Wavelet and Optical Features Sparkling NLP

Anonymous EMNLP submission

Abstract

Computational resources are vital in natural language processing (NLP) development. Since the physical limit of transistors is approaching a saturation point due to the outspace of Moore's Law and Dennard scaling, we look for alternative computing power from optical devices. As an initial step in this research direction, we facilitate feature extraction using optical computing and integrate optical extracted features to enhance NLP baselines on conventional electronic GPUs. Unlike another one of a kind of features extracted from Transformer, such as lexical embeddings, we extend the feature space beyond traditional embeddings using Wavelet functions that can run on optical toolkits. These extracted features, alongside the original input text, provide additional information that enhances model performance in NLP tasks. We employ two different feature extraction methods: a direct approach involving Wavelet or FFT transformations, and a novel method employing optical computing for NLP feature extraction. Our evaluation encompasses fice GLUE tasks - CoLA, SST-2, STSB, MRPC, and RTE - and reveals a notable improvement of up to $+2.8\%$ in classification accuracy.

1 Introduction

The extensive use of computational resources is a major challenge in defining the performance of neural network-based models. The ever-increasing model size and training data ask for an extremely high GPU supply. As one way to look for alternative computing power, we will introduce Wavelet-based methods that are compatible with and can run on optical devices to enhance NLP performance.

Recently, optical computing has been receiving rising attention on various tasks such as image classification (Mirek et al., 2021; Lupo and Massar, 2021). They offer energy efficiency (Wang et al., 2022) and time efficiency of more than ten trillion operations per second (Xu et al., 2021). Photonic neuromorphic computing systems demonstrate that photonic machine learning can realize NLP system on sentiment analysis tasks (Valensise et al., 2022) but the performance does not reach (20% less) the state-of-the-art result on GPUs, for example, the embedding is based on TF-ITF instead of Transformer. We aim to combine the advantage of optical device and electric device for a better accuracy, and furthermore, our model design can be applied on any neural network architecture.

We propose using Wavelet transformation features as additional inputs to learn NLP models. Wavelet features have been commonly used for feature extraction in images (Kingsbury and Magarey, 1998) and dimension reduction (Coifman et al., 1994; Qureshi et al., 2008). However, previous research has mainly been focused on using Wavelet features for statistical methods (Mahajan et al., 2015; Kristomo et al., 2016a); little research has been carried out to use Wavelet for NLP tasks so far, and there is much potential to explore. We use simulated optical computing to extract Wavelet transformation-based features and add them into the electronic computing (CPU/GPU) hosted neural networks to enhance the model performance. We combine these features with the original input during the model training process and interpret these features with linguistic meanings.

Specifically, we use the BERT$_{\text{BASE}}$ model to represent the input text. This representation is then passed to our non-linear optical simulation of the optical device to extract the optical features. After that these features are then concatenated with the original sentence embeddings to finetune the model. We use the BERT model as the baseline and show how these extracted optical or Wavelet features help improve the performance of the baseline model on the GLUE Benchmark (Wang et al., 2018), which is a standard benchmark containing multiple NLP tasks and has been used to show performance for various large language models, e.g.

BERT (Devlin et al., 2018), GPT (Brown et al., 2020) etc. Our major contributions include:

1. We propose two novel methods, i.e., optical-simulation-based and Wavelet-transformation-based feature extraction from sentence embeddings to improve NLP accuracy;
2. To our knowledge, we are the first to observe that features from non-linear optical device can improve the performance for NLP tasks;
3. We apply our Wavelet method to the BERT language model on the GLUE benchmark and achieve an improvement of up to +2.8% in BLEU over the baseline method.

2 Method

This section describes the Wavelet feature extraction methods and how we incorporate them into the neural network models. Firstly, Section 2.1 describes our method to extract the sentence embedding using a pre-trained transformer-based model given an input sequence. Section 2.2 describes the two methods we use to extract the features from the sentence embedding. (i) 'Direct Method' directly applies the Wavelet/Fast Fourier (FFT) transformation on the embedding matrix, and (ii) 'Optical Method' for feature extraction uses non-linear optical computing to extract the features. Section 2.3 describes how we combine the extracted features with the original input. These combined features are then used to fine-tune and improve the model's performance.

Algorithm 1 gives the details of our method including feature extraction and model fine-tuning. For a given input sequence of n tokens $w = \{w_1, \cdots, w_n\}$, we propose using optical feature extraction method to get m output features $f = \{f_1, \cdots, f_m\}$. Then, instead of fine-tuning a neural model $\mathbb{M}(w)$, we fine-tune on $\mathbb{M}(w \oplus f)$. Here M is a pre-trained transformer model and $\oplus$ is the concatenation of original tokens with the extracted features. Now, we will detail step by step.

2.1 Extracting Sentence Embeddings

Each input sentence is converted to a sentence-level embedding matrix using the pre-trained BERT (Devlin et al., 2018) model. For each training, validation, and test sample of the specific NLP task, we tokenize them and put them as input to the pre-trained language model and get the final layer's sentence embedding matrix as output. The final output is a $len \times dim$ matrix where len is the length of the tokenized input sequence[1] and dim is the dimension of the Transformer language model. Note that for embedding matrix extraction, we use the pre-trained language model without any fine-tuning. Additionally, we freeze the model layers to ensure the models remain the same throughout the extraction phase. We then pass each data sample to get the final layer's output matrix. The model weights are only updated during the fine-tuning phase.

```
Algorithm 1 Model Fine-tuning
Input: input data ($w = \{w_1, \cdots, w_n\}$), pre-trained model ($\mathbb{M}$)
Output: fine-tuned model ($\mathbb{M}'$)
$w'$ : tokenize the data $w$ using the tokenizer for $\mathbb{M}$
$W$ : last layer's output of $\mathbb{M}(w')$
$F$ : feature extraction using $W$
$f$ : column-wise average of $F$
$X$ : concatenate $w$ and $f$
$\mathbb{M}'$ : fine-tune $\mathbb{M}$ using $X$
return $\mathbb{M}'$
```

2.2 Feature Extraction

Once we extract the sentence embeddings, we use two different methods to extract the features: a 'Direct Method' and an 'Optical Method'.

2.2.1 Direct Method

In our first method, 'Direct Method' we directly apply the Haar Wavelet transformation or the FFT transformation to get the Wavelet or FFT features respectively. For the Wavelet features, we use the high-pass filter output (HH), however other outputs give similar results. The algorithm in Appendix E describes the 'Direct Method' for feature extraction. We then perform a column-wise average of the transformation output to get the feature vector.

2.2.2 Optical Method

Our second method, 'Optical Method' is a simulation of the non-linear optics method to extract the features. For each data sample, we extract its sentence embedding matrix as input. We use a simulation of the original method to extract the final features by utilizing the 2-D partial Fourier transform. We also experiment with the optical computation hardware which the simulation code is built upon. The steps for feature extraction are the same in both hardware and optical simulation, however, due to noise, humidity, and temperature changes, the hardware can give slightly different results.

[1] The length of the sentence includes the special tokens, i.e. [CLS] and [SEP].

Algorithm 2 Optical feature extraction

```
Input: sentence embedding matrix ($W$), azimuthal index range ($l_1, l_2$), radial index range ($p_1, p_2$)
Output: optical features ($f$)
$N : 1200$
$W'$ : resize $W$ to $N \times N$ using bicubic interpolation, then normalize values to be between 0 and $2\pi$
$b_1, b_2$ : Gaussian beams
$beam_{se} : b_1 e^{W'}$
$signal : PPLN(SLM(beam_{se}))$
$LG : generate\_LG\_modes(l_1, l_2, p_1, p_2)$
for $lg_i \in LG$ do
    $beam_{lg} : b_2 e^{lg_i}$
    $pump : PPLN(SLM(beam_{lg}))$
    $F : signal \cdot A \cdot \sin(B \cdot pump \cdot C)$
    $f_i$ : normalized sum of absolute values of $F$
end for
return $f$
```

The optical simulation algorithm takes three inputs: the sentence embedding matrix W, and the two index ranges used for generating the set of Laguerre-Gaussian (LG) modes (azimuthal l and radial p index ranges). We define this set of LG modes as LG. The phase patterns of the sentence embedding matrix and a set of LG modes are uploaded to the simulation separately ($beam_{se}$ and $beam_{lg}$) to two spatial light modulators (SLM) and then propagated through the Magnesium-doped Periodic Poled Lithium Niobate crystal ($PPLN$).[2] These two beams are then merged, and the sum frequency values are outputted by the simulation. These sum-frequency values are computed using the slowly varying envelope approximation (SVAE). A, B, and C are hardware parameters for the sum frequency generation and are treated as constants for our experiments. Further details on this algorithm along with the parameter settings is mentioned in Appendix A and Appendix B respectively.

2.3 Combining Extracted Features

There are various ways to combine additional features to improve model performance (Feng et al., 2021; Wei and Zou, 2019; Zhang et al., 2017; Sennrich et al., 2015). We concatenated the sentence's extracted features with the original input sequence. The model is then fine-tuned on this concatenated sentence-level embedding matrix. For tasks that have more than one input sentence, e.g. RTE, etc, we concatenate the feature embeddings with both sentences separately and then give them as input to the model for fine-tuning. For example, if we have an input sequence $w_1, w_2, \cdots$, and w_n, then the combined sequence becomes $w_1, w_2, \cdots, w_n, f_1, f_2, \cdots$, and f_m. Here w_i are the tokens in the input sequence and f_i are the extracted features.

3 Experiments

3.1 Tasks and Settings

Datasets: We applied our method on the GLUE Benchmark (Wang et al., 2018), which consists of nine text classification tasks. We apply our methods to total six of the nine tasks including, CoLA (grammatical acceptability), SST-2(sentiment classification), STSB(sentence similarity), RTE(natural language inference), and MRPC(paraphrase task). The sentence-level statistics for each of the tasks are mentioned in Table 6.

Tools: We use Pytorch library (Paszke et al., 2019) for all the experiments along with the huggingface (Wolf et al., 2019) toolkit for the preprocessing and fine-tuning of the models. We also used the PyWavelet (Lee et al., 2019) and the scipy (Virtanen et al., 2020) packages to perform the Wavelet and the FFT transformations respectively. All the experiments are carried out on a single NVIDIA Tesla V100 GPU.

Baseline: For each of the five GLUE tasks, we carry out two sets of fine-tuning. We initially fine-tuned the BERT$_{\text{BASE}}$ model using the data for the specific GLUE task to create a baseline BERT model. Then we further fine-tune using the combined original text and the extracted features.

Parameter settings: For each experiment, we pad the sentences to the maximum length of 512 or truncate them if greater than 512. Each training is run for 3 epochs (Devlin et al., 2018) with a learning rate of $2e^{-5}$. We use Adam optimizer with betas set as 0.900 and 0.999.

Evaluation criteria: For the evaluation, we report Matthew's correlation for the CoLA task, Pearson's correlation for STSB, and accuracy scores for all the other tasks.

3.2 Experimental Results

Direct Method: Table 1 shows the results for all five GLUE tasks using the BERT$_{\text{BASE}}$ cased model. We show how the model's performance changes when we use the additional Wavelet or FFT features to fine-tune the model. We observe that fine-tuning the model on the combined original input sentences and the extracted Wavelet/FFT features improves

[2] We simulate SLM and $PPLN$ using a 2D partial Fourier Transform over the input.

Task	B	B+Wavelet	B+FFT
CoLA	51.8	**54.6 (+2.8)**	52.2 (+0.4)
SST-2	93.4	93.5 (+0.1)	**94.0 (+0.6)**
STSB	84.3	84.9 (+0.6)	**85.9 (+1.6)**
RTE	65.8	**67.9 (+2.1)**	67.8 (+2.0)
MRPC	82.3	83.5 (+1.2)	**84.2 (+1.9)**

Table 1: Results of GLUE Benchmark Tasks for the direct method. 'B' is the baseline model when fine-tuned on only the original input. 'B+Wavelet' and 'B+FFT' are the result of fine-tuning the Baseline model with additional Wavelet and FFT features respectively.

the performance for all five tasks. We get an average improvement of 1.3% points with a maximum improvement of 2.8% points above the baseline for the CoLA task.

Optical Method: Table 3 compares the results for the CoLA and RTE tasks using different feature extraction methods. We can see that optical features ('B+Optical Method') give similar improvement as compared to the FFT features ('B+FFT') for the CoLA task and give a higher improvement in comparison with both the Direct Methods for the RTE task. On average, the optical features give a similar improvement compared to Wavelet features (approx. 2.4% points above the baseline) and higher than the FFT features.

Table 2 compares the results of Optical Simulation and Optical Hardware for the WNLI dataset. We can observe that Optical hardware got a +4.2% points improvement while Optical simulation got a +15.8% improvement above the baseline experiment.

Task	B	B+Optical Hardware	B+Optical Simulation
WNLI	46.5	50.7 (+4.2)	**62.3 (+15.8)**

Table 2: Results comparison of WNLI task for optical simulation and optical hardware.

4 Related Work

There is extensive work on using alternative representations of text input to improve model performance across NLP tasks (García-Martínez et al., 2016; Khan et al., 2020). Researchers have used these representations as additional features which include statistical (Jing et al., 2002), attention-based features (Tang et al., 2022), and phonetic features (Liu et al., 2018a). We add to this research area by utilizing Wavelet transformations and optical methods to create additional features from the original text to improve model performance.

Wavelet transformations been used for various tasks like data mining (Li et al., 2002), data compression (Coifman et al., 1994), and computer vision (Li et al., 2020; Liu et al., 2019, 2018b; Alemohammad et al., 2017; Wang et al., 1995; Qureshi et al., 2008). Within NLP, Wavelet has been used within model architecture (Aggarwal, 2002), for dimensionality reduction (Xexéo et al., 2008), and to reduce word embedding computational cost (Dahab et al., 2021). We focus on Wavelet feature extraction, which has been used previously for specific NLP tasks like language identification (Al-Dubaee et al., 2010) as well as other ML/neural-network based tasks (Mahajan et al., 2015; Huang and Fang, 2022; Kristomo et al., 2016b; Hidayat et al., 2015). We focus on evaluating Wavelet feature extraction's general usage within neural network-based NLP tasks by using multiple standardized benchmarks.

Task	B	Direct Method		B+Optical Method
		B+Wavelet	B+FFT	
CoLA	51.8	**54.6 (+2.8)**	52.2 (+0.4)	52.3 (+0.5)
RTE	65.8	67.9 (+2.1)	67.8 (+2.0)	**68.3 (+2.5)**

Table 3: Performance comparison of GLUE tasks for different feature extraction methods. 'B' is the baseline model. 'B+Wavelet', 'B+FFT', and 'B+Optical Method' are the result of fine-tuning the Baseline model with Wavelet, FFT, and optical features respectively.

Optical neural networks (ONNs) have been utilized as an alternative to artificial neural networks for computational speed and efficiency (Liu et al., 2021). ONNs have been used to speed up operations in neural networks such as CNNs (Mehrabian et al., 2018). To the best of our knowledge, we are the first to apply non-linear optics to extract features from text using a standardized benchmark.

5 Conclusion

As optical technology becomes widely used, we will see the transfer of the current models from traditional to optical hardware due to its time and energy efficiency. This work is an initial step in this direction where we propose a novel approach to using Optical Computing to extract features from a transformer-based language model. We are the first to use non-linear optical computation (Optical Method) as a feature extraction method for NLP tasks. For comparison, we use Wavelet and FFT transformation (Direct Method) to extract these features. We combine these extracted features with the original input to improve performance on five text classification tasks. We also analyze how the Direct and Optical methods give a similar improvement in performance.

6 Limitations

Current optical computation equipment pieces are slow to convert electrical signals to optical signals and vise versa which shows a much higher extraction time. However, we expect usage of state-of-the-art hardware will bridge this generation gap.

References

Charu C Aggarwal. 2002. On effective classification of strings with wavelets. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and Data Mining*, pages 163–172.

Shawki A Al-Dubaee, Nesar Ahmad, Jan Martinovic, and Vaclav Snasel. 2010. Language identification using wavelet transform and artificial neural network. In *2010 International Conference on Computational Aspects of Social Networks*, pages 515–520.

Milad Alemohammad, Jasper R Stroud, Bryan T Bosworth, and Mark A Foster. 2017. High-speed all-optical haar wavelet transform for real-time image compression. *Optics express*, 25(9):9802–9811.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ronald R Coifman, Yves Meyer, Steven Quake, and M Victor Wickerhauser. 1994. Signal processing and compression with wavelet packets. In *Wavelets and their applications*, pages 363–379. Springer.

Mohamed Yehia Dahab, Omar A Batar, Muazzam Siddiqui, and Reda Mohamed Salama Khalifa. 2021. Word embeddings based on spectral analysis: A novel approach. *International Journal of Information Technology and Language Studies*, 5(1).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *arXiv preprint arXiv:1609.04621*.

Risanuri Hidayat, Priyatmadi, and Welly Ikawijaya. 2015. Wavelet based feature extraction for the vowel sound. In *2015 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–4.

Hao Huang and Yi Fang. 2022. Adaptive wavelet transformer network for 3d shape representation learning. In *International Conference on Learning Representations*.

Li-Ping Jing, Hou-Kuan Huang, and Hong-Bo Shi. 2002. Improved feature selection approach tfidf in text mining. In *Proceedings. International Conference on Machine Learning and Cybernetics*, volume 2, pages 944–946. IEEE.

Abdul Rafae Khan, Jia Xu, and Weiwei Sun. 2020. Coding textual inputs boosts the accuracy of neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1350–1360.

Nick Kingsbury and Julian Magarey. 1998. Wavelet transforms in image processing. In *Signal analysis and prediction*, pages 27–46. Springer.

Domy Kristomo, Risanuri Hidayat, and Indah Soesanti. 2016a. Feature extraction and classification of the indonesian syllables using discrete wavelet transform and statistical features. In *2016 2nd International Conference on Science and Technology-Computer (ICST)*, pages 88–92. IEEE.

Domy Kristomo, Risanuri Hidayat, and Indah Soesanti. 2016b. Feature extraction and classification of the indonesian syllables using discrete wavelet transform and statistical features. In *2016 2nd International Conference on Science and Technology-Computer (ICST)*, pages 88–92.

Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. 2019. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237.

Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. 2020. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7245–7254.

Tao Li, Qi Li, Shenghuo Zhu, and Mitsunori Ogihara. 2002. A survey on wavelet applications in data mining. *ACM SIGKDD Explorations Newsletter*, 4(2):49–68.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2018a. Robust neural machine translation with joint textual and phonetic embedding. *arXiv preprint arXiv:1810.06729*.

Jia Liu, Qiuhao Wu, Xiubao Sui, Qian Chen, Guohua Gu, Liping Wang, and Shengcai Li. 2021. Research progress in optical neural networks: theory, applications and developments. *PhotoniX*, 2.

Pengju Liu, Hongzhi Zhang, Wei Lian, and Wangmeng Zuo. 2019. Multi-level wavelet convolutional neural networks. *IEEE Access*, 7:74973–74985.

Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. 2018b. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782.

Alessandro Lupo and Serge Massar. 2021. Parallel extreme learning machines based on frequency multiplexing. *Applied Sciences*, 12(1):214.

Anuj Mahajan, Sharmistha Jat, and Shourya Roy. 2015. Feature selection for short text classification using wavelet packet transform. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, pages 321–326.

Armin Mehrabian, Yousra Al-Kabani, Volker J Sorger, and Tarek El-Ghazawi. 2018. Pcnna: A photonic convolutional neural network accelerator. In *2018 31st IEEE International System-on-Chip Conference (SOCC)*, pages 169–173.

Rafał Mirek, Andrzej Opala, Paolo Comaron, Magdalena Furman, Mateusz Król, Krzysztof Tyszka, Bartłomiej Seredyński, Dario Ballarini, Daniele Sanvitto, Timothy CH Liew, et al. 2021. Neuromorphic binarized polariton networks. *Nano letters*, 21(9):3715–3720.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Hammad Qureshi, Olcay Sertel, Nasir Rajpoot, Roland Wilson, and Metin Gurcan. 2008. Adaptive discriminant wavelet packet transform and local binary patterns for meningioma subtype classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 196–204. Springer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Xuting Tang, Abdul Rafae Khan, Shusen Wang, and Jia Xu. 2022. Learning by interpreting. In *Proceedings of the 31th International Joint Conference on Artificial Intelligence (IJCAI 2022)*.

Carlo M Valensise, Ivana Grecco, Davide Pierangeli, and Claudio Conti. 2022. Large-scale photonic natural language processing. *Photonics Research*, 10(12):2846–2853.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Tianyu Wang, Shi-Yuan Ma, Logan G Wright, Tatsuhiro Onodera, Brian C Richard, and Peter L McMahon. 2022. An optical neural network using less than 1 photon per multiplication. *Nature Communications*, 13(1):123.

Wenlu Wang, Guofan Jin, Yingbai Yan, and Minxian Wu. 1995. Image feature extraction with the optical haar wavelet transform. *Optical Engineering*, 34(4):1238–1242.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Geraldo Xexéo, Jano de Souza, Patrícia F Castro, and Wallace A Pinheiro. 2008. Using wavelets to classify documents. In *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 272–278. IEEE.

Xingyuan Xu, Mengxi Tan, Bill Corcoran, Jiayang Wu, Andreas Boes, Thach G Nguyen, Sai T Chu, Brent E Little, Damien G Hicks, Roberto Morandotti, et al. 2021. 11 tops photonic convolutional accelerator for optical neural networks. *Nature*, 589(7840):44–51.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

A Optical Method Hardware Definitions

The optical method used is a simulation of a nonlinear optics method that depends on specific hardware. The hardware components within the original optical method are a Spatial Light Modulator and a Magnesium-doped Periodic Poled Lithium Niobate crystal. A Spatial Light Modulator is used to manage the properties of the input light beam. In the optical method, two Spatial Light Modulators are used: one to control the beam propagating the sentence embedding matrix and another for the beam propagating the LG modes generated. A Magnesium-doped Periodic Poled Lithium Niobate crystal is also used to convert beams sent through it. These two hardware components essentially perform 2D partial Fourier Transformations on the beams representing the sentence embeddings and the LG modes, as shown in the footnote on page 3.

B Parameters for Optical Method

The optical simulation uses an input size of 1200×1200. As the embedding matrices have a size of sentence $len\times$ 768 (the model dimensions), we resize the embedding matrix using the bicubic interpolation method. By default, we set the azimuthal (l) and radial (p) index ranges to [0,7] and [-2,2] respectively to generate a set of 40 LG modes. These 40 LG modes, therefore, extract a 40-length feature vector for each data sample.

We set the default ranges of the azimuthal (l) and radial (p) indexes to [-2,+2] and [0,7] respectively. These values create 40 different LG modes and therefore output 40 length feature vectors.

C Time Comparison for feature extraction

Table 4 shows the comparison of feature extraction time of the 'Direct Method' compared with the 'Optical Method'.

GLUE Task	Direct Method		Optical
	Wavelet	FFT	Method
CoLA	3	4	117
RTE	2	3	68

Table 4: Comparison of time (minutes) for feature extraction using each of the methods.

Table 5 shows the feature extraction time for the Optical hardware compared to Optical simulation. The much longer feature extraction time for the Optical hardware is attributed to the conversion of sentence embedding from electrical signals to light beams. Faster hardware equipment, e.g. Digital Micrometer Device (DMD) with faster modulation speeds compared to SLM, can potentially reduce this time by more than 100 times.

Optical Method	Time (min)
Hardware	723
Simulation	289

Table 5: Feature extraction time comparison of WNLI task for optical simulation and optical hardware.

D Dataset statistics

Table 6 shows the stats for all the GLUE tasks we experimented with.

Task	Train	Valid	Test
CoLA	8,551	1,043	1,063
SST-2	67,349	872	1,821
STSB	5,749	1,500	1,379
RTE	2,490	277	3,000
MRPC	3,668	408	1,725
WNLI	635	71	146

Table 6: Number of sentences for each GLUE task.

E Direct Method Algorithm

Algorithm 3 describes the algorithm for extracting the wavelet features given the sentence embedding W as well as the mother wavelet m as inputs. By default we use the high-pass features only.

Algorithm 3 Direct Method feature extraction

```
Input: sentence embedding ($W$), mother wavelet ($\psi$)
Output: High-pass Wavelet features ($F$)
$Z =$ apply 2D wavelet transformation on $W$ using $\psi$
$F =$ extract high-pass features from $Z$
return $F$
```

F Functional Analysis for Wavelet Transformation

The Wavelet transform is a multi-resolution representation of an input function. Unlike the Fourier transformation, the Wavelet transform has a rich wavelet basis that can be utilized for functions with different characteristics. Additionally, we can analyze the input both in the time as well as the frequency domain. It is widely used in engineering, astronomy, neuroscience, and other fields.

The Wavelet transform represents the input function based on two concepts; scaling and translation. Scaling stretches or compresses the frequency of a given wavelet in the time axis, and translation moves (shifts) the wavelet along the time axis.

A 'mother wavelet' $\psi(t)$ with a Fourier transformation of $\psi(\omega)$ can be represented concerning a set of basis functions known as the 'daughter wavelets'. The daughter wavelet is a scaled and translated representation of the mother wavelet. It can be defined as:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t)\psi(\frac{t-\tau}{s}) \qquad (1)$$

Here, $\tau, s \in \mathbb{R}; s \neq 0$ are the translation and scaling factors, and $\psi_{\tau,s}(t)$ is generated by the mother wavelet $\psi(t)$.

The wavelet should satisfy an admissibility condition,

$$C_\psi = \int_R \frac{|\hat{\psi}(w)|^2}{|w|} dw < \infty \qquad (2)$$

Given a function, $\psi = L^2(\mathbb{R})$, and input function $f(x)$ with a Fourier transform $\psi(\omega)$, and using the concepts from Equation 1 and 2, the wavelet transform in the time domain can be defined as follows:

$$[W_\psi f](\tau, s) = \frac{1}{\sqrt{s}} \int_{-\infty}^{\infty} f(x)\psi\Big(\frac{x-\tau}{s}\Big)\, dx \qquad (3)$$

The equivalent frequency domain representation of Equation 3 is

$$[W_\psi f](\tau, s) = \frac{\sqrt{s}}{2\pi} \int_{-\infty}^{+\infty} F(\omega)\psi(s\omega)e^{j\omega\tau}\, d\omega \qquad (4)$$