or more alleles extracted from DNA sequences, which are the most widely adopted genetic markers in GS.[1]

[1]Stripping away segments that exhibit the same patterns from DNA sequences to obtain SNP sequences is advantageous for analyzing genotype-phenotype associations, as phenotypic differences

A mainstream solution to GS is to mine the statistical knowledge of SNP data. GBLUP [VanRaden, 2008] and SSBLUP [Goddard *et al.*, 2011] utilize kinship matrices to weight and aggregate the phenotypic values of different individuals, thereby obtaining their estimated breeding values (EBVs) but may not fully explore and utilize genomic information. Another category of statistical methods involves estimating marker effects on a reference population and then predicting marker effects on a candidate population, followed by linear aggregation to obtain EBVs for the candidates [Meuwissen *et al.*, 2001; Tipping and Faul, 2003; De Los Campos *et al.*, 2009; Kärkkäinen and Sillanpää, 2012]. These methods typically demand substantial computational resources and lack parallelization capabilities. Their practicality in time-sensitive breeding scenarios is constrained. In addition, two common limitations of statistical methods are the strong statistical priors (*e.g.*, Gaussian distribution) and linear relationship assumption between markers. Though these two assumptions simplify the statistical computation, statistical models suffer a setback when the underlying data distribution differs.

In order to capture the non-linear relationships between markers, efforts have been made by applying deep learning to process SNP sequences, especially with convolutional neural networks (CNN) [Ma *et al.*, 2017; Liu *et al.*, 2019; Xie *et al.*, 2023] and Transformers [Wu *et al.*, 2024]. However, two major data challenges threaten the robustness of deep learning models. First, the number of samples for each crop species is limited. This can easily cause overfitting for deep learning models, while statistical models are commonly deemed as robust solutions. Second, SNP are long sequences. Besides, plant genomes, unlike those of animals, demonstrate a higher frequency of long insertions and deletions attributed to the activity of transposable elements [Ramakrishnan *et al.*, 2022], leading to a greater abundance of SNP data among different plant genomes. The locality of convolution introduces a strong inductive bias for local dependencies, thus cannot catch the long-range interactions between markers. On the contrary, while Transformers can better tackle long-range dependencies in sequences, attention's quadratic complexity becomes the major bottleneck for long sequence [Dao *et al.*, 2022].

To mitigate the effect of long sequences, an intuitive way is to statistically pre-process the data before building deep learning models. However, the limitations of statistical methods still remain. For example, DNNGP utilizes principal component analysis (PCA), a statistical dimensionality reduction technique, and condenses the sequence to only several hundred dimensions for CNN modeling [Wang *et al.*, 2023]. However, the viability of this hinges on fulfilling the assumption of PCA which states that there exist linear correlations among variables in the dataset, but this might not be true for the majority of SNP datasets. Thus, the linear dimensionality reduction techniques may result in information loss regarding *loci* that influences phenotypes. Another statistical way to shorten the sequence is traditional feature selection, *i.e.*, selecting important features out of hundreds of thousands of SNPs [Manor and Segal, 2013]. For example, a promising method is to employ genome-wide association studies (GWAS) to identify major effect loci within gene sequences [Liu *et al.*, 2019]. However, these methods heavily rely on the accuracy of GWAS and are not conducive to modeling and analyzing quantitative traits.

To address the aforementioned challenges of statistical and deep learning methods, we propose a simple yet effective Transformer-based method that supports end-to-end training. The proposed method leverages simple tricks such as k-mer tokenization, and random masking to achieve robust predictive performance from genotypes to phenotypes on crop breeding datasets. It is worth mentioning that, though similar natural language processing (NLP) techniques have been investigated on DNA data due to the similarities between DNA sequences and text sequences [Ji *et al.*, 2021], there is no such application on SNP data yet. **To the best of our knowledge, we are the first to successfully adapt these simple NLP techniques to address the GS problem.** Compared with statistical methods, our method does not require rigid statistical priors or linear assumptions, and thus can better capture the non-linear relationships. Supported by GPU, our method has a significantly shorter inference time. Compared with existing deep learning methods, our method can better leverage the attention mechanism [Vaswani *et al.*, 2017] to assist contextual understanding. We summarize the differences between our method and seminal methods in Tab. 1.

We extensively evaluate the robustness of our method on two crop datasets, *rice3k* [Wang *et al.*, 2018] and *wheat3k*. *rice3k* is a public dataset on rice with SNP data. On the *rice3k* dataset, we outperform the current best-performing method (a hybrid method of GWAS and Transformer) on average over 1.05% in the accuracy metric. *wheat3k* is a private cereal grain dataset to be released, which contains 3032 SNP sequences following a similar setup of *rice3k*. Our method also consistently outperforms the seminal baselines on *wheat3k*.

- We propose an end-to-end Transformer-based framework for genomic selection that enables capturing non-linear relationships between genotypes and phenotypes.
- We show that in genotype-to-phenotype prediction tasks, using k-mer tokenization and random masking can effectively reduce the data complexity while enhancing model prediction performance.
- We conduct extensive experiments on the *rice3k* and *wheat3k* datasets, where our method achieves the state-of-the-art performance on both datasets against mainstream methods.

2 Related Work

2.1 Analysis-based Genomic Selection

Currently, analysis-based genomic selection methods are mainly classified into two families: *best linear unbiased prediction* (BLUP) methods and Bayesian methods [Gualdrón Duarte *et al.*, 2020]. BLUP-based methods evaluate random effects using a linear model. GBULP [VanRaden, 2008] constructs a pedigree relationship matrix using genomic information, incorporating it as a random ef-

often arise from distinct segments between two DNA sequences.

fect into the model to estimate genetic values or predict trait values of individuals. RRBLUP [Endelman, 2011] builds upon GBLUP by incorporating the concept of ridge regression. SSBLUP [Goddard *et al.*, 2011] integrates both the genomic relationship matrix and pedigree relationship matrix, along with phenotypic data, into a unified mixed model. For Bayesian methods, such as [Tipping and Faul, 2003; De Los Campos *et al.*, 2009; Kärkkäinen and Sillanpää, 2012], the marker effects are assumed to follow different Gaussian distributions. For example, BayesA [Meuwissen *et al.*, 2001] assumes that each marker has its own distribution and variance. Though Bayesian methods can reveal the relationship between genotype and phenotype to some extent, they are constrained by strong statistical priors and linear assumptions. In addition to the two families of methods, statistical learning methods [Ke *et al.*, 2017] may also improve prediction accuracy on specific datasets. However, this improvement might not generalize, especially when the number of samples is significantly smaller than the number of feature dimensions.

2.2 Deep Learning-based Genomic Selection

Fueled by the recent success of deep learning in vision and language tasks, researchers have attempted to integrate deep learning into the Genome Selection (GS) field. DeepGS [Ma *et al.*, 2017] proposes a genome-wide selection framework based on CNN, while DualCNN [Liu *et al.*, 2019] utilizes a dual-stream CNN to predict quantitative traits. ResGS [Xie *et al.*, 2023] uses ResNet [He *et al.*, 2016] to extract gene sequence information without using pooling layers. DNNGP [Wang *et al.*, 2023], a seminal work, performs PCA to reduce the high dimensionality of gene data to extract effective information. However, the limited convolutional receptive field hinders the model's ability to capture linkage disequilibrium loci in long SNP sequences.

Transformer [Vaswani *et al.*, 2017] has exhibited excellent performance in modeling global interactions within sequences. GPformer [Wu *et al.*, 2024] applies Informer [Zhou *et al.*, 2021], a model from the long-term time series prediction field, to aggregate periodic information and capture region-specific features in SNP sequences. However, the computational complexity of Transformer also has a quadratic relationship with the sequence length. The current Transformer-based methods [Wu *et al.*, 2024] are unable to handle the whole genetic SNP sequences for most crop species. There are hybrid methods [De Los Campos *et al.*, 2009; Liu *et al.*, 2019] that use GWAS to pre-process the raw SNP sequence, but the results heavily depend on the results of GWAS and exhibit unstable performance across species. Instead, we propose to leverage tokenization and optimization acceleration techniques to achieve full attention computation on long SNP sequences, thereby mining long-range interactions and obtaining more accurate and robust predictive models.

2.3 Sequence Representation

A straightforward method to handle SNP data is to directly model loci using additive encoding, which is widely adopted by seminal methods [VanRaden, 2008; Aguilar *et al.*, 2010; Mittag *et al.*, 2015]. But, it is overly rigid to represent genotypes by the number of non-reference alleles.

LETTER	A	T	C	G	Y	K	W	R	S	M	N
A	2	0	0	0	0	0	1	1	0	1	-
T	0	2	0	0	1	1	1	0	0	0	-
C	0	0	2	0	1	0	0	0	1	1	-
G	0	0	0	2	0	1	0	1	1	0	-
N	0	0	0	0	0	0	0	0	0	0	1/2

Table 2: Coding rule for SNP data. **A**, **T**, **C**, and **G** represent four basic nucleotides, while **N** represents an undetermined nucleotide that was not sequenced. Given a reference DNA sequence and a sample DNA sequence, at each nucleotide position, we use a set of LETTERs to represent possible combinations of two nucleotides. The numbers represent the frequency of occurrences. For example, A represents the combination **AA** and W represents the combination **AT** (**AT** and **TA** are equivalent). N means there is at least one **N**.

An emerging research topic is to leverage language models to model DNA data. However, unlike human language, where each word carries rich meaning, DNA's lexical units consist of only four basic nucleotide bases (**A**, **T**, **C**, **G**) with relatively ambiguous meanings, making them lower-level abstractions. For example, DNABERT [Ji *et al.*, 2021] and DNAGPT [Zhang *et al.*, 2023] have shown that encoding DNA sequences into k-mer patterns can improve the performance on downstream tasks without losing information. DNABERT-2 [Zhou *et al.*, 2023] uses random masking for unsupervised pre-training on DNA sequences. However, as there are fundamental differences between DNA and SNP sequences, it is still unclear whether these NLP techniques work on SNP data. In this work, we present the first empirical study.

3 Problem Formulation

We formalize the genomic selection problem as a mapping learning task. Given a genotype-phenotype pair (x, y), the final goal is to learn a mapping (or function) $f : x \mapsto y$. In this work, the genotype data x is the raw SNP sequence. Let $S = \{s_1, s_2, \ldots, s_{N_l}\}$ denote the SNP sequence, where N_l is the sequence length. We have $s_i \in \{\mathtt{A}, \mathtt{T}, \mathtt{C}, \mathtt{G}, \mathtt{N}, \mathtt{Y}, \mathtt{K}, \mathtt{W}, \mathtt{R}, \mathtt{S}, \mathtt{M}\}$, where different letters represent different combinations of alleles [Johnson, 2010]. In this work, the definition of each letter is presented in Tab. 2. The phenotype data y represents the corresponding phenotype values, which can either be discrete or continuous, depending on the task of interest (*i.e.* classification or regression).

Let N_s denote the number of samples for the species of interest in a crop dataset. In contrast to common machine learning tasks, we have $N_s \ll N_l$. This poses a non-trivial challenge for overfitting-prone methods such as Transformer, and thus Transformer cannot be directly used.

4 Methodology

The focus of our framework is to mitigate the limitations of Transformer and capture the contextual information and language patterns of SNP sequence. To this end, we first preprocess raw SNP sequence by an initial mapping which does

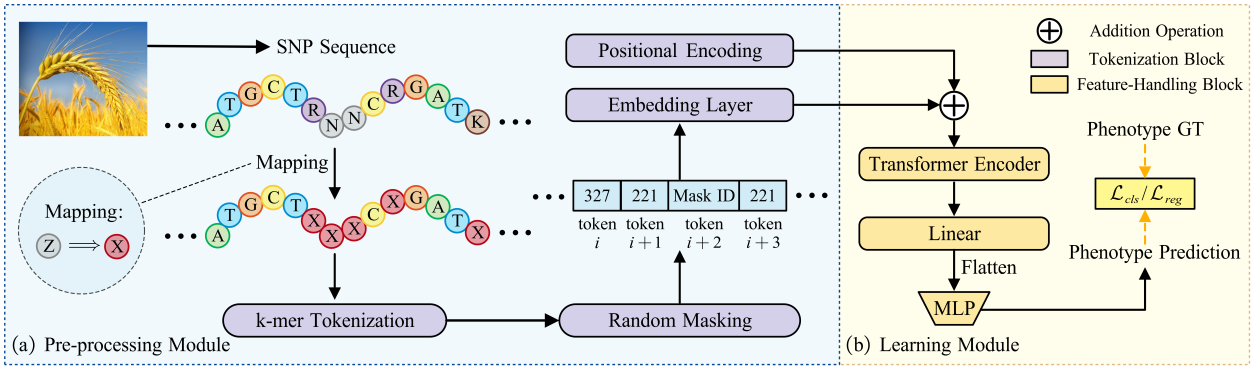

Figure 1: Illustration of the proposed framework, consisting of two modules, (a) a pre-processing module and (b) a learning module. (a) The raw SNP sequence is first pre-processed by a many-to-one mapping rule. Let Z denote an arbitrary index that does not belong to the set, *i.e.*, {A, T, C, G}, which are the four most frequent letters in SNP sequence. Z is mapped to X to reduce computational cost while it does not influence performance. The pre-processed sequence is then input into the tokenizer composed of k-mer and random masking to get token ID sequence and then the embedding layer. (b) Transformer encoder and MLP layers are adopted on embedding vectors to predict phenotype.

not change its length. After that, we implement k-mer tokenization and random masking to convert SNP sequence to token ID sequence. Then, embedding vectors are generated by the embedding layer and positional encoding [Vaswani *et al.*, 2017]. Finally, the embedding vectors are mapped to phenotype predictions by a neural network. The overall end-to-end training architecture is elaborated in Fig. 1.

4.1 Pre-processed SNP Sequence

In the candidate list of the sequence, {A, T, C, G} is the majority and {N, Y, K, W, R, S, M} is the minority. To avoid the difficulty of excessively large k-mer vocabularies contributed by the minority and improve the computational efficiency of subsequent self-attention, we generate the pre-processed SNP sequence S^p by the mapping rule:

$$s_i^p = \begin{cases} s_i, & s_i \in \{\mathtt{A},\mathtt{T},\mathtt{C},\mathtt{G}\} \\ \mathtt{X}, & s_i \notin \{\mathtt{A},\mathtt{T},\mathtt{C},\mathtt{G}\} \end{cases} \tag{1}$$

4.2 Sequence Tokenizer

Considering the biological significance of SNP sequences and mitigating the attention computational cost on long sequences, we choose non-overlapping k-mer as a tokenization technique on pre-processed SNP sequence S^p. Let $T = \{\dots, \underline{t_i}, t_{i+1}, t_{i+2}, \dots\}$ denote the token ID sequence processed by k-mer operation. Here is an illustration for a 4-mer operation with simulated sequence and token IDs.

$$\begin{aligned} T &= mer(S^p, k=4) \\ &= \{\dots, \underline{\mathtt{A},\mathtt{C},\mathtt{G},\mathtt{T}}, \underline{\mathtt{X},\mathtt{A},\mathtt{A},\mathtt{C}}, \underline{\mathtt{G},\mathtt{T},\mathtt{X},\mathtt{A}}, \dots\} \\ &= \{\dots, \underline{39}, \underline{502}, \underline{346}, \dots\} \in \mathbb{R}^{1\times\lfloor\frac{N_l}{k}\rfloor} \end{aligned}$$

$mer = (\cdot, k)$ is a function that treats the substring of length k as a whole entity and converts it to a unique ID. Although k-mer shortens the SNP sequence to $1/k$ of original length N_l, the problem of $N_s \ll N_l$ still exists.

We introduce randomness by *random masking* on token ID sequence to reduce the risk of overfitting:

$$T^m = Mask(T) = \{\dots, t_i^m, t_{i+1}^m, t_{i+2}^m, \dots\}. \tag{2}$$

$Mask(\cdot)$ is a function that has the pre-defined probability p to convert the original token ID to $mask\_id$, which is defined separately. For the i^{th} position in T^m, we have

$$t_i^m = Mask(t_i) = \begin{cases} t_i, & p \\ mask\_id, & 1-p \end{cases} \tag{3}$$

The final embedding vector is the sum of the embedded masked token ID sequence and the positional encoding.

$$E_0 = Embed(T_m) + E_{pos} \tag{4}$$

$Embed(\cdot)$ denotes the embedding layer, where the vocabulary size is 5^k+1. The "5^k" represents the number of all possible combinations and "1" denotes the specific Mask ID for unknown cases. $E_{pos} \in \mathbb{R}^{1\times\lfloor\frac{N_l}{k}\rfloor\times D_w}$ is the sinusoidal positional encoding [Vaswani *et al.*, 2017], where D_w denotes embedding width.

4.3 Phenotype Learning

We use Transformer encoder to learn the contextual information of encoded SNP sequences, which consists of L layers of multi-head self-attention (MSA) [Vaswani *et al.*, 2017] and multi-layer perceptron (MLP) blocks.

$$E_l' = MSA(LN(E_{l-1})) + E_{l-1} \tag{5}$$

$$E_l = MLP(LN(E_l')) + E_l' \tag{6}$$

$LN(\cdot)$ is the layer normalization [Vaswani *et al.*, 2017].

After computing the attention, a linear projector $LP(\cdot)$ is used to map the output to a lower-dimensional space, thus lowering the computational complexity in the regressor:

$$E_P = LP(E_L) \tag{7}$$

Finally, the phenotype prediction y_p can be output as:

$$y_p = MLP(Flatten(E_P)), \tag{8}$$

where the projected vector is flattened and inputted into MLP layer to predict the phenotype.

There are two types of learning tasks in predicting phenotype from genotype information: variety recognition and

Phenotype	Description
APCO_REV_REPRO	Apiculus color at reproductive
CUST_REPRO	Culm strength at reproductive - cultivated
LPCO_REV_POST	Lemma and palea color at post-harvest
LSEN	Leaf senescence
PEX_REPRO	Panicle exsertion at reproductive
PTH	Panicle threshability

Table 3: Description of each phenotype of interest in the *rice3k* dataset.

Phenotype	Description
COLD	Resistance score to cold
FHB	Resistance score to red rot disease
LODGING	Lodging severity
STERILSPIKE	Number of sterile spikes
TKW	Thousand kernel weight
YIELD	Wheat yield

Table 4: Description of each phenotype of interest in the *wheat3k* dataset.

trait prediction, corresponding to classification and regression tasks, respectively. For the classification task, we use the cross entropy (CE) as the loss function:

$$\mathcal{L}_{cls} = CE(y_p, y_l), \tag{9}$$

where y_l denotes the label.

For the regression task, we define the loss function by Mean Squared Error (MSE):

$$\mathcal{L}_{reg} = MSE(y_p, y_l). \tag{10}$$

5 Experiments

5.1 Setup

Datasets We use two crop datasets: *rice3k* [Wang *et al.*, 2018] and *wheat3k*, representing the two most significant staple crops worldwide. Each dataset includes raw SNP sequences and multiple traits of interest (phenotype).

rice3k is a public dataset that comprises genomic data and the corresponding phenotype data for a total of 3024 samples collected from 89 countries. For genomic data, the SNP sequence for each sample has a length of 404388. *rice3k* encompasses multiple quality traits, *i.e.* discrete variables. We select six valuable and well-structured phenotypes for experimentation, including APCO_REV_REPRO, CUST_REPRO, LPCO_REV_POST, LSEN, PEX_REPRO, and PTH. The definitions of these phenotypes are described in Tab. 3.

wheat3k is a private dataset to be released. *wheat3k* is collected following the setup of *rice3k* to provide a comprehensive study on wheat. It comprises genomic data and the corresponding phenotype data for a total of 3032 samples. Each SNP sequence for each sample has a length of 201740. In contrast to *rice3k*, *wheat3k* focuses on quantitative traits, *i.e.* continuous variables. We select six valuable and well-structured phenotypes for experimentation, including COLD, FHB, LODGING, STERILSPIKE, TKW, and YIELD. The definitions of these phenotypes are described in Tab. 4.

Implementation We implement the proposed framework using PyTorch and conduct experiments on NVIDIA 3090 GPU. There are a total 80 of epochs with early stops. The dimension of the sequence embedding width after embedding layer is set as 32 (*i.e.*, $D_w = 32$). Note that the vocabulary size of embedding layer is $5^k + 1$, where 1 additional ID for unknown words. Following the settings in DNABERT [Ji *et al.*, 2021], we use 6-mer (*i.e.*, $k = 6$) and the default random masking ratio is 15% (*i.e.*, $p = 0.15$). We use 3 Transformer encoder blocks in the learning module and an Adam optimizer [Kingma and Ba, 2014] with an initial learning rate of 0.0001 and weight decay of 0.01. All experiments are repeated for five-fold cross-validation, following the standard practice in GS. The reported numbers are the average metrics and standard deviations. Following [Wang *et al.*, 2023; Wu *et al.*, 2024], each phenotype of interest is considered an independent task for training.

Evaluation Metrics Following the setups of [Wang *et al.*, 2023; Wu *et al.*, 2024], we adopt accuracy (ACC) as the evaluation metric for the classification task (*rice3k* dataset) and Pearson correlation coefficient (PCC) to evaluate the regression performance (*wheat3k* dataset) of our model. PCC is defined as.

$$PCC = \frac{\sum_{i=1}^{n}(y_{p,i} - \bar{y}_p)(y_{l,i} - \bar{y}_l)}{\sqrt{\sum_{i=1}^{n}(y_{p.i} - \bar{y}_p)^2 \sum_{i=1}^{n}(y_{l,i} - \bar{y}_l)^2}}. \tag{11}$$

Baselines To demonstrate the robustness of our method, we select four seminal methods for comparison, including GBLUP [VanRaden, 2008; Yin *et al.*, 2023], RidgeClassifier [Quyet Nguyen *et al.*, 2024], BayesA [Meuwissen *et al.*, 2001; Lilin *et al.*, 2022], DNNGP [Wang *et al.*, 2023], and DLGWAS [Liu *et al.*, 2019]. GBLUP is a representative BLUP-based method and performs well across various regression tasks. To suit the classification tasks of *rice3k*, we replace GBLUP with RidgeClassifier, which shares similar statistical assumptions. BayesA is a robust Bayesian method with simple assumptions. We conduct 1500 iterations to ensure convergence. DNNGP and DLGWAS are two state-of-the-art deep learning methods. We use the source code of DNNGP[2]. We pre-process SNP sequences using PCA and retain 3000 principal components to preserve 95% variance information. We implement DLGWAS as an integration of GWAS and Transformer. GWAS uses the *mixed linear model* [Muhammad *et al.*, 2021] to analyze the correlation of SNP loci, retaining the top 10% of loci based on effect size. After locus screening, the relative order of loci retained in the original sequence is preserved for Transformer modeling. The model configuration information is kept as consistent as possible for all baselines.

5.2 Results

Performance on Classification Tasks Tab. 5 elaborates the classification results of different models on *rice3k* dataset. On the phenotypes of APCO_REV_REPRO, CUST_REPRO, LPCO_REV_POST, LSEN, PEX_REPRO, and PTH, our model achieves the best performance compared to the existing models in ACC metric. On LPCO_REV_POST, we also

[2]https://github.com/AIBreeding/DNNGP

Method	APCO_REC_REPRO	CUST_REPRO	LPCO_REV_POST	LSEN	PEX_REPRO	PTH
RidgeClassifier	0.4872±0.0116	0.4472±0.0192	**0.5930±0.0163**	0.4858±0.0120	0.5820±0.0145	0.5106±0.0237
BayesA	0.4597±0.0154	0.3521±0.0169	0.5113±0.0176	0.4133±0.0132	0.5764±0.0101	0.4975±0.0178
DNNGP	0.2808±0.0166	0.2441±0.0146	0.2755±0.0121	0.2272±0.0108	0.1430±0.0174	0.1675±0.0258
DLGWAS	0.4863±0.0154	0.4538±0.0126	0.5826±0.0109	0.4955±0.0126	0.6139±0.0059	0.5346±0.0266
Ours	**0.5018±0.0180**	**0.4685±0.0135**	0.5922±0.0103	**0.5058±0.0108**	**0.6189±0.0059**	**0.5422±0.0242**

Table 5: Comparison with state-of-the-art methods on the *rice3k* dataset. On average, our method is 2.06% superior to the best-performing statistical baseline RidgeClassifier and 1.05% superior to the best-performing deep learning baseline DLGWAS on average across 6 traits.

Method	COLD	FHB	LODGING	STERILSPIKE	TKW	YIELD
GBLUP	0.3587±0.0324	0.2447±0.0486	0.6195±0.0178	0.3983±0.0359	0.5676±0.0204	0.5896±0.0355
BayesA	0.4691±0.0194	0.3027±0.0451	0.8041±0.0138	0.6813±0.0215	0.7462±0.2147	0.6968±0.0278
DNNGP	0.2687±0.0282	0.0122±0.0427	0.5029±0.0200	0.3089±0.0304	0.5698±0.2315	0.0926±0.0192
DLGWAS	0.4781±0.0085	0.3354±0.0524	0.8166±0.0203	0.6794±0.0154	0.7395±0.0255	0.7000±0.0223
Ours	**0.4937±0.0120**	**0.3408±0.0534**	**0.8270±0.0136**	**0.6862±0.0175**	**0.7541±0.0213**	**0.7163±0.0159**

Table 6: Comparison with state-of-the-art methods on the *wheat3k* dataset. On average, our method is 1.97% superior to the best-performing statistical baseline BayesA and 1.15% superior to the best-performing deep learning baseline DLGWAS on average across 6 traits.

Method		FHB	STERILSPIKE	YIELD
DLGWAS		0.3354±0.0524	0.6794±0.0154	0.7000±0.0223
k-mer	Random Masking			
		0.3216±0.0360	0.5881±0.0098	0.6663±0.0201
	✓	0.3290±0.0406	0.5921±0.0090	0.6699±0.0205
✓		0.3340±0.0539	0.6723±0.0179	0.7044±0.0123
✓	✓	**0.3408±0.0534**	**0.6862±0.0175**	**0.7163±0.0159**

Table 7: Ablation study of the different component combinations on the *wheat3k* dataset. In this table, we use 6-mer, and random masking proportion is set as 15%. We find both components are beneficial to enhance model capacity, while k-mer is more important.

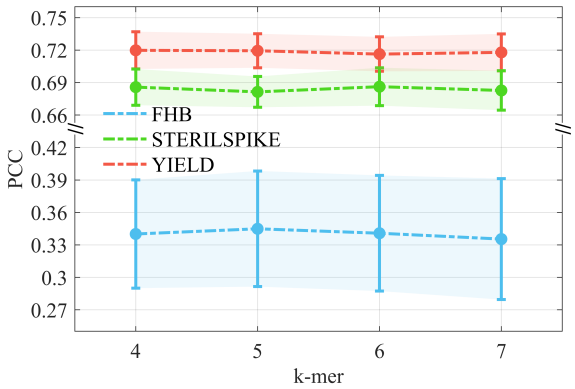

Figure 2: Sensitivity analysis on k for k-mer tokenization on the *wheat3k* dataset. The optimal value of k might be dependent on the task (phenotype).

obtain competitive results. Specifically, we surpass the current best-performing method DLGWAS over 1.05% across these 6 traits on average. The robustness and advancement of our simple approach to enhancing Transformer are evidenced by the overall elevation in mean values and reduction in standard deviation. These outcomes position our method as a pioneering benchmark in the field.

Performance on Regression Tasks We further analyze each baseline's capacity on *wheat3k* dataset, tabulated in Tab. 6. Similar to the outstanding performance on the *rice3k* dataset, our model also achieves the best performance compared to existing methods, where the higher PCC metric and lower standard deviation clearly prove the superiority. On YIELD task, which is important as it signifies the meaning of wheat yield, our method is 1.63% superior to the current state-of-the-art DLGWAS, which holds the potential in the application of genomic selection for crop breeding.

5.3 Ablation Studies

To investigate the contribution of each key component of our proposed method, we conduct a series of ablation experiments on the *wheat3k* dataset.

Analysis of Pre-Processing We conduct ablation experiments to have a deep insight into key components in the tokenization. Note, we choose the model without pre-processing module as baseline to study the impact of each component in Tab. 7. It is found that k-mer plays a more significant role than random masking, which enhances the performance of baseline model by 1.24%, 8.42%, and 3.81% in PCC on three sub-tasks of *wheat3k* dataset. This suggests that using k-mer in pre-processing can assist attention mechanisms to better capture contextual information. Another finding is that when utilizing k-mer as tokenizer, random masking can yield greater improvements than using random masking on baseline alone. The difference lies in whether the smallest unit of the masking is 1 bit or k bits, which relates to the influence of masking on context. This proves the importance of context extraction on k-mer. Compared with DLGWAS, Transformer without k-mer and random masking shows inferior performance and our model enhances the performance. This proves the validity of key components.

Analysis of k-mer Tokenization We study the influence of k in k-mer tokenizer in Fig. 2. Note, when $k = 5$, our model obtains the best results on the FHB and YIELD traits, but

Method	Vocab Size	FHB	STERILSPIKE	YIELD
Character-level	5 + 1 (for unknown)	0.3216±0.0360	0.5881±0.0098	0.6663±0.0201
Byte-Pair Encoding	8000	0.0562±0.0123	0.3159±0.0341	0.3656±0.0447
Learnable Tokenizer (MLP)	-	0.3092±0.0576	0.6669±0.0307	0.6747±0.0590
Learnable Tokenizer (Transformer)	-	0.3329±0.0560	0.6592±0.0219	0.7086±0.0166
Ours (5-mer+15% random masking)	5^5+ 1 (for unknown)	**0.3449±0.0534**	**0.6814±0.0142**	**0.7194±0.0157**

Table 8: Comparison of different tokenizer methods on the *wheat3k* dataset. In character-level tokenizer and our method, there exists an additional token in vocabulary for unknown cases. Under 5-mer and 15% random masking, our model presents 1.20%, 2.22%, and 1.08% improvements over the state-of-the-art tokenizers on FHB, STERILSPIKE, and YIELD, respectively. In this table, we set $k = 5$ in k-mer for our model in comparison as it holds the overall best performance in Fig. 2.

Proportion	FHB	STERILSPIKE	YIELD
0%	0.3340±0.0539	0.6723±0.0179	0.7044±0.0123
15%	0.3408±0.0534	0.6862±0.0175	0.7163±0.0159
30%	**0.3456±0.0509**	**0.6883±0.0188**	**0.7169±0.0134**
45%	0.3394±0.0533	0.6861±0.0170	0.7147±0.0165
60%	0.3380±0.0508	0.6738±0.0204	0.7093±0.0170

Table 9: Impact of the masking proportion for the capacity of our method on the *wheat3k* dataset. When 30% tokens are masked for a SNP sequence, our model achieves the best performance.

the best results on the STERILSPIKE occur at $k = 6$. We hypothesize that the value of k might be task-oriented.

Analysis on Random Masking We conduct experiments on random masking proportion to study the effect of randomness in the tokenization. In Tab. 9, the PCC value of three subtasks improves as the random masking proportion increases, reaching its peak value when the proportion equals to 30%. Through randomly masking tokens, the data diversity is enhanced, thereby promoting the model's generalization ability and alleviating overfitting issues arising from data scarcity, especially in our problem formulation ($N_s \ll N_l$). When the masking ratio becomes too high, *i.e.*, 45%, it can lead to diminishing returns and worsen the effectiveness of the model because excessively masking a large portion of tokens may disrupt the coherence of the input sequence to make it harder for the model to learn meaningful representations.

Choice of Tokenizer We compare our method with seminal tokenization algorithms to demonstrate the effectiveness of k-mer and random masking, shown in Tab. 8. Character-level tokenizer treats each letter in SNP sequence as an independent feature, where we use $\{\texttt{A}, \texttt{T}, \texttt{C}, \texttt{G}, \texttt{X}\}$ and one additional ID for unknown words as the vocabulary. Byte-Pair Encoding builds a sub-word vocabulary of size 8000. We also implement two learnable tokenizers: one MLP layer and one Transformer encoder, where the input is SNP sequence and the embedding width of encoding output is 20. Compared with the second-best tokenizer (Learnable Transformer), our method achieves 1.20%, 2.22%, and 1.08% PCC improvements on FHB, STERILSPIKE, and YIELD, respectively, providing better SNP sequence representation for learning module.

Computational Cost Lastly, to comprehensively assess a model, computational cost is considered. Take *wheat3k* as an example, the original sequence length is 201740. Tabulated in Tab. 10, we utilize parameter count (Parameters), GPU memory consumption (Memory), and inference time (Time) to measure all models. GBLUP shows fast inference time as a lightweight linear model. But it sacrifices the degree of freedom, thus leading to lower accuracy in Tab. 6. After PCA, DNNGP only retains 3000 principal components for CNN, thus generating low computational costs. However, DNNGP reports the lowest performance on both two datasets. For *vanilla* Transformer without tokenization, the original SNP sequence of length 201740 is input into attention mechanism. For DLGWAS and Ours (6-mer), the inputs for both Transformer encoder consist of 33623 dimensions. For Ours (5-mer), this value is 40348. Compared with the statistical method BayesA, our method provides a quicker inference time, and compared with *vanilla* Transformer, our method efficiently reduces GPU memory cost.

Method	Parameters (M)	Memory (MB)	Time (ms)
GBLUP	-	-	11.00
BayesA	-	-	1143.00
DNNGP	0.35	0.36	0.40
DLGWAS	34.54	185.31	27.10
Transformer	206.69	1514.94	899.22
Ours (5-mer)	41.52	246.16	39.29
Ours (6-mer)	35.04	205.68	26.91

Table 10: Computational cost of the state-of-the-art methods on *wheat3k* dataset. Mentioned that our computational cost are statistically based on individual samples. Compared with the statistical method BayesA and *vanilla* Transformer (without tokenization), our method successfully reduces computation cost.

6 Conclusion

In this study, we present a simple yet effective approach to enhance Transformer's performance in the realm of genomic selection for crop breeding. Specifically, we propose the preprocessing module composed of k-mer tokenizer and random masking to assist contextual understanding of SNP sequence. Experiments on the *rice3k* and *wheat3k* datasets demonstrate promising performance on genotype-to-phenotype prediction. The empirical findings of this work not only suggest the potential of DL to handle long sequences but also pose a new research direction on end-to-end genomic selection.

Meanwhile, we shall notice that though tokenization plays an important role in genomic selection, a comprehensive understanding is still needed in future work.

Acknowledgements

This work was supported by Shanghai Artificial Intelligence Laboratory and the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No.XDA0450203.

Contribution Statement

Renqi Chen, Wenwei Han, and Haohao Zhang have equal contributions. Nanqing Dong is the corresponding author.

References

[Aguilar *et al.*, 2010] I Aguilar, I Misztal, DL Johnson, Andres Legarra, S Tsuruta, and TJ Lawlor. Hot topic: a unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of holstein final score. *Journal of dairy science*, 93(2):743–752, 2010.

[Boichard *et al.*, 2012] Didier Boichard, Francois Guillaume, Aurélia Baur, Pascal Croiseau, Marie-Noelle Rossignol, Marie Yvonne Boscher, Tom Druet, LUCIE Genestout, JJ Colleau, L Journaux, et al. Genomic selection in french dairy cattle. *Animal Production Science*, 52(3):115–120, 2012.

[Dao *et al.*, 2022] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[De Los Campos *et al.*, 2009] Gustavo De Los Campos, Hugo Naya, Daniel Gianola, José Crossa, Andrés Legarra, Eduardo Manfredi, Kent Weigel, and José Miguel Cotes. Predicting quantitative traits with regression models for dense molecular markers and pedigree. *Genetics*, 182(1):375–385, 2009.

[Endelman, 2011] Jeffrey B Endelman. Ridge regression and other kernels for genomic selection with r package rrblup. *The plant genome*, 4(3), 2011.

[García-Ruiz *et al.*, 2016] Adriana García-Ruiz, John B Cole, Paul M VanRaden, George R Wiggans, Felipe J Ruiz-López, and Curtis P Van Tassell. Changes in genetic selection differentials and generation intervals in us holstein dairy cattle as a result of genomic selection. *Proceedings of the National Academy of Sciences*, 113(28):E3995–E4004, 2016.

[Goddard *et al.*, 2011] Michael E Goddard, Ben J Hayes, and Theo HE Meuwissen. Using the genomic relationship matrix to predict the accuracy of genomic selection. *Journal of animal breeding and genetics*, 128(6):409–421, 2011.

[Gualdrón Duarte *et al.*, 2020] José Luis Gualdrón Duarte, Ann-Stephan Gori, Xavier Hubin, Daniela Lourenco, Carole Charlier, Ignacy Misztal, and Tom Druet. Performances of adaptive multiblup, bayesian regressions, and weighted-gblup approaches for genomic predictions in belgian blue beef cattle. *BMC genomics*, 21(1):1–18, 2020.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[Ji *et al.*, 2021] Yanrong Ji, Zhihan Zhou, Han Liu, and Ramana V Davuluri. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112–2120, 2021.

[Johnson, 2010] Andrew D Johnson. An extended iupac nomenclature code for polymorphic nucleic acids. *Bioinformatics*, 26(10):1386–1389, 2010.

[Kärkkäinen and Sillanpää, 2012] Hanni P Kärkkäinen and Mikko J Sillanpää. Back to basics for bayesian model building in genomic selection. *Genetics*, 191(3):969–987, 2012.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Kingsolver and Pfennig, 2007] Joel G Kingsolver and David W Pfennig. Patterns and power of phenotypic selection in nature. *Bioscience*, 57(7):561–572, 2007.

[Lilin *et al.*, 2022] Yin Lilin, Zhang Haohao, Li Xinyun, Zhao Shuhong, and Liu Xiaolei. hibayes: An r package to fit individual-level, summary-level and single-step bayesian regression models for genomic prediction and genome-wide association studies. *bioRxiv*, 2022.

[Liu *et al.*, 2019] Yang Liu, Duolin Wang, Fei He, Juexin Wang, Trupti Joshi, and Dong Xu. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. *Frontiers in genetics*, 10:1091, 2019.

[Ma *et al.*, 2017] Wenlong Ma, Zhixu Qiu, Jie Song, Qian Cheng, and Chuang Ma. Deepgs: Predicting phenotypes from genotypes using deep learning. *BioRxiv*, page 241414, 2017.

[Manor and Segal, 2013] Ohad Manor and Eran Segal. Predicting disease risk using bootstrap ranking and classification algorithms. *PLoS computational biology*, 9(8):e1003200, 2013.

[Meuwissen *et al.*, 2001] Theo HE Meuwissen, Ben J Hayes, and ME1461589 Goddard. Prediction of total genetic value using genome-wide dense marker maps. *genetics*, 157(4):1819–1829, 2001.

[Mittag *et al.*, 2015] Florian Mittag, Michael Römer, and Andreas Zell. Influence of feature encoding and choice of classifier on disease risk prediction in genome-wide association studies. *PloS one*, 10(8):e0135832, 2015.

[Muhammad *et al.*, 2021] Ali Muhammad, Jianguo Li, Weichen Hu, Jinsheng Yu, Shahid Ullah Khan, Muhammad Hafeez Ullah Khan, Guosheng Xie, Jibin Wang, and Lingqiang Wang. Uncovering genomic regions controlling plant architectural traits in hexaploid wheat using different gwas models. *Scientific reports*, 11(1):6767, 2021.

[Quyet Nguyen *et al.*, 2024] Chien Quyet Nguyen, Tuyen Thi Tran, Trang Thanh Thi Nguyen, Thuy Ha Thi Nguyen, et al. Mapping of soil erosion susceptibility using advanced machine learning models at nghe an, vietnam. *Journal of Hydroinformatics*, 26(1):72–87, 2024.

[Ramakrishnan *et al.*, 2022] Muthusamy Ramakrishnan, Lakkakula Satish, Anket Sharma, Kunnummal Kurungara Vinod, Abolghassem Emamverdian, Mingbing Zhou, and Qiang Wei. Transposable elements in plants: Recent advancements, tools and prospects. *Plant Molecular Biology Reporter*, 40(4):628–645, 2022.

[Ribaut and Hoisington, 1998] Jean-Marcel Ribaut and David Hoisington. Marker-assisted selection: new tools and strategies. *Trends in Plant Science*, 3(6):236–239, 1998.

[Siepielski *et al.*, 2013] Adam M Siepielski, Kiyoko M Gotanda, Michael B Morrissey, Sarah E Diamond, Joseph D DiBattista, and Stephanie M Carlson. The spatial patterns of directional phenotypic selection. *Ecology letters*, 16(11):1382–1392, 2013.

[Tipping and Faul, 2003] Michael E Tipping and Anita C Faul. Fast marginal likelihood maximisation for sparse bayesian models. In *International workshop on artificial intelligence and statistics*, pages 276–283. PMLR, 2003.

[United Nations, 2023a] United Nations. 17 sustainable development goals. *https://sdgs.un.org/goals*, 2023.

[United Nations, 2023b] United Nations. Leave no one behind. *https://unsdg.un.org/2030-agenda/universal-values/leave-no-one-behind*, 2023.

[United Nations, 2023c] United Nations. *The State of Food Security and Nutrition in the World 2023*. FAO, Rome, 2023. Urbanization, agrifood systems transformation and healthy diets across the rural–urban continuum.

[VanRaden, 2008] Paul M VanRaden. Efficient methods to compute genomic predictions. *Journal of dairy science*, 91(11):4414–4423, 2008.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[Wang *et al.*, 2018] Wensheng Wang, Ramil Mauleon, Zhiqiang Hu, Dmytro Chebotarov, Shuaishuai Tai, Zhichao Wu, Min Li, Tianqing Zheng, Roven Rommel Fuentes, Fan Zhang, et al. Genomic variation in 3,010 diverse accessions of asian cultivated rice. *Nature*, 557(7703):43–49, 2018.

[Wang *et al.*, 2023] Kelin Wang, Muhammad Ali Abid, Awais Rasheed, Jose Crossa, Sarah Hearne, and Huihui Li. Dnngp, a deep neural network-based method for genomic prediction using multi-omics data in plants. *Molecular Plant*, 16(1):279–293, 2023.

[Wu *et al.*, 2024] Cuiling Wu, Yiyi Zhang, Zhiwen Ying, Ling Li, Jun Wang, Hui Yu, Mengchen Zhang, Xianzhong Feng, Xinghua Wei, and Xiaogang Xu. A transformer-based genomic prediction method fused with knowledge-guided module. *Briefings in Bioinformatics*, 25(1):bbad438, 2024.

[Xie *et al.*, 2023] Zhengchao Xie, Xiaogang Xu, Ling Li, Cuiling Wu, Yinxing Ma, Jingjing He, Sidi Wei, Jun Wang, and Xianzhong Feng. Residual networks without pooling layers improve the accuracy of genomic predictions. 2023.

[Xu and Crouch, 2008] Yunbi Xu and Jonathan H Crouch. Marker-assisted selection in plant breeding: From publications to practice. *Crop science*, 48(2):391–407, 2008.

[Yin *et al.*, 2023] Lilin Yin, Haohao Zhang, Zhenshuang Tang, Dong Yin, Yuhua Fu, Xiaohui Yuan, Xinyun Li, Xiaolei Liu, and Shuhong Zhao. HIBLUP: an integration of statistical models on the BLUP framework for efficient genetic evaluation using big genomic data. *Nucleic Acids Research*, 51(8):3501–3512, 02 2023.

[Zhang *et al.*, 2023] Daoan Zhang, Weitong Zhang, Bing He, Jianguo Zhang, Chenchen Qin, and Jianhua Yao. Dnagpt: A generalized pretrained tool for multiple dna sequence analysis tasks. *bioRxiv*, pages 2023–07, 2023.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11106–11115, 2021.

[Zhou *et al.*, 2023] Zhihan Zhou, Yanrong Ji, Weijian Li, Pratik Dutta, Ramana Davuluri, and Han Liu. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. *arXiv preprint arXiv:2306.15006*, 2023.