Next-Value Prediction of Beam Waists From Mixed Pitch Grating Using Simplified Transformer Model

Yu Dian Lim, *Member, IEEE*, Peng Zhao, *Member, IEEE*, Luca Guidoni, Jean-Pierre Likforman, and Chuan Seng Tan, *Fellow, IEEE*

***Abstract*—In this study, a simplified transformer model is used to predict the beam waist of 1,092 nm light coupled out from SiN-based mixed pitch gratings at various heights. The beam waists data at various heights above the grating is first compiled. Then, we used a sequence of the current beam waist values, z-positions, and the computed mathematical indicators (features) to predict the next beam waist value (labels). Optimized transformer model yields average percentage error (APE) of 6.6% between the predicted and actual beam waists, which corresponds to 93.4% prediction accuracy. This study provides a pioneering approach to using natural language processing model to perform predictive modelling on photonics data, and possible extrapolation of photonics data using transformer model.**

***Index Terms*—Attention, gratings, multi-head attention, photonics integrated circuits, quantum computing, self-attention, silicon photonics, transformer model.**

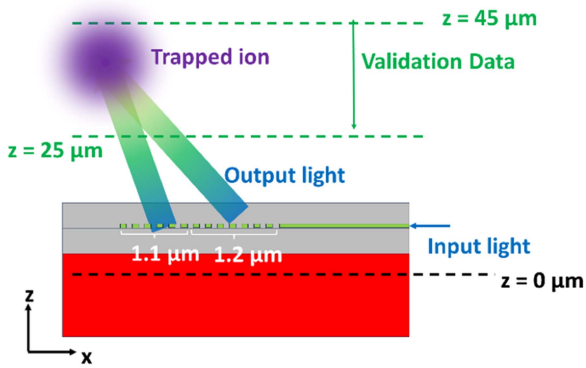


Fig. 1. Illustration of optical addressing using mixed pitch grating.

I. Introduction

RECENT breakthroughs in quantum technologies have sparked a surge in investments and research in quantum computing. Due to its exceptional computing performance, much attention has been placed on quantum computing as the next-generation computing technology [1], [2], [3]. Generally, quantum computing operation involves continuously-changing the quantum state of quantum bits (qubits) [4], [5]. Among various qubits, one of the popular choices is the trapped ion qubit, due to its high fidelity, CMOS-compatible scalability, and its feasibility in room temperature operation. Fundamentally, the computing operation of trapped ion qubits involves optical addressing of trapped ions, where laser light of specific wavelengths are required to alter the energy level of the ions that corresponds to the $|\ 1\ \rangle$ and $|\ 0\ \rangle$ quantum states [6]. For instance, $^{88}Sr^{+}$ ion requires 405, 422, 674, 1033, and 1092 nm for ionization, doppler cooling, qubit transition, quenching of metastable excited state, and clear-out during doppler cooling, respectively [7].

To further miniaturize the ion trap quantum computers, silicon photonics components have been integrated into the planar electrode ion trap for the optical addressing ion qubits. For instance, it has been reported that grating couplers can be buried underneath the planar electrode of ion trap, to couple light towards the trapped ions for optical addressing purposes [4], [8]. To achieve this, the light beam coupled out from the grating should have high degree of focusing with narrow beam waist. In the effort of enhanced focusing, our research group has reported several strategies, such as optimizing the radius of curvature [9], [10] and implementing mixed pitch gratings [11]. For mixed pitch gratings, the focusing mechanism is illustrated in Fig. 1. It can be seen that grating pitches of 1.1 and 1.2 µm, which correspond to different output directions for 1,092 nm light, are placed next to each other to form focused light beam. The details about the mechanism of the mixed pitch grating are discussed in our previous work [11].

In order to "aim" the light towards the trapped ion, the direction where the light coupled out from the grating should be thoroughly investigated for accurate positioning and integration of grating structure into the ion trap. Meanwhile, the understanding of beam waist progression above the grating is crucial. Ideally, the ions should be trapped in the position where optimal focusing occurs. In our previous works, we have reported the

Manuscript received 16 May 2024; revised 3 August 2024; accepted 8 August 2024. Date of publication 13 August 2024; date of current version 16 August 2024. This work was supported in part by the Ministry of Education of Singapore AcRF Tier 2 under Grant T2EP50121-0002 (MOE-000180-01) and in part by AcRF Tier 1 under Grant RG135/23, RT3/23. *(Corresponding author: Yu Dian Lim.)*

Yu Dian Lim is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: yudian.lim@ntu.edu.sg).

Peng Zhao is with the Interuniversity Microelectronics Centre (IMEC), 3001 Leuven, Belgium (e-mail: peng.zhao@imec.be).

Luca Guidoni and Jean-Pierre Likforman are with the Laboratoire Matériaux et Phénomènes Quantiques (MPQ), Université de Paris, F-75205 Paris, France (e-mail: luca.guidoni@univ-paris-diderot.fr; jean-pierre.likforman@univ-paris-diderot.fr).

Chuan Seng Tan is with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798, and also with the Institute of Microelectronics, Agency for Science, Technology and Research (A∗STAR), Singapore 117685 (e-mail: tancs@ntu.edu.sg).

Digital Object Identifier 10.1109/JPHOT.2024.3442169

analysis of photonics data simulated by finite difference time domain (FDTD) method using Python-based data science techniques [10], [11]. For more versatile design and integration of grating into the planar electrode ion trap, the optical properties of light coupled out from the grating can be modelled using machine learning technique. In the state-of-the-art, pioneering works on the usage of machine learning in photonics have been reported [12]. For instance, Hooten et al. reported the usage of reinforcement learning on the inverse design of grating couplers [13]. Meanwhile, Tu et al. reported using deep neural network (DNN) technique for similar applications [14]. Nevertheless, the research works of machine learning in photonics are still lacking, where more studies are called for.

In our previous work, we used DNN technique to perform predictive modeling on light beams coupled out from gratings with various radius of curvature. We first simulated and computed the beam waists of 1,092 nm light coupled out from 6 different design of gratings between z = 0 to 45 μm (ref. Fig. 1) to obtain 6 datasets. We trained the DNN models using 5 datasets (from 5 grating designs), then predict the beam waists coupled out from the remaining grating (z = 0 to 45 μm) [15]. Besides this approach, another possible approach to use machine learning on FDTD-simulated beam waists is to extrapolate the beam waist coupled out from grating. By creating a machine learning model that can perform accurate next-value prediction on the beam waists, extrapolations can be performed.

To perform the next-value prediction, the beam waist propagation should be shaped into one dimensional (1D) data, with z values in the x-axis and their corresponding beam waists in the y-axis. Traditionally, 1D data can be modelled with recurrent neural network (RNN) technique. However, RNN suffers from the problem of vanishing gradients. Upon extensive computation and modelling, the gradient in RNN tends to reduce significantly, where the parameter updates become insignificant [16]. To resolve the vanishing gradient problem, an alternative to RNN in modeling 1D data is the transformer model. Transformer model was first introduced by Google Inc., and was widely used in natural language processing (NLP). Generally, a transformer model takes in a sequence of data, compute the relevance between each vector using a self-attention mechanism, and predict the output corresponding to the input data [17].

Due to its ability to model long data sequence without vanishing gradient, the usage of transformer in NLP applications, such as language translation and next-word prediction has been widely reported. A well-known example would be the generative artificial intelligence software, ChatGPT (Chat Generative Pre-Trained Transformer), where its key algorithm is based on transformer model [18]. In the context of next-word prediction in NLP, the transformer first tokenizes an initial sentence into a sequence of numbers. Next, the sequence of numbers is inserted into the trained transformer model as the input data. Then, the model basically predicts the numbers associated to the next words, and produces the corresponding words as the output data. Similar concept can also be applied in 1D data. In extrapolating 1D data, the transformer model takes in a sequence of earlier numbers as the initial input data, and predicts the subsequent numbers for extrapolation. In the state-of-the-art, Li et al. reported usage of transformer-based generative adversarial network in time-series anomaly detection [19]. At the same time, Zhao et al. reported implementation of attention-based mechanism in transformer model in time series data imputation [20]. Nevertheless, there has been limited studies on the usage of transformer models on photonics-related applications.

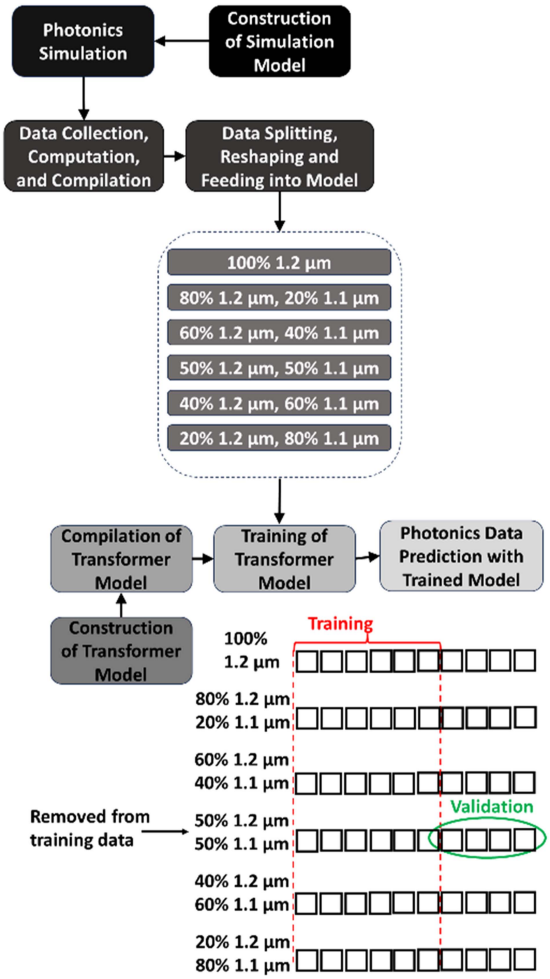


Fig. 2. General workflow used in this study. Six sets of simulated data ("100% 1.2 μm" to "20% 1.2 μm/80% 1.1 μm") are split into training and validation data. The validation data is fixed to be 0.4 of "50% 1.2 μm/50% 1.1 μm" dataset. Six training datasets are inserted into the model set-by-set.

In this study, a simplified transformer model is used to model and predict the beam waist coupled out from mixed pitch gratings. The beam waists are first simulated and computed for 1.2 μm pitch grating and 1.1/1.2 μm mixed pitch gratings. Then, mathematical indicators, including relative strength index (RSI) and exponential moving average (EMA), are computed. The obtained beam waists and indicators are then compiled into datasets. The datasets are then used to train, optimize, and evaluate the prediction accuracy of transformer models.

II. Data Shaping and Model Construction

Fig. 2 illustrates the overall workflow of this study. For the "Construction of Simulation Model" and "Photonics Simulation" parts, the beam waists of the light coupled out from the

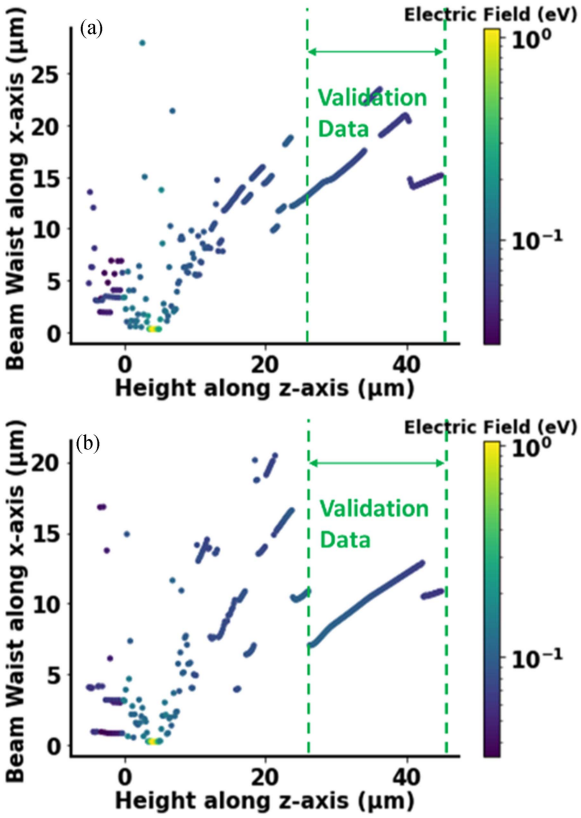


Fig. 3. Simulated beam waist along z-axis (a): 1.2 μm grating (b) 1.1/1.2 μm mixed pitch grating.

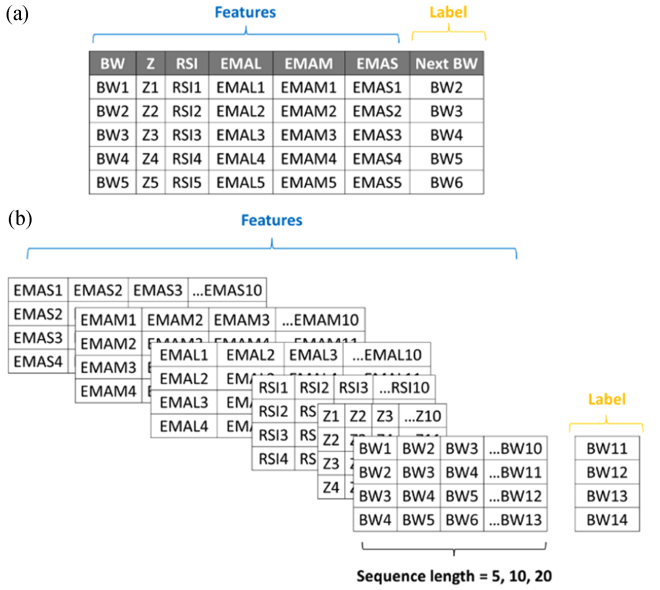


Fig. 4. Compiled features and labels for model training. BW and RSI are beam waist and relative strength index, while EMAL, EMAM, and EMAS are the exponential moving average of the past 40, 30, and 20 BW. (b) Features rearranged into 3D data array to be fed into the transformer to compute their respective label.

gratings are simulated using Fourier-difference time-domain (FDTD) technique (1,092 nm wavelength, z = 0 to 45 μm) and computed using Python-based data analysis techniques. Laser light with 1,092 nm wavelength is selected for this study, as it is used for the clear-out function in trapped $^{88}Sr^{+}$ ion qubit. The grating used is 0.4 μm thick SiN etched-through grating. In a typical FDTD simulation, 1.092 nm laser light (labeled as "Input light" in Fig. 1) is first coupled into the waveguide and propagates towards the grating. After that, the light is coupled out (labeled as "Output light" in Fig. 1) from the grating, where the electric field distribution of the 1,092 nm light of z = 0 – 45 μm is captured by a 3D monitor. The beam waist at each z-position is then computed using Python-based data analysis technique. Similar FDTD simulation and beam waist computation is performed repeatedly for gratings with pitches of 100% 1.2 μm, 80% 1.2 μm/20% 1.1 μm, 60% 1.2 μm/40% 1.1 μm, 50% 1.2 μm/50% 1.1 μm, 40% 1.2 μm/60% 1.1 μm, and 20% 1.2 μm/80% 1.1 μm. The details of the FDTD simulation and beam waist computation is described in our previous work [11].

Fig. 3 compares the beam waists of 1,092 nm light from 1.2 μm grating and 1.1/1.2 μm mixed pitch grating. It can be observed that beam waists from 1.1/1.2 μm mixed pitch grating are approximately ~25% smaller than 1.2 μm grating along z-axis due to the focusing mechanism of mixed pitch grating. It can be observed that the beam waists below z = 25 μm exhibits high level of irregularities. As referred to Fig. 1, the top oxide layer above the grating ends at z = 10 μm, which means that be beam forming generally starts from z = 10 μm. The z = 10 μm to 25 μm region represents the reactive field region, approximately ranges between the grating to $0.62\left(\sqrt{\frac{D^3}{\lambda}}\right)$ above the grating, where D is the maximum linear dimension of grating and λ is the wavelength [21]. Meanwhile, region above z = 25 μm represents the far-field region, approximately ranges above $0.62\left(\sqrt{\frac{D^3}{\lambda}}\right)$ [22]. Thus, in our study of machine learning modeling and prediction using transformer model, beam waists in the reactive field region will be used as the features (data used to make predictions), while beam waists above the reactive field region will be used as the labels (data to be predicted).

The "Data Collection, Computation, and Compilation" part of Fig. 2 is illustrated in Fig. 4(a). The primary data collected from the FDTD simulation and Python-based computation are beam waist (BW) and height (z) above grating, with total 317 rows of primary data. To better model the beam waist, we used mathematical indicators such as Relative Strength Index (RSI) [23] and exponential moving average (EMA) [24] of the BW values. The RSI can be calculated by the following equation:

$$RSI = 100 - \frac{100}{1 + \frac{Avg.\ Increment}{Avg.\ Reduction}} \quad (1)$$

where Avg. Increment/Avg. Reduction of a specific Z/BW row are the average increase/decrease in BW values of the past 10

data points. Meanwhile, EMA can be calculated using:

$$EMA_n = (BW_n)\left(\frac{2}{1+N}\right) + (EMA_{n-1})\left(1 - \frac{2}{1+N}\right) \quad (2)$$

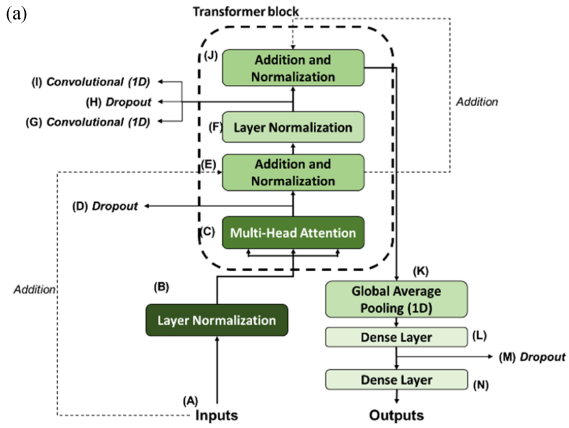
where the EMAS, EMAM, and EMAL shown in Fig. 4(a) are calculated with N = 20, 30, and 40, respectively (S, M, L in EMA represents 'Short', 'Medium, 'Long', respectively). In this work, the RSI, EMAS, EMAM, and EMAL values are calculated using 'pandas_ta' package, which is a built-in package for technical analysis in Python. The usage of mathematical indicators, EMAS, EMAM EMAL and RSI reflects the moving averages of 10 – 40 prior beam waist values. These indicators are widely-used in financial modelling, and prediction of financial indicators such as stock price and index values [25]. In the context of next-value prediction of beam waist along z-axis, these indicators illustrate the moving trend of the beam waist, 10 – 40 values prior to the current beam waist values. Thus, it can be deduced that these indicators can be a useful tool in performing next-value predictions on beam waist values along z-axis.

As shown in Fig. 4(a), each dataset is split into feature columns and label column, where features are the data used for prediction, while label is the data to be predicted. Note that 'ratio', which indicates the mixing ratio of 1.1/1.2 µm pitches, is also included as one of the features. The label is set to be the BW value of the next data row. This implies that the transformer model takes in the features of a current data point to predict the beam waist of the next data point. Along the rows of the datasets, the datasets are split into training and validation datasets. Training datasets are the data used to train the transformer model, while the validation datasets are the data used to evaluate the prediction performance of the trained model.

As shown in Figs. 2 and 5 datasets are used as training data to train the transformer model, with mixing ratio of "50% 1.2 µm/50% 1.1 µm" singled out (not fed into the transformer model for training). Meanwhile, 60% of each dataset (including BW, z, RSI, EMAS etc.) from the SiO_2/SiN region and the reactive field region is used as the training data, while 40% of the dataset is used as the validation data. The purpose of having a validation dataset is to evaluate the predictive performance of the trained transformer model.

The "Data Splitting, Reshaping, and Feeding into Model" in Fig. 2 is illustrated in Fig. 4. Each feature is rearranged from 1D data arrays to 2D data arrays. Note that the number of columns in each 2D data array represents the sequence length inserted into the transformer model during the model training. As illustrated in Fig. 4(b), when the sequence length is 5, the features of the past 5 data points will be considered. In this study, sequence length of 5, 10, and 20 will be explored. The details of sequence length variations will be discussed in the next sections.

The "Construction of Transformer Model" part in Fig. 2 is illustrated in Fig. 5. The simplified transformer model illustrated in Fig. 5(a) is inspired by the full transformer model proposed by Google Inc., without the word embedding and the



(b)

Layer	Layer Name	Parameters	Description
(A)	Input Layer	0	Layer to insert features described in Fig. 4(b)
(B), (F)	Layer Normalization	14, 14	Normalizing inputs to values between 0 to 1
(C)	Multi-head Attention	999	Performing multi-head attention for features with sequence length of 5, 10 or 20.
(D), (H), (M)	Dropout	0	Randomly converting 25% of incoming data to NaN (not a number) to avoid overfitting
(E), (J),	Addition and Normalization	0	Adding outputs from two layers, then normalize the product to values between 0 to 1 (E) = (A) + (D); (J) = (E) + (I)
(G), (I)	!D Convolutional	16, 21	Perform 1D convolution by multiplying each element in the input with one randomly-generated vector
(K)	1D Global Average Pooling	0	Computing mean of elements in each vector
(L), (N)	Dense Layer	768, 129	Conventional fully-connected neural network with 128 nodes in each layer

Fig. 5. (a) Outline of the simplified transformer model. The transformer blocks are repeated 2, 3, and 4 times for model optimization, (b) Number of parameters and layer-by-layer description for each layer in the simplified transformer model (model with 1 transformer block). The layers in red are layers belong to the transformer block.

encoder/decoder labels in the model [17]. The simplified transformer model uses a 2-head input for the multi-head attention layer, where the head size of each head is fixed to be 16. In the simplified model, the reshaped features illustrated in Fig. 4(b) are first inserted as a sequence of input data. For instance, input data with sequence length of 5 will be inserted as a series of 5 vectors, with each vector consists of 6 elements (EMAS, EMAM, EMAL, RSI, Z, and current BW). The layer-by-layer route for the input data in the transformer model, the function of each layer, and the corresponding parameters in each layer, are illustrated in Fig. 5(a) and (b). For models with 2, 3 and 4 transformer blocks, the details of the transformer model can be found out by expanding the layers in red as shown in Fig. 5(b). Alternatively, 'model.summary()' function can be used in the full Python code provided, which will be mentioned later. The details of the transformer model and multi-head attention are discussed in ref [17], [26]. The datasets shown in Fig. 2(100% 1.2 µm, 20% 1.2 µm/80% 1.1 µm … etc.) are inserted into the transformer model set-by-set for model training.

Fundamentally, one important consideration when constructing a machine learning model is the availability of training data. When a small amount of training data is available, the model should be kept small, with less trainable parameters. In this study,

we started with a big transformer model, with 4 heads in the multi-head attention, head size of 64, and 4 transformer blocks. The number of trainable parameters is 30,817. As we train the transformer model set-by-set, the dimension of each dataset is (146, 5, 6), indicating 146 rows, sequence length of 5, each sequence has vectors with 6 elements. This result in only 4,380 datapoints, where 5 training datasets only yield a total of 21,900 datapoints. As the number of trainable parameters exceeds the number of training data available, the model training is unstable, and often result in inconsistent outcome. To resolve this, we used small transformer models with 2 heads in multi-head attention layer, where the size of each head is only 16. As a result, the number of trainable parameters reduced to 3,025 – 7,073 for transformer models with 2 – 4 transformer blocks and 5 – 20 sequence length.

After constructing the small transformer model, the model is then compiled using mean squared error as the loss indicator and 'Adam' optimizer with learning rate of 10^{-4}, as referred to the "Compilation of Transformer Model" in Fig. 2. For the model training ("Training of Transformer Model" in Fig. 2), the number of epochs is fixed at 100. The 5 training datasets illustrated in Fig. 2 are compiled into 5 separated tables similar to Fig. 4(a). Then, we reshaped the datasets into 5 3D data arrays similar to Fig. 4(b). The 5 data arrays are then fed into the transformer model for model training array-by-array. After the preparation of training datasets and transformer model, the model is then trained. The training is carried out using a workstation with 16 cores processor with up to 5.7 GHz boost clock; and a gaming graphic card with 24 GB GDDR6X RAM running at 2520 MHz clock speeds.

Upon training of the model, the accuracy of the prediction using the trained model is tested using the validation data. The features of the validation data (reshaped BW, z, RSI, EMAL, EMAM, and EMAS) are used to predict the label of the validation data (Next BW), as referred to Fig. 4. As mentioned earlier, "50% 1.2 μm/50% 1.1 μm" dataset is used as the validation data. The abovementioned processes are carried out in Python platform. The Python codes for transformer model construction, model training and model prediction used in this study extends from the source code presented in [27].

III. Prediction of Beam Waist

The "Photonics Data Prediction with Trained Model" illustrated in Fig. 2 is presented in the following sections. For the optimization of simplified transformer model, we employed two variations: sequence length (5, 10, 20) and number of transformer blocks (2, 3, 4). The average percentage error (APE) between actual (simulated) beam waist and predicted beam waist is used as the main figure-of-merit to determine the predictive accuracy of the model.

Fig. 6(a) shows the APEs between predicted/actual data; while Fig. 6(b) shows the average difference between training and validation data (from 50th epoch onwards, 5th set of training data "20% 1.2 μm/80% 1.1 μm"). Generally, it can be observed that model trainings with a sequence length of 5 exhibit the lowest

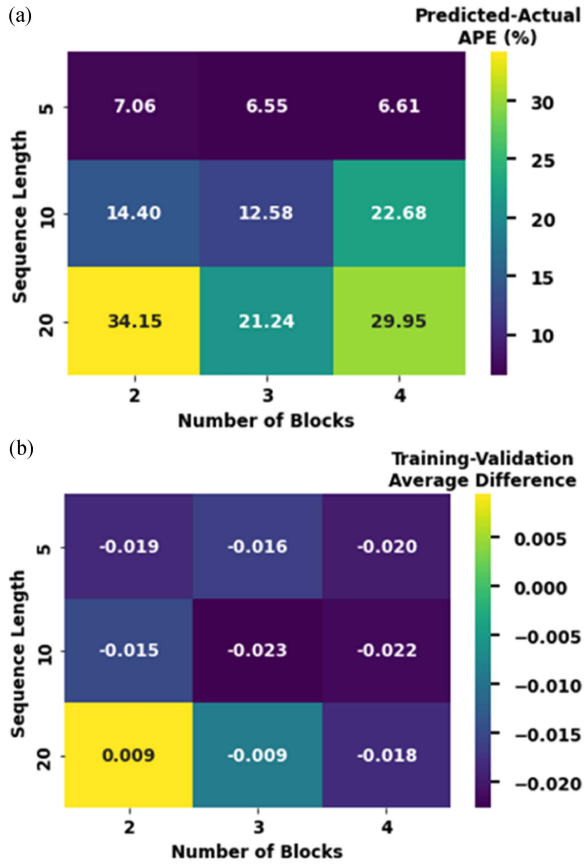


Fig. 6. (a) APE between prediction-actual data, (b) Average difference between training-validation data.

APEs, ranging between 6.6% and 7.06%. This is followed by sequence length of 10, with APEs ranging between 12.58% and 22.68%. The highest APEs, ranging from 21.24% to 34.15%, are observed in model trainings using a sequence length of 20. The high APEs using large sequence length can be due to the over-considerations of beam waists data from the 0 to 25 μm region (training data) at the early stage of the validation data (∼25 to 30 μm). As the 0 to 25 μm region tends to have higher beam waist data and higher degree of fluctuation, as shown in Fig. 3, over-consideration of these prior data (with large sequence length of 20) will cause the transformer model to predict higher-than-actual beam waist values.

The prediction of higher-than-actual beam waist values can be further visualized in Fig. 7. In Fig. 7(a), it is shown that the predicted beam waists fitted closely to the actual beam waist, with minimal deviations between these two sets of data. Generally, the percentage errors between actual and predicted beam waist are 0.6 to 9% from $z = 30$ μm to $z = 39$ μm. When the actual beam waist experiences sudden drop at $z = 39$ μm, the trained transformer model could not 'catch-up' with the sudden change. Thus, the percentage error spiked to 61%, then slowly reduced to 9 – 15% range. In contrast, in Fig. 7(b), there is a spike in the predicted data in the range of 26 to 28 μm. As a result, the percentage error spiked to $> 100\%$ at this stage. This implies the over-consideration of prior beam waist data using

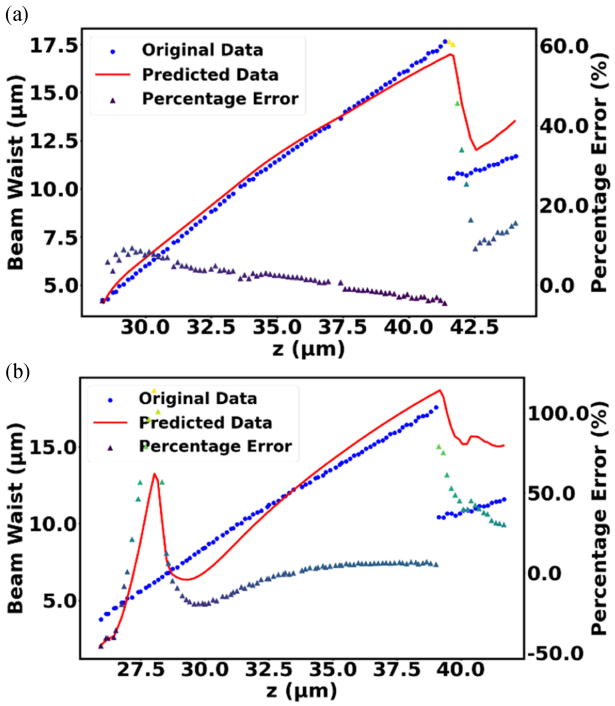


Fig. 7. Original data, predicted data and percentage error along z-axis: (a) sequence length = 5, number of blocks = 3, APE = 6.6% (b) sequence length = 20, number of blocks = 2, APE = 34.15%.

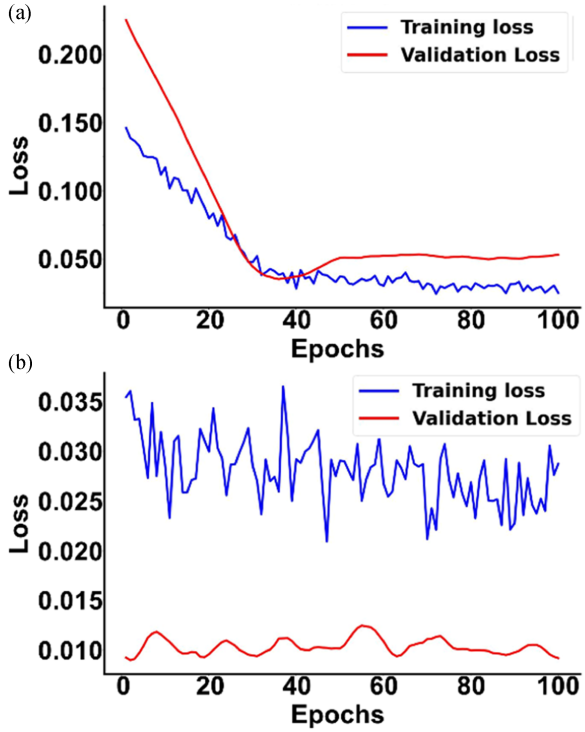


Fig. 8. Training loss-Validation loss curves of transformer model with 3 transformer blocks with sequence length = 5: (a) 1st training using "100% 1.2 μm" dataset, (b) 5th training using "50% 1.2 μm/50% 1.1 μm" dataset.

sequence length = 20, which result in higher-than-actual prediction of beam waist. Between z = 32 to 39 μm, the percentage errors reduced and stabilities to 0 – 6%. The higher-than-usual prediction is also observed between 40 to 45 μm. In this region, the actual beam waist experienced an abrupt drop. However, the predicted beam waist did not 'catch-up' with this drop, which can be attributed to the large sequence length of 20. The percentage error increases to 30 – 79% again.

Meanwhile, as referred to Fig. 6(a), it can be observed that transformer models with 3 transformer blocks yields the lowest APE for sequence length = 5, 10 and 20. As the accuracy of transformer model scales with both the size of training data and the size of the transformer model, it can be deduced that models with 3 transformer blocks with 4089 to 6009 trainable parameters is the most optimized combination for the size of 5 datasets (with 21,900 datapoints) used in this study. [28], [29].

Fig. 6(b) shows the average difference between the training and validation losses. Fundamentally, a negative value indicates that the training loss is larger than the validation loss, implying that the model is overfitted. Meanwhile, a positive value implies that the model is underfitted [30], [31], [32]. As mentioned earlier, the transformer models are trained with the 5 training datasets labeled in Fig. 2. The 5 training datasets are inserted into the transformer model set-by-set, with the epoch of each training fixed at 100. Thus, it can be deduced that the training-validation average difference is higher during the 1st training (using "100% 1.2 μm" dataset, when the model is under-trained), and the value of average difference reduces during the 5th training (using "50% 1.2 μm/50% 1.1 μm" dataset, when the model is trained). For instance, the training loss and validation loss curves for the transformer model with 3 transformer blocks and sequence length = 5 (combination with lowest APE in Fig. 6(a)) are shown in Fig. 8. At the 1st model training using "100% 1.2 μm" dataset, the average difference between training and validation data is 0.02, where the positive value indicates that the model is under trained. At the 5th model training using "50% 1.2 μm/50% 1.1 μm", the average difference dropped to -0.016, where the negative value indicates that the model is trained. In this study, the values shown in Fig. 6(b) are the average difference taken from the 5th training.

As mentioned earlier, in Fig. 6(b), a negative value implying that the model is overfitted, while a positive value implying that the model is underfitted. Generally, most transformer blocks/sequence length combinations shown in Fig. 6(b) are slightly overfitted. This indicates that further training does not further improve the prediction accuracy of the model. However, for transformer model with 2 transformer blocks with sequence length = 20, the model is slightly underfitted. As mentioned earlier, the accuracy of transformer model scales with both the size of training data and the size of the transformer model. Thus, for this transformer model, further training with more training data may improve its prediction accuracy. Nevertheless, the magnitude of the average difference between training-validation losses shown in Fig. 6(b) is rather small. Thus, it can be deduced that the models are sufficiently trained to achieve their respective optimized predictive performance. To further understand the

impact of key parameters, including the number of transformer blocks, sequence length, transformer size, size of training data, on the prediction performance of the transformer model, further studies are called for. Authors can use the full Python code (with linked dataset) provided in [33] to optimize the above-mentioned parameters. At the same time, the Python code for the transformer modelling can also be used for extended datasets, to investigate the impact of dataset size on the prediction performance of the transformer model. Nevertheless, the above-mentioned outcomes demonstrated modelling and prediction of transformer model on beam waist data, with accuracy of up to 93.45%.

IV. Discussion

In this work, we introduced a pioneering attempt in using transformer models on photonics data, a quantitative dataset which has much smaller scale than the conventional language processing application. Transformer model has been widely-used as large language models (LLM) in natural language processing (NLP) applications. Due to its 'data hungry' nature, transformer model often requires large amount of data to train [34]. This is due to the presence of massive parameters in a transformer-based LLM. In a typical NLP application, the parameters to trained often range from 70 million to over 16 billion [29], [35]. However, as we have limited photonics data available, we scaled down our simplified transformer model to have 4945 (2 transformer blocks) and 7073 (4 transformer blocks). Thus, despite limited data available, good fitting of the training dataset into the transformer model can be achieved.

This work can be viewed as an extension to a research work that our group has previously-reported. In [11], we reported using Python-based data science technique to compute the beam waists along x-axis and y-axis (ref. Fig. 1(b)) of 1,092 nm light coupled out from 1.1/1.2 μm mixed pitch grating. In this work, we used simplified transformer models to predict the beam waist coupled out from 50% 1.1 μm/50% 1.2 μm mixed pitch grating. This work can be benchmarked against another previously-reported work, where we used deep neural network (DNN) model to predict the beam waists coupled out from gratings with various radius of curvature [15].

As compared to [15], this work demonstrated slight improvement in APE values from 7.2% to 6.6% for well-fitted models. The concepts of predictive modelling of both works ([15] and current work) are fundamentally different. For the predictive modelling of beam waist using DNN models, we used only grating design indicator (radius of curvature) and the z-position to as the features, to predict the corresponding beam waists. Meanwhile, for this work, we performed next-value prediction using transformer models. This means that we used the current beam waist and other mathematical indicators (EMAL, EMAM, EMAS, RSI etc.) to predict the next beam waist values. The mathematical indicators are computed based on the moving trends of 10 – 40 beam waist values prior to the current beam waist. This implies that we include more features in this work than our prior work in [15]. At the same time, the nature of the DNN model is different from the transformer model. For the DNN model, the features (radius of curvature and z-position) are inserted into the model as isolated values. Upon training of the model, the parameters in the model are optimized without explicitly considering the relationship between the inserted features. On the other hand, the features of transformer model are inserted into the model as a sequence of vectors, where the correlation between each vector in the sequence is computed in the multi-head attention layer. In the context of our work, for sequence length = 5, a sequence of 5 vectors is inserted into the transformer model. Each vector consists of information including current beam waist, z-position, RSI, EMAL, EMAM, and EMAS. Then, the correlation between each vector is computed in the multi-head attention layer. Due to the different in prediction nature, inclusion of more features, and the presence of multi-head attention to compute the correlation between the vectors in the features, this work achieved slightly better APE values than our prior work in [15] for well-fitted models.

Another key differences between this work and [15] is the splitting of training and validation data. In [15], six beam waist datasets, similar to the dataset presented in Fig. 4(a), are simulated. Five datasets are used as the training data, and one dataset is used as the validation data. Similar to the previous work, six beam waist datasets are simulated in this work. However, among the simulated beam waist datasets, 60% of all six datasets are used as the training data, the unused 40% of the "50% 1.1 μm/50% 1.2 μm" dataset is used as the validation data. Another key difference is the inclusion of grating properties as the feature data. For [15], we include the radius of curvature, a key determining factor of the beam waist, as the feature data. However, in our current work, we treated each dataset neutrally, as we did not include mixing ratio as one of the features.

The exclusion of mixing ratio implies that we use a sequence of current beam waists (and other associated indicators) to predict the next beam waist value. Without considering the mixing ratio, the next-value prediction performed by in "50% 1.2 μm/50% 1.1 μm" dataset exhibits high prediction accuracy even under two conditions: (1) "50% 1.2 μm/50% 1.1μ m" beam waist data is not included when training the transformer model, (2) the next-value prediction is carried out in the z = 30 – 45 μm region, however, the training datasets described in Fig. 4 do not include this beam waists from this region at all. For the extrapolation of FDTD simulation, the model can be further modified, such as training to predict the next-5 or next-10 beam waist values. Due to scarcity of training data as we have limited computing resources in performing FDTD simulation, the proposed approach is not attempted in this study. Nevertheless, we demonstrate a possibility of extrapolating the beam waist using transformer model, where the fundamentals can be further expanded by fellow researchers with better computing resources, and mass amount of beam waist datasets.

A key limitation of this study is the lack of experimental data. As demonstrated earlier, the modelling and next-value prediction of beam waists from mixed-pitch gratings are carried out solely from data simulated by finite-difference time-domain (FDTD) technique. For mixed-pitch gratings, we have previously reported the comparison of simulated and experimentally-measured beam waists [11]. The obtained beam waists from mixed-pitched gratings show similar focusing phenomenon in both simulated and experimentally measurement settings. However, due to the limitations in experimental resources, we are only able to obtain beam waist from one z-position experimentally. As mentioned earlier, transformer model is generally "data-hungry". Thus, to perform similar transformer modelling and prediction on experimentally-measurement beam waists, highly-magnified beam profiles are required. To achieve this, high pixel infrared camera should be used. At the same time, a set of beam profile data should be taken for every 1 μm elevation of the infrared camera. Thus, a complex measurement system with high precision z-axis motorized stage is needed. Nevertheless, the obtained outcome from this work can serve as a solid foundation to achieve the abovementioned experiments, where similar transformer modelling can be used on experimental beam profile datasets.

As mentioned earlier, one of the distinctive differences between transformer model and deep neural network (DNN) model is the nature of the input features. For transformer model, correlations between all vectors in a sequence of input data are computing by the multi-head attention layer. The nature of this multi-head attention mechanism can be applied to optimizing other silicon photonics (SiPh) devices. For instance, when optimizing SiPh grating couplers, a series of grating parameters, including pitch, duty cycle, thickness, etch depth, and operational wavelength should be optimized. These parameters are related to each other, but the correlations are not explicitly addressed when modelling the parameters and corresponding grating performance using DNN model. Thus, the simplified transformer models introduced in this work can be used. For instance, a series of grating parameters mentioned earlier can be converted to a series of vectors. The parameters can be varied, to generate more combinations of gratings and its corresponding performance. The series can then serve as the input to the transformer model. Similar mechanism can also be applied on optimizing other SiPh devices, such as photodetectors, modulators etc.

A key improvement can be done to improve the prediction accuracy of the transformer model. For each dataset illustrated in Fig. 2, the dimension is (146, 5, 6), which translates to 4380 datapoints. As mentioned earlier, as transformer model is widely used in NLP, it is capable of handling millions of datapoints. Thus, to increase the prediction accuracy, FDTD simulation with higher mesh count and more data points should be used to train larger transformer models, with more trainable parameters. However, such improvement requires better computing resources. Thus, to enable the expansion of exploratory work in this direction, we have included the full Python code of the transformer model (with linked datasets) in [33].

V. Conclusion

In this study, simplified transformer models have been employed to perform modelling and prediction of beam waists of 1,092 nm coupled out from SiN mixed pitch gratings. The primary data used in the modeling is the simulated and computed beam waist (BW) and its associated height (z). Mathematical indicators, such as relative strength index (RSI) and exponential moving average (EMA) are computed to be included in the modeling and prediction. The modelling and prediction of transformer models uses a sequence of the current beam waist values, z-levels, and the abovementioned mathematical indicators (features) to predict the next beam waist value (labels). Sequence length of 5, 10, and 20; with 2, 3 and 4 blocks of transformer blocks, are explored. Predictions with lowest average percentage errors (APEs) are obtained from sequence length = 5, with APEs of 6.6 to 7.1%, corresponding to prediction accuracy of 92.9 to 93.4%. Transformer modelling with sequence length = 20 yields highest APEs, ranging from 21.2 to 34.2%. Nevertheless, the optimized transformer demonstrated a prediction accuracy of up to 93.4%. The obtained outcomes provide insightful inputs on the design and integrating of gratings for the optical addressing of trapped ion qubits. At the same time, the demonstration of using simplified transformer model to predict optical properties from gratings opens up a pioneering path on using natural language processing (NLP) models and mathematical indicators (RSI, EMA, etc.) in predicting photonics data. The next-value prediction underlies a foundation to the extrapolation of photonics data, where the framework can be further expanded to prediction the next-5 or next-10 beam waists. Besides, the multi-head attention of the transformer model can be used for the optimization of silicon photonics devices, where the correlations among the key parameters in the device can be considered.

Acknowledgment

The preparation of Python codes in [33] is partly assisted by the generative AI tool, ChatGPT

References

[1] B.-H. Wu, R. N. Alexander, S. Liu, and Z. Zhang, "Quantum computing with multidimensional continuous-variable cluster states in a scalable photonic platform," *Phys. Rev. Res.*, vol. 2, no. 2, 2020, Art. no. 023138, doi: 10.1103/physrevresearch.2.023138.

[2] J. M. Pino et al., "Demonstration of the QCCD trapped-ion quantum computer architecture," 2020. [Online]. Available: http://arxiv.org/abs/2003.01293

[3] J. M. Pino et al., "Demonstration of the trapped-ion quantum CCD computer architecture," *Nature*, vol. 592, pp. 209–213, 2021.

[4] K. K. Mehta, C. D. Bruzewicz, R. Mcconnell, R. J. Ram, J. M. Sage, and J. Chiaverini, "Integrated optical addressing of an ion qubit," *Nature Nanotechnol.*, vol. 11, no. 12, pp. 1066–1070, 2016, doi: 10.1038/nnano.2016.139.

[5] J. P. Covey, H. Weinfurter, and H. Bernien, "Quantum networks with neutral atom processing nodes," *npj Quantum Inf*, vol. 9, no. 1, 2023, Art. no. 90, doi: 10.1038/s41534-023-00759-9.

[6] P. Zhao, Y. D. Lim, H. Y. Li, L. Guidoni, and C. S. Tan, "Advanced 3D integration technologies in various quantum computing devices," *IEEE Open J. Nanotechnol.*, vol. 2, pp. 101–110, 2021, doi: 10.1109/ojnano.2021.3124363.

[7] J. P. Likforman, V. Tugayé, S. Guibal, and L. Guidoni, "Precision measurement of the branching fractions of the 5p P1/2 2 state in Sr + 88 with a single ion in a microfabricated surface trap," *Phys. Rev. A*, vol. 93, no. 5, pp. 1–9, 2016, doi: 10.1103/PhysRevA.93.052507.

[8] K. K. Mehta, C. Zhang, M. Malinowski, T.-L. Nguyen, M. Stadler, and J. P. Home, "Integrated optical multi-ion quantum logic," *Nature*, vol. 586, pp. 533–537, 2020. [Online]. Available: http://arxiv.org/abs/2002.02258

[9] Y. D. Lim, H. Y. Li, P. Zhao, J. Tao, L. Guidoni, and C. S. Tan, "Design and fabrication of grating couplers for the optical addressing of trapped ions," *IEEE Photon. J*, vol. 13, no. 4, Aug. 2021, Art. no. 2200306, doi: 10.1109/jphot.2021.3094646.

[10] Y. D. Lim, P. Zhao, L. Guidoni, J. -P. Likforman, and C. S. Tan, "Development of grating for the optical addressing of 88Sr+ ions with data analysis techniques," *IEEE Photon. J*, vol. 15, no. 4, Aug. 2023, Art. no. 6601907, doi: 10.1109/JPHOT.2023.3281134.

[11] Y. D. Lim, P. Zhao, L. Guidoni, J. P. Likforman, and C. S. Tan, "Development of mixed pitch grating for the optical addressing of trapped Sr+ ions with data analysis techniques," *Opt. Exp.*, vol. 31, no. 15, pp. 23801–23812, 2023, doi: 10.1109/JPHOT.2023.3281134.

[12] T. F. De Lima et al., "Machine learning with neuromorphic photonics," *J. Light. Technol.*, vol. 37, no. 5, pp. 1515–1534, Mar. 2019, doi: 10.1109/JLT.2019.2903474.

[13] S. Hooten, R. G. Beausoleil, and T. Van Vaerenbergh, "Inverse design of grating couplers using the policy gradient method from reinforcement learning," *Nanophotonics*, vol. 10, no. 15, pp. 3843–3856, 2021, doi: 10.1515/nanoph-2021-0332.

[14] X. Tu et al., "Analysis of deep neural network models for inverse design of silicon photonic grating coupler," *J. Light. Technol.*, vol. 39, no. 9, pp. 2790–2799, May 2021, doi: 10.1109/JLT.2021.3057473.

[15] Y. D. Lim, P. Zhao, L. Guidoni, J. -P. Likforman, and C. S. Tan, "Predictive modelling of optical beams from grating structure using deep neural network," *J. Light. Technol.*, vol. 42, no. 2, pp. 696–703, Jan. 2024, doi: 10.1109/JLT.2023.3319692.

[16] J. Wang, X. Li, J. Li, Q. Sun, and H. Wang, "NGCU: A new RNN model for time-series data prediction," *Big Data Res*, vol. 27, 2022, Art. no. 100296, doi: 10.1016/j.bdr.2021.100296.

[17] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2016, pp. 6000–6010.

[18] V. Goar, N. S. Yadav, and P. S. Yadav, "Conversational AI for natural language processing: An review of ChatGPT," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, pp. 109–117, 2023, doi: 10.17762/ijritcc.v11i3s.6161.

[19] Y. Li, X. Peng, J. Zhang, Z. Li, and M. Wen, "DCT-GAN: Dilated convolutional transformer-based GAN for time series anomaly detection," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3632–3644, Apr. 2023, doi: 10.1109/TKDE.2021.3130234.

[20] J. Zhao, C. Rong, C. Lin, and X. Dang, "Multivariate time series data imputation using attention-based mechanism," *Neurocomputing*, vol. 542, 2023, Art. no. 126238, doi: 10.1016/j.neucom.2023.126238.

[21] K. Li, K. Sasaki, K. Wake, T. Onishi, and S. Watanabe, "Quantitative comparison of power densities related to electromagnetic near-field exposures with safety guidelines from 6 to 100 GHz," *IEEE Access*, vol. 9, pp. 115801–115812, 2021, doi: 10.1109/ACCESS.2021.3105608.

[22] I. Demirtzioglou et al., "Apodized silicon photonic grating couplers for mode-order conversion," *Photon. Res*, vol. 7, no. 9, 2019, Art. no. 1036, doi: 10.1364/prj.7.001036.

[23] J. Fernando, "Relative strength index (RSI) indicator explained with formula," *Investopedia*, 2023, Accessed: Sep. 29, 2023. [Online]. Available: https://www.investopedia.com/terms/r/rsi.asp

[24] C. Jason, "What is EMA? How to use exponential moving average with formula," *Investopedia*, 2023, Accessed: Sep. 29, 2023. [Online]. Available: https://www.investopedia.com/terms/e/ema.asp

[25] J. Stanković, I. Marković, and M. Stojanović, "Investment strategy optimization using technical analysis and predictive modeling in emerging markets," *Procedia Econ. Finance*, vol. 19, no. 15, pp. 51–62, 2015, doi: 10.1016/s2212-5671(15)00007-6.

[26] N. Ketkar, "Feed forward neural network," in *Deep Learning with Python: A Hands-On Introduction*. Berlin, Germany: Springer, 2017, pp. 17–33.

[27] T. Ntakouris, "Timeseries classification with a Transformer model," *KEras*, 2021, Accessed: Sep. 29, 2023. [Online]. Available: https://keras.io/examples/timeseries/timeseries_transformer_classification/

[28] J. Wei, N. Kim, Y. Tay, and Q. Le, "Inverse scaling can become U-shaped," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2023, pp. 15580–15591, doi: 10.18653/v1/2023.emnlp-main.963.

[29] J. Hoffmann et al., "Training compute-optimal large language models," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 30016–30030.

[30] M. M. Bejani and M. Ghatee, "A systematic review on overfitting control in shallow and deep neural networks," *Artif. Intell. Rev.*, vol. 54, no. 8, pp. 6391–6438, 2021, doi: 10.1007/s10462-021-09975-1.

[31] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[32] Z. Li, J. Zhang, X. Yao, and G. Kou, "How to identify early defaults in online lending: A cost-sensitive multi-layer learning framework," *Knowl.-Based Syst*, vol. 221, 2021, Art. no. 106963, doi: 10.1016/j.knosys.2021.106963.

[33] Y. D. Lim, "Python code for transformer modelling (linked dataset)," *Github*, 2024, Accessed: Jan. 22, 2024. [Online]. Available: https://raw.githubusercontent.com/yd145763/Mixed_pitch_ML_full_data/main/Transformer_Photonics_TrainMixtureOnebyOne_farfield_horizontal_narrowed.py

[34] W. Wang, J. Zhang, Y. Cao, Y. Shen, and D. Tao, "Towards data-efficient detection transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 88–105, doi: 10.1007/978-3-031-20077-9_6.

[35] Y. Tay et al., "Scale efficiently: Insights from pre-training and fine-tuning transformers," in *2022 10th Int. Conf. Learn. Representations*, 2022, pp. 1–18.