

# Kolmogorov-Arnold Networks for Genomic Tasks

Oleksandr Cherednichenko<sup>1</sup> Maria Poptsova<sup>1</sup>

## Abstract

Kolmogorov-Arnold Networks (KANs) emerged as a promising alternative for multilayer perceptrons in dense fully connected networks. Multiple attempts have been made to integrate KANs into various deep learning architectures in the domains of computer vision and natural language processing. Integrating KANs into deep learning models for genomic tasks has not been explored. Here, we tested linear KANs (LKANs) and convolutional KANs (CKANs) as replacement for MLP in baseline deep learning architectures for classification and generation of genomic sequences. We used three genomic benchmark datasets: Genomic Benchmarks, Genome Understanding Evaluation, and Flipon Benchmark. We demonstrated that LKANs outperformed both baseline and CKANs on almost all datasets. CKANs can achieve comparable results but struggle with scaling over large number of parameters. Ablation analysis demonstrated that the number of KAN layers correlates with the model performance. Overall, linear KANs show promising results in improving the performance of deep learning models with relatively small number of parameters. Unleashing KAN potential in different SOTA deep learning architectures currently used in genomics requires further research.

## 1. Introduction

Deep learning models have been successfully applied to a wide range of genomic tasks including prediction of variant effect (Zhou & Troyanskaya, 2015), (Avsec et al., 2021), and location of functional genomic elements, such as promoters (Oubounyt et al., 2019a), enhancers (Le et al., 2021), transcription factor binding sites (Quang & Xie, 2019), histone marks (Yin et al., 2021), (Wen et al., 2024), flipons (Bek-

nazarov et al., 2020), (Umerenkov et al., 2023), splice sites (de Sainte Agathe et al., 2023), and others. Neural networks also proved to be efficient in the field of RNA biology (see (Sato & Hamada, 2023) and (Hwang et al., 2024) for review) predicting RNA-protein binding (Shen et al., 2020), RNA structure (Franke et al., 2024), non-coding RNA (Georgakilas et al., 2020).

The evolution of deep learning applications in genomics followed the path of the evolution of deep learning architectures. It started with pioneering application of Convolutional Neural Networks (CNNs) (Zhou & Troyanskaya, 2015; Chen et al., 2022) and Recurrent Neural Networks (RNNs) to DNA sequences (Shen et al., 2018), and then continued with Transformer-based Large Language Models (LLMs) such as DNABERT (Ji et al., 2021), DNABERT2 (Zhou et al., 2024b), Nucleotide Transformer (Dalla-Torre et al., 2023). DNABERT was trained on one reference human genome while DNABERT2 and Nucleotide transformer were trained on the multi-species genome comprising up to 40B of bases pairs. After training foundation models are fine-tuned on different downstream tasks however even with parameter efficient fine-tuning techniques, resource capability can impose significant limitations on the task realization.

Transformers require substantial computational resources due to quadratic attention scaling, and the next architecture Hyena, based on long convolutions, incorporated longer context up to 1M nucleotides by scaling subquadratically, which resulted in HyenaDNA (Nguyen et al., 2023). Another promising alternative to overcome transformers' computational inefficiency is Mamba architecture (Gu & Dao, 2023) which lie at the basis of Caduceus model in genomics (Schiff et al., 2024). Both Hyena and Mamba achieved a certain tradeoff between computational resources and performance quality. In comparison to the latest genomic foundational model DNABERT2, HyenaDNA has 10x less parameters and on some genomic tasks outperforms DNABERT2.

With ever-growing size of deep learning architectures, it has been a challenge to find smaller architectures that would perform equally well. Here we aim to investigate the potential of small neural networks that utilize only few MLPs layers with recent innovations in deep learning such as Kolmogorov-Arnold Networks (KANs) (Liu et al., 2024b).

<sup>1</sup>International Laboratory of Bioinformatics, HSE University, Moscow, Russia. Correspondence to: Maria Poptsova <mpoptsova@hse.ru>.

## KANs for Genomics

The neural networks discussed above have one thing in common: they rely on the universal approximation theorem. One of the recent advances in deep learning architectures, Kolmogorov-Arnold Network (Liu et al., 2024b), leverages the Kolmogorov-Arnold theorem to incorporate splines into a neural network architecture, offering a compelling alternative to traditional Multi-Layer Perceptrons (MLPs). KANs have already been successfully applied in different areas such as mechanics (Liu et al., 2024a; Abueidda et al., 2024), computer vision (Bodner et al., 2024; Drokin, 2024; Yang & Wang, 2024; Aghaei, 2024; Cheon, 2024), NLP (Yu et al., 2024; Genet & Inzirillo, 2024b), time series (Genet & Inzirillo, 2024b;a; Zhou et al., 2024a), physics (Liu et al., 2024b;a; Ta, 2024), speech enhancement (Xu et al., 2024a), molecular representations (Li et al., 2024; Nagai & Okumura, 2024), and recommendation systems (Park et al., 2024). Inspired by this advancement, multiple modifications have emerged that attempt to overcome various issues associated with the spline-based approach, namely computational overhead and a large number of trainable parameters.

In the presented study we aim at evaluating the potential of KAN-based models in genomics. Given the limitations in computational resources, we applied KAN to simple convolutions and dense networks with a relatively small number of parameters. We tested KANs to the task of classification and generation by replacing layers in the baseline model with Linear KAN layers (LKAN) and Convolutional KAN layers (CKAN). The test was performed on three benchmark datasets: Genomic Benchmarks (Gresova et al., 2023), Genome Understanding Evaluation (GUE) and our collection of datasets for flipons. or non-B DNA structures (Herbert, 2024).

### 1.1. Kolmogorov-Arnold Networks

#### 1.1.1. LINEAR KOLMOGOROV-ARNOLD NETWORKS

Here we briefly introduce KAN and present key features of this architecture. Multi-Layer Perceptrons (Hornik et al., 1989) are inspired by the universal approximation theorem that states that a feed-forward network with a single hidden layer containing a finite number of neurons can approximate continuous functions on compact subsets of  $\mathbb{R}^d$ . Kolmogorov-Arnold Network (Liu et al., 2024b) focuses on the Kolmogorov-Arnold representation theorem (Kolmogorov, 1956) which states that any multivariate continuous function can be represented as a composition of univariate functions and the addition operation

$$f(\mathbf{x}) = \sum_{k=1}^{2n+1} \Phi_k \left( \sum_{l=1}^n \phi_{k,l}(x_l) \right) \quad (1)$$

where  $\phi_{k,l}$  are univariate functions that map each input variable  $x_l$  as follows:  $\phi_{k,l} : [0, 1] \rightarrow \mathbb{R}$ . The authors of

the original KAN paper proposed how to extend the KAN application to deep learning networks by stacking KAN layers.

$$f(\mathbf{x}) = \sum_{i_{L-1}=1}^{n_{L-1}} \phi_{L-1,i_{L-1}} \left( \sum_{i_{L-2}=1}^{n_{L-2}} \left( \sum_{i_0=1}^{n_0} \phi_{0,i_1,i_0}(x_{i_0}) \right) \right) \quad (2)$$

Similarly it is possible to rewrite it in a way

$$\text{KAN}(\mathbf{x}) = (\Phi_{L-1} \circ \dots \circ \Phi_1 \circ \Phi_0)\mathbf{x} \quad (3)$$

Implementation of Kolmogorov-Arnold Networks is provided by the authors of (Liu et al., 2024b), however there are several aspects that could limit the scalability of KANs: a large number of trainable parameters, training time, and inference time. On top of that, in (Yu et al., 2024) the authors claim that weights have different initialization that makes it impossible to keep variance-preserving initialization (Yang & Wang, 2024). To overcome these issues, in (Yang & Wang, 2024) the authors suggest an efficient implementation of KANs with the straightforward matrix multiplication and L1 regularization for the model's weights.

#### 1.1.2. KOLMOGOROV-ARNOLD CONVOLUTIONS

Kolmogorov-Arnold Convolutions are proposed in (Bodner et al., 2024) and designed to be similar to CNNs. The difference is that the convolutional layers are replaced by KAN convolutional layers, and, after flattening, one can use either KAN or MLP. The advantage of the Convolutional KANs is that they have significantly fewer parameters compared to other architectures. This is provided by the network architecture, because B-Splines are capable of smooth representations of arbitrary activation functions, which cannot be determined with ReLU. The authors concluded that Kolmogorov-Arnold Convolutions are capable of achieving high performance as compared to original convolutional networks (Drokin, 2024).

### 1.2. Genomics Tasks

#### 1.2.1. DNA CLASSIFICATION

Classification of DNA sequences is one of the most common tasks in genomics solved with deep learning models. In this work, we perform classification on three large benchmark datasets: Genomic Benchmarks (Gresova et al., 2023), Genomic Understanding Evaluation (Zhou et al., 2024b), and Flipon Benchmark assembled by us. For each of the datasets there available a train and a test sets.

#### 1.2.2. DNA GENERATION

Another important application of deep learning in genomics is generative modeling. Generative models are usually used in data augmentation approaches to generate new samples

from the learned distribution.

There are many types of generative models, but here we focus on the most popular models for DNA sequence generation: denoising diffusion implicit model (DDIM) (Song et al., 2021) and generative adversarial network (GAN) (Arjovsky et al., 2017). In generative approaches, first, a generative model is trained on a train set, and the trained model produces synthetic samples of the same size as the test set. Secondly, synthetic data is supplied to a classifier trained on the real data to evaluate how good are generative models in capturing main patterns of the real data.

We will use the Flipons collection (Kouzine, 2017; Shin, 2016; Qian, 2024; Zhang et al., 2019; Chambers et al., 2015; Lyu et al., 2016) for this task (see Methods).

## 2. Methods

### 2.1. Models

We use pytorch efficient implementation of KAN (EKAN) (Yang & Wang, 2024) as linear KAN layers. We use KAN-Conv (Bodner et al., 2024) implementation as KAN convolutional neural network. We incorporate KAN layers into convolutional neural network that takes its core from LegNet (Penzar et al., 2023). We compare the impact of KAN layers by replacing MLP modules with KANs (see Tables 1-5). To make a comparison more fair and to exclude model size effect, in all the experiments for classification we used models of equal size: 22M parameters. We used the following notation for our tested models: Baseline for LegNet-based convolutional neural network, LKAN for LegNet-based convolutional neural network (Tan & Le, 2024) with incorporated Linear KANs, and CKAN for LegNet-based convolutional neural network with incorporated Convolutional KANs. Since CKANs scales with larger grid size (see Results), combining CKAN and LKAN in our architecture was impossible due to limitations in resources, thus hybrid CKAN-LKAN architectures were not conducted.

In experiments with DNA sequence generation we utilized denoising diffusion implicit model (Song et al., 2021) designed specifically for DNA (DaSilva, 2024). We replaced linear layers with KAN layers in Unet like architecture.

### 2.2. Datasets

#### 2.2.1. GENOMIC BENCHMARKS

We use the following notations for all datasets: DMEE - dummy mouse enhancers ensemble, DCIS - demo coding vs intergenomic seqs, DHW - demo human or worm, DES - drosophila enhancers stark, HEC - human enhancers cohn, HEE - human enhancers ensembl, HER - human ensembl regulatory, HNP - human nontata promoters, HOE - human ocr ensembl.

#### 2.2.2. FLIPONS, OR NON-B DNA STRUCTURES

Inspired by Genomic Benchmarks we created a collection of datasets for detecting flipons (Herbert, 2024), or non-B DNA structures: Z-flipons, or Z-DNA (Kouzine, 2017; Shin, 2016); G-flipons, or G-quadruplexes (GQs) (Qian, 2024; Zhang et al., 2019; Chambers et al., 2015; Lyu et al., 2016); H-flipons, or H-DNA (triplexes) (Kouzine, 2017). We provide train and test sets trying to prevent data leakage. Full description of datasets are available in Supplementary Table 1. Similar to Genomic Benchmarks datasets we use the following notations for all flipons datasets: ENDO - Endoquad GQs; G4seq - G4-seq experimental dataset for GQs; G4ChIP - G4 ChIP-seq experimental data set for GQs; G4cut - G4 CUT&Tag experimental dataset for GQs; ZKOU - Kouzine et al. experimental dataset for Z-DNA; ZShin - Shin et al. ChIP-seq experimental dataset for Z-DNA; HKou - Kouzine experimental dataset for H-DNA.

#### 2.2.3. GENOME UNDERSTANDING EVALUATION

We follow (Zhou et al., 2024b) authors in evaluation on Genome Understanding Evaluation (GUE) and extended version (GUE+) to study models' performance on a wider range of genomic tasks: prediction of promoter, core promoter, splice sites, epigenetic marks, transcription factor binding sites on human and mouse, and covid variant classification.

## 3. Results

### 3.1. Benchmarks on DNA classification

#### 3.1.1. GENOMIC BENCHMARKS

KAN performance in comparison to the baseline model for nine datasets from Genomic Benchmarks (Gresova et al., 2023) are presented in Table 1 (MCC score averaged from 5 folds). Baseline CNN model is of equal size with LKAN and CKAN. Other classification metrics such as accuracy, ROC-AUC, precision, recall, and F1 are provided in Supplementary Tables 2-14. We observe that both LKAN and CCKAN improve the quality of the model, yet CKAN requires higher value of grid size parameter.

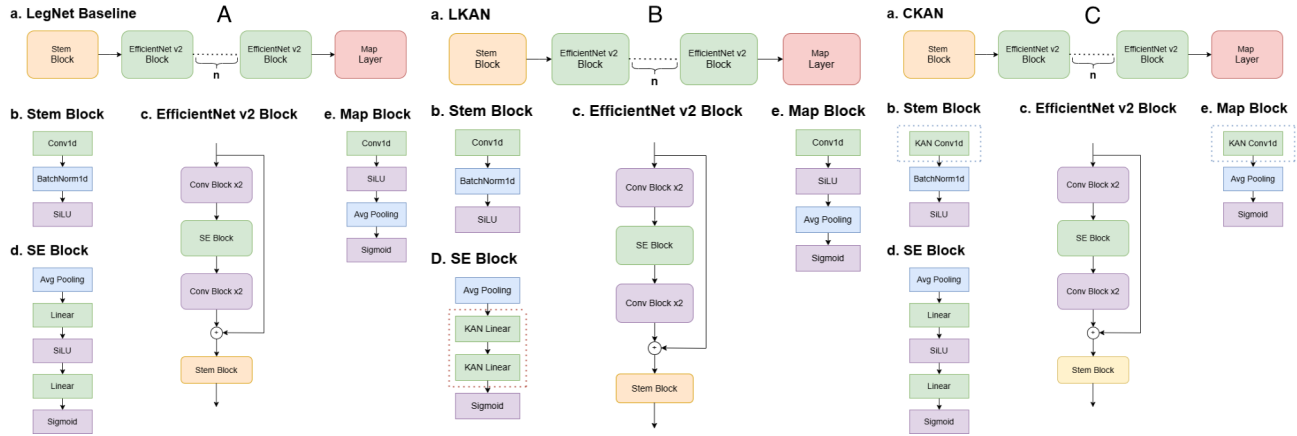
#### 3.1.2. FLIPONS, OR NON-B DNA STRUCTURES

KAN performance on seven datasets from Flipon collection (Kouzine, 2017; Shin, 2016; Qian, 2024; Zhang et al., 2019; Chambers et al., 2015; Lyu et al., 2016) is presented in Table 2. Baseline CNN model is of equal size with LKAN and CKAN.

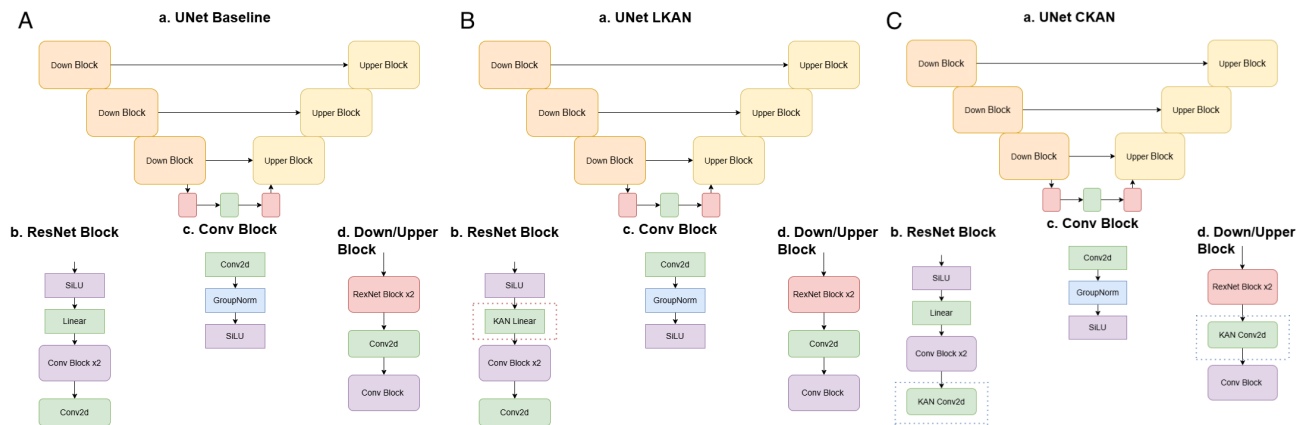
#### 3.1.3. GENOME UNDERSTANDING EVALUATION

Results of KAN performance in comparison to the baseline model for nine datasets from Genome Understanding Eval-

## KANs for Genomics



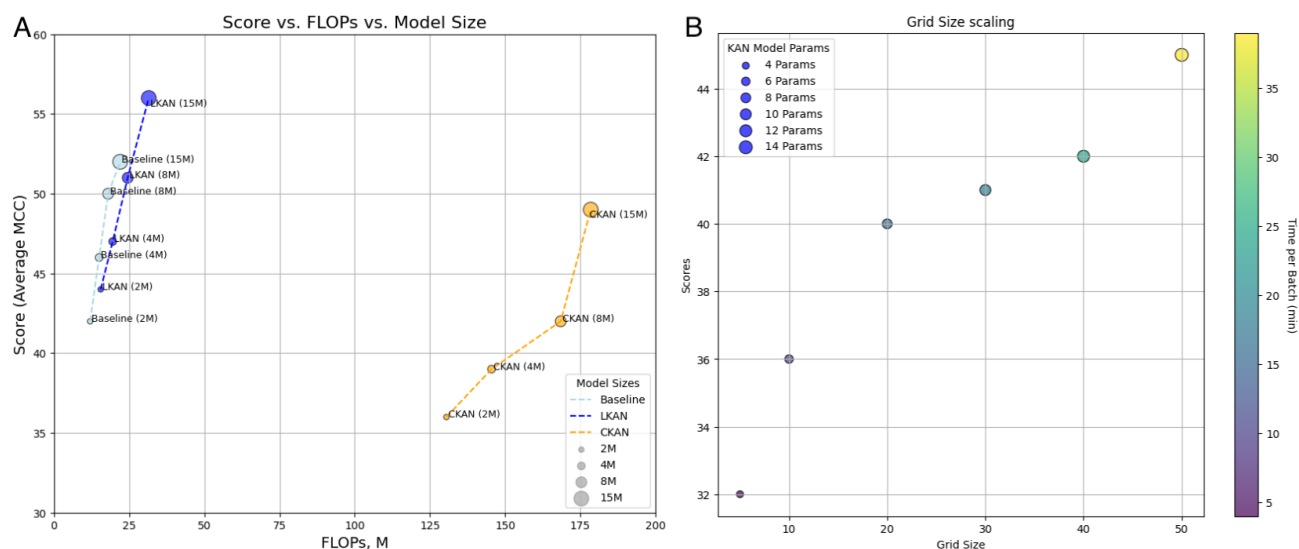
**Figure 1. A. LegNet-based Baseline.** a. Architecture of Baseline LegNet-based convolutional neural network. The Number of EfficientNet-like block are defined as  $n$  and we use  $n = 6$ . Other values were tested in Ablation study. b. Stem block incorporated in Baseline architecture. c. Main convolutional EfficientNet-like (Tan & Le, 2024) block incorporated in Baseline architecture. Plus sign refers to per channel concatenation. d. Mid SE block with bilinear layer incorporated in Baseline architecture. e. Final block incorporated in Baseline architecture to access final prediction. **B. LegNet-based LKAN.** a. Architecture of Baseline LegNet-based convolutional neural network with LKAN. The number of EfficientNet like block are defined as  $n$  and we use  $n = 6$ . Other values were tested in Ablation study. b. Stem block incorporated in Baseline architecture. c. Main convolutional EfficientNet-like (Tan & Le, 2024) block incorporated in Baseline architecture. Plus sign refers to per channel concatenation. d. Mid SE block with replaced LKAN layers incorporated in Baseline architecture. e. Final block incorporated in Baseline architecture to access a final prediction. **C. LegNet-based CKAN.** a. Architecture of Baseline LegNet-based convolutional neural network with CKAN. The number of EfficientNet-like block are defined as  $n$  and we use  $n = 6$ . Other values were tested in Ablation study. b. Stem block with replaced CKAN layer incorporated in Baseline architecture. c. Main convolutional EfficientNet-like (Tan & Le, 2024) block incorporated in Baseline architecture. Plus sign refers to per channel concatenation. d. Mid SE block with bilinear layer incorporated in Baseline architecture. e. Final block with replaced CKAN layer incorporated in Baseline architecture to access a final prediction.



**Figure 2. A. Unet-based Baseline** a. Architecture of Baseline DNA-Diffusion Unet-based model. Model's core consists of upper and down blocks. b. Base ResNet block gathered from Convolutional Blocks. c. Convolutional Block is incorporated into base ResNet block and at the end of Upper and Down blocks. d. Structure of Down and Upper Blocks with ResNet Blocks. **B. Unet-based LKAN** a. Architecture of Linear Kolmogorov Arnold Network (LKAN) modification of DNA-Diffusion Unet-based model. Model's core consists of upper and down blocks. b. Base ResNet block gathered from Convolutional Blocks. Dense layer in midst replaced with Linear KAN. c. Convolutional Block is incorporated into base ResNet block and at the end of Upper and Down blocks. d. Structure of Down and Upper Blocks with ResNet Blocks. Convolutional layer replaced with KAN. **C. Unet-based CKAN** a. Architecture of Convolutional Kolmogorov Arnold Network (CKAN) modification of DNA-Diffusion Unet-based model. Model's core consists of upper and down blocks. b. Base ResNet block gathered from Convolutional Blocks with replaced CKANs. c. Convolutional Block is incorporated into base ResNet block and at the end of Upper and Down blocks. d. Structure of Down and Upper Blocks with ResNet Blocks. Convolutional layer replaced with CKAN.



## KANs for Genomics



**Figure 3.** Scaling of models. **A.** Comparison of the tested models of various sizes, evaluated on different collections of datasets. Scores are calculated as average MCC and FLOPs are calculated per input batch. We observe that Linear KAN performs the best and scales the same as Baseline. In contrast, CKAN requires more computations per second and shows relatively poor performance. Achieving better performance with CKAN requires heavier model in terms of parameters. **B.** Linear KAN grid size scaling on model's parameters and training time. We used GUE+ collection for this experiment. As one can see, grid size significantly increases training time.

**Table 1.** KAN performance on classification task for Genomic Benchmarks based on MCC-score averaged over 5-folds. Best values are in **bold**, second best value is in *italics*.

DATA SET	BASELINE	LKAN	CKAN
DMEE	70.2 ± 0.5	<b>73.2 ± 0.2</b>	64.8 ± 1.3
DCIS	81.1 ± 0.5	<b>85.5 ± 1.4</b>	81.7 ± 1.2
DHW	89.2 ± 0.5	<b>90.2 ± 0.5</b>	88.1 ± 0.8
DES	<b>27.3 ± 0.9</b>	28.9 ± 0.6	25.3 ± 0.6
HEC	66.6 ± 0.8	<b>68.2 ± 0.5</b>	67.9 ± 0.8
HEE	63.2 ± 0.5	<b>69.9 ± 0.5</b>	59.0 ± 1.0
HER	<b>90.0 ± 0.8</b>	89.3 ± 0.8	88.4 ± 0.6
HNP	87.9 ± 0.9	<b>88.8 ± 0.9</b>	85.0 ± 1.0
HOE	70.8 ± 0.6	<b>74.4 ± 0.7</b>	68.9 ± 1.0

**Table 2.** KAN performance on classification task for flipons based on MCC-score averaged over 5-folds. Best values are in **bold**, second best value is in *italics*.

DATA SET	BASELINE	LKAN	CKAN
ENDO	89.1 ± 0.5	<b>93.5 ± 0.4</b>	89.0 ± 0.8
G4SEQ	83.3 ± 0.6	<b>89.0 ± 0.8</b>	84.5 ± 0.2
G4CHIP	90.2 ± 0.9	<b>92.2 ± 0.7</b>	91.3 ± 1.5
G4CUT	91.1 ± 0.8	<b>94.8 ± 0.7</b>	85.6 ± 1.2
ZKOU	94.5 ± 1.0	<b>96.3 ± 0.8</b>	94.0 ± 0.8
ZSHIN	<b>97.9 ± 0.5</b>	97.0 ± 0.5	95.7 ± 1.1
HKOU	90.5 ± 0.6	<b>95.2 ± 1.1</b>	88.7 ± 0.9

uation (Zhou et al., 2024b) are presented in Table 3 (MCC score averaged from 5 folds). Baseline CNN model is of equal size with LKAN and CKAN.

Considering results for the classification tasks we observe that Linear KAN scales better than Convolutional KAN of the same size. We present visualization of averaged model's performance for models of various size in Figure 3. Additionally, we provide results of experiments on how grid size parameter affects the performance of LKAN and the number of total parameters along with the time per batch training. We evaluated these results on GUE datasets.

### 3.2. Benchmarks on DNA generative models

To test KAN's capability in producing synthetic DNA sequences we incorporated Linear KAN layers into denoising Unet from DNA Diffusion framework (DaSilva, 2024). We compared final loss for various models (Table 5) and observed that LKAN is capable of reaching the lowest validation loss given equal amount of parameters with baseline Unet. CKAN requires x2 parameters to achieve the comparable metrics.

Additionally, we observe that KAN affects training stability of Unet, as it is presented in Figure 4. Both trainings utilize exponential moving average for smoother performance. We conclude that Linear KANs provide better performance than Convolutional KANs, yet Convolutional KANs requires

## KANs for Genomics

Table 3. KAN performance on classification task for GUE based on MCC averaged over 5-folds. Best values are in **bold**, second best value is in *italics*.

DATA SET	BASELINE	LKAN	CKAN
H3	70.7 ± 1.5	<b>71.3 ± 1.0</b>	69.4 ± 1.5
H3K14AC	40.2 ± 0.8	<b>43.2 ± 0.7</b>	41.0 ± 1.0
H3K36ME3	44.3 ± 0.9	<b>47.7 ± 0.6</b>	42.9 ± 0.9
H3K4ME1	35.6 ± 0.4	<b>40.0 ± 0.8</b>	35.8 ± 1.1
H3K4ME2	27.2 ± 0.5	<b>29.3 ± 0.5</b>	24.3 ± 1.2
H3K4ME3	28.4 ± 0.5	<b>29.8 ± 0.4</b>	27.2 ± 0.7
H3K79ME3	60.8 ± 0.9	<b>62.1 ± 0.8</b>	61.1 ± 1.5
H3K9AC	50.5 ± 0.6	<b>51.9 ± 0.5</b>	50.0 ± 0.9
H4	75.6 ± 0.9	<b>77.2 ± 0.7</b>	70.9 ± 0.8
H4AC	33.9 ± 0.8	<b>38.1 ± 0.8</b>	36.2 ± 0.9
PD ALL	75.9 ± 1.1	<b>81.2 ± 1.4</b>	77.2 ± 1.2
PD NOTATA	88.2 ± 1.1	<b>89.9 ± 0.7</b>	87.1 ± 1.2
PD TATA	56.1 ± 0.5	<b>58.2 ± 0.8</b>	57.9 ± 0.9
TF HUMAN 1	69.9 ± 1.0	<b>72.4 ± 0.8</b>	66.3 ± 1.3
TF HUMAN 2	49.6 ± 0.4	<b>54.3 ± 0.7</b>	47.2 ± 0.7
TF HUMAN 3	43.2 ± 0.5	<b>46.8 ± 0.5</b>	44.4 ± 0.9
TF HUMAN 4	72.6 ± 0.8	<b>72.9 ± 0.5</b>	71.3 ± 0.9
TF HUMAN 0	65.1 ± 0.7	<b>67.4 ± 0.6</b>	65.5 ± 0.9
CPD ALL	64.8 ± 0.9	<b>68.2 ± 0.6</b>	66.2 ± 1.1
CPD NOTATA	64.5 ± 0.7	<b>67.2 ± 0.8</b>	63.1 ± 1.1
CPD TATA	72.3 ± 0.9	<b>78.2 ± 0.9</b>	74.7 ± 1.2
TF MOUSE 1	81.0 ± 0.7	<b>82.8 ± 0.6</b>	80.1 ± 0.8
TF MOUSE 2	79.8 ± 0.6	<b>85.5 ± 0.8</b>	81.2 ± 0.7
TF MOUSE 3	77.6 ± 1.2	<b>81.4 ± 0.7</b>	79.2 ± 0.9
TF MOUSE 4	42.6 ± 0.7	<b>48.7 ± 0.5</b>	40.1 ± 0.7
TF MOUSE 0	38.2 ± 0.7	<b>40.1 ± 0.6</b>	39.0 ± 0.9
VIRUS	50.3 ± 0.9	<b>58.2 ± 1.5</b>	50.5 ± 0.6
SPICE	78.2 ± 0.8	<b>80.0 ± 0.8</b>	75.2 ± 0.7

higher values for grid size, which significantly heavies up the model in terms of parameters. For instance, CKAN requires 820M parameters while LKAN requires 440M to outperform baseline Unet.

We compare the performance of KAN in generative design by evaluating Kullback–Leibler divergence and Wasserstein distance. Since diffusion model require substantial resources for training and inference (see Supplementary Materials for details), we conduct these experiments only for flipon’s datasets (Cherednichenko & Poptsova, 2025). We compare the diversity calculated on edit distance within a sample of synthetic sequences for various models (Table 6). Higher values ensure more diversity within a sample. Trained generative models have captured the prior distributions of flipons based on calculated distances between real and generated distributions (Figures 3-4).

### 3.3. Ablation study

Additionally, we investigated the impact of the total amount of replaced MLP layers with KANs. For this experiment we

Table 4. KAN performance on classification task for GUE+ based on MCC averaged over 5-folds. Best values are in **bold**, second best value is in *italics*.

DATA SET	BASELINE	LKAN	CKAN
GM12878	36.6 ± 1.1	<b>41.2 ± 0.9</b>	37.0 ± 1.2
HELA-S3	34.3 ± 1.0	<b>40.8 ± 1.2</b>	31.3 ± 0.9
HUVEC	37.2 ± 0.7	<b>39.6 ± 0.9</b>	30.2 ± 0.8
IMR90	34.5 ± 0.9	<b>42.0 ± 0.9</b>	33.3 ± 1.3
K692	38.2 ± 0.9	<b>44.4 ± 0.7</b>	37.9 ± 1.1
NHEK	33.7 ± 1.1	<b>40.1 ± 0.9</b>	34.4 ± 0.6
FUNGI	51.1 ± 0.6	<b>55.5 ± 1.0</b>	50.4 ± 1.2
VIRUS	20.7 ± 0.5	<b>35.3 ± 1.9</b>	22.1 ± 1.8

Table 5. KANs for DNA generation: KAN loss (SP-MSE) comparison on flipon generation. Best values are in **bold**, second best value is in *italics*

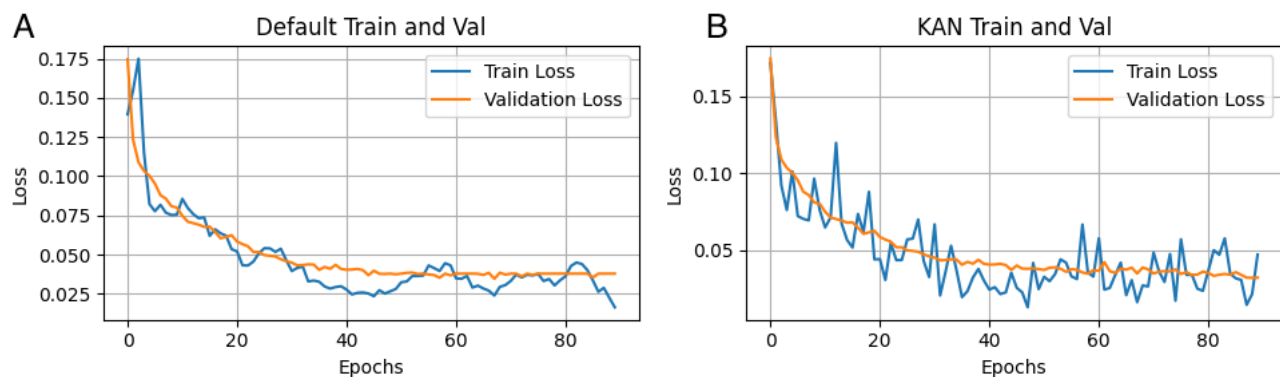
TASK	UNET (410M)	LKAN (440M)	CKAN (820M)
G4	0.0334	<b>0.0276</b>	0.0340
ZDNA	0.0301	<b>0.0222</b>	0.0299
HDNA	0.0388	<b>0.0281</b>	0.0379

used human enhancers cohn dataset from Genomic Benchmarks collection. Taking into consideration the total number of EfficientNet like blocks (Supplementary Table 15) we decided to determine if the same quality can be achieved by replacing only few blocks with LKANs. Model performance based on F1 score with the number of replaced MLP layers and the grid size is presented in Supplementary Table 15. We observe a steady increase in performance with the number of replaced layers. We also see that increasing grid size positively impacts the overall performance. Thus, we found that the best performance is achieved when all blocks are replaced with LKANs, however, with higher grid size values it is possible to achieve fair performance with  $N = 4$  or  $N = 5$  (see Supplementary Tables 15-16). Yet, increasing grid size makes model heavier, therefore the training process takes longer. Moreover, higher grid size is prone to certain overfit, when training metrics are close to high values while testing metrics remain constant (Supplementary Figure 1).

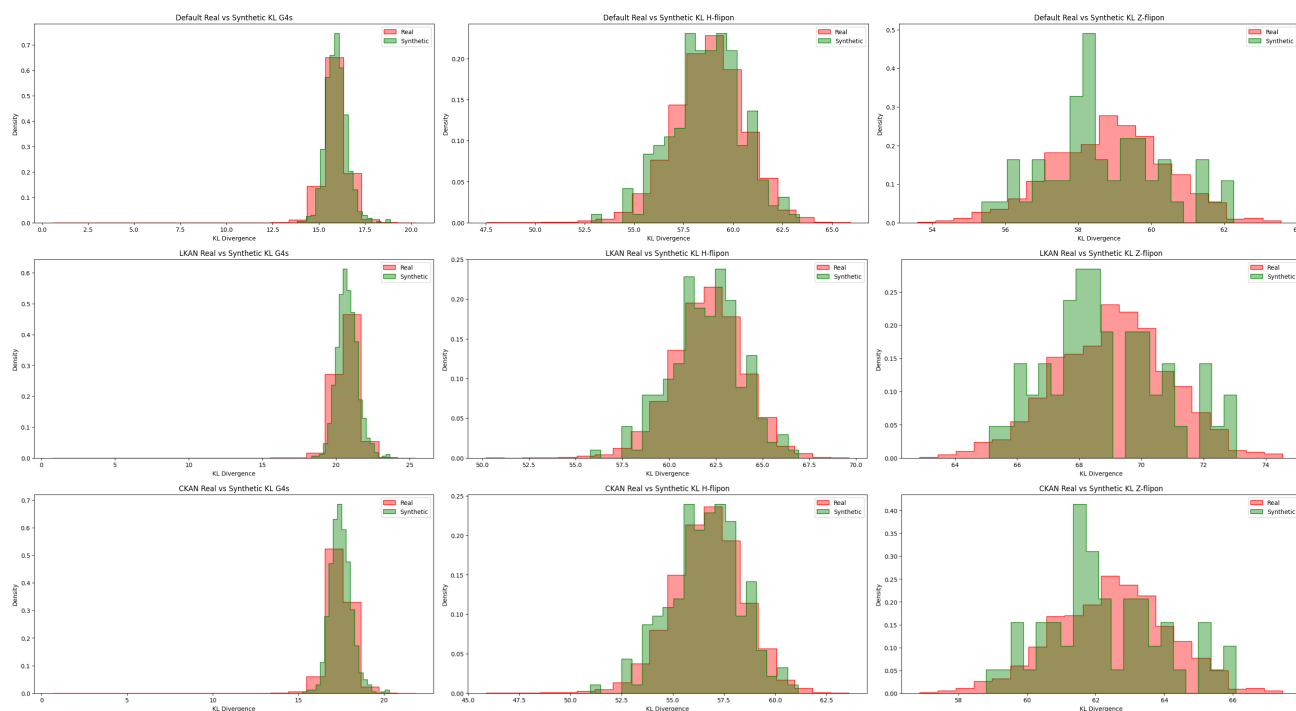
## 4. Discussions and Conclusions

In the presented study, we evaluated performance of the recently developed Kolmogorov-Arnold Networks in the domain of genomics. For that we used a wide range of datasets to cover different types of genomic functional elements including promoters, enhancers, histone marks, and others collected in two published genomic benchmark datasets and one novel benchmark dataset for flipons, or non-B DNA

## KANs for Genomics

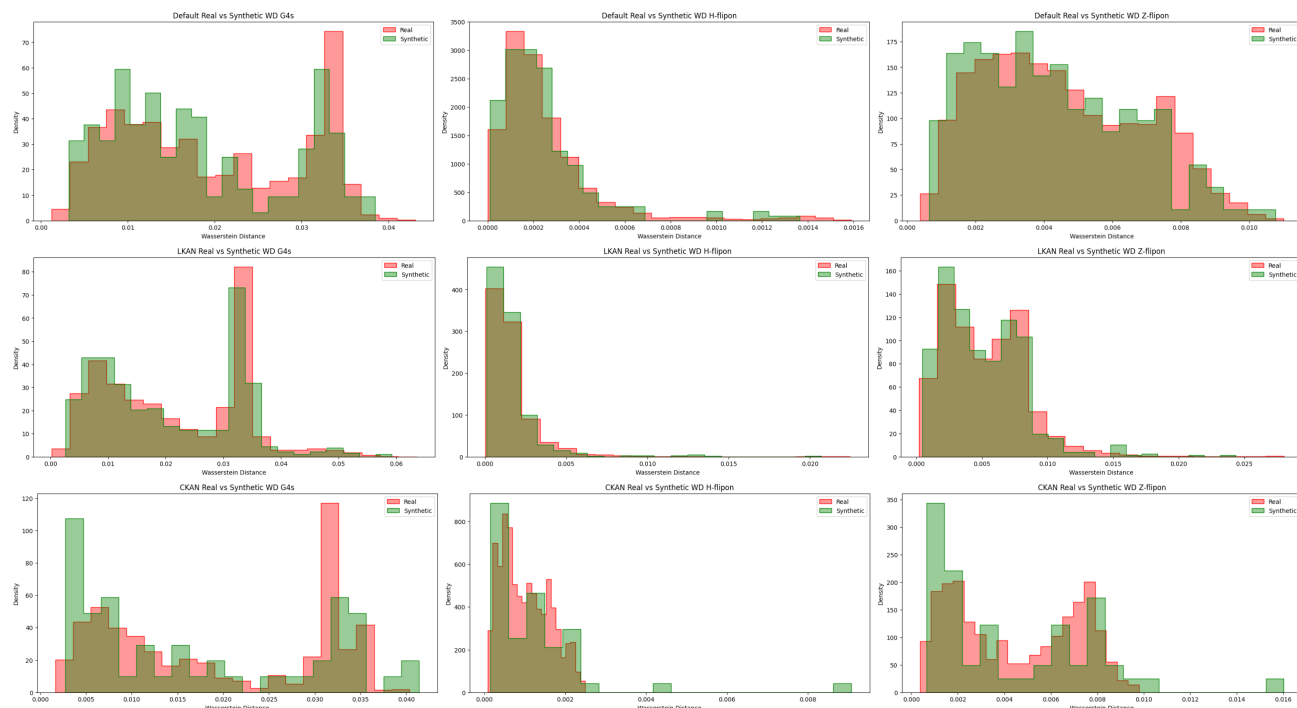


**Figure 4.** Loss functions of generative models. **A.** Baseline Unet with Exponential Moving Average (EMA). **B.** Unet with replaced MLP with Linear KAN (LKAN) also with EMA. While LKAN generally achieves better loss values, it is challenging to stabilize training, even with EMA it gets more spiky.



**Figure 5.** Distributions of Kullback–Leibler divergence (KL) for real and synthetic data. Synthetic data are produced by Baseline Unet (Default) and Unet with Linear KAN (LKAN). We conducted this experiment for flipon benchmark datasets and presnet distributions for Z-DNA, H-DNA and G4s. Results show that both generative models learned to approximate prior distribution as green and red distributions are close.

## KANs for Genomics



**Figure 6.** Distributions of Wasserstein Distance (WD) for real and synthetic data. Synthetic data are produced by Baseline Unet (Default) and Unet with Linear KAN (LKAN). We conducted this experiment for flipon benchmark datasets and present distributions for Z-DNA, H-DNA and G4s. Results show that both generative models learned to approximate prior distribution as green and red distributions are close.

**Table 6.** KANs for DNA generation: metrics of diversity for flipons. Best values are in **bold**, second best value is in *italics*.

TASK	UNET (410M)	LKAN (440M)	CKAN (820M)
G4	69.2	<i>70.3</i>	<b>73.2</b>
ZDNA	70.4	<b>72.2</b>	<i>71.1</i>
HDNA	64.8	<b>64.9</b>	64.8

structures, assembled by us. For our baseline models i.e. models without KAN, we utilized LegNet-based convolutional network for DNA classification (Penzar et al., 2023) and denoising diffusion implicit model DNA-Diffusion with a core Unet module for generating synthetic sequences (DaSilva, 2024). MLP and convolutional layers in these baselines were replaced with linear and convolution KAN layers correspondingly.

Our benchmarking show that Linear KANs show promising results in replacing MLPs in a broad range of genomic tasks, while Convolutional KANs struggle with scaling over large number of parameters, which require more computational resources. In sequence generation we proved that all models, baseline and KANs, can learn distributions of the input data (Figures 5-6), however LKAN outperforms baseline and

CKAN in terms of Loss, while CKAN requires x2 more parameters to achieve the same performance as LKAN. In diversity of generated samples both LKANs and CKANs outperform baseline.

In different domains, such as computer vision and natural language processing it is not yet clear if KANs significantly outperform MLPs as it is the topic of active discussions (Yu et al., 2024; Drokin, 2024; Zhou et al., 2024a; Xu et al., 2024b). Our results are preliminary since we have not tested models with large grid size parameters and did not test other types of KANs. Recent advancements in the KAN field include RNN-based and Transformer-based KANs. Temporal Kolmogorov-Arnold Networks (TKAN) using Long Short Term Memory (LSTM) mechanism to leverage time dependency in KANs were proposed in (Genet & Inzirillo, 2024b). Kolmogorov-Arnold Transformer (KAT) architecture is based on Vision Transformer (Dosovitskiy et al., 2021) and utilizes rational polynomial functions to replace splines (Yang & Wang, 2024). KATs were applied to computer vision tasks and, to our knowledge, have not been yet adapted for natural language processing making it difficult for testing on genomic tasks.

Current implementations of Linear KAN lack the same interpretability as original KAN (Liu et al., 2024b) offers: an ability to plot and prune latest layers and suggest symbolic



## KANs for Genomics

expression for outlying dependencies between variables. Potentially, this could bring even more information about inner structure of a biological task of interest.

Research in the KAN field is evolving rapidly as more and more studies emerge and suggest various technical improvements to address challenges related to KAN usage. The future directions may involve training a Transformer-based models with KANs to handle task of studying latent representations of a language model's hidden states - embeddings. It was shown (Zhou et al., 2024c; Nguyen et al., 2023) that visualization of language model embeddings could enhance biological understanding of model's utility. Of interest is to test more frameworks as state-space models (Gu & Dao, 2023), and transformers with GPT-like architecture (Cui et al., 2023). In addition, there are multiple advancements in sequence generative modeling like Discrete Diffusion for DNA (Sarkar et al., 2024). Despite our first positive experiments with linear and convolutional KANs, unleashing the potential of KAN for the entire spectrum of genomic tasks require further extensive research.

## Acknowledgments

The work was supported by the Basic Research Program of the National Research University Higher School of Economics.

## References

- Abueidda, D. W., Pantidis, P., and Mobasher, M. E. Deepokan: Deep operator network based on kolmogorov arnold networks for mechanics problems. <https://doi.org/10.48550/arXiv.2405.19143>, 2024.
- Aghaei, A. A. fkan: Fractional kolmogorov-arnold networks with trainable jacobi basis functions. <https://doi.org/10.48550/arXiv.2406.07456>, 2024.
- Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. In <https://doi.org/10.48550/arXiv.1701.07875>, 2017.
- Avsec, , Agarwal, V., and Visentin, D. Effective gene expression prediction from sequence by integrating long-range interactions. *Nat Methods*, 18(1), 2021.
- Beknazarov, N., Jin, S., and Poptsova, M. Deep learning approach for predicting functional z-dna regions using omics data. *Scientific reports*, 10(1), 2020.
- Bodner, A. D., Tepsich, A. S., Spolski, J. N., and Pourteau, S. Convolutional kolmogorov-arnold networks. <https://doi.org/10.48550/arXiv.2406.13155>, 2024.
- Chambers, V., Marsico, G., Boutell, J., Antonio, M. D., Smith, G., and Balasubramanian, S. High-throughput sequencing of dna g-quadruplex structures in the human genome. *Nat Biotechnol*, 2015.
- Chen, K., Wong, A., and Troyanskaya, O. A sequence-based global map of regulatory activity for deciphering human genetics. *Nat Genet*, 54(1):940949, 2022.
- Cheon, M. Demonstrating the efficacy of kolmogorov-arnold networks in vision tasks. <https://doi.org/10.48550/arXiv.2406.14916>, 2024.
- Cherednichenko, O. and Poptsova, M. Data augmentation with generative models improves detection of non-b dna structures. *Computers in Biology*, 184(1):109440, 2025.
- Cui, H., Wang, C., Maan, H., and Wang, B. scgpt: Towards building a foundation model for single-cell multi-omics using generative ai. *bioRxiv* 2023.04.30.538439; doi: <https://doi.org/10.1101/2023.04.30.538439>, 2023.
- Dalla-Torre, H., Gonzalez, L., Revilla, J. M., Carranza, N. L., Grzywaczewski, A. H., Oteri, F., Christian Dallago, Evan Trop, H. S., Richard, G., Skwark, M., Beguir, K., Lopez, M., and Pierrot, T. The nucleotide transformer: Building and evaluating robust foundation models for human genomics. In *bioRxiv* 2023.01.11.523679; doi: <https://doi.org/10.1101/2023.01.11.523679>, 2023.
- DaSilva, e. a. Dna-diffusion: Leveraging generative models for controlling chromatin accessibility and gene expression via synthetic regulatory elements. *bioRxiv* 2024.02.01.578352, 2024.
- de Sainte Agathe, J., Filser, M., and Isidor, B. Spliceai-visual: a free online tool to improve spliceai splicing variant interpretation. *Hum Genomics*, 17(7), 2023.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houshy, N. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representation (ICLR 2021)*, Austria, 2021.
- Drokin, I. Kolmogorov-arnold convolutions: Design principles and empirical studies. <https://doi.org/10.48550/arXiv.2407.01092>, 2024.
- Franke, J. K., Runge, F., Köksal, R., Matus, D., Backofen, R., and Hutter, F. Rnaformer: A simple yet effective model for homology-aware rna secondary structure prediction. *bioRxiv* 2024.02.12.579881; 2024. doi: <https://doi.org/10.1101/2024.02.12.579881>.
- Genet, R. and Inzirillo, H. A temporal kolmogorov-arnold transformer for time series forecasting. <https://doi.org/10.48550/arXiv.2406.02486>, 2024a.

# KANs for Genomics

- Genet, R. and Inzirillo, H. Tkan: Temporal kolmogorov-arnold networks. <https://doi.org/10.48550/arXiv.2405.07344>, 2024b.
- Georgakilas, G., Grioni, A., Liakos, K., Chalupova, E., Plessas, F., and Alexiou, P. Multi-branch convolutional neural network for identification of small non-coding rna genomic loci. *Sci Rep.*, 10(1):1—10, 2020.
- Gresova, K., Martinek, V., Cechak, D., Simecek, P., and Alexiou, P. Genomic benchmarks: a collection of datasets for genomic sequence classification. *BMC Genomic Data*, 21(1), 2023.
- Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. <https://doi.org/10.48550/arXiv.2312.00752>, 2023.
- Herbert, A. Flipons: The discovery of z-dna and soft-wired genomes, 2024.
- Hornik, K., Stinchcombe, M., and White, H. Multilayer feedforward networks are universal approximators. *Neural networks*, 2:359—366, 1989.
- Hwang, H., Jeon, H., and Yeo, N. Big data and deep learning for rna biology. *Exp Mol Med*, 56(1), 2024.
- Ji, Y., Zhou, Z., Liu, H., and Davuluri, R. V. Dnabert: pre-trained bidirectional encoder representations from transformers model for dna-language in genome. *Bioinformatics*, 37(15):2112—2120, 2021.
- Kolmogorov, A. On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables. *Doklady Akademii Nauk*, 114:953—956, 1956.
- Kouzine, F. Permanganate/s1 nuclease footprinting reveals non-b dna structures with regulatory potential across a mammalian genome. *Cell Syst*, 4(1):344—356, 2017.
- Le, N., Ho, Q., Nguyen, T., and Ou, Y. A transformer architecture based on bert and 2d convolutional neural network to identify dna enhancers from sequence information. *Brief Bioinform*, 2021.
- Li, R., Li, M., Liu, W., and Chen, H. Gnn-skan: Harnessing the power of swallowkan to advance molecular representation learning with gnns. <https://doi.org/10.48550/arXiv.2408.01018>, 2024.
- Liu, Z., Ma, P., Wang, Y., Matusik, W., and Tegmark, M. Kan 2.0: Kolmogorov-arnold networks meet science. <https://doi.org/10.48550/arXiv.2408.10205>, 2024a.
- Liu, Z., Wang, Y., Vaidya, S., Fabian Ruehle, J. H., Soljačić, M., Hou, T. Y., and Tegmark, M. Kan: Kolmogorov-arnold networks. <https://doi.org/10.48550/arXiv.2404.19756>, 2024b.
- Lyu, J., Shao, R., Kwong, Y., and Elsasser, S. Genome-wide mapping of g-quadruplex structures with cuttag. *Nucleic Acids Res*, 50, 2016.
- Nagai, Y. and Okumura, M. Kolmogorov-arnold networks in molecular dynamics. <https://doi.org/10.48550/arXiv.2407.17774>, 2024.
- Nguyen, E., Poli, M., Faizi, M., Thomas, A., Birch-Sykes, C., Wornow, M., Patel, A., Rabideau, C., Mas-saroli, S., Bengio, Y., Ermon, S., Baccus, S. A., and Ré, C. Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution. In *doi: 10.48550/ARXIV.2306.15794.*, 2023.
- Oubounyt, M., Louadi, Z., and Tayara, H. Deepromoter: robust promoter predictor using deep learning. *Front Genet.*, 2019a.
- Oubounyt, M., Louadi, Z., and Tayara, H. Deepromoter: robust promoter predictor using deep learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207—1216, Stanford, CA, 2019b. Morgan Kaufmann.
- Park, J.-D., Kim, K.-M., and Shin, W.-Y. Cf-kan: Kolmogorov-arnold network-based collaborative filtering to mitigate catastrophic forgetting in recommender systems. <https://doi.org/10.48550/arXiv.2409.05878>, 2024.
- Penzar, D., Nogina, D., Noskova, E., Zinkevich, A., Meshcheryakov, G., Lando, A., Rafi, A. M., de Boer, C., and Kulakovskiy, I. V. Legnet: a best-in-class deep learning model for short dna regulatory regions. *Bioinformatics*, 39, 2023.
- Qian, S. Endoquad: a comprehensive genome-wide experimentally validated endogenous g-quadruplex database. *Nucleic Acids Res*, 52, 2024.
- Quang, D. and Xie, X. Factornet: a deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods*, 160:40—7, 2019.
- Sarkar, A., Tang, Z., Zhao, C., and Koo, P. K. Designing dna with tunable regulatory activity using discrete diffusion. *bioRxiv* 2024.05.23.595630; [doi: https://doi.org/10.1101/2024.05.23.595630](https://doi.org/10.1101/2024.05.23.595630), 2024.
- Sato, K. and Hamada, M. Recent trends in rna informatics: a review of machine learning and deep learning for rna secondary structure prediction and rna drug discovery. *Briefings in Bioinformatics*, 24(4), 2023.
- Schiff, Y., Kao, C.-H., Gokaslan, A., Dao, T., Gu, A., and Kuleshov, V. Caduceus: Bi-directional equivariant long-range dna sequence modeling. In *Proceedings of the 41nd*

# KANs for Genomics

- International Conference on Machine Learning (ICML 2024)*, 2024.
- Shen, Z., Bao, W., and Huang, D. Recurrent neural network for predicting transcription factor binding sites. *Sci Rep*, 8, 2018.
- Shen, Z., Zhang, Q., Han, K., and Huang, D. A deep learning model for rnaprotein binding preference prediction based on hierarchical lstm and attention network. *IEEE/ACM Trans Comput Biol Bioinforma.*, 19(2):753–62, 2020.
- Shin, S. Zdna-forming sites identified by chip-seq are associated with actively transcribed regions in the human genome. *DNA Res*, 2016.
- Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *Proceedings of the 9th International Conference on Learning Representation (ICLR 2021)*, Austria, 2021.
- Ta, H.-T. Bsrbf-kan: A combination of b-splines and radial basis functions in kolmogorov-arnold networks. <https://doi.org/10.48550/arXiv.2406.11173>, 2024.
- Tan, M. and Le, Q. V. Efficientnetv2: Smaller models and faster training. <https://doi.org/10.48550/arXiv.2104.00298>, 2024.
- Umerenkov, D., Herbert, A., Konovalov, D., Danilova, A., Beknazarov, N., Kokh, V., Fedorov, A., and Poptsova, M. Z-flipon variants reveal the many roles of z-dna and z-rna in health and disease. *Life Science Alliance*, 6(7), 2023.
- Wen, W., Zhong, J., Zhang, Z., Jia, L., Chu, T., Wang, N., Danko, C. G., and Wang, Z. dhica: a deep transformer-based model enables accurate histone imputation from chromatin accessibility. *Briefings in Bioinformatics*, 25 (6), 2024.
- Xu, A., Zhang, B., Kong, S., Huang, Y., Yang, Z., Srivastava, S., and Sun, M. Effective integration of kan for keyword spotting. <https://doi.org/10.48550/arXiv.2409.08605>, 2024a.
- Xu, J., Chen, Z., Li, J., Yang, S., Wang, W., Hu, X., and Ngai, E. C. H. Fourierkan-gcf: Fourier kolmogorov-arnold network – an effective and efficient feature transformation for graph collaborative filtering. <https://doi.org/10.48550/arXiv.2406.01034>, 2024b.
- Yang, X. and Wang, X. Kolmogorov-arnold transformer. <https://doi.org/10.48550/arXiv.2409.10594>, 2024.
- Yin, Q., Wu, M., Q. Liu, H. L., and Jiang, R. Deephistone: a deep learning approach to predicting histone modifications. *BMC Genomics*, 20(2):11–23, 2021.
- Yu, R., Yu, W., and Wang, X. Kan or mlp: A fairer comparison. <https://doi.org/10.48550/arXiv.2407.16674>, 2024.
- Zhang, Y., Yang, L., and Kucherlapati, M. Global impact of somatic structural variation on the dna methylome of human cancers. *Genome Biol*, 2019.
- Zhou, J. and Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.
- Zhou, Q., Pei, C., Sun, F., Han, J., Gao, Z., Pei, D., Zhang, H., Xie, G., and Li, J. Kan-ad: Time series anomaly detection with kolmogorov-arnold networks. <https://doi.org/10.48550/arXiv.2411.00278>, 2024a.
- Zhou, Z., Ji, Y., Li, W., Dutta, P., Davuluri, R., and Liu, H. Dnabert-2: Efficient foundation model and benchmark for multi-species genome. In *Proceedings of the 12th International Conference on Learning Representation (ICLR 2024)*, Vienna, Austria, 2024b.
- Zhou, Z., Wu, W., Ho, H., Wang, J., Shi, L., Davuluri, R. V., Wang, Z., and Liu, H. Dnabert-s: Learning species-aware dna embedding with genome foundation models. <https://doi.org/10.48550/arXiv.2402.08777>, 2024c.