INDIAN INSTITUTE OF TECHNOLOGY HYDERABAD

DEPARTMENT OF PHYSICS

# APPLICATION OF DEEP LEARNING IN SUPERCONTINUUM GENERATION

*A project report submitted in partial fulfillment of degree of B.Tech in Engineering Physics*

Author

M Dinesh Kumar
EP20BTECH11011

Supervisor

Dr. K. Nithyanandan
Assistant Professor, IITH

February, 2023

**Acknowledgement**

It gives me immense satisfaction to present the report of my project work undertaken during the sixth semester of the Engineering Physics B.Tech. program. I would like to express my deepest thanks to my project guide Dr. K. Nithyanandan, Assistant Professor, Department of Physics, IIT Hyderabad for their support and guidance throughout the course of my work. Their constant efforts have helped me progress.

I would like to express my special thanks to my faculty advisor Dr. Kirit D. Makwana, Assistant Professor, Department of Physics, IIT Hyderabad for the efforts they provided throughout the course of EP3105. I would also like to thank Mr. Rajkumar Joseph, Ph.D. for their useful suggestions and they were helpful to me during the project's development. In this aspect, I am extremely grateful to them.

I would like to acknowledge that this project was done entirely by me and not by someone else.

## Abstract

With the rapid development of fiber optics and photonic technologies, it's becoming increasingly important to model the nonlinear dynamics in optical fibers effectively. Supercontinuum light, generated by a train of laser pulses propagating through an optical fiber, is a classic example. The forward process of determining the spectrum based on pulse parameters is governed by nonlinear equations that can be reproduced through numerical simulations. However, numerical simulations are often computationally intensive and require time to reproduce. Neural networks like Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been employed to address the pitfalls.

Recently, another neural network has been gaining popularity among engineers and researchers alike, and that is the Transformer Neural Network. Since its formalisation, the Transformer architecture has outperformed traditional neural networks in many tasks across various domains and has been particularly useful for modeling large datasets. In this report, we utilise the transformer architecture to model supercontinuum light and possibly demonstrate the versatility of transformer architecture to model nonlinear phenomena in nature as well.



Keywords: Nonlinear dynamics; Super-continuum; Regression; Deep-Learning; Transformer architecture; Self-attention;

# Table of Contents

# 1 Introduction

In the last few years, machine learning has made big strides in science and technology, impacting areas like healthcare, autonomous cars, and language processing. One exciting application of machine learning is in studying complex systems, which often behave in unpredictable ways and can be sensitive to small changes. One such area is supercontinuum generation, where light is generated in optical fibers. This process can be difficult to model using traditional numerical methods because it requires a lot of computational resources and is sensitive to various parameters.

## 1.1. Scope

The application of machine learning techniques, such as neural networks and evolutionary algorithms, has garnered attention for analysing and optimising nonlinear systems [1]. While many conventional approaches have shown effectiveness in optimising specific characteristics of nonlinear systems, they often suffer from slow convergence rates and may fail to properly showcase the relationships between various variables in the process. This research seeks to bridge the gap between traditional methodologies and advanced machine learning techniques, specifically targeting the limitations present in existing optimisation strategies.

## 1.2. Objective

The objective is to enhance the modeling and prediction of supercontinuum generation dynamics [2] through the implementation of a Transformer-based architecture. By exploiting the unique capabilities of Transformers such as the coveted self-attention mechanism, we aim to capture complex relationships and patterns within the data more effectively. This approach is expected to yield more accurate and efficient regression of spectral intensity profiles, ultimately reducing the reliance on computationally intensive simulations in real-time experiments. The implementation of the architecture can be found at https://github.com/dnshkmr7/deep-learning-photonics

# 2 Background

In this section, several related concepts will be explained for better understanding the process of the project.

## 2.1. Transformer Neural Network

The Transformer neural network is a novel architecture designed to do sequence-to-sequence tasks while efficiently managing long-range dependencies. Introduced in the influential paper "Attention Is All You Need," the Transformer has emerged as a state-of-the-art technique in various applications. Its effectiveness stems from three key innovations: positional encodings, attention mechanism, and self-attention [3].

### 2.1.1. Positional Encoding

Positional encoding [3] injects information about the relative or absolute position of tokens in a sequence into the model. This is crucial when dealing with sequences, as the order of tokens often carries significant meaning. In our model, we employ sinusoidal positional encodings, which are calculated using sine and cosine functions of different frequencies. This approach allows the model to easily learn to attend to relative positions, enabling it to handle sequences longer than those seen during training.

### 2.1.2. Attention Mechanism

Attention is a mechanism that allows a model to focus on relevant parts of the input sequence. It involves mapping a query and a set of key-value pairs to an output [3]. The output is a weighted sum of the values, where the weights are determined by the compatibility between the query and the corresponding keys. By attending to different parts of the input, the model can capture complex relationships and dependencies within the data.

### 2.1.3. Self-Attention

Self-attention [3] is a mechanism that allows a model to weigh the importance of different parts of its input sequence when generating an output. It works by calculating attention scores between pairs of input elements, determining which elements are most relevant to the current output position. This approach enables the model to capture long-range dependencies and parallelise computations.

## 2.2. Progress in Transformers

Numerous studies [4,5,6] have demonstrated the superior performance of Transformer architectures compared to other models. The following examples represent just a few of the extensive research findings that support the choice of Transformers in this project and showcase their advantages.

| Methods | | **Informer** | | LSTMa | | DeepAR | | ARIMA | |
|---|---|---|---|---|---|---|---|---|---|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| ETTh1 | 24 | **0.098** | **0.247** | 0.114 | 0.272 | 0.107 | 0.280 | 0.108 | 0.284 |
| | 48 | **0.158** | **0.319** | 0.193 | 0.358 | 0.162 | 0.327 | 0.175 | 0.424 |
| | 168 | **0.183** | **0.346** | 0.236 | 0.392 | 0.239 | 0.422 | 0.396 | 0.504 |
| | 336 | **0.222** | **0.387** | 0.590 | 0.698 | 0.445 | 0.552 | 0.468 | 0.593 |
| | 720 | **0.269** | **0.435** | 0.683 | 0.768 | 0.658 | 0.707 | 0.659 | 0.766 |
| . . . | | | | | | | | | |
| ECL | 24 | 0.239 | 0.359 | 0.493 | 0.539 | **0.204** | **0.357** | 0.879 | 0.764 |
| | 48 | 0.447 | 0.503 | 0.723 | 0.655 | **0.315** | **0.436** | 1.032 | 0.833 |
| | 96 | 0.489 | 0.528 | 1.212 | 0.898 | **0.414** | **0.519** | 1.136 | 0.876 |
| | 288 | **0.540** | **0.571** | 1.511 | 0.966 | 0.563 | 0.595 | 1.251 | 0.933 |
| | 672 | **0.582** | **0.608** | 1.545 | 1.006 | 0.657 | 0.683 | 1.370 | 1.982 |
| Count | | 32 | | 0 | | 6 | | 0 | |

**Table. 1:** Univariate long sequence time-series forecasting results on four datasets (five cases). [4]

| | LinearAR | LSTM | MTGNN | Temporal | **Spacetimeformer** |
|---|---|---|---|---|---|
| 40 hours | | | | | |
| MSE | 18.84 | 14.29 | 13.32 | 13.29 | **12.49** |
| MAE | 3.24 | 2.84 | 2.67 | 2.67 | **2.57** |
| RRSE | 0.40 | 0.35 | 0.34 | 0.34 | **0.33** |
| 80 hours | | | | | |
| MSE | 23.99 | 18.75 | 19.27 | 19.99 | **17.9** |
| MAE | 3.72 | 3.29 | 3.31 | 3.37 | **2.57** |
| RRSE | 0.45 | **0.40** | 0.41 | 0.41 | **0.40** |
| 160 hours | | | | | |
| MSE | 28.84 | 22.11 | 24.28 | 24.16 | **21.35** |
| MAE | 4.13 | 3.63 | 3.78 | 3.77 | **3.51** |
| RRSE | 0.50 | **0.44** | 0.46 | 0.46 | **0.44** |

**Table. 2:** NY-TX Weather Results. [5]

**Note:** Best performance is highlighted in **bold**. Complete tables are available in the respective original papers.

# 3 Implementation

## 3.1. SpectrumTransformer

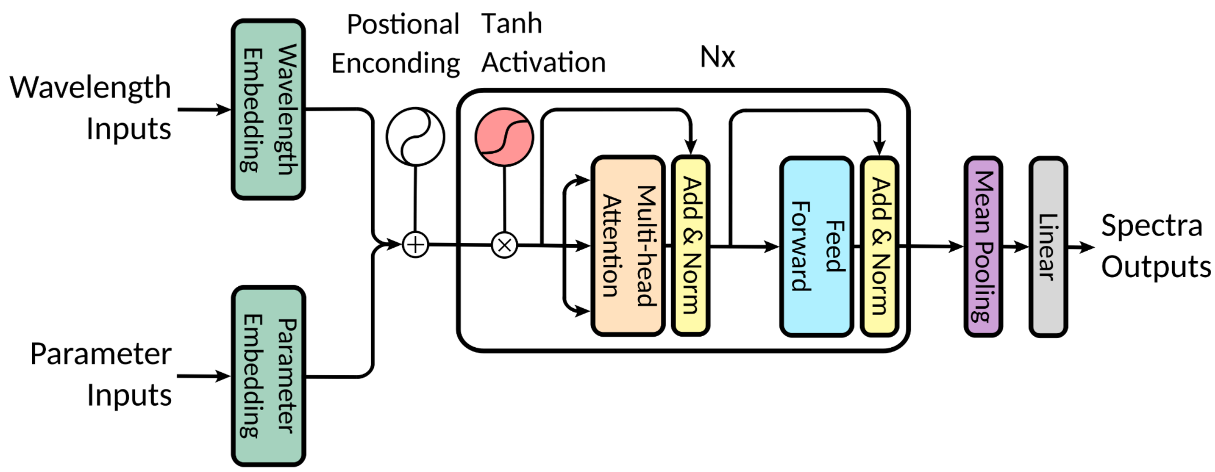


**Fig. 1:** Overview of the Spectrum Transformer, showcasing the key components involved in transforming inputs into predicted spectra.

The SpectrumTransformer is a specialised neural network designed for spectral data analysis, integrating spectral sequences and parametric information within a Transformer-based framework. This architecture modifies traditional Transformer elements to suit spectral data characteristics and enhance performance in spectral analysis tasks.

The SpectrumTransformer architecture is organised into distinct components, each contributing to the processing pipeline.

### 3.1.1. Input Processing Layer

The model begins with an embedding layer that transforms both the wavelength sequence and parametric input into a shared embedding space. The sequence embedding projects input data dimension ($d_{input}$) into an embedding dimension ($d_{embed}$) that aligns with the model's multi-head attention. For parameters, a separate linear projection maps parameter vectors from parameters data dimension ($d_{param}$) to the embedding dimension, which is then broadcast across the sequence length, ensuring compatibility for fusion with spectral data.

### 3.1.2. Feature Integration

The SpectrumTransformer incorporates the parametric information by performing element-wise addition between the input and parameter embeddings. This integration produces a combined representation that fuses spectral and parametric features while preserving dimensional consistency, allowing effective downstream processing in subsequent layers.

### 3.1.3. Positional Encoding

To encode positional information within the sequence, the SpectrumTransformer utilises sinusoidal positional encoding as described by Vaswani et al. (2017). This encoding method applies fixed patterns to indicate position within a sequence, using alternating sine and cosine functions based on position indices. Specifically, positional encoding for an embedding dimension $(d_{embed})$ is defined as:

$$PE(pos, 2i) = sin\left(\frac{pos}{10000^{\frac{2i}{d_{embed}}}}\right), \; PE(pos, 2i + 1) = cos\left(\frac{pos}{10000^{\frac{2i}{d_{embed}}}}\right)$$

This encoding allows the model to retain sequence ordering information while learning positional dependencies. Dropout with a rate of 0.1 is applied to the positional encodings to improve regularisation and prevent overfitting.

### 3.1.4. Transformer Encoder Stack

The core of the architecture is a stack of Transformer encoder layers, each containing a multi-head self-attention mechanism, a feed-forward network, layer normalisation, and residual connections. Each attention head operates over a scaled dot-product attention mechanism, allowing parallel computation across multiple heads. The feed-forward network within each layer transforms the embedded input through two linear transformations, separated by a nonlinear activation. This structure enables complex feature extraction from the spectral and parametric fusion.

### 3.1.5. Outputs

After passing through the encoder stack, the model aggregates the sequence dimension using mean pooling, reducing each sequence to a single vector. A final linear decoder layer then maps this vector to the output dimension ($d_{output}$). This setup allows for flexible adaptation to various spectral analysis tasks, producing an output ready for further processing or evaluation.

## 3.2. Dataset & Preprocessing

Details about the dataset are available in the Zenodo record titled *Supercontinuum Spectra* [2]. https://zenodo.org/records/6241344.

For preprocessing, spectra and wavelengths were binned to save memory and improve runtime efficiency. The spectra and wavelengths were divided into a specified number of bins (*n_bins*), reducing the dataset size while preserving essential information. For each bin, the function average of the spectral values across the wavelength range was taken, and the average wavelength for each bin was calculated to capture information efficiently.

$$\overline{S} = \frac{1}{\lambda_b - \lambda_a}\int_{\lambda_a}^{\lambda_b} S(\lambda)\, d\lambda \,;\, \overline{\lambda} = \frac{1}{b-a}\sum_{a}^{b} \lambda_i \,,\ \textit{where b-a is the bin size}$$

## 3.3. Results

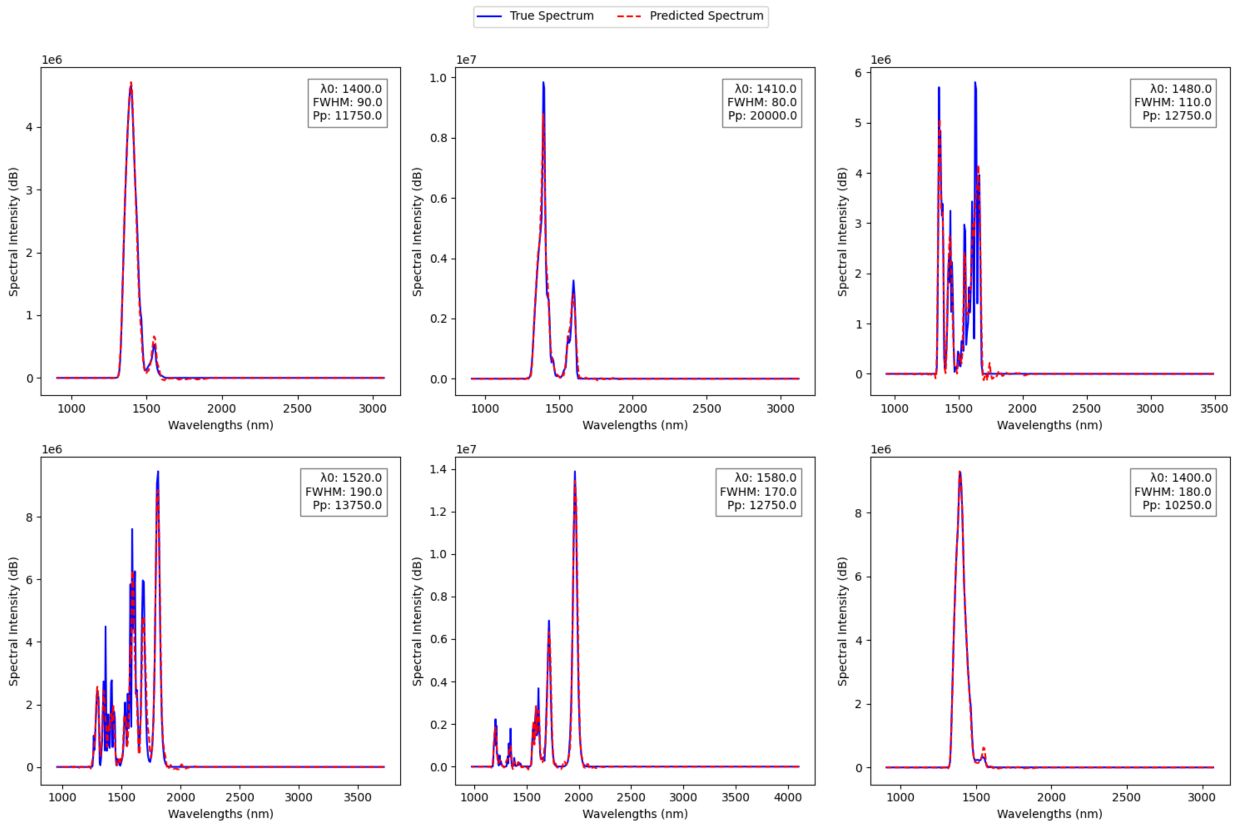


**Fig. 2:** Comparison of true and predicted spectra generated by the Spectrum Transformer model for various samples.

The SpectrumTransformer was used to process spectral data using a self-attention mechanism to capture long-range dependencies across binned wavelengths while considering three constant parameters (λ0, FWHM, and Peak Power).

### 3.3.1. Hyperparameters

For training, spectra and wavelengths were preprocessed into 512 bins to reduce data size while preserving essential information. The model was trained with a 100-epoch limit and early stopping with a patience of 10 to prevent overfitting. The initial learning rate of 1e-4 was managed with a ReduceLROnPlateau scheduler, reducing the rate by 0.5 upon validation loss plateaus for smoother convergence. AdamW was selected as the optimiser with a weight decay of 1e-5 to aid generalization. SmoothL1Loss was chosen as the best criterion, balancing sensitivity to outliers and maintaining stable predictions across noisy spectral data.

### 3.3.2. Loss functions

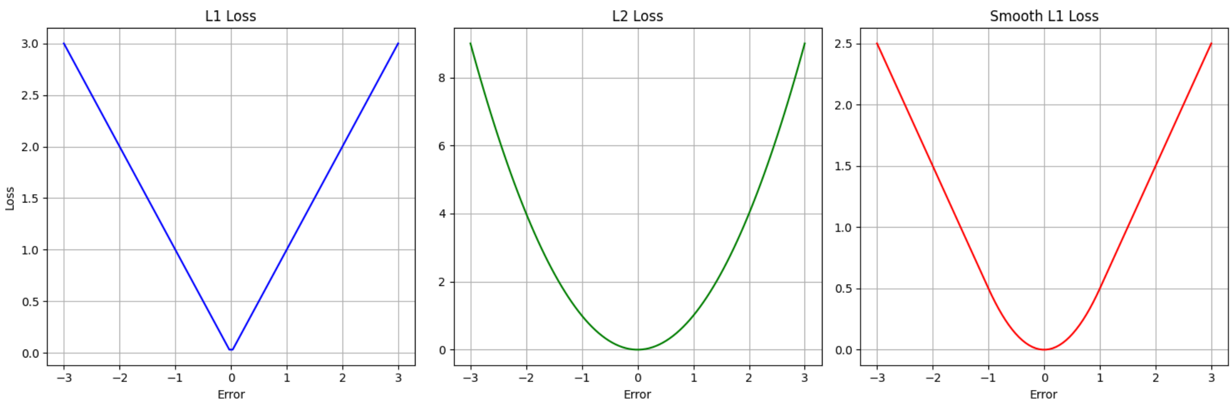


**Fig. 3:** Comparison of the L1, L2, and Smooth L1 loss functions, illustrating their behavior for varying error magnitudes.

| Metric | Loss | |
|---|---|---|
| | *bins = 512* | *bins = 256* |
| MSE (L2) loss | 1.3926 | 1.5851 |
| MAE (L1) loss | 0.0790 | 0.0659 |
| SmoothL1 loss | 0.0338 | 0.0250 |

**Table. 3:** Comparison of loss metrics for the Spectrum Transformer model on two different bin sizes.

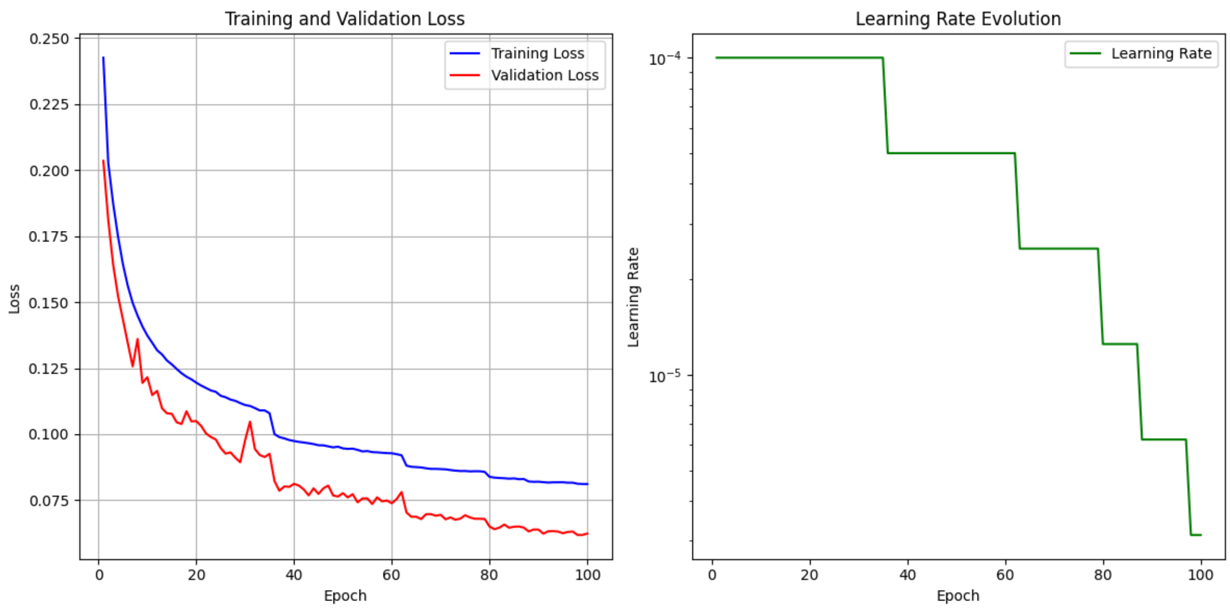


**Fig. 4:** Visualisation of the training and validation loss, and learning rate schedule during the training process of the model.

# 4 Conclusion

In conclusion, this research introduces a novel Spectrum Transformer architecture for predicting supercontinuum generation. The model effectively exploits self-attention to capture complex relationships within the input parameters and output spectra, leading to improved prediction accuracy. The experimental results demonstrate the model's potential for optimising supercontinuum generation processes. Future work may explore further refinements to the architecture, such as incorporating more sophisticated attention mechanisms or exploring alternative loss functions. Additionally, applying the model to a wider range of experimental conditions and fiber types could provide valuable insights into its generalisability and robustness.

## 4.1. Future Work

- Implementing Num2Vec embeddings instead of positional encoding seems to improve model performance in many use cases [7].
- Incorporating more advanced attention mechanisms may improve the model's performance [8].
- Training other architectures and comparing their performances, along with researching hybrid models, may provide valuable insights and enhance results [9,10]

# Bibliography


[1] Salmela, L., Tsipinakis, N., Foi, A., Billet, C., Dudley, J. M., & Genty, G. (2021). Predicting ultrafast nonlinear dynamics in fibre optics with a recurrent neural network. *Nature Machine Intelligence*, *3*(4), 344-354.

[2] Stucchi Diego, Corsini Andrea, Genty Goery, Boracchi Giacomoand Foi Alessandro, “Supercontinuum Spectra”. Zenodo, Feb. 23, 2022. doi: 10.5281/zenodo.6241344.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

[4] Zhou, Haoyi, et al. "Informer: Beyond efficient transformer for long sequence time-series forecasting." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. No. 12. 2021.

[5] Grigsby, J., Wang, Z., & Qi, Y. (2021). Long-range transformers for dynamic spatiotemporal forecasting. *arXiv preprint arXiv:2109.12218*.

[6] Huang, L., Mao, F., Zhang, K., & Li, Z. (2022). Spatial-Temporal Convolutional Transformer Network for Multivariate Time Series Forecasting. *Sensors*, *22*(3), 841.

[7] Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., ... & Brubaker, M. (2019). Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*.

[8] Chao Long, Yanmin Zhu, Haobing Liu, and Jiadi Yu. Efficient feature interactions learning with gated attention transformer, 2021.

[9] Yanglin Huang, Donghui Tan, Yuan Zhang, Xuanya Li, and Kai Hu. Transmixer: A hybrid transformer and cnn architecture for polyp segmentation, 2022

[10] Cao, K., Zhang, T. & Huang, J. Advanced hybrid LSTM-transformer architecture for real-time multi-task prediction in engineering systems. Sci Rep 14, 4890 (2024). https://doi.org/10.1038/s41598-024-55483-x