DFT-Trans: A Bidirectional Encoder for Efficient Fusion of Time-Frequency Domain Textual Features

Anonymous ACL submission

Abstract

Despite the remarkable achievements of BERT-style encoder models in NLP research, the high computational costs make it challenging to pre-train specific BERTs from scratch. This work proposes a novel BERT-style encoder model called DFT-Trans, addressing the critical question of enhancing performance while reducing training costs. The DFT-Trans model is primarily composed of the trainable Fourier operator and the attention operator. The novel trainable Fourier operator, which consists of the unique Blending Token and Mixing Token methods, is developed, given that frequency domain features are seldom considered in text representation extraction. This operator utilizes fast Fourier transform(FFT) to capture data features in the frequency domain, integrating frequency information into the network's structure and computations, enabling more robust feature extraction capabilities. The attention operator is designed by combining FlashAttention and Attention with Linear Bias to address the quadratic time and memory complexity inherent to self-attention while efficiently extracting features from time-domain data. When pre-trained from scratch on large-scale corpora, DFT-Trans achieves an average downstream GLUE(dev) score of 80.6% using a single RTX 4090 GPU in one day, with a cost of approximately \$5. Furthermore, we experimented on the Long-Range Arena(LRA) benchmark, where DFT-Trans achieved an average task score of 75.94%, demonstrating its effectiveness in long-text scenarios. Code is available at this repository: `https://anonymous.4open.science/r/DFT-Trans-3FDD`.

1 Introduction

BERT-style encoder models, as bidirectional encoders, are widely utilized in natural language processing(NLP). Primarily composed of self-attention mechanisms, these models achieve notable performance across downstream tasks such as text classification, sequence labeling, and semantic similarity matching(Devlin et al., 2019; Liu et al., 2019; Lan et al., 2019; Joshi et al., 2020; He et al., 2020; Yang et al., 2019) when pre-trained on the large-scale corpus. In recent years, the release and success of prominent models like T5(Raffel et al., 2020), ChatGPT(Achiam et al., 2023), GLM(Zeng et al., 2022), and Llama(Touvron et al., 2023) have led to a surge in the interest and research of large language models (LLMs). However, BERT-style encoder models remain highly relevant even in the LLM era. For example, encoder models are used in tasks such as data vectorization, retrieval-augmented generation, and intent recognition(Lewis et al., 2020; Wang et al., 2022; Weld et al., 2022). These tasks often demand shorter training times and improved performance, posing significant challenges for BERT-style encoder models.

Many BERT-style encoder models have been designed to enhance performance while reducing training costs(Tay et al., 2022). Recent studies(Izsak et al., 2021; Geiping and Goldstein, 2023; Portes et al., 2023; Belcak and Wattenhofer, 2024) have aimed to achieve high-performance models with minimal training costs, primarily relying on the vanilla self-attention mechanism(Vaswani et al., 2017). Due to the quadratic complexity of self-attention mechanisms(Lin et al., 2017), substituting them with multi-layer perceptrons (MLPs) has shown promising results without pretraining(Mai et al., 2023; Tolstikhin et al., 2021a). However, MLPs generally struggle to improve performance on downstream tasks with pretraining. Recently, models such as Mamba(Gu and Dao, 2024) have been based on structured state spaces for efficient sequence modeling. At the same time, TNN has utilized Toeplitz matrices with relative position encoding to model sequences, leveraging the logarithmic complexity of Toeplitz matrix computations(Qin et al., 2023). Compared to vanilla self-attention-based BERT-style encoder models, these studies(Gu and Dao, 2024; Qin et al., 2023) have relatively reduced computational complexity but did not fully explore performance after pretraining. In the domain of computer vision, modeling frequency domain features via Fourier transforms is both common and effective. Models such as (Rao et al., 2023; Patro and Agneeswaran, 2023; Guibas et al., 2021) have applied filtering in the frequency domain to extract richer representations from images. In NLP, some studies have replaced self-attention mechanisms with Fourier transforms to reduce computational costs and enhance performance after pretraining, including FNet(Lee-Thorp et al., 2022), Fourier Transformer(He et al., 2023), FAN(Dong et al., 2024), and FSRU(Lao et al., 2024). However, the integration and distinction of text features in the frequency domain remain underexplored. Dependencies between features in the frequency domain vary (e.g., low-frequency vs high-frequency features), and blending similar features can capture more comprehensive representations and vice versa. For instance, declarative sentences (low-frequency) and turnaround sentences (high-frequency) exhibit distinct features in the frequency domain. As shown in Figure 1(A), the orange square represents declarative sentences, and the purple diamond represents turnaround sentences. The figure indicates that compared to declarative sentences, turnaround sentences are more symmetrical, with lower and more stable magnitudes.

In this work, we propose a novel BERT-style encoder model called DFT-Transforms (DFT-Trans), designed to dynamically learn text features in the frequency domain while preserving those in the time domain. DFT-Trans is composed primarily of optimized trainable Fourier operators and attention operators. Given the limited use of frequency domain features in text representation, we designed the trainable Fourier operator to process data transformed via fast Fourier transform (FFT)(Cooley and Tukey, 1965), integrating frequency information into the network's structure and training process. The trainable Fourier operator comprises 1-D discrete Fourier transform (DFT), unique Blending Token and Mixing Token methods, and 1-D inverse Fourier transform (iDFT). The 1-D DFT converts the input text features from the time domain to the frequency domain. The Blending Token performs Einstein multiplication(Patro and Agneeswaran, 2023) between frequency domain features and dynamic mixing matrices to extract local features. The Mixing Token performs matrix multiplication between frequency domain features and trainable matrices to extract global features. The 1-D iDFT maps the features back to the time domain. By learning global and local frequency domain features, the trainable Fourier operator enables DFT-Trans to capture both long-range and short-range dependencies across texts in the frequency domain. Previous studies have demonstrated the importance of time-domain text features(Lipton, 2015; Shi et al., 2015; Vaswani et al., 2017). To this end, we constructed the attention operator to process time-domain features. The attention operator, built by combining FlashAttention(Dao et al., 2022) and Attention with Linear Bias(ALiBi)(Press et al., 2022), reduces computational complexity while extending input length, thereby enhancing the inference ability to text longer.

We conducted diverse experiments on the GLUE(Wang et al., 2019) and Long Range Arena(LRA)(Tay et al., 2021b) benchmarks to evaluate the effectiveness and efficiency of DFT-Trans. To verify the impact of pretraining on the performance of DFT-Trans, we pretrained the model on large-scale corpora and tested its downstream task performance. Experimental results demonstrate that DFT-Trans outperforms other models, such as Mamba(Gu and Dao, 2024), MosaicBERT(Portes et al., 2023), and Cramming BERT(Geiping and Goldstein, 2023), on the GLUE benchmark. Similarly, without pretraining, DFT-Trans surpasses MLP-based models(Tolstikhin et al., 2021a; Mai et al., 2023) on the GLUE benchmark. Furthermore, to validate the model's capability in long-text scenarios, we conducted evaluations on the LRA benchmark. Results show that DFT-Trans achieves state-of-the-art performance among Transformer-based efficient models while maintaining a short runtime. These findings indicate that DFT-Trans reduces training costs while enhancing performance across both general and long-text scenarios.

In summary, our contributions can be enumerated as follows:

- Based on FFT, we propose a novel BERT-style encoder model that effectively integrates time and frequency domain information.
- We introduce the trainable Fourier operator, including Blending Token and Mixing Token methods, which extract global and local features.
- We combine FlashAttention and ALiBi to construct the attention operator, improving both training speed and accuracy.

- We analyze the performance of DFT-Trans against other BERT-style encoder models and advanced alternatives on the GLUE and LRA benchmarks.

Finally, the goal of this work is to show relative improvements in performance and training costs in comparison with Bert-style encoder models. We do not compare our model with the current optimal LLMs on GLUE benchmark(Wang et al., 2019). Because LLMs are trained for much longer, which is far superior to the models we explore in this work.

2 Related work

Many researchers are exploring improvements to the BERT-style model with the aim of reducing the cost of the model. The directions for improvement fall into (1) Exploration of model structure and pretraining methods under the reserved Attention.(Exploration of model and pretraining) (2) Dropping Attention and using simpler feature extraction methods.(Replacing Attention with MLPs)

2.1 Exploration of model and pretraining

Most of the BERT-style models have mostly retained Attention(Vaswani et al., 2017), and its training processes are: (1)Self-supervised pretraining allows the model to learn the general feature representation of the sentence. (2)Supervised fine-tuning allows the model to learn the representation of features in a specific domain.

The process of self-supervised pretraining is time and GPUs-consuming; for example, in the study by BERT(Devlin et al., 2019), the authors trained their model on 16 TPUs for about four days to complete. Due to the large parameters of the BERT model, researchers have proposed Albert(Lan et al., 2019), which uses parameter sharing to reduce the model parameters. Roberta(Liu et al., 2019) and SpanBERT(Joshi et al., 2020) have removed the NSP task and improved MLM to speed up training while improving performance. XLNet(Yang et al., 2019) has improved the model's ability to learn bidirectional context and achieved good results on many tasks. Recently study(Izsak et al., 2021) have improved on pretraining, reducing the training time of BERT to 24 hours. MosaicBERT(Portes et al., 2023), Cramming BERT(Geiping and Goldstein, 2023) have adapted the Attention structure in BERT to reduce significantly the pretraining time and match BERT in performance. NarrowBert(Li et al., 2023) has sparsified the encoder so that it can focus on the Masked Token. SpikingBERT(Bal and Sengupta, 2024) has introduced a spiking attention mechanism, which reduces the computational cost of the model. The model we designed is based on the above model approach is considered.

2.2 Replacing Attention with MLPs

Recently studies have found attention to have a great deal of complexity (Choromanski et al., 2021; Zhai et al., 2021; Tay et al., 2021a). Star-Transformer(Guo et al., 2019) has proposed a star topology instead of fully connected attention, significantly reducing the complexity. In addition, considering that FFT has low time complexity, FFT has been introduced into long text classification (He et al., 2023) aiming to make the attention mechanism scale better via sparsity patterns(Child et al., 2019; Qiu et al., 2020; Parmar et al., 2018; Beltagy et al., 2020; Ainslie et al., 2020; Zaheer et al., 2020; Wang et al., 2020) or linearization of the attention matrix(Katharopoulos et al., 2020; Choromanski et al., 2021; Peng et al., 2021). In the field of computer vision and multimodality, there are many approaches that have proposed to use the MLP-Like model instead of Attention (Chen et al., 2022; Tolstikhin et al., 2021b; Hou et al., 2023; Lao et al., 2024), which not only reduced the cost of the model, but also further improved the performance. gMLP (Liu et al., 2021), pNLP-Mixer (Fusco et al., 2023) and hyperMixer (Mai et al., 2023) have applied standard MLP-like on NLP to simulate the effect of attention and achieved good results on specific tasks. The disadvantage of this type of MLP-Like based model is that it is not possible to improve the performance of the model by pretraining.

3 Methods

In this section, we introduce DFT-Trans in detail, which is implemented based on the Trainable Fourier operator and time domain Attention operator. Our model is the Bert-style encoder model, with the objective of allowing the model to extract frequency information. As shown in Figure 2, the model has $M + N$ layers.

3.1 The Overall Structure

The input sentence $\mathbf{S}$ is encoded into a feature vector $\mathbf{X} \in \mathbb{R}^{L \times H}$ by embedding layer, L represents the length of the input sentence, and H represents

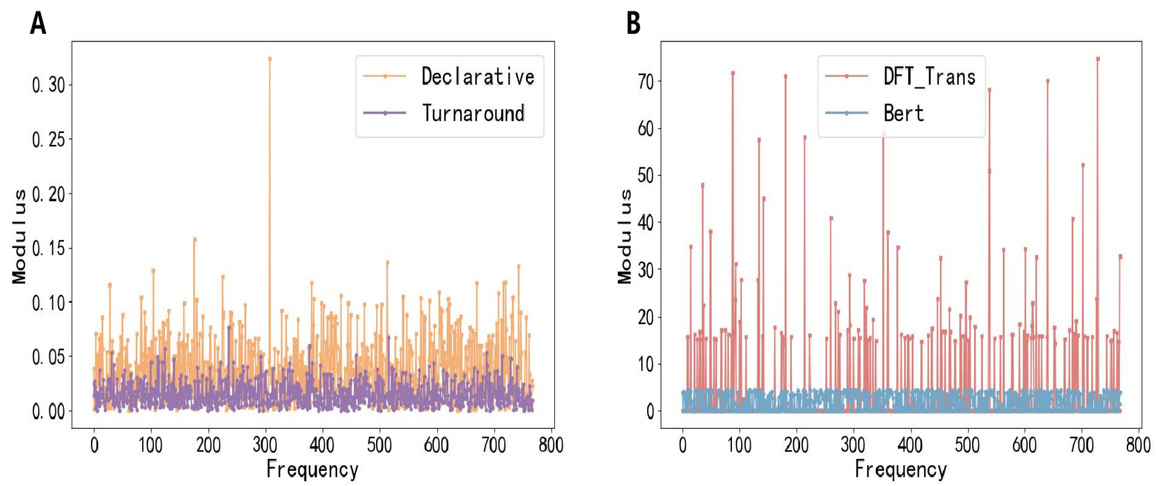

Figure 1: (A) study of turnaround(The sun sets in the west, casting a golden glow across the sky.) and declarative(He was tired; nevertheless, he continued working He was tired; nevertheless, he continued working through the night.) sentence in the frequency domain.(B) The difference between the DFT-Trans and Bert (Devlin et al., 2019) models in the frequency domain when dealing with the same sentence.

the size of the hidden state. Then, we input the feature vector $\mathbf{X}$ into subsequent network layers to extract frequency and temporal feature.

The purpose of the trainable Fourier operator is to allow DFT-Trans to handle high and low-frequency information. By computing DFT of the feature vectors $\mathbf{X} \in \mathbb{R}^{L \times H}$(The calculation methodology has already been proposed in (Cooley and Tukey, 1965), and is known as FFT), we get $\mathcal{Z} = \mathcal{F}_{FFT}(\mathbf{X})$. Since the input data is a sequence of real numbers, $\mathcal{Z}$ is split into two parts: real part $\mathcal{Z}_{real} \in \mathbb{R}^{L \times H}$, and imaginary part $\mathcal{Z}_{imag} \in \mathbb{R}^{L \times H}$. The frequency components are defined as:

$$\mathcal{Z} = \mathcal{Z}_{real} + j\mathcal{Z}_{imag} \tag{1}$$

$\mathcal{Z}_{real}$ and $\mathcal{Z}_{imag}$ both contain high and low-frequency information, and we need to fuse them in the Trainable Fourier operator accordingly.

3.2 Details of Attention Operator

To allow the model to be pretrained quickly and to be able to handle long text scenarios, referring to the model designed by MosaicBERT (Portes et al., 2023), we introduce Flash Attention and ALiBi.

Flash Attention: Flash Attention (Dao et al., 2022) was proposed to reduce the number of reads and writes between GPU HBM and GPU SRAM.

Attention with Linear Biases(ALiBi): ALiBi eliminates positional embedding and adds positional encoding information to the Attention operation. It does this by adding a negative bias to the attention score of the token for each text that grows linearly as the relative distance between tokens increases. Following the notation in (Press et al., 2022), the Attention block calculates the ith query $q_i \in \mathbb{R}^d$ as well as the key $K \in \mathbb{R}^{L \times d}$, where d is the head dimension and L is the length of the sequence, using the following equation:

$$Softmax(q_i \mathbf{K}^T - m \cdot \mathbf{abs}([i-1, i-2, ..., i-L])) \tag{2}$$

where m is the slope of each header used to control the growth of the bias. The slopes m follow a geometric sequence such that for n heads, each head has a ratio of $2^{\frac{-8}{n}}$, where $d = H/n$.

3.3 Trainable Fourier Operator

Inspired by AFNO-transformers (Guibas et al., 2021) in images, we design a novelty trainable Fourier Operator in the frequency domain. Trainable Fourier operators are used to efficiently extract the global and local frequency domain features of the text information after the Fourier transform.

In Figure 2, the frequency-domain text features $\mathcal{Z}^0$ converted by DFT are iteratively integrated and distinguished through the Blending Token and Mixing Token in Spectrum within the trainable Fourier operator. Assuming that the input and output of Blending Token and Mixing Token in Spectrum respectively are $\mathcal{Z}^{\mathbb{l}-1} \in \mathbb{C}^{L \times N_{block} \times H_{block}}$ and $\mathcal{Z}^{\mathbb{l}} \in \mathbb{C}^{L \times N_{block} \times H_{block}}$, where $N_{block} \times H_{block} = H$. The current output $\mathcal{Z}^{\mathbb{l}}$ is obtained by fusing the current input $\mathcal{Z}^{\mathbb{l}-1}$ with either the dynamic mixing matrix $\mathcal{W}_{\psi}^{\mathbb{l}} \in \mathbb{C}^{N_{block} \times H_{block} \times H_{block}}$ in Blending Token or the trainable matrix $\mathcal{W}_{\phi}^{\mathbb{l}} \in \mathbb{C}^{H_{block} \times H_{block}}$ in Mixing Token. The fusion process can be formulated as follows:

$$\mathcal{Z}^{\mathbb{l}} = \begin{cases} \Phi(\mathcal{Z}^{\mathbb{l}-1}, \mathcal{W}_{\phi}^{\mathbb{l}}) & when\ \mathbb{l} \geq 2 \\ \Psi(\mathcal{Z}^{\mathbb{l}-1}, \mathcal{W}_{\psi}^{\mathbb{l}}) & otherwise \end{cases} \quad (\mathbb{l} = 1, 2, ..., \mathbb{L}) \tag{3}$$

$$\mathcal{Z}^0 = \mathcal{F}_{FFT}(\mathbf{X}^l) \qquad \mathbf{X}^l \in \mathbb{R}^{L \times N_{block} \times H_{block}} \tag{4}$$

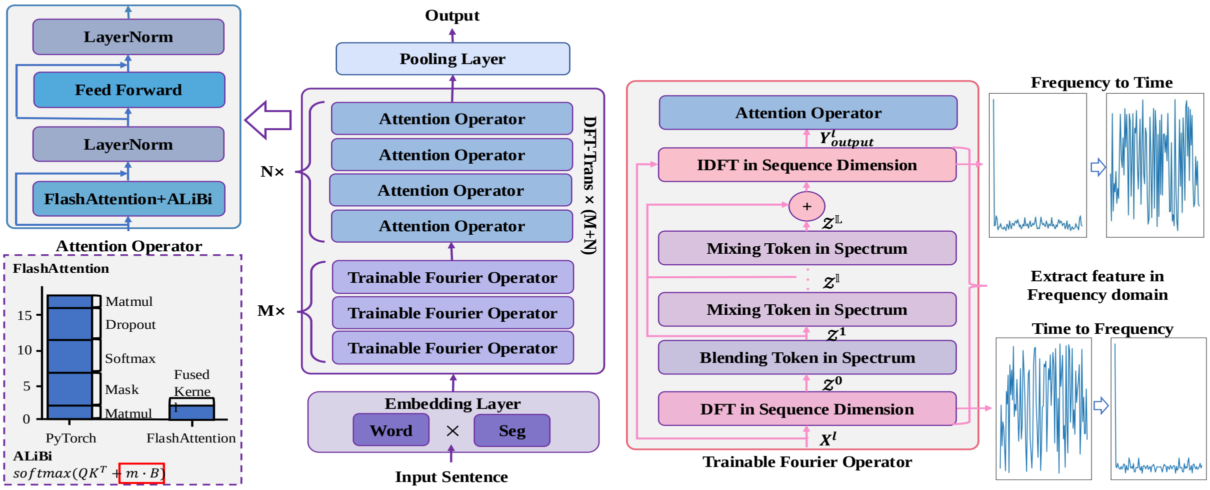

Figure 2: Overall architecture of DFT-Trans. The model consists of M+N layers, which include trainable Fourier operators to extract frequency-domain features and attention operators to capture temporal-domain features.

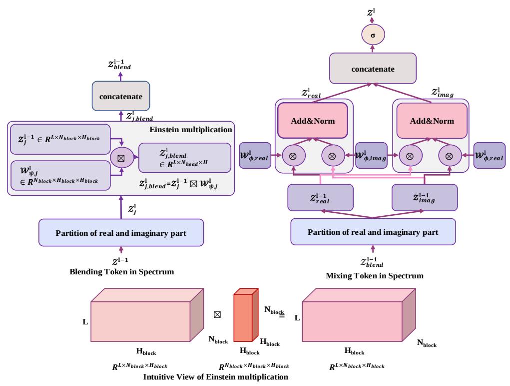

Figure 3: Implementation of Blending Token in Spectrum and Mixing Token in Spectrum in Spectrum.

$$\mathbf{Y}^l_{output} = i\mathcal{F}_{FFT}\left(\sum_{\mathbb{l}=1}^{\mathbb{L}} \sigma_{soft}(\mathcal{Z}^{\mathbb{l}}) + \mathcal{Z}^0\right) \qquad (5)$$

In equation (3), $\Phi(\mathcal{Z}^{\mathbb{l}-1}, \mathcal{W}^{\mathbb{l}}_{\phi})$ is detailed in equation (6), (7), (8) and $\Psi(\mathcal{Z}^{\mathbb{l}-1}, \mathcal{W}^{\mathbb{l}}_{\psi})$ is detailed in equation (9). $\mathbb{L}$ denotes the total number of layers of the Blending Token and the Mixing Token in the Spectrum. $\mathbf{X}^l$, $\mathbf{Y}^l_{output}$ respectively are the inputs and outputs of the Extract feature in the Frequency domain on the far right of Figure 2. l denotes the layer of Trainable Fourier Operator; note the difference with $\mathbb{l}$. $\mathcal{Z}^0$ is obtained by performing 1D FFT accordingly on the text features X^l in the time domain along the dimension L and N_{block}.

Since both $\mathcal{Z}^{\mathbb{l}-1}$ and $\mathcal{W}^{\mathbb{l}}_{\phi}$ are vectors represented in complex form, the vector multiplication between them requires separate operations for the real and imaginary parts(proof in Appendix A). Figure 3 shows in detail the multiplication operation performed by Equation 3. For the Mixing Token, we use the matrix multiplication:

$$\mathcal{Z}^{\mathbb{l}}_{real} = \sigma(\mathcal{Z}^{\mathbb{l}-1}_{real} \cdot \mathcal{W}^{\mathbb{l}}_{\phi,real} - \mathcal{Z}^{\mathbb{l}-1}_{imag} \cdot \mathcal{W}^{\mathbb{l}}_{\phi,imag} + \mathbb{B}^{\mathbb{l}}_{real}) \quad (6)$$

$$\mathcal{Z}^{\mathbb{l}}_{imag} = \sigma(\mathcal{Z}^{\mathbb{l}-1}_{real} \cdot \mathcal{W}^{\mathbb{l}}_{\phi,imag} + \mathcal{Z}^{\mathbb{l}-1}_{imag} \cdot \mathcal{W}^{\mathbb{l}}_{\phi,real} + \mathbb{B}^{\mathbb{l}}_{imag}) \quad (7)$$

$$\mathcal{Z}^{\mathbb{l}} = \mathcal{Z}^{\mathbb{l}}_{real} + j\mathcal{Z}^{\mathbb{l}}_{imag} \quad (8)$$

We use the Einstein Blending Method(Patro and Agneeswaran, 2023)(EBM) for the Blending Token. We perform EBM between $\mathcal{Z}^{\mathbb{l}-\mathbb{1}}$ and $\mathcal{W}^{\mathbb{l}}_{\psi}$ along the last two dimensions:

$$\begin{gathered} \mathcal{Z}^{\mathbb{l}}_{j} = \mathcal{Z}^{\mathbb{l}-1} \boxtimes \mathcal{W}^{\mathbb{l}}_{\psi} + \mathbb{B}^{\mathbb{l}}_{j}, j\ means\ real\ or\ imag \\ \mathcal{Z}^{\mathbb{l}}_{j} \in \mathbb{R}^{L \times N_{block} \times H_{block}}\ \mathcal{Z}^{\mathbb{l}-1} \in \mathbb{R}^{L \times N_{block} \times H_{block}} \\ \mathcal{W}^{\mathbb{l}}_{\psi} \in \mathbb{R}^{N_{block} \times H_{block} \times H_{block}} \end{gathered} \quad (9)$$

The dynamic mixing matrix $\mathcal{W}_{\psi}$ in the Blending Token introduces additional parameters to cover all frequency domain features, which is used to extract local features in the frequency domain. The trainable matrix $\mathcal{W}_{\phi}$ in the Mixing Token uses straightforward matrix multiplication to directly fuse frequency domain features, which is used to extract global features in the frequency domain. Through the efficient combination of them, the trainable Fourier operator is capable of extracting both global and local features in the frequency domain. We will further discuss the advantages of our models in the following subsection.

3.4 Advantages of DFT-Trans

To explore the advantages of DFT-Trans in integrating and distinguishing frequency-domain information, we have respectively plotted the frequency-domain text feature maps for declarative and turnaround sentences(shown in Figure 1(A)), as well as comparative diagrams of DFT-Trans and BERT in the frequency domain when processing the same sentence(shown in Figure 1(B)).

In Figure 1(A), we utilize a general text embedding model(Xiao et al., 2023) to extract the semantic features of declarative and turnaround sentences separately. Subsequently, these features are converted into the frequency domain using FFT, and the modulus is obtained. It can be observed that its low-frequency components carry greater weight for declarative sentences because of thematic monotony. In contrast, the turnaround sentences exhibit a more balanced distribution between low- and high-frequency components, reflecting its thematic diversity. Additionally, the modulus values of the turnaround sentence are smaller, demonstrating stability. The key to enhancing the model's effectiveness lies in dynamically extracting frequency domain features of different types of sentences.

In Figure 1(B), we utilized DFT-Trans and BERT(Devlin et al., 2019) to extract the semantic features from the same sentence, followed by converting them into the frequency domain using FFT, and the modulus is obtained. It can be observed that the feature distribution extracted by DFT-Trans is more uneven, which is attributed to its capability to distinguish the importance of different frequency-domain features. In contrast, the feature distribution extracted by BERT is more uniform, indicating its limited effectiveness in processing frequency-domain information. When handling different types of sentences, the trainable Fourier operator can dynamically fuse these frequency-domain features, thereby extracting features(both global and local) more efficiently.

4 Experiment

To evaluate DFT-Trans, we design a series of controlled experiments on the GLUE(Wang et al., 2019) and LRA(Tay et al., 2021b) benchmarks, comparing it against various BERT-style encoder models and other advanced alternatives. The experiments include: (1) Performance on the GLUE benchmark with completed pretraining;(Result on GLUE with c-pretraining) (2) Performance on the GLUE benchmark without pretraining;(Result on GLUE w/o pretraining) (3) Performance on the GLUE benchmark with fixed time pretraining;(Result on GLUE with f-pretraining) (4) Performance on the LRA benchmark.(Result on LRA)

Dataset: (1) Pretraining datasets, including C4 (Raffel et al., 2020), Wikipedia, Bookcorpus (Zhu et al., 2015)[1]. (2)GLUE benchmark, using eight tasks. (3) LRA benchmark, using five tasks with input lengths ranging from 1K to 8K. More dataset details are in Appendix C.1.

Baseline: We use different baselines in various experiments for the GLUE and LRA benchmarks due to different applicability conditions of models. Additional details about the hyperparameters and baselines can be found in Appendix C.2 and C.3.

4.1 Main Experimental Results

4.1.1 Results on GLUE with c-pretraining

RTX 4090 is used in experiments with a batch size of 64, and the pretraining tasks are the MLM and NSP tasks proposed in BERT(Devlin et al., 2019). The results are shown in Table 1. DFT-Trans and FNet-Hybrid achieve 80.6% on the GLUE benchmark, outperforming other models. Compared to FNet and FNet-Hybrid, DFT-Trans performs significantly worse on the CoLA dataset. This is because CoLA is a few-shot dataset, whereas FNet and FNet-Hybrid utilize larger pretraining corpora, which enhances their performance on few-shot datasets. As the amount of training data for downstream tasks increases, our model is better able to bridge the gap caused by differences in pretraining. Compared to UltraSparseBERT, DFT-Trans shows inferior performance on the QQP dataset. This is because UltraSparseBERT is capable of distinguishing key information, which allows it to achieve good results in binary classification tasks. However, its performance is not as effective in other tasks. Compared to BERT_{base}, DFT-Trans performs poorly on the MRPC and MNLI datasets due to the larger scale of pretraining corpora used by BERT_{base}. To substantiate this conclusion, we implement BERT_{base} using the same pretraining corpus as DFT-Trans, and its performance is inferior to our model.

[1]The dataset is available at https://anonymous.4open.science/r/DFT-Trans-3FDD

Model	Params	RTE	SST-2	CoLA	STS-B	MRPC	QQP	MNLI	QNLI	Avg
BERT$_{base}$(Devlin et al., 2019)	108M	**66.4**	93.5	52.1	85.8	**88.9**	71.2	**84.6**	90.5	79.1
FNet(Lee-Thorp et al., 2022)	83M	63.0	**95.0**	69.0	79.0	83.0	83.0	72.0	80.0	77.8
FNet-Hybrid(Lee-Thorp et al., 2022)	88M	60.0	94.0	**76.0**	86.0	79.0	85.0	78.0	88.0	**80.6**
NarrowBert(Li et al., 2023)	105M	56.0	91.0	42.0	86.0	81.0	87.0	81.0	89.0	76.6
TNN(Qin et al., 2023)	126M	-	90.6	49.9	-	83.0	88.3	76.7	85.1	78.9
Mamba(Gu and Dao, 2024)	130M	57.0	91.6	56.7	86.8	75.2	87.8	82.5	87.9	78.7
SpikingBERT(Bal and Sengupta, 2024)	-	66.1	88.2	-	81.9	82.2	86.8	78.1	85.2	-
UltraSparseBERT(Belcak and Wattenhofer, 2024)	108M	56.7	92.3	48.4	86.3	**88.9**	88.0	82.9	**92.3**	79.9
BERT* (Devlin et al., 2019)	108M	63.9	90.4	48.9	83.7	84.3	84.9	77.4	84.3	77.2
DFT-Trans(our)	95M	64.6	91.2	58.1	**87.1**	84.3	**88.9**	81.3	88.3	**80.6**

* indicates that the pretraining of the model uses the same corpus as DFT-Trans. - indicates that we don't get the result from this task.

Table 1: Results on GLUE with c-pretraining, where the metrics on the MRPC and QQP tasks are means of accuracy and F1 scores, CoLA is the Mathews correlation coefficient, Spearman correlations for STS-B, and accuracy scores for other tasks. MNLI is reported by dev-matched only. Bolded results are the optimal, underlined results are sub-optimal, and the same applies to the subsequent ones.

Model	Params	RTE	SST-2	CoLA	STS-B	MRPC	QQP	MNLI	QNLI	Avg
BERT(Devlin et al., 2019)	108M	52.71	78.44	0.00	12.24	73.18	68.64	60.79	60.10	50.76
HyperMixing(Mai et al., 2023)	25M	**56.31**	78.78	0.00	15.79	74.99	72.65	56.38	**63.68**	52.32
MosaicBERT(Portes et al., 2023)	137M	54.15	82.34	9.86	20.08	74.35	74.26	64.45	61.94	55.19
Mamba(Gu and Dao, 2024)	130M	52.71	80.05	**18.22**	**20.17**	74.74	77.71	59.51	60.75	55.48
DFT-Trans(our)	95M	53.43	**83.03**	11.00	19.95	**75.03**	**82.35**	**68.62**	62.69	**57.01**

Table 2: Results of the unpretrained model on the GLUE benchmark.

4.1.2 Result on GLUE w/o pretraining

Considering the massive cost of pretraining, we explore the performance of models without pretraining. The experimental setup is similar to section 4.1.1. The results are shown in Table 2. DFT-Trans generally outperforms the compared models by about 1.5% point to 5% point. Although HyperMixing is four times smaller than ours, its performance is worse than our model on most tasks. MosaicBERT does not have particularly significant performance on most tasks. Mamba is a new architecture to replace transformers; its average performance is only second to our model.

4.1.3 Result on GLUE with f-pretraining

This experiment explores the performance of models at fixed pretraining time. The experimental setup is the same as above. The results are shown in Table 3. DFT-Trans achieves the best GLUE score of 80.6%. MosaicBERT has a shorter training time, and it achieves a score of 79.6%, but MosaicBERT's experimental environment is far superior to ours in terms of both the performance and number of GPUs. Cramming BERT achieves a good result with consumer GPUs.

4.1.4 Result on LRA

We adopt the same experimental configurations from the (Xiong et al., 2021). The experimental results are shown in Table 4. DFT-Trans outperforms the previous SOTA model, achieving the best scores on average score. Compared to TNN, our model performs worse on the Image and Text datasets. This is attributed to TNN's superior performance on image classification tasks. Additionally, the Text dataset has a smaller amount of data, and DFT-Trans underperforms relative to TNN on few-shot datasets.

After that, we use a byte-level text classification task (Text dataset) to evaluate the time and memory consumption of the models at lengths of 512, 1k, 2k, and 4k in the environment with a batch size of 8. All models are performed under RTX 4090. The results are shown in Table 9 in the Appendix C.4. DFT-Trans maintains the best performance under better time and memory consumption. FNet has the lowest time and memory consumption, but its performance on the LRA benchmark is significantly inferior to that of our model.

4.2 Ablation Experiment

We further perform ablation experiments on the GLUE benchmark to investigate whether the design of DFT-Trans is optimal. The experiments in this section primarily include: (1) Replacing all Attention operators with trainable Fourier operators.(Replacing Attention with Fourier) (2) Replacing all attention operators with MLPs.(Replacing Attention with MLPs) In addition, we conducted an exploration of the model architecture, focusing on two primary aspects: (1) the proportional relationship between the number of layers for trainable

Model	Params	Training time(hours)	Hardware	Batch Size	GLUE Score
BERT(Devlin et al., 2019)	108M	24	1 RTX A6000	64	52.2
BERT(Izsak et al., 2021)	108M	24	1 RTX A6000	64	72.9
MosaicBERT(Portes et al., 2023)	137M	1.13	8 A100-80	4096	79.6
Cramming BERT(Geiping and Goldstein, 2023)	145M	24	1 RTX A6000	64	78.6
DFT-Trans(our)	95M	24	1 RTX 4090	64	**80.6**

Table 3: Results on the GLUE benchmark after 24 hours of pretraining.

Model	Listops	Image	Pathfinder	Retrieval	Text	Avg
Transformer(Vaswani et al., 2017)	36.37	42.44	71.40	57.46	64.27	54.39
Linformer(Wang et al., 2020)	35.70	38.56	76.34	52.27	53.94	51.36
BigBird(Zaheer et al., 2020)	36.05	40.83	74.87	59.29	64.02	55.01
Performer(Choromanski et al., 2021)	18.01	42.77	77.50	53.82	65.40	51.41
Nystromformer(Xiong et al., 2021)	37.15	41.58	70.94	79.56	65.52	58.95
FNet(Lee-Thorp et al., 2022)	35.33	38.67	77.80	59.61	65.11	55.30
Fourier Transformers(He et al., 2023)	40.73	53.17	**83.43**	85.35	75.02	67.54
TNN(Qin et al., 2023)	47.33	**77.84**	73.89	**89.40**	**86.39**	74.97
IceFormer(Mao et al., 2024)	41.53	40.46	74.42	65.41	59.78	56.78
Mamba★(Gu and Dao, 2024)	38.02	69.82	69.26	72.14	82.98	66.44
Griffin★(De et al., 2024)	32.34	61.15	73.38	66.58	71.75	61.04
DFT-Trans(our)	**57.81**	68.78	82.61	88.63	81.85	**75.94**

Table 4: Results on the LRA benchmark. ★ indicates the results reported by (Alonso et al., 2024).

Fourier operators and attention operators. The results are presented in Appendix C.5. (2) The impact of layer numbers of Blending Token and Mixing Token in Spectrum within the trainable Fourier operators. The results are shown in Appendix C.6.

4.2.1 Replacing Attention with Fourier

DFT-Trans employs trainable Fourier operators to extract frequency-domain features and Attention operators to extract time-domain features. Experiments on the GLUE benchmark demonstrate that the combination of time-frequency features enhances performance on downstream tasks. The results are shown in Table 5. DFT-Trans effectively leverages both time-frequency features, outperforming models that utilize only frequency-domain features (DFT-DFT) or time-domain features (BERT).

4.2.2 Replacing Attention with MLPs

Considering that many image studies have used MLP-like networks (Tolstikhin et al., 2021b; Chen et al., 2022; Hou et al., 2023) in place of Attention and achieved better results, DFT-Trans is initially designed with the use of MLPs. The results are shown in Table 5 for DFT-MLP. It performs poorly on all the tasks.

5 Conclusion

In this work, we propose DFT-Trans, a novel BERT-style encoder model for NLP. The core of our model consists of trainable Fourier operators and Attention operators, which extract frequency-domain and time-domain features, respectively. By simply combining these features, we can capture more comprehensive information, including the theme of the text, long-range and short-range dependencies within sentences, and semantic features. Experiments on the GLUE and LRA benchmarks demonstrate that DFT-Trans outperforms other BERT-style encoder models and other state-of-the-art models. In future work, we will further consider the efficient integration of time-frequency domain features (e.g., how frequency domain information changes over time (wavelet transform)). We hope that by making BERT-style encoder models train faster, an amount of research in specific domains such as biomedicine(Lee et al., 2020; Gu et al., 2021), math(Shen et al., 2021), chemistry(Horawalavithana et al., 2022), and finance(Shah et al., 2022) will convert from fine-tuning general models to pretraining private models on specific data.

Model	M	N	Params	Avg
DFT-Trans(our)	DFT	Attention	95M	**80.6**
DFT-DFT	DFT	DFT	115M	75.3
DFT-MLP	DFT	MLPs	37M	72.4

Table 5: Results of Ablation Experiment. Three models share the same M layers but differ in N. DFT and Attention denote the trainable Fourier operator and Attention operator, respectively, as illustrated in Figure 2, while MLPs refers to multilayer perceptron networks.

6 Limitations

Although our model has the advantage of being more lightweight as well as performing well, the time complexity for each trainable Fourier operator layer is still quadratic complexity. Although the fast Fourier transform can reduce the time complexity to $O(L \cdot logL)$, the combination of Fourier transform with multi-head attention rolls the time complexity back to $O(L^2)$. Additionally, the matrix $\mathcal{W}_{\psi}^{\mathbb{1}}$ using in equation 3 in Blending Token in Spectrum increase the number of parameters and also the computational cost(GFLOPs).These are very unfavourable in long text scenarios. How to further improve the time complexity of the proposed model is still under study.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 268–284, Online. Association for Computational Linguistics.

Carmen Amo Alonso, Jerome Sieber, and Melanie N Zeilinger. 2024. State space models as foundation models: A control theoretic overview. arXiv preprint arXiv:2403.16899.

Malyaban Bal and Abhronil Sengupta. 2024. Spikingbert: Distilling bert to train spiking language models using implicit differentiation. Proceedings of the AAAI Conference on Artificial Intelligence, 38(10):10998–11006.

Peter Belcak and Roger Wattenhofer. 2024. UltraSparseBERT: 99% conditionally sparse language modelling. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 104–108, Bangkok, Thailand. Association for Computational Linguistics.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

S Chen, E Xie, C Ge, DY Liang, and P Luo. 2022. Cyclemlp: A mlp-like architecture for dense prediction. In International Conference on Learning Representation (ICLR), Oral. IEEE.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In International Conference on Learning Representations.

James W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex fourier series. Mathematics of Computation, 19:297–301.

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

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. 2024. Griffin: Mixing gated linear recurrences with local attention for efficient language models. arXiv preprint arXiv:2402.19427.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yihong Dong, Ge Li, Yongding Tao, Xue Jiang, Kechi Zhang, Jia Li, Jing Su, Jun Zhang, and Jingjing Xu. 2024. Fan: Fourier analysis networks. arXiv preprint arXiv:2410.02675.

Francesco Fusco, Damian Pascual, Peter Staar, and Diego Antognini. 2023. pNLP-mixer: an efficient all-MLP architecture for language. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track), pages 53–60, Toronto, Canada. Association for Computational Linguistics.

Jonas Geiping and Tom Goldstein. 2023. Cramming: Training a language model on a single gpu in one day. In International Conference on Machine Learning, pages 11117–11143. PMLR.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In First Conference on Language Modeling.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. ACM Trans. Comput. Healthcare, 3(1).

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. 2021. Adaptive fourier neural operators: Efficient token mixers for transformers. arXiv preprint arXiv:2111.13587.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. 2019. Star-transformer. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 1315–1325.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In International Conference on Learning Representations.

Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, and Zhouhan Lin. 2023. Fourier transformer: Fast long range modeling by removing sequence redundancy with fft operator. In Findings of the Association for Computational Linguistics: ACL 2023, pages 8954–8966.

Sameera Horawalavithana, Ellyn Ayton, Shivam Sharma, Scott Howland, Megha Subramanian, Scott Vasquez, Robin Cosbey, Maria Glenski, and Svitlana Volkova. 2022. Foundation models of scientific knowledge for chemistry: Opportunities, challenges and lessons learned. In Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models, pages 160–172, virtual+Dublin. Association for Computational Linguistics.

Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. 2023. Vision permutator: A permutable mlp-like architecture for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(1):1328–1334.

Peter Izsak, Moshe Berchansky, and Omer Levy. 2021. How to train BERT with an academic budget. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10644–10652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In International conference on machine learning, pages 5156–5165. PMLR.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In International Conference on Learning Representations.

An Lao, Qi Zhang, Chongyang Shi, Longbing Cao, Kun Yi, Liang Hu, and Duoqian Miao. 2024. Frequency spectrum is more effective for multimodal representation and fusion: A multimodal spectrum rumor detector. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 18426–18434.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics, 36(4):1234–1240.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. FNet: Mixing tokens with Fourier transforms. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems, 33:9459–9474.

Haoxin Li, Phillip Keung, Daniel Cheng, Jungo Kasai, and Noah A. Smith. 2023. NarrowBERT: Accelerating masked language model pretraining and inference. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 1723–1730, Toronto, Canada. Association for Computational Linguistics.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING. In International Conference on Learning Representations.

Zachary Chase Lipton. 2015. A critical review of recurrent neural networks for sequence learning. arXiv Preprint, CoRR, abs/1506.00019.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to mlps. Advances in neural information processing systems, 34:9204–9215.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Florian Mai, Arnaud Pannatier, Fabio Fehr, Haolin Chen, Francois Marelli, Francois Fleuret, and James Henderson. 2023. HyperMixer: An MLP-based low cost alternative to transformers. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15632–15654, Toronto, Canada. Association for Computational Linguistics.

Yuzhen Mao, Martin Ester, and Ke Li. 2024. Iceformer: Accelerated inference with long-sequence transformers on CPUs. In The Twelfth International Conference on Learning Representations.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In International conference on machine learning, pages 4055–4064. PMLR.

Badri Narayana Patro and Vijay Srinivas Agneeswaran. 2023. Scattering vision transformer: Spectral mixing matters. In Thirty-seventh Conference on Neural Information Processing Systems.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah A Smith, and Lingpeng Kong. 2021. Random feature attention. In 9th International Conference on Learning Representations, ICLR 2021.

Jacob Portes, Alexander R Trott, Sam Havens, DANIEL KING, Abhinav Venigalla, Moin Nadeem, Nikhil Sardana, Daya Khudia, and Jonathan Frankle. 2023. Mosaicbert: A bidirectional encoder optimized for fast pretraining. In Thirty-seventh Conference on Neural Information Processing Systems.

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In International Conference on Learning Representations.

Zhen Qin, Xiaodong Han, Weixuan Sun, Bowen He, Dong Li, Dongxu Li, Yuchao Dai, Lingpeng Kong, and Yiran Zhong. 2023. Toeplitz neural network for sequence modeling. In The Eleventh International Conference on Learning Representations.

Jiezhong Qiu, Hao Ma, Omer Levy, Wen-tau Yih, Sinong Wang, and Jie Tang. 2020. Blockwise self-attention for long document understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 2555–2565.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jie Zhou, and Jiwen Lu. 2023. Gfnet: Global filter networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(9):10960–10973.

Raj Shah, Kunal Chawla, Dheeraj Eidnani, Agam Shah, Wendi Du, Sudheer Chava, Natraj Raman, Charese Smiley, Jiaao Chen, and Diyi Yang. 2022. When FLUE meets FLANG: Benchmarks and large pretrained language model for financial domain. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2322–2335, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jia Tracy Shen, Michiharu Yamashita, Ethan Prihar, Neil Heffernan, Xintao Wu, Ben Graff, and Dongwon Lee. 2021. Mathbert: A pre-trained language model for general nlp tasks in mathematics education. In NeurIPS 2021 Math AI for Education Workshop.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun Woo. 2015. Convolutional lstm network: a machine learning approach for precipitation nowcasting. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, page 802–810, Cambridge, MA, USA. MIT Press.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2021a. Synthesizer: Rethinking self-attention for transformer models. In International conference on machine learning, pages 10183–10192. PMLR.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021b. Long range arena : A benchmark for efficient transformers. In International Conference on Learning Representations.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. ACM Comput. Surv., 55(6).

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. 2021a. Mlp-mixer: An all-mlp architecture for vision. In Advances in Neural Information Processing Systems, volume 34, pages 24261–24272. Curran Associates, Inc.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021b. Mlp-mixer: An all-mlp architecture for vision. Advances in neural information processing systems, 34:24261–24272.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding. In 7th International Conference on Learning Representations, ICLR 2019.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. arXiv preprint arXiv:2212.03533.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768.

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2022. A survey of joint intent detection and slot filling models in natural language understanding. ACM Comput. Surv., 55(8).

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. Preprint, arXiv:2309.07597.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 14138–14148.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. Advances in neural information processing systems, 32.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. Advances in neural information processing systems, 33:17283–17297.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414.

Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. 2021. An attention free transformer. arXiv preprint arXiv:2105.14103.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In Proceedings of the IEEE international conference on computer vision, pages 19–27.

A Complex Multiplication

For two complex values $\mathcal{Z}_1 = \mathbf{r}_1 + \mathbf{u}_1 j$ and $\mathcal{Z}_2 = \mathbf{r}_2 + \mathbf{u}_2 j$, $\mathbf{r}_i$ is the real part of the complex numbers and $\mathbf{u}_i$ is the imaginary part of the complex numbers. We multiply the two complex numbers as follows:

$$\begin{aligned} \mathcal{Z}_1 \times \mathcal{Z}_2 &= (\mathbf{r}_1 + \mathbf{u}_1 j)(\mathbf{r}_2 + \mathbf{u}_2 j) \\ &= \mathbf{r}_1\mathbf{r}_2 + \mathbf{r}_1\mathbf{u}_2 j + \mathbf{u}_1\mathbf{r}_2 j - \mathbf{u}_1\mathbf{u}_2 \\ &= (\mathbf{r}_1\mathbf{r}_2 - \mathbf{u}_1\mathbf{u}_2) + (\mathbf{r}_1\mathbf{u}_2 + \mathbf{u}_1\mathbf{r}_2) j \end{aligned} \tag{10}$$

note that $j^2 = -1$.

B Proof of the Convolution Theorem

The convolution theorem states that the Fourier transform of the convolution of the function is equal to the product of the Fourier transform of the function. Suppose we have two functions $f_1(t)$ and $f_2(t)$ in the time domain whose corresponding frequency domain representations after the Fourier transformer are $\mathcal{F}_1(w)$ and $\mathcal{F}_2(w)$, $\mathcal{F}(\cdot)$ denotes the Fourier transform, and the formula is expressed as follows:

$$\mathcal{F}(f_1(t) * f_2(t)) = \mathcal{F}_1(w) \times \mathcal{F}_2(w) \tag{11}$$

The proof is as follows:

According to the definition of convolution there is:

$$f_1(t) * f_2(t) = \int_{-\infty}^{+\infty} f_1(\tau) f_2(t-\tau) d\tau \tag{12}$$

Expand the left-hand side of equation (11) according to the definition of the Fourier transform and the definition of convolution:

$$\begin{aligned} &\mathcal{F}(f_1(t) * f_2(t)) \\ &= \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} f_1(\tau) f_2(t-\tau) d\tau\right) e^{-j2\pi wt} dt \\ &= \int_{-\infty}^{+\infty} f_1(\tau) \left(\int_{-\infty}^{+\infty} f_2(t-\tau) e^{-j2\pi wt} dt\right) d\tau \end{aligned} \tag{13}$$

We suppose $u = t - \tau$, $du = dt$, $t = u + \tau$, and get:

$$\int_{-\infty}^{+\infty} f_1(\tau)(\int_{-\infty}^{+\infty} f_2(u)e^{-j2\pi w(u+\tau)}du)d\tau$$
$$= \int_{-\infty}^{+\infty} f_1(\tau)e^{-j2\pi w\tau}d\tau \int_{-\infty}^{+\infty} f_2(u)e^{-j2\pi wu}du$$
$$= \mathcal{F}_1(w) \times \mathcal{F}_2(w) \tag{14}$$

Certification completed. It can be seen that all our operations in the frequency domain can be equated to a global convolution in the time domain, allowing us to have a global view to extract input features.

C Experimental details

C.1 Datasets

C4: It is a dataset created based on Common Crawl, which grabs about 156 billion tokens from 365 million domains, making it one of the largest available corpora. We can get this dataset at https://huggingface.co/datasets/allenai/c4.

Wikipedia and Bookcorpus: Its earliest use for Bert's pretraining, Wikipedia has collected information on Wikipedia to organize into a large-scale corpus, and the dataset can be accessed at https://huggingface.co/datasets/wikipedia. Bookcorpus is a collection of 11,000 unpublished books organized on the Internet, containing about 985 million words, including a wide range of book types, and the dataset can be accessed at https://huggingface.co/datasets/bookcorpus.

C4BookC: Approximately 110 million sentences. Consists of part of the C4 dataset and the BookCorpus dataset used to pretrain DFT-Trans. This dataset is available at https://anonymous.4open.science/r/DFT-Trans-3FDD/.

SQuAD: This is a reading comprehension dataset. The dataset contains 100,000 (question, original text, answer) triples, with the original text from 536 Wikipedia articles.

We used eight datasets from the GLUE(Wang et al., 2019) benchmark on natural language inference, textual entailment, sentiment analysis, and semantic similarity, which is designed to measure the ability of models in natural language understanding. GLUE benchmark can be accessed at https://gluebenchmark.com/. We utilized five datasets from the LRA benchmark, encompassing mathematical computation, text classification, document retrieval, image classification, and long-range spatial dependency. These datasets are specifically designed to evaluate a model's capability in handling long-text modeling. The LRA benchmark can be accessed at https://paperswithcode.com/dataset/lra.

CoLA: This is a single-sentence binary classification task, derived from books and journals in linguistics, where each sentence needs to be judged as grammatical or not. Number of samples: 8551 in the training set and 1043 in the development set.

SST-2: This is a single sentence binary classification task. The sentiment of a given sentence needs to be recognized and categorized into positive and negative. Number of samples: training set 67350, development set 873.

MRPC: This is a multiple sentence binary classification task. A corpus of sentence pairs is automatically extracted from online news sources, and we need to determine whether these sentence pairs are semantically equivalent. Number of samples: 3668 in the training set and 408 in the development set.

STS-B: This is a multiple sentence regression task. Sentence pairs are extracted from news headlines, video captions, image captions, and natural language inference data, and the model is needed to predict the similarity of these sentences. The similarity score is 0-5. number of samples: 5749 in the training set and 1379 in the development set.

QQP: This is a multi-sentence binary classification task. Derived from a collection of question pairs in the community Q &A website Quora, it is necessary to determine whether a pair of questions is semantically equivalent or not. This dataset has an uneven distribution of positive and negative samples, with 63% negative samples and 37% positive samples. Number of samples: 363,870 in the training set and 40,431 in the development set.

MNLI: This is a multi-sentence multi-categorization task. Given premise and hypothesis statements, we needs to predict whether the premise statement contains the hypothesis, or contradicts the hypothesis, or is neutral to the hypothesis. Number of samples: 392,702 for the training set, and the development set is divided into dev-matched 9815 and dev-mismatched 9832. matched means that the data sources of the training set and the development set are the same, and mismatched means that the sources of the training set and the development set are not the same.

QNLI: This is a multi-sentence binary classification task. It is derived from the SQuAD(Rajpurkar et al., 2016) dataset, where given a question and paragraph text, it is necessary to determine whether the answer to the question is embedded in the paragraph text. Number of samples: 104743 in the training set and 5463 in the development set.

RTE: This is a multi-sentence binary classification task. Given a sentence pair, determine whether sentence 1 and sentence 2 entail each other. Number of samples: 2491 in the training set and 277 in the development set.

ListOps: Adataset of math expressions that asks the model to calculate the output value of a math expression with sequence lengths up to 2K.

Text: Abyte-level text classification task, with a fixed sequence length 4K which requires the model to deal with compositionality.

Retrieval: A byte-level document retrieval task with a maximum length of 8K which test the model's ability to compress long sequences.

Image: Animageclassification task of which requires the model to learn the 2D spatial relations between input pixels by sequentially reading the pixels. The sequence length is fixed to 1K.

Pathfinder: An synthetic image classification task with a fixed input length of 1K which requires the model to capture long range spatial dependencies.

C.2 Finetuning Hyperparameters and Pretrain Detail

The hyperparameters we used in the fine-tuning process are displayed in Table 6. The maximum sequence length of 256 is used for all the datasets. We found that these can help the model to reach convergence very quickly. For large datasets, a small learning rate is needed, while for small datasets, a large learning rate is needed. This is the principle by which we choose the learning rate. Large datasets with large amounts of data need a small learning rate to converge slowly, while small datasets with small amounts of data using a large learning rate can help the model to learn better. The parameters of models in the pretraining phase are given in the Table 8. Since the pretraining period is relatively long, we set the warmup step to improve the effectiveness of the pretraining. Different training corpus is used for different models to ensure the performance of the models.

On the SQuAD(Rajpurkar et al., 2016) dataset, we compare only DFT-Trans and BERT(Devlin et al., 2019), and the results are displayed in Table 7. It can be seen that our model is superior to Bert in both metrics.

Task	lr	beta	epsilon	wd	epoch
RTE	4e-5	[0.9, 0.98]	1e-12	0.01	30
CoLA	4e-5	[0.9, 0.98]	1e-12	0.01	30
SST-2	3e-5	[0.9, 0.98]	1e-12	0.01	10
STS-B	4e-5	[0.9, 0.98]	1e-12	0.01	30
MRPC	4e-5	[0.9, 0.98]	1e-12	0.01	30
QQP	3e-5	[0.9, 0.98]	1e-12	0.01	5
MNLI	3e-5	[0.9, 0.98]	1e-12	0.01	5
QNLI	3e-5	[0.9, 0.98]	1e-12	0.01	10

Table 6: Hyperparameters used for finetuning. lr represents learning rate and wd represents weight decay.

Model	Params	lr	EM	F1
BERT(Devlin et al., 2019)	108M	3e-5	65.87	74.88
DFT-Trans(our)	95M	3e-5	**66.91**	**77.51**

Table 7: The performance of DFT-Trans and Bert on SQuAD(Rajpurkar et al., 2016) dataset and training details. lr represents learning rate.

C.3 Baseline

BERT(Devlin et al., 2019): a traditional Bert-style model that has referenced the design of the encoder in transformers and works well in most NLP tasks after pretraining.

BERT(Izsak et al., 2021): a proposal by Microsoft to implement a functionally similar model to Bert on a low budget, making Bert less expensive to train.

FNet(Lee-Thorp et al., 2022): aims to explore whether Attention can be replaced by implementing the model using the Fourier transform without parameterization, but this way leads to performance loss during the Fourier transform.

NarrowBert(Li et al., 2023): aims to speed up training and allow the model to focus more on masked Token.

HyperMixing(Mai et al., 2023): uses MLP-like architecture instead of Attention and simulates the effect of Attention to achieve good results without pretraining. The disadvantage is that the model performance cannot be improved by pretraining.

MosaicBERT(Portes et al., 2023): applying current techniques to Bert to shorten model training.

Cramming BERT(Geiping and Goldstein, 2023): aims to further reduce the training cost of Bert-style models and explore the performance of Bert-style models that can approximate the original Bert as much as possible with one day of pretraining.

TNN(Qin et al., 2023): utilizing Toeplitz matrices to capture the relationships between each token pair, thereby replacing relative position encoding. Due to the $O(n \log n)$ complexity of Toeplitz matrices, it can reduce the computational complexity of the model and achieve commendable results in

Model	Params	Learning rate	Corpus	warmup step
FNet(Lee-Thorp et al., 2022)	83M	1e-4	C4	5000
BERT(Devlin et al., 2019)	108M	1e-4	C4BookC[1]	5000
BERT(Izsak et al., 2021)	108M	1e-4	Wiki+BookCorpus	5000
NarrowBert(Li et al., 2023)	105M	1e-4	Wikipedia	5000
MosaicBERT(Portes et al., 2023)	137M	1e-4	C4	5000
Cramming BERT(Geiping and Goldstein, 2023)	145M	1e-4	Wiki+BookCorpus	5000
DFT-Trans(our)	95M	1e-4	C4BookC[1]	10000

Table 8: Pretrain detail for different models

long-sequence modeling tasks.

Mamba(Gu and Dao, 2024): selective processing of information, coupled with hardware-aware acceleration algorithms at the hardware level, combined with a simpler SSM architecture, forms Mamba. It addresses the quadratic complexity issue of Transformer.

C.4 Analysis of model training speed and memory usage

To investigate the training speed of DFT-Trans compared to other state-of-the-art (SOTA) models on long-text tasks, we select the byte-level text classification task(Text dataset) from the LRA benchmark for evaluation. This is because the input text length in the Text datasets task is 4k, which aligns with the maximum length we aim to test. All models are trained under 512, 1k, 2k, and 4k input lengths. The results are presented in Table 9. For fairness, we maintain the same configuration settings in Nyströmformer. The basic Transformer model exhibits the poorest performance in this test. Other improved models, such as Linformer, Performer, and Nyströmformer, demonstrate relatively better performance. FNet, due to its abandonment of the attention mechanism, significantly improves training speed but shows inferior task accuracy compared to other models. DFT-Trans demonstrates commendable performance across all tasks while maintaining a competitive training speed.

C.5 The proportion of Layers

We conduct experiments without pre-training to investigate the layer allocation between trainable Fourier operators and Attention operators. The results are shown in Figure 4. The layer allocation of (2, 6) achieves the highest average performance on the GLUE benchmark and performs well on all tasks except MRPC. However, the layer allocation of (3, 6) performs the worst, showing poor results on nearly all tasks. The layer allocations of (1, 4) and (3, 4) show strong performance on few-shot datasets.

C.6 Exploring the Layers of Mixing Token in Spectrum

In equation 5, we propose a hyperparameter $\mathbb{L}$ to denote the number of layers of the Mixing Token Operator in the Spectrum, as illustrated in Figure 2. A series of experiments without pretraining are conducted to explore the reasonableness of the $\mathbb{L}$ selection. The results are presented in Figure 5. It can be seen from Figure 5(A) that the average performance of the model is best for $\mathbb{L} = 3$. The difference in performance for other value of L is not particularly large. In Figure 5(B), the best results are achieved at $\mathbb{L} = 3$ on both the SST-2 and QNLI tasks. Other values except $\mathbb{L} = 1$ achieve good results on specific tasks.

Model	Steps per second↑				Peak Memory Usage↓			
	512	1K	2K	4K	521	1K	2K	4K
Transformer(Vaswani et al., 2017)	13	10	3	1	0.8	1.3	3.3	11.4
Linformer(Wang et al., 2020)	15	15	13	**10**	1.5	2.5	4.5	8.4
Performer(Choromanski et al., 2021)	16	16	12	6	1.7	3.0	5.9	10.4
Nystromformer(Xiong et al., 2021)	10	8	7	6	1.2	1.6	2.3	3.8
FNet(Lee-Thorp et al., 2022)	**27**	**24**	**14**	8	**0.5**	**0.7**	**0.9**	**1.3**
Fourier Transformers(He et al., 2023)	10	11	9	5	0.9	1.9	5.9	21.2
DFT-Trans(ours)	16	15	12	7	0.8	1.1	1.6	2.8

Table 9: The speed and memory consumption on LRA benchmark over Text task

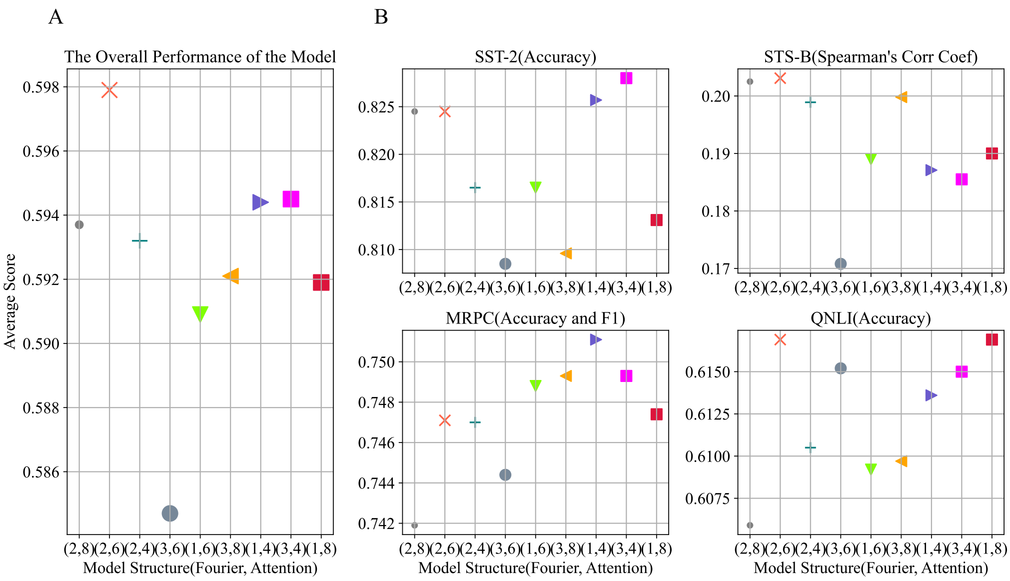

Figure 4: (A) represents the average performance of the model on GLUE with a different number of layers; here, we just used four tasks, namely SST-2, STS-B, MRPC, and QNLI. (B) is the performance of the models on a single task.

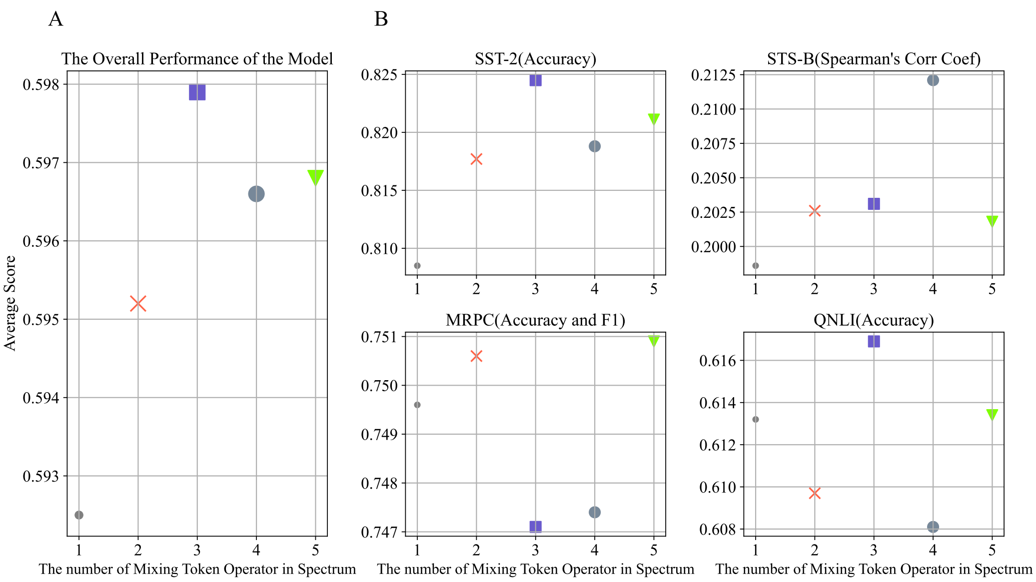

Figure 5: (A) represents the average performance of the model on GLUE with different number of Mixing Token Operator in Spectrum, here we just uesd four tasks, namely SST-2, STS-B, MRPC, QNLI. (B) is the performance of the models on single task.