FourierKAN outperforms MLP on Text Classification Head Fine-tuning

Abdullah Al Imran*
University of Liverpool
a.al-imran@liverpool.ac.uk

Md Farhan Ishmam*
Islamic University of Technology
farhanishmam@iut-dhaka.edu

Abstract

In resource constraint settings, adaptation to downstream classification tasks involves fine-tuning the final layer of a classifier (i.e. classification head) while keeping the rest of the model weights frozen. Multi-Layer Perceptron (MLP) heads fine-tuned with pre-trained transformer backbones have long been the de facto standard for text classification head fine-tuning. However, the fixed non-linearity of MLPs often struggles to fully capture the nuances of contextual embeddings produced by pre-trained models, while also being computationally expensive. In our work, we investigate the efficacy of KAN and its variant, Fourier KAN (FR-KAN), as alternative text classification heads. Our experiments reveal that FR-KAN significantly outperforms MLPs with an average improvement of 10% in accuracy and 11% in F1-score across seven pre-trained transformer models and four text classification tasks. Beyond performance gains, FR-KAN is more computationally efficient and trains faster with fewer parameters. These results underscore the potential of FR-KAN to serve as a lightweight classification head, with broader implications for advancing other Natural Language Processing (NLP) tasks.

1 Introduction

Classification head fine-tuning, also known as *linear probing*, is a widely adopted strategy that involves training the final classification layer while the backbone model remains frozen. This approach allows efficient adaptation to downstream tasks especially in resource constraint settings [1] and improved robustness in out-of-domain distributional shifts [2], compared to standard fine-tuning. In text classification tasks, Multi-Layer Perceptron (MLP) [3] classification heads are usually fine-tuned with pre-trained transformer backbones [4].

MLPs are fully connected or dense neural networks, used across domains including time series analysis [5], computer vision [6, 7], and speech processing [8]. MLP classifiers are able to capture the non-linearity and aggregate the high-dimensional contextualized embedding produced by the feature extractor to the fixed set of output classes. While it is undeniable that MLPs have revolutionized deep learning, they have a few noticeable limitations [9].

MLPs account for most of the trainable parameters in the transformer architecture while being less interpretable compared to methods, *e.g.* self-attention [10]. Reducing the parameter count while maintaining performance has been explored by various strategies, such as network pruning [11], and quantization [12], but with significant tradeoffs. Recent advancements in MLP alternatives, such as Kolmogorov-Arnold Networks (KANs) [13], show promising results to replace MLPs by *learning the non-linearity* [14] instead of relying on the fixed non-linear activations used in MLPs.

Our work aims to investigate the adaptation of the KAN as a text classification head in resource-constraint settings by exploring its efficacy in linear probing. We primarily focus on the potential

*Equal Contribution

38th Workshop on Fine-Tuning in Machine Learning (NeurIPS 2024).

application of Fourier-KAN (FR-KAN) [15], a modification of the spline-based KAN variant using the Fourier series. To the best of our knowledge, we are the first to adapt the KAN architecture as an MLP alternative in linear probing. We also observed that the simple modification of using FR-KANs instead of MLPs as the classification head resulted in an average increase of 10% in accuracy and 11% in F1 score across 7 text classification datasets using pre-trained transformer backbones.

2 Methodology

2.1 Text Classification Head

For the contextual embedding, $\mathbf{H} = f(\mathbf{x}; \theta_f)$, produced by the pre-trained language model f where $\mathbf{x}$ is the input text embedding and θ_f is the frozen model parameters, we formulate the predicted answer class,

$$\hat{y} = \arg\max_{c \in \mathcal{C}} \text{Head}(\mathbf{H}) \tag{1}$$

where, Head : $\mathbb{R}^{|\mathbf{H}|} \rightarrow \mathbb{R}^{|\mathcal{C}|}$, Head $\in \{\text{MLP}, \text{KAN}, \text{FR-KAN}\}$, and c represents an answer class from the answer class set $\mathcal{C}$. It should be noted that the pre-trained language model f acts as a feature extractor only. We utilize the cross-entropy or the negative log-likelihood loss during the classifier head fine-tuning, mathematically expressed,

$$L(\hat{y}, y) = -\sum_{i=1}^{|\mathcal{C}|} y_i \log(\hat{y}_i) \tag{2}$$

2.2 Kolmogorov-Arnold Network (KAN)

Following the formulation of KANs with arbitrary depth and width [13], a single KAN layer is defined as:

$$\text{KAN}(\mathbf{H}) = f(\mathbf{H}) = \sum_{j=1}^{2n+1} \Phi_j \left(\sum_{i=1}^{n} \phi_{ij}(h_i) \right) \tag{3}$$

where, ϕ_{ij} are univariate continuous functions mapping the input vector x, such that, $\phi_{ij} : [0, 1] \rightarrow \mathbb{R}$ and Φ_j are learnable activation functions, such that, $\Phi_j : \mathbb{R} \rightarrow \mathbb{R}$. The KAN layer can be analogous to a 2-layer MLP where the first layer computes the inner sum $\sum_{i=1}^{n} \phi_{ij}(x_i)$ and the next layer applies and sums Φ_j to the previous layer output. The original implementation of KAN [13] follows a residual layer formulation of the learnable activation function:

$$\phi_b(x) = w(b(x) + \text{spline}(x)) \tag{4}$$

where, the basis function $b(x)$ are defined as,

$$b(x) = \text{silu}(x) = \frac{x}{1 + e^{-x}} \tag{5}$$

and the splines can be formulated as the weighted sum of B-splines,

$$\text{spline}(x) = \sum_{i=1}^{G} c_i B_i(x) \tag{6}$$

where, c_i are trainable parameters and G is the grid size.

Theorem 1. *Assume with Fourier coefficients a_k, b_k and grid size G, the Fourier series for the function $f(x)$ taking the form:*

$$f_G(x) = \sum_{k=0}^{G} (a_k \cdot \cos(kx) + b_k \cdot \sin(kx))$$

converges to a corresponding univariate function over a finite interval $[a, b]$ as $G \rightarrow \infty$, given the function is continuous.

Proof. The convergence of the Fourier series to a univariate function can be proved via pointwise, uniform, or mean square (or L^2) convergence. We are particularly interested in uniform convergence, which implies pointwise and mean square convergence. For pointwise convergence, Dirichlet's proof states $f_G(x)$ converges at points of continuities and takes the average value of $(f(d^+) + f(d^-))/2$ when jump discontinuity is observed at $x = d$.

We generalize the Fourier coefficients a_k and b_k as c_k and consider the Riemann-Lebesgue lemma on the Fourier coefficients *i.e.* $c_k \to 0$ as $k \to 0$. From pointwise convergence, we state for $x \in [a, b]$,

$$|f(x) - f_G(x)| = \left| \sum_{k=G+1}^{\infty} a_k \cdot \cos(kx) + \sum_{k=G+1}^{\infty} b_k \cdot \sin(kx) \right| \tag{7}$$

Since, $\cos(kx) \leq 1$ and $\sin(kx) \leq 1$,

$$|f(x) - f_G(x)| \leq \sum_{k=G+1}^{\infty} (|a_k| + |b_k|) \tag{8}$$

To ensure uniform convergence, the sum of the trailing elements is required to be bounded. As the Fourier coefficients are square summable *i.e.* $\sum_{k=1}^{\infty} |c_k|^2 < \infty$, we apply the Cauchy-Schwarz inequality:

$$\left(\sum_{k=G+1}^{\infty} |c_k| \right)^2 \leq \left(\sum_{k=G+1}^{\infty} 1^2 \right) \left(\sum_{k=G+1}^{\infty} |c_k|^2 \right) \tag{9}$$

$|c_k|^2$ converges as the Fourier coefficients are square summable. Hence, the tail sum of $\sum_{k=G+1}^{\infty} |c_k|^2$ can be arbitary small as $G \to \infty$ and the Fourier series converges uniformly to $f(x)$ on $[a, b]$. □

Corollary 1. *As $G \to \infty$, the truncation error of the Fourier series, $E_G \to 0$.*

2.3 FouRier KAN (FR-KAN)

The residual formulation using B-splines in Eq. 4 can be replaced with the Fourier series following the convergence of the series up to G terms in Theorem 1. Hence, the univariate continuous function in Eq. 3 can be defined as:

$$\phi_f(x) = \sum_{k=0}^{G} (a_k \cdot \cos(kx) + b_k \cdot \sin(kx)) \tag{10}$$

where, a_k and b_k are the trainable Fourier coefficients and G is the grid size.

2.4 Multi Layer Perceptron (MLP)

We define MLP_L as an L-layer perceptron with the trainable weights, W_{L-1} and bias, b_{L-1}. We formulate a 1-layer perceptron:

$$\text{MLP}_1(\mathbf{H}) = \text{softmax}(W_0\mathbf{H} + b_0) \tag{11}$$

where the softmax function is defined as:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{12}$$

We also define a 2-layer perceptron:

$$h_0 = \sigma(W_0\mathbf{H} + b_0) \tag{13}$$

$$\text{MLP}_2(\mathbf{H}) = \text{softmax}(W_1 h_0 + b_1) \tag{14}$$

where, the non-linear sigmoid, σ function is defined:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

Dataset	Task Name	Type	#Classes	Avg. Len	Max Len	#Train	#Val	#Test
AgNews	News Classification	TC	4	44	221	10.5k	2.25k	2.25k
DBpedia	Ontology Classification	TC	14	67	3841	10.5k	2.25k	2.25k
IMDb	Movie Sentiment	SA	2	292	3045	10.5k	2.25k	2.25k
Papluca	Language Identification	LI	20	111	2422	10.5k	2.25k	2.25k
SST-5	General Sentiment	SA	5	103	283	8.3k	1.78k	1.78k
TREC-50	Question Classification	QC	50	11	39	4.17k	893	893
YELP-Full	Review Sentiment	SA	5	179	2342	10.5k	2.25k	2.25k

Table 1: Statistics of the text classification datasets used in our work. Full form of the types – SA: Sentiment Analysis, TC: Topic Classification, QC: Question Classification, LI: Language Identification.

3 Experiments

3.1 Tasks and Datasets

We chose four types of text classification tasks and seven datasets to evaluate our models. The overall statistics of the datasets are shown in Tab. 1.

Sentiment Analysis We use three datasets – IMDb [16], SST-5 [17], and Yelp-full [18] for sentiment analysis. The IMDb is a binary classification dataset on movie reviews, while SST-5 and YELP-full are multi-class classification datasets on movie reviews and general reviews, respectively.

Topic Classification We use two datasets AgNews [18] and DBpedia [18] for topic classification. The AgNews dataset classifies news topics from over 2000 news sources into 4 topic classes. DBpedia introduces ontology classification as a form of topic classification on 14 ontology classes, each with 40k training samples and 5k test samples.

Question Classification We use the 50 class or fine-grained variant of the TREC dataset [19] consisting of open-domain questions for question classification.

Language Identification The Papluca dataset [20] classifies the text language into 20 uniformly distributed classes.

3.2 Models

We utilize seven variants of pre-trained transformer models to generate the contextual embedding as seen in Tab. 2. BART [21] is the only encoder-decoder model while BERT [22], DeBERTa [23], DistilBERT [24], ELECTRA [25], RoBERTa [26], and XLNet [27] are encoder-only models. BART has 12 layers encoder and 12 decoder layers, while the other models have 12 encoder layers with the exception of DistilBERT with 6 encoder layers.

Model	Arch	#L	#PC (M)
BART	ED	12+12	140
BERT	E	12	110
DeBERTa	E	12	139
DistilBERT	E	6	66
ELECTRA	E	12	110
RoBERTa	E	12	125
XLNet	E	12	110

Table 2: Architecture type [E: Encoder, D: Decoder], number of layers, and parameter count of the transformer models.

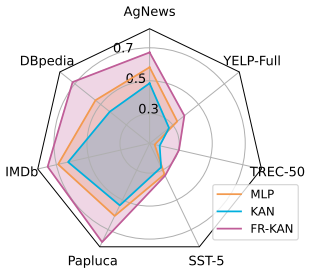

Figure 1: Comparison of average accuracy of different classification heads.

3.3 Evaluation Metrics

We evaluate the classification performance using four key metrics: accuracy, macro-averaged F1 score (simply F1 score), micro-averaged F1 score, and Cohen's kappa coefficient.

3.4 Experimental Setup

All models were fine-tuned on an A4000 GPU with 16 GiB of GPU memory. The default tokenizer and embedding layers corresponding to each model were used. Each model features a hidden dimension of 768 and 12 self-attention heads. The implementation and training configurations followed the HuggingFace library [28]. The classifiers were fine-tuned using Adam optimizer with the max length set to 512 and the batch size fixed at 64. Identical training configurations were used across dataset-model pairs to ensure fair evaluation of the classification heads.

3.5 Hyperparameters

To ensure fairness of evaluation, all classification heads are defined as 1-layer architectures, excluding the input layer. 2 layer MLPs (Eq. 14) have also been later specified to align the number of trainable parameters with that of the other heads. The width of the hidden layer of MLP varied depending on the classification dataset. Unless specified, the original KAN and FR-KAN layers use a grid size of 1 and 5 respectively. To evaluate in resource-constraint settings, all the classifiers were fine-tuned for 5 epochs at the learning rate of $2e-5$.

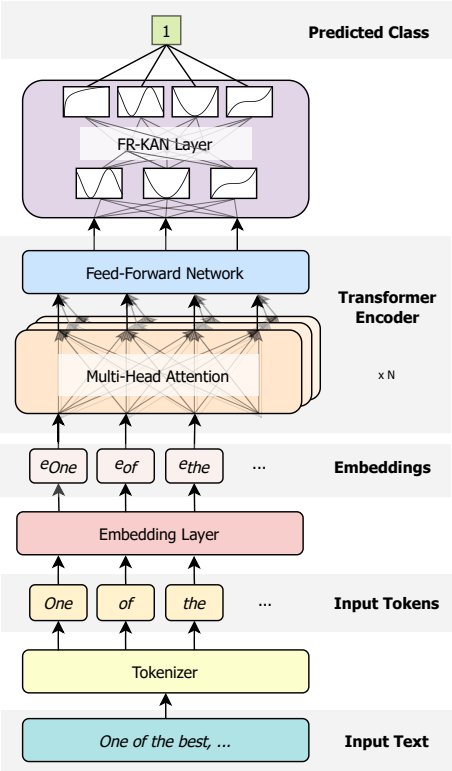

Figure 2: Overview of the architecture with FR-KAN classification head – following the standard tokenization and embedding, the input text is passed to a pre-trained transformer encoder. The FR-KAN layer maps the contextualized embedding produced by the transformer to the output classes.

Dataset	Classifier	#PC(k) ↓	Acc ↑	F1 ↑
AgNews	MLP-40	30.9	0.835	0.831
	KAN-1	30.7	0.812	0.813
	FR-KAN-5	30.7	**0.877**	**0.876**
	Diff	-0.2	+0.04	+0.05
Dbpedia	MLP-138	108.1	0.892	0.891
	KAN-1	107.5	0.843	0.842
	FR-KAN-5	107.5	**0.970**	**0.971**
	Diff	-0.6	+0.08	+0.08
IMDb	MLP-20	15.4	0.778	0.777
	KAN-1	15.4	0.739	0.739
	FR-KAN-5	15.4	**0.831**	**0.830**
	Diff	0.0	+0.05	+0.05
Papluca	MLP-196	154.6	0.816	0.819
	KAN-1	153.6	0.730	0.733
	FR-KAN-5	153.6	**0.986**	**0.986**
	Diff	-1.0	+0.17	+0.17
SST-5	MLP-50	38.7	0.351	0.176
	KAN-1	38.4	0.307	0.231
	FR-KAN-5	38.4	**0.401**	**0.336**
	Diff	-0.3	+0.05	+0.16
TREC-50	MLP-477	512.8	0.179	0.006
	KAN-1	384	0.188	0.017
	FR-KAN-5	384	**0.351**	**0.057**
	Diff	-0.8	+0.17	+0.05
YELP-Full	MLP-50	38.7	0.458	0.456
	KAN-1	38.4	0.338	0.324
	FR-KAN-5	38.4	**0.492**	**0.475**
	Diff	-0.3	+0.03	+0.02

Table 3: Parameter count, accuracy, and F1 score of DistilBERT using different classification heads across all datasets. MLP-x represents a 2-layer perceptron with x hidden layer width. KAN-x and FR-KAN-x represent the corresponding 1-layer network with grid size x. The differences between the FR-KAN head and MLP have been shown.

Method		AgNews		DBpedia		IMDb		Papluca		SST-5		TREC-50		YELP-Full	
Backbone	**Head**	**Acc ↑**	**F1 ↑**	**Acc ↑**	**F1 ↑**	**Acc ↑**	**F1 ↑**	**Acc ↑**	**F1 ↑**	**Acc ↑**	**F1 ↑**	**Acc ↑**	**F1 ↑**	**Acc ↑**	**F1 ↑**
BART	MLP	0.612	0.610	0.808	0.806	**0.769**	**0.769**	0.835	0.833	**0.303**	0.181	0.254	0.042	0.364	0.357
	KAN	0.351	0.352	0.459	0.459	0.594	0.594	0.657	0.654	0.260	0.216	0.283	0.055	0.254	0.250
	FR-KAN	**0.653**	**0.651**	**0.872**	**0.872**	0.749	0.749	**0.880**	**0.879**	0.273	**0.237**	**0.451**	**0.122**	**0.390**	**0.385**
	Diff	+0.04	+0.04	+0.06	+0.07	-0.02	-0.02	+0.05	+0.05	-0.03	+0.06	+0.20	+0.08	+0.03	+0.03
BERT	MLP	0.772	0.770	0.752	0.746	0.763	0.762	0.579	0.581	0.374	0.197	0.181	0.008	0.440	0.433
	KAN	0.722	0.721	0.674	0.672	0.730	0.729	0.391	0.348	0.317	0.260	0.186	0.027	0.336	0.321
	FR-KAN	**0.834**	**0.833**	**0.939**	**0.938**	**0.812**	**0.811**	**0.946**	**0.945**	**0.406**	**0.339**	**0.378**	**0.069**	**0.471**	**0.450**
	Diff	+0.06	+0.06	+0.19	+0.19	+0.05	+0.05	+0.37	+0.36	+0.03	+0.14	+0.20	+0.06	+0.03	+0.02
DeBERTa	MLP	0.554	0.549	0.567	0.568	0.710	0.708	0.848	0.850	**0.417**	0.289	0.199	0.015	0.393	0.380
	KAN	0.400	0.400	0.357	0.347	0.666	0.666	0.751	0.753	0.292	0.271	0.167	0.037	0.319	0.314
	FR-KAN	**0.595**	**0.594**	**0.648**	**0.648**	**0.770**	**0.770**	**0.920**	**0.923**	0.367	**0.333**	**0.377**	**0.076**	**0.412**	**0.409**
	Diff	+0.04	+0.04	+0.08	+0.08	+0.06	+0.06	+0.07	+0.07	-0.05	+0.04	+0.18	+0.06	+0.02	+0.03
DistilBERT	MLP	0.836	0.834	0.865	0.863	0.795	0.794	0.878	0.877	0.352	0.184	0.096	0.009	0.429	0.420
	KAN	0.812	0.813	0.843	0.842	0.739	0.739	0.730	0.733	0.307	0.231	0.188	0.017	0.338	0.324
	FR-KAN	**0.877**	**0.876**	**0.970**	**0.971**	**0.831**	**0.830**	**0.986**	**0.986**	**0.401**	**0.336**	**0.351**	**0.057**	**0.492**	**0.475**
	Diff	+0.04	+0.04	+0.11	+0.11	+0.04	+0.04	+0.11	+0.11	+0.05	+0.15	+0.26	+0.05	+0.06	+0.06
ELECTRA	MLP	0.480	0.471	0.364	0.338	0.592	0.590	0.539	0.509	0.326	0.175	0.096	0.009	0.291	0.284
	KAN	0.384	0.378	0.296	0.255	0.558	0.552	0.477	0.459	0.295	0.229	0.179	0.016	0.240	0.233
	FR-KAN	**0.612**	**0.606**	**0.610**	**0.609**	**0.745**	**0.745**	**0.672**	**0.670**	**0.326**	**0.285**	**0.338**	**0.064**	**0.370**	**0.352**
	Diff	+0.13	+0.14	+0.25	+0.27	+0.15	+0.16	+0.13	+0.16	0.00	+0.11	+0.24	+0.06	+0.08	+0.07
RoBERTa	MLP	0.420	0.304	0.258	0.155	0.578	0.546	0.327	0.226	0.269	0.085	0.132	0.008	0.209	0.088
	KAN	0.448	0.434	0.253	0.214	0.568	0.521	0.584	0.528	0.269	0.125	0.179	0.006	0.203	0.190
	FR-KAN	**0.836**	**0.832**	**0.844**	**0.829**	**0.819**	**0.819**	**0.925**	**0.910**	**0.328**	**0.189**	**0.179**	**0.006**	**0.369**	**0.306**
	Diff	+0.42	+0.53	+0.59	+0.67	+0.24	+0.27	+0.60	+0.68	+0.06	+0.10	+0.05	+0.00	+0.16	+0.22
XLNet	MLP	**0.399**	**0.375**	**0.163**	**0.156**	**0.616**	**0.613**	**0.248**	**0.195**	0.255	0.190	**0.105**	**0.019**	0.218	0.212
	KAN	0.275	0.261	0.123	0.112	0.536	0.528	0.166	0.153	0.225	0.211	0.102	0.015	0.207	0.203
	FR-KAN	0.300	0.293	0.133	0.124	0.552	0.533	0.157	0.138	0.255	**0.237**	0.048	0.018	**0.223**	**0.221**
	Diff	-0.10	-0.08	-0.03	-0.03	-0.06	-0.08	-0.09	-0.06	0.00	+0.05	-0.06	0.00	+0.01	+0.01
Average	MLP	0.582	0.559	0.540	0.519	0.689	0.683	0.608	0.582	0.328	0.186	0.152	0.016	0.335	0.311
	KAN	0.485	0.480	0.429	0.414	0.627	0.618	0.537	0.518	0.281	0.220	0.183	0.025	0.271	0.262
	FR-KAN	**0.672**	**0.669**	**0.717**	**0.713**	**0.754**	**0.751**	**0.784**	**0.779**	**0.337**	**0.279**	**0.303**	**0.059**	**0.390**	**0.371**
	Diff	+0.09	+0.11	+0.18	+0.19	+0.07	+0.07	+0.18	+0.20	+0.01	+0.09	+0.15	+0.04	+0.05	+0.06

Table 4: Accuracy and F1 score of MLP, KAN, and FR-KAN classification heads on different backbones, evaluated across text classification datasets. FR-KAN consistently outperformed the other heads in both Accuracy and F1 score, with a few exceptions, such as for XLNet. The performance difference between the FR-KAN and the MLP head has been shown.

4 Result Analysis

4.1 Efficacy of FR-KAN head fine-tuning

The FR-KAN head significantly outperformed the MLP (Eq. 11) and KAN (Eq. 3 and 4) classification heads across most dataset-model pairs (Tab. 4 and Appendix Tab. 5), while requiring similar time to fine-tune. Among the models, RoBERTa showed the largest improvement with an average increase of 0.3 in accuracy and 0.35 in F1 score over the MLP heads. The only exception was XLNet, where FR-KAN classifiers performed similarly or slightly worse compared to the MLPs, with an average decrease of 0.05 in accuracy and 0.03 in F1 score. Nonetheless, across all the datasets, the FR-KAN classifiers substantially outperformed both MLPs and KANs with an average increase of 10% in accuracy and 11% in F1 score over MLPs.

4.2 Convergence of FR-KAN head

Following the training and validation loss, the FR-KAN head (Fig. 3c) converges significantly faster than the other two heads (Fig. 3a and 3b) when trained for 50 epochs. Additionally, Fig. 3d and 3e show a substantial increase in both accuracy and F1 score at every training epoch. For instance - the accuracy of the FR-KAN head in the 4th epoch is achieved by the other heads after fine-tuning for more than 20 epochs. The faster convergence of FR-KAN, especially in comparison to spline-based KAN, can be attributed to the smoother functional representation by utilizing the Fourier series.

4.3 Parameter Efficiency of FR-KAN head

We adjust the trainable parameter count of the MLP classifier by varying the hidden layer width of the MLP defined in Eq. 14, to match or exceed the trainable parameter count of the FR-KAN head. The KAN heads with a grid size of 1 have the same number of trainable parameters as the FR-KAN

Method		AgNews		DBpedia		IMDb		Papluca		SST-5		TREC-50		YELP-Full	
Backbone	**Head**	**mF1** ↑	κ ↑	**mF1** ↑	κ ↑	**mF1** ↑	κ ↑	**mF1** ↑	κ ↑	**mF1** ↑	κ ↑	**mF1** ↑	κ ↑	**mF1** ↑	κ ↑
BART	MLP	0.612	0.483	0.808	0.793	**0.769**	**0.538**	0.835	0.826	**0.303**	0.057	0.254	0.109	0.364	0.205
	KAN	0.351	0.136	0.459	0.417	0.594	0.188	0.657	0.639	0.260	0.035	0.283	0.180	0.254	0.068
	FR-KAN	**0.653**	**0.538**	**0.872**	**0.862**	0.749	0.498	**0.880**	**0.873**	0.273	**0.058**	**0.451**	**0.396**	**0.390**	**0.238**
	Diff	+0.04	+0.06	+0.06	+0.07	-0.02	-0.04	+0.04	+0.05	-0.03	0.00	+0.20	+0.29	+0.03	+0.03
BERT	MLP	0.772	0.696	0.752	0.732	0.763	0.527	0.579	0.556	0.374	0.147	0.181	0.003	0.440	0.300
	KAN	0.722	0.630	0.674	0.649	0.730	0.459	0.391	0.357	0.317	0.107	0.186	0.061	0.336	0.169
	FR-KAN	**0.834**	**0.778**	**0.939**	**0.934**	**0.812**	**0.624**	**0.946**	**0.943**	**0.406**	**0.221**	**0.378**	**0.294**	**0.471**	**0.339**
	Diff	+0.06	+0.08	+0.19	+0.20	+0.05	+0.10	+0.37	+0.39	+0.03	+0.07	+0.20	+0.29	+0.03	+0.04
DeBERTa	MLP	0.554	0.405	0.567	0.533	0.710	0.421	0.848	0.840	**0.417**	**0.216**	0.199	0.030	0.393	0.241
	KAN	0.400	0.201	0.357	0.308	0.666	0.333	0.751	0.738	0.292	0.095	0.167	0.064	0.319	0.148
	FR-KAN	**0.595**	**0.460**	**0.648**	**0.621**	**0.770**	**0.539**	**0.920**	**0.916**	0.367	0.178	**0.377**	**0.294**	**0.412**	**0.265**
	Diff	+0.04	+0.05	+0.08	+0.09	+0.06	+0.12	+0.07	+0.08	-0.05	-0.04	+0.18	+0.26	+0.02	+0.02
DistilBERT	MLP	0.836	0.781	0.865	0.854	0.795	0.591	0.878	0.871	0.352	0.117	0.096	0.019	0.429	0.287
	KAN	0.812	0.750	0.843	0.831	0.739	0.477	0.730	0.716	0.307	0.081	0.188	0.033	0.338	0.172
	FR-KAN	**0.877**	**0.836**	**0.970**	**0.968**	**0.831**	**0.662**	**0.986**	**0.985**	**0.401**	**0.208**	**0.351**	**0.258**	**0.492**	**0.365**
	Diff	+0.04	+0.06	+0.11	+0.11	+0.04	+0.07	+0.11	+0.11	+0.05	+0.09	+0.25	+0.24	+0.06	+0.08
ELECTRA	MLP	0.480	0.306	0.364	0.315	0.592	0.186	0.539	0.514	0.326	0.082	0.096	0.007	0.291	0.114
	KAN	0.384	0.178	0.296	0.242	0.558	0.118	0.477	0.450	0.295	0.069	0.179	0.020	0.240	0.051
	FR-KAN	**0.612**	**0.482**	**0.610**	**0.580**	**0.745**	**0.490**	**0.672**	**0.655**	0.326	**0.124**	**0.338**	**0.249**	**0.370**	**0.213**
	Diff	+0.13	+0.18	+0.25	+0.26	+0.15	+0.30	+0.13	+0.14	0.00	+0.04	+0.24	+0.24	+0.08	+0.10
RoBERTa	MLP	0.420	0.224	0.258	0.197	0.578	0.151	0.327	0.292	0.269	0.000	0.132	-0.008	0.209	0.013
	KAN	0.448	0.263	0.253	0.193	0.568	0.129	0.584	0.561	0.269	0.005	0.179	0.000	0.203	0.002
	FR-KAN	**0.836**	**0.781**	**0.844**	**0.832**	**0.819**	**0.639**	**0.925**	**0.921**	**0.328**	**0.089**	**0.179**	0.000	**0.369**	**0.212**
	Diff	+0.42	+0.56	+0.59	+0.64	+0.24	+0.49	+0.60	+0.63	+0.06	+0.09	+0.05	+0.01	+0.16	+0.20
XLNet	MLP	**0.399**	**0.198**	**0.163**	**0.099**	**0.616**	**0.234**	**0.248**	**0.207**	0.255	0.011	**0.105**	**0.022**	0.218	0.021
	KAN	0.275	0.035	0.123	0.055	0.536	0.070	0.166	0.122	0.225	0.013	0.102	-0.001	0.207	0.008
	FR-KAN	0.300	0.066	0.133	0.066	0.552	0.101	0.157	0.112	0.255	**0.048**	0.048	0.003	**0.223**	**0.029**
	Diff	-0.10	-0.13	-0.03	-0.03	-0.06	-0.13	-0.09	-0.09	0.00	+0.04	-0.06	-0.02	0.00	+0.01
Average	MLP	0.582	0.442	0.540	0.503	0.689	0.378	0.608	0.587	0.328	0.090	0.152	0.026	0.335	0.169
	KAN	0.485	0.313	0.429	0.385	0.627	0.253	0.537	0.512	0.281	0.058	0.183	0.051	0.271	0.088
	FR-KAN	**0.673**	**0.563**	**0.717**	**0.695**	**0.754**	**0.508**	**0.784**	**0.772**	**0.337**	**0.132**	**0.303**	**0.213**	**0.390**	**0.237**
	Diff	+0.09	+0.12	+0.18	+0.19	+0.06	+0.13	+0.18	+0.19	+0.01	+0.04	+0.15	+0.19	+0.05	+0.07

Table 5: Micro F1 and Kappa score of MLP, KAN, and FR-KAN classification heads on different backbones, evaluated across text classification datasets. Similar to Table 4, the FR-KAN head consistently outperforms the MLP and KAN heads.

heads with a grid size of 5. As observed in Tab. 3, the FR-KAN heads not only outperform MLP and KAN heads in terms of performance but do so while requiring equal or fewer parameters.

4.4 Impact of Grid Size on Performance

The grid size signifies how fine-grained the FR-KAN coefficients will be in a single FR-KAN layer. Aligning with Corollary 1, increasing the grid size leads to performance improvement, though, the gains diminish at higher grid sizes due to convergence, as evident from Fig. 3f. Models may also experience overfitting at larger grid sizes, as shown in Tab. 6, where a slight drop in performance is observed at higher grid sizes.

5 Discussion

5.1 B-Splines vs Fourier Series

Both the B-splines of the original KAN implementation and the Fourier series of the FR-KAN head are used to approximate continuous univariate functions. Both methods work well with smooth, continuous, and low-dimensional functions while being computationally inexpensive. At points of discontinuities, splines require higher degree functions while Fourier series suffer from oscillations due to the Gibbs phenomenon.

The Fourier series has several key advantages over splines. By nature, the Fourier representation uses sines and cosines instead of the piece-wise polynomials in splines. Hence, the Fourier series can represent smoother periodic functions which can be advantageous as smoothness can improve KAN performance [29]. The Fourier series also has better global control compared to the better local control of splines which can contribute to improved parameter efficiency in certain tasks. However, one key downside of the Fourier series is the decline of interpretability in comparison to splines – although both methods are significantly more interpretable than perceptrons.

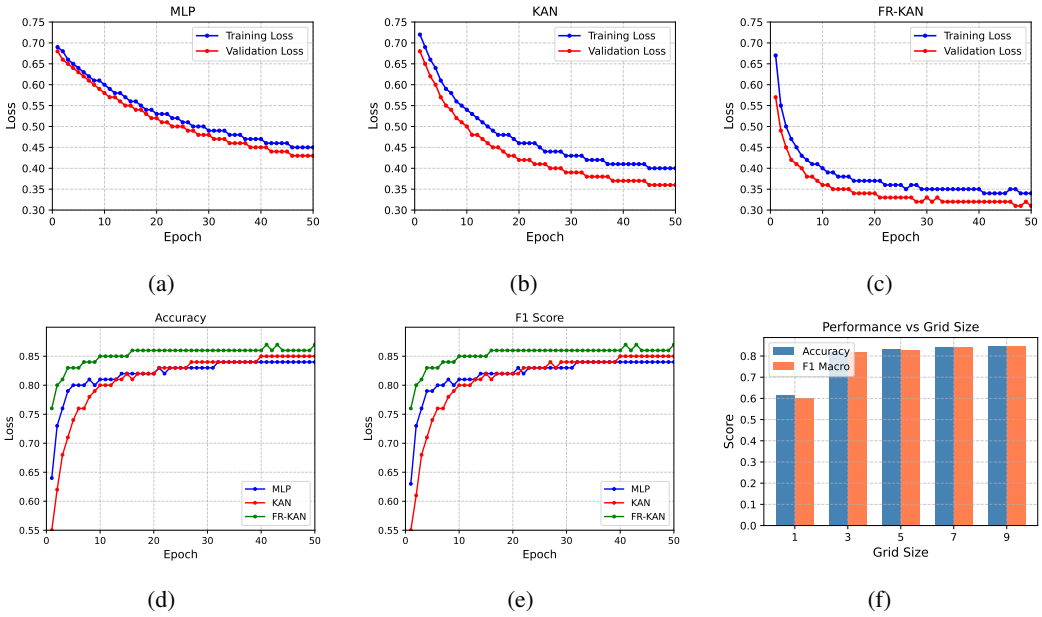

Figure 3: Results of the DistilBERT model on the IMDb dataset. For different classification heads, (a)-(c) training and validation loss, (d) accuracy, and (e) F1 score. For the FR-KAN head, (f) accuracy and F1 score at varying grid sizes.

5.2 Evaluation Fairness

Throughout the work, we gave utmost importance to ensuring that all three classification heads were evaluated on equal grounds. This involved using the same backbone architecture, dataset splits, and training configuration across all experiments. We further attempted to evaluate the classifiers with an equal number of trainable parameters to mitigate biases induced by training larger models. The experiments were conducted using multiple seed values in the same hardware configuration and the average result had been taken. We concur that the results might differ based on the pre-trained model weights, dataset splits, and training configuration but the differences are expected to be negligible.

6 Broader Impact

6.1 Greener Approach

Pre-training and fully fine-tuning large networks can be attributed to high energy consumption and substantial carbon footprint [30]. Classification head fine-tuning offers a greener alternative, achieving slightly worse or better [2] performance compared to fully fine-tuning. These transfer learning strategies also ensure the reusability of pre-trained weights, consequently avoiding redundant pre-training. The proposed FR-KAN heads train faster than the standard MLP heads, thereby consuming less resources and leaving less carbon footprint.

6.2 Universal MLP Alternative

The promising empirical results of FR-KAN heads in the domain of text classification affirms their potential as a generalized MLP alternative. We envision that FR-KANs will not be limited to classification heads only and can be incorporated as neural network layers, *e.g* within the transformer architecture. The adaptability can be explored in other domains *e.g* computer vision, time series analysis, and speech analysis.

Grid Size		1		2		3		4		5	
Dataset	**Method**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**	**Acc**	**F1**
AgNews	DistilBERT	0.812	0.813	0.832	0.829	0.852	0.852	0.864	0.862	0.877	0.876
DBpedia	DistilBERT	0.739	0.710	0.892	0.890	0.951	0.951	0.961	0.961	0.970	0.971
IMDb	DistilBERT	0.616	0.601	0.808	0.808	0.817	0.817	0.830	0.830	0.831	0.830
Papluca	DistilBERT	0.768	0.754	0.933	0.934	0.976	0.977	0.984	0.983	0.986	0.986
SST-5	BERT	0.310	0.197	0.376	0.240	<u>0.396</u>	0.316	0.394	0.330	0.406	0.339
TREC-50	BART	0.343	0.046	0.412	0.086	0.430	0.096	<u>0.467</u>	<u>0.125</u>	0.451	0.122
YELP-Full	DistilBERT	0.385	0.328	0.420	0.401	0.457	0.439	0.466	0.453	0.492	0.475

Table 6: Change in accuracy and F1 score with grid size for the best performing FR-KAN models from Tab. 4. While the performance of most models improves with the increase in grid size, a few cases where a smaller grid size outperforms the larger grid size are <u>underlined</u>. The grid size of 5 usually shows the best performance for all the benchmarks and has been chosen as the default grid size for all experiments.

7 Related Work

Multi-layer Perceptrons (MLPs) Inspired by the neurons in the biological brain, the original unilayer perceptron dates back to the 50s and was intended as a machine for pattern recognition [31]. Initially constrained by its single-layer architecture, the methodology was later expanded to multi-layer perceptrons based on the universal approximation theorem which states that a continuous function can be approximated by a feedforward network with a finite number of neurons [3]. During the deep learning era, researchers quickly adopted MLPs as a fundamental module in several foundational deep learning architectures [6, 7]. The versatility and adaptability of MLPs have made them the most popular choice for classification heads in virtually all classification-based tasks [32].

Kolmogorov-Arnold Networks (KANs) KAN [13] is based on the Kolmogorov-Arnold representation theorem [33] stating that a continuous multivariate function is a composition of multiple continuous univariate functions and addition operations. Boasting faster neural scaling laws and interpretability KANs became popular in multiple domains including time series analysis [34] and forecasting [35, 36], satellite image classification [37], mechanics problems [38], and quantum architecture search [39]. KANs have developed multiple variations utilizing the wavelet transform [40], Jacobi basis functions [41], radial basis functions [42], and several functional combinations [43, 44]. We highlight the work of [15], where the Fourier KAN was introduced to improve graph collaborative filtering in recommendation tasks.

Transformers in Text Classification The transformer architecture [4], initially introduced for sequence-to-sequence generation, has revolutionized various domains of NLP, including text classification [45]. While primarily proposed as encoder-decoder architecture, pre-trained transformer encoders, such as BERT [22] and its variants [24, 26, 23] have been popular in natural language sequence classification tasks.

Linear Probing Full fine-tuning demands significant computational resources and is susceptible to overfitting on smaller datasets [46, 1]. In contrast, finetuning the classification head, *i.e.* linear probing, can be a resource-efficient alternative with enhanced robustness on out-of-distribution data [2]. Linear probing can be enhanced via several strategies, *e.g.* parameter-efficient tuning [47].

8 Conclusion

Our work explores Fourier-KAN as a promising alternative to MLPs for text classification using pre-trained transformer classifiers. We find that FR-KANs perform better, require fewer parameters, and train faster compared to MLPs. In the future, we wish to investigate the potential of FR-KANs replacing MLPs inside the transformer architectures.

Limitations

Although significant strides have been made by the FR-KAN head over its KAN predecessor, the improved performance comes at the expense of interpretability. We were also unable to evaluate the classification heads with a lower parameter count, as the number of parameters in KAN heads is constrained by the grid size, which cannot be reduced beyond the minimum value of 1. Finally, our study is limited to the use of B-splines and Fourier series as the univariate function in the KAN representation. Alternative methods of function approximation might produce interesting results.

References

[1] M. Gao, Q. Wang, Z. Lin, P. Zhu, Q. Hu, and J. Zhou, "Tuning pre-trained model via moment probing," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 803–11 813.

[2] A. Kumar, A. Raghunathan, R. Jones, T. Ma, and P. Liang, "Fine-tuning can distort pretrained features and underperform out-of-distribution," *arXiv preprint arXiv:2202.10054*, 2022.

[3] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

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

[5] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Transactions on industrial electronics*, vol. 66, no. 6, pp. 4788–4797, 2018.

[6] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[8] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates *et al.*, "Deep speech: Scaling up end-to-end speech recognition," *arXiv preprint arXiv:1412.5567*, 2014.

[9] H. Mühlenbein, "Limitations of multi-layer perceptron networks-steps towards genetic neural networks," *Parallel Computing*, vol. 14, no. 3, pp. 249–260, 1990.

[10] H. Cunningham, A. Ewart, L. Riggs, R. Huben, and L. Sharkey, "Sparse autoencoders find highly interpretable features in language models," *arXiv preprint arXiv:2309.08600*, 2023.

[11] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.

[12] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[13] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark, "Kan: Kolmogorov-arnold networks," *arXiv preprint arXiv:2404.19756*, 2024.

[14] V. Dhiman, "Kan: Kolmogorov–arnold networks: A review," 2024.

[15] J. Xu, Z. Chen, J. Li, S. Yang, W. Wang, X. Hu, and E. C.-H. Ngai, "Fourierkan-gcf: Fourier kolmogorov-arnold network–an effective and efficient feature transformation for graph collaborative filtering," *arXiv preprint arXiv:2406.01034*, 2024.

[16] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, 2011, pp. 142–150.

[17] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[18] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," *Advances in neural information processing systems*, vol. 28, 2015.

[19] E. M. Voorhees and D. M. Tice, "The TREC-8 question answering track," in *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, M. Gavrilidou, G. Carayannis, S. Markantonatou, S. Piperidis, and G. Stainhauer, Eds. Athens, Greece: European Language Resources Association (ELRA), May 2000. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2000/pdf/26.pdf

[20] L. Papariello, "Papluca language identification," https://huggingface.co/datasets/papluca/language-identification, accessed: 2024-06-14.

[21] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[23] P. He, X. Liu, J. Gao, and W. Chen, "Deberta: Decoding-enhanced bert with disentangled attention," *arXiv preprint arXiv:2006.03654*, 2020.

[24] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[25] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," *arXiv preprint arXiv:2003.10555*, 2020.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[27] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[29] M. E. Samadi, Y. Müller, and A. Schuppert, "Smooth kolmogorov arnold networks enabling structural knowledge representation," *arXiv preprint arXiv:2405.11318*, 2024.

[30] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, and J. Dean, "Carbon emissions and large neural network training," *arXiv preprint arXiv:2104.10350*, 2021.

[31] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[32] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, 2018.

[33] V. Tikhomirov, "On the representation of continuous functions of several variables as superpositions of continuous functions of one variable and addition," in *Selected Works of AN Kolmogorov*. Springer, 1991, pp. 383–387.

[34] C. J. Vaca-Rubio, L. Blanco, R. Pereira, and M. Caus, "Kolmogorov-arnold networks (kans) for time series analysis," *arXiv preprint arXiv:2405.08790*, 2024.

[35] K. Xu, L. Chen, and S. Wang, "Kolmogorov-arnold networks for time series: Bridging predictive power and interpretability," *arXiv preprint arXiv:2406.02496*, 2024.

[36] R. Genet and H. Inzirillo, "A temporal kolmogorov-arnold transformer for time series forecasting," *arXiv preprint arXiv:2406.02486*, 2024.

[37] M. Cheon, "Kolmogorov-arnold network for satellite image classification in remote sensing," *arXiv preprint arXiv:2406.00600*, 2024.

[38] D. W. Abueidda, P. Pantidis, and M. E. Mobasher, "Deepokan: Deep operator network based on kolmogorov arnold networks for mechanics problems," *arXiv preprint arXiv:2405.19143*, 2024.

[39] A. Kundu, A. Sarkar, and A. Sadhu, "Kanqas: Kolmogorov arnold network for quantum architecture search," *arXiv preprint arXiv:2406.17630*, 2024.

[40] Z. Bozorgasl and H. Chen, "Wav-kan: Wavelet kolmogorov-arnold networks," *arXiv preprint arXiv:2405.12832*, 2024.

[41] A. A. Aghaei, "fkan: Fractional kolmogorov-arnold networks with trainable jacobi basis functions," *arXiv preprint arXiv:2406.07456*, 2024.

[42] Z. Li, "Kolmogorov-arnold networks are radial basis function networks," *arXiv preprint arXiv:2405.06721*, 2024.

[43] H.-T. Ta, D.-Q. Thai, A. B. S. Rahman, G. Sidorov, and A. Gelbukh, "Fc-kan: Function combinations in kolmogorov-arnold networks," *arXiv preprint arXiv:2409.01763*, 2024.

[44] H.-T. Ta, "Bsrbf-kan: A combination of b-splines and radial basic functions in kolmogorov-arnold networks," *arXiv preprint arXiv:2406.11173*, 2024.

[45] W. Cunha, F. Viegas, C. França, T. Rosa, L. Rocha, and M. A. Gonçalves, "A comparative survey of instance selection methods applied to non-neural and transformer-based text classification," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–52, 2023.

[46] D. Lian, D. Zhou, J. Feng, and X. Wang, "Scaling & shifting your features: A new baseline for efficient model tuning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 109–123, 2022.

[47] Z. Yang, M. Ding, Y. Guo, Q. Lv, and J. Tang, "Parameter-efficient tuning makes a good classification head," *arXiv preprint arXiv:2210.16771*, 2022.