Twisted Photons and Machine Learning

Benjamin Collins
Level 4 Project, MPhys Theoretical Physics
Supervisors: Dr R. Potvliege and Dr D. Maitre
Department of Physics, Durham University
Submitted: April 21, 2020

The use of the orbital angular momentum (OAM) of light as a carrier of information in free space optical communication creates the potential for higher bit rates and increased security in quantum key distribution. However, OAM shows high susceptibility to atmospheric turbulence. This paper seeks to simulate the distortion of OAM beams propagating over a long distance, using a Kolmogorov model of turbulence. Convolutional neural networks were then trained to distinguish individual modes from resulting intensity patterns, achieving accuracies of over 71% for up to 32 distinct OAM modes in high turbulence.

Contents

1. INTRODUCTION

Twisted photons are quanta of light carrying orbital angular momentum (OAM) [1]. One of the ways in which they are of interest stems from their potential use in optical communications systems as information carriers [2]. Specific attention is being paid to their use in free space optical communication (FSO): a method using optical wavelengths which depends on the atmosphere as a transmission medium [3]. Each mode, or a multiplex of modes, of angular momentum can be utilised to form a component of a communication alphabet, thus forming a complete orthonormal communication basis [2]. This has the potential for a theoretically unlimited increase in channel capacity; this is significant when compared to spin angular momentum (SAM), which constrains each photon to only communicating one bit [4]. The reason behind this is the infinite range of OAM available, which have an angular momentum contribution of $L_z = l\hbar$ per photon, with the azimuthal mode index $l \in \mathbb{Z}$ [1].

However, due to the nature of FSO passing through the constantly varying medium of the atmosphere, countering the effects of turbulence is critical to unlocking the full potential OAM-carrying photons provide [2]. Atmospheric turbulence (AT) distorts the wavefront of OAM beams in changing the refractive index of air, preventing the accurate classification of the beams mode. This contrasts SAM, which is relatively unaffected by AT. It is possible to distinguish L_z modes using experimental and computational techniques. Currently, due to the lack of an efficient physical sorter, use at a single-photon level is prevented [4]. This restricts its use in quantum key distribution, which due to the higher-order entanglement possible with OAM offers additional robustness against eavesdroppers and noise [5]. The entanglement allows twisted photons to reduce the error tolerance imposed when using SAM [6]. A computational technique currently utilised to successfully counter turbulence is the use of machine learning algorithms, which analyse the distorted wavefront patterns to discern the mode [3].

In 2014, the practical use of twisted photons was experimentally demonstrated with the transmission of 16 OAM superpositions 3 km over Vienna, using Laguerre-Gauss beams (LGB) with only a 1.7% error rate [7]. In 2016, the same group achieved transmission over 143 km between two Canary Islands, using up to $|l| = 3$ with an 82% average success probability, transmitting a message with "speed comparable to that of smoke-signalling" [5]. Both used superpositions of $+l$ and $-l$ known as petal modes carried by LGB's for increased phase stability this superposition causes and had the light analysed by artificial neural networks (ANN) at the receiving end [7]. Since this, many different machine learning methods have been used in a drive for improved accuracy, with an increased number of modes used. This has led to the wide use of convolution neural networks (CNN), which are more effective than other types of learning algorithms, such as K nearest neighbour (KNN) and back propagation artificial neural networks (BP-ANN) [8]. Further modifications of the raw images are used in networks, using combinations of convolutional and pooling layers to get feature maps before passing through fully connected layers.

Simulations of these experiments are frequently used, and have produced comparable results to those on physically generated and artificially or naturally distorted [9–11]. Most of the studies to date utilise a Kolmogorov power law distribution or a modified von Karman power density spectrum. These simulate AT numerically by generating phase screens to propagate the OAM beams through replicating the refraction index perturbations caused in turbulence [12].

2. THEORY

A simulation of an OAM beam propagating through AT and its treatment when received follow several key stages, as shown in Fig.1. Critical considerations for this simulation are: the beam used, the atmospheric turbulence, the method of propagation, and its analysis. The analysis for this work comprises of the use of the computational technique of neural networks. There are many types of OAM carrying beams which can be used: Hermite–Gaussian, Bessel–Gaussian, Ince–Gaussian, Mathieu–Gaussian and Laguerre–Gaussian beams [2].

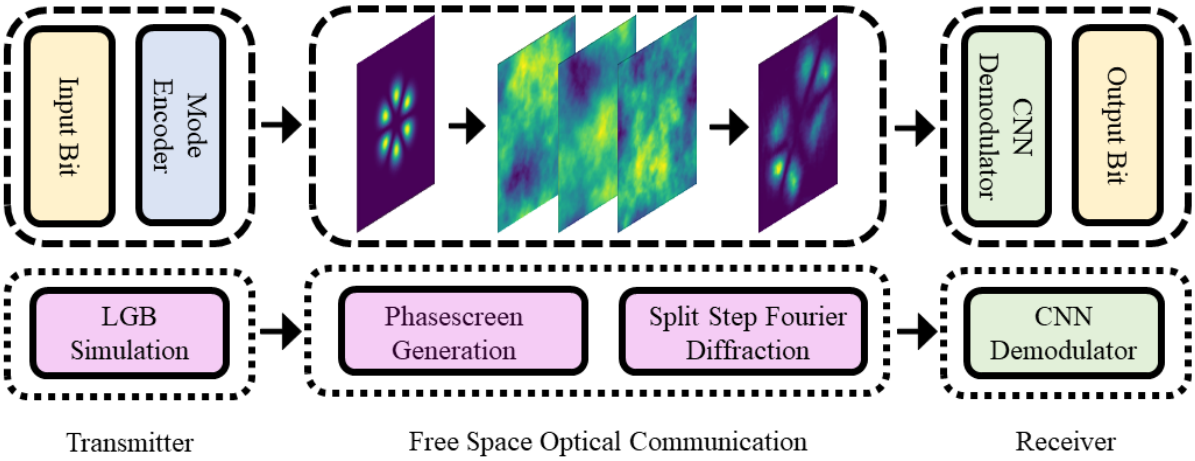

FIG. 1: A diagram of a free–space optical OAM based communication set-up with each key step: the transmission of the signal of a Laguerre–Gauss Beam (LGB), its propagation through turbulence, its reception and translation into working data using a convolutional neural network (CNN). Below this is the corresponding section of the simulation and their key areas.

2.1. Laguerre-Gauss Beams

For the simulations, LGB's were used to carry the OAM. This type of beam was chosen due to being physically realisable and the petal mode structures formed when two beams of opposite polarity are superimposed, which can be seen in Fig.2. These have a relative phase which is relatively unaffected by turbulence and produce a clear distinctive pattern to be analysed [7].

The complex electric component of a LGB, $U_{l,p}(r, \phi, z)$, can be written as

$$U_{l,p}(r,\phi,z) = \sqrt{\frac{2p!}{\pi(p+|l|)}}\frac{1}{w(z)}\left(\frac{r\sqrt{2}}{w(z)}\right)^{|l|} exp(-\frac{r^2}{w^2(z)} - \frac{ikr^2z}{2(z^2+z_R^2)})L_p^{|l|}\left(\frac{2r^2}{w^2(z)}\right) exp(-il\phi)exp(i(|l|+2p+1)\psi(z)), \quad (1)$$

written in cylindrical coordinates, where r is the radial distance from the propagation axis, ϕ is the azimuthal angle around the axis and z is the distance along the propagation axis. This expression contains the beam radius $w(z) = w_0\sqrt{1+(z/z_R)}$, w_0 is the beam waist, the

Rayleigh range is $z_R = \pi w_0^2/\lambda$, the wavenumber $k = 2\pi/\lambda$ with λ as the wavelength, $L_p^{|l|}(.)$ is the generalized Laguerre-Gauss polynomial of p (radial mode) and l (angular mode), and $\psi(z) = \arctan(z/z_R)$ is the Gouy phase. A notable feature in the LGB equation is the Gaussian phase profile $\exp(-il\phi)$ which is the key component allowing these beams to exhibit OAM [13]. In this paper, the radial mode p is set to 0 as it was not of critical interest.

To create the petal modes desired, waves with angular modes $\pm l$ are superimposed, producing a pattern with $2l$ peaks which act as clear indicators of the angular mode used. Equally weighted superpositions can be written as

$$U_{\pm l}(r, \phi, z) \propto (U_{+l}(r, \phi, z) + U_{-l}(r, \phi, z)), \tag{2}$$

which leads trivially to the intensity [14]

$$I_{\pm l}(r, \phi, z) \propto |U_{\pm l}(r, \phi, z)|^2 = |U_{+l}(r, \phi, z) + U_{-l}(r, \phi, z)|^2. \tag{3}$$

These equations are used in the simulation to generate the OAMs beam at the source, $z = 0$, as the complex value at each point on an $N \times N$ grid [13]. Each pixel on the grid is separated by a distance $\delta x = L/N$ where L is the physical distance of the $L \times L$ screen the simulation is computed on.

2.2. Propagation

To simulate the atmosphere's effect on an OAM beam travelling through it, the wave is propagated using a Fourier transform method, with first the phase of the wave perturbed by application of a planar phase screen (described in section 2.3.) and then propagated through spatial frequency space. The overall process consists of repeated steps of the Fresnel transfer function know as the split-step Fourier method (SSFM) where a single propagation is

$$U_l(r, \phi, z + \Delta z) = \mathfrak{F}^{-1}[exp(-i\frac{\kappa_x^2 + \kappa_y^2}{2k}\Delta z)\mathfrak{F}[U_l(r, \phi, z)e^{i\Theta}]], \tag{4}$$

where $exp(-i(\kappa_x^2 + \kappa_y^2)\Delta z/2k)$ is the transmission operator [15]. This is achieved in multiple iterations of the transfer function, where the total distance of the propagation is divided into

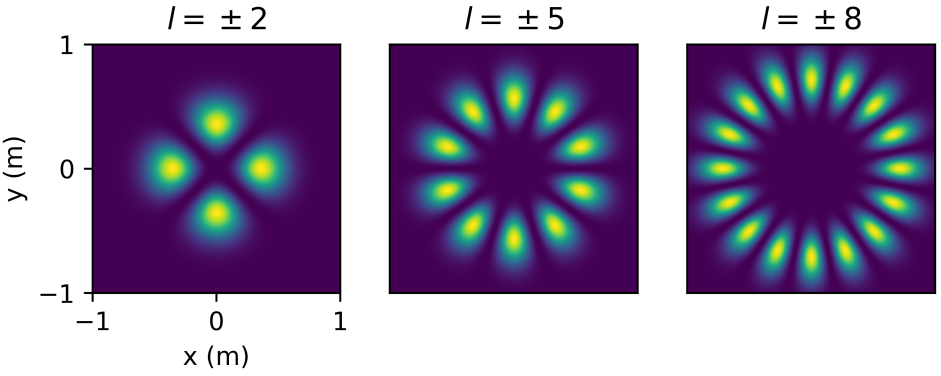

FIG. 2: Intensity plot of a selection of Laguerre–Gauss Beam petal modes caused by the superposition of two beams with orbital angular momentum modes, l, of opposite values.

equally spaced distances depending on the number of steps used. A true propagation through turbulence would require an infinitesimally small Δz, leading to infinite steps of propagation. However, as concluded by Chatterjee and Mohamed, "a trade-off between accuracy and processing time" is necessary to prevent producing unrealistic results [16], the details of which depend on the hardware used.

2.3. Atmospheric Turbulence

Due to the realities of FSO, AT presents a key hurdle in the use of OAM beams by destroying the orthogonality of OAM modes. AT is a random thermal process caused by the transfer of heat. Turbulence is modelled by Kolmogorov as consisting of cells of air with various temperatures and sizes which collapse down into smaller cells until they dissipate from viscosity [17]. These cells, having different densities, create a screen varying in space and time. This critically refracts the spatially dependent modes of OAM beams, causing crosstalk between modes [18]. Kolmogorov's model parametrises the air cells with having an outer scale L_0 and an inner scale l_0 [16].

The AT was simulated by generating phase screens to use in conjunction with Fourier transforms when propagating the OAM beam. This was achieved analytically with the widely used Kolmogorov power spectrum model. The power spectrum is applied to a calculation of the perturbation of the atmospheric refractive index; this perturbation is subsequently used to generate a phase screen.

For this simulation, the modified Kolmogorov turbulence model of Andrews [19] was used for the power spectrum Φ, described by

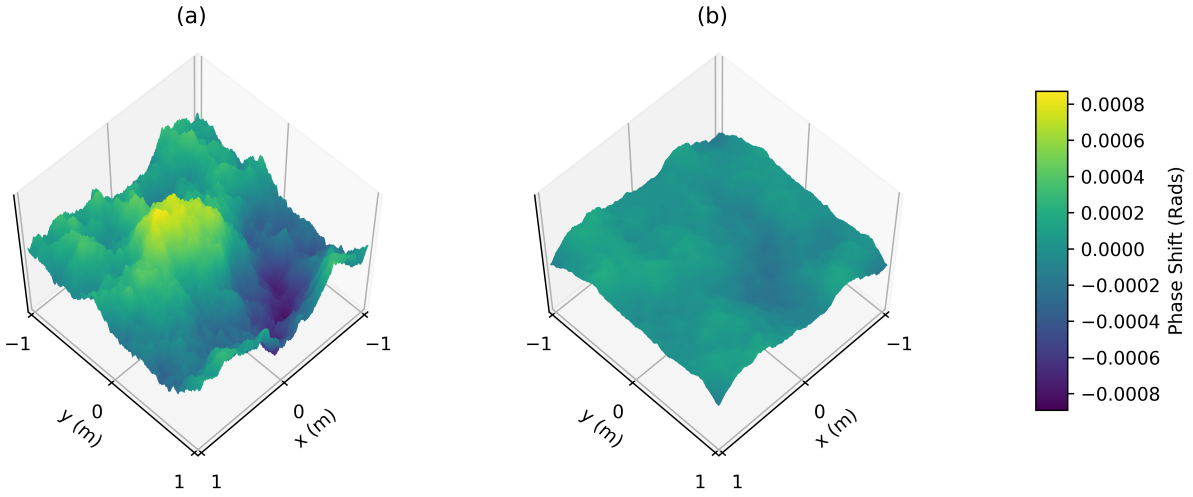

FIG. 3: A side by side comparison of phasescreens generated with the stated numerical method. (a) is generated with a C_n^2 of $10^{-14}m^{-2/3}$,(b)is generated with a C_n^2 of $10^{-15}m^{-2/3}$, both for a $2 \times 2\ m$ cross section.

$$\Phi(\boldsymbol{\kappa}) = 0.33C_n^2 \left(1 + 1.802\left(\frac{\kappa_x^2+\kappa_y^2}{\kappa_l^2}\right)^{1/2} - 0.254\left(\frac{\kappa_x^2+\kappa_y^2}{\kappa_l^2}\right)^{7/12}\right) exp(-\frac{\kappa_x^2+\kappa_y^2}{\kappa_l^2}) \left(\kappa_x^2+\kappa_y^2+\frac{1}{L_0}\right)^{11/6}, \quad (5)$$

where $\boldsymbol{\kappa} = (\kappa_x, \kappa_y, \kappa_z)$ is the spatial frequency, with its corresponding components in frequency space, $\kappa_l = 3.3/l_0$, and C_n^2 is the structure constant of the refraction index, a measure of the strength of turbulence [3]. This is then used to calculate the perturbation of the refractive index γ, which is defined as

$$\gamma^2(\boldsymbol{\kappa}) = \left(\frac{2\pi}{N\Delta\kappa}\right)^2 2\pi k^2 \Delta z \Phi(\boldsymbol{\kappa}), \quad (6)$$

where $\Delta\kappa$ is the grid interval in frequency space, and Δz is the physical distance of propagation the perturbation occurs over [20]. The random phase screen Θ is then generated using the refractive index perturbation, with

$$\Theta = \mathbb{R}[\mathfrak{F}^{-1}[\gamma C_{NN}]], \quad (7)$$

where c_{NN} is an array filled with random complex numbers with a Gaussian distribution, and $\mathfrak{F}^{-1}$ is the inverse Fourier transform [10]. This is realised in simulation using the fast Fourier transform algorithms for efficiency, examples of which can be seen in Fig.3.

This method is ideal for producing a single phase screen. However, each phase screen generated will be unrelated to the previous, which does not accurately represent the effect of the atmosphere over the short distances propagation occurs through. Consequently, the simulation utilises an autoregressive method of generating random sequential phase screens. Each Fourier mode has a complex autoregressive parameter, α, with $|\alpha| \leq 1$, this encodes wind speed and its direction. The modulus can be scaled to determine the quantity of phase from the initial screen conserved, with $|\alpha| = 1$ producing pure frozen flow and $|\alpha| = 0$ representing no change from the prior screen. A new phase screen can be generated from the previous with perturbation of the refractive index and a new complex random Gaussian array using

$$\Theta_i = \mathfrak{F}^{-1}[\alpha\tilde{\Theta}_{i-1} + \sqrt{1-|\alpha^2|}\gamma(\boldsymbol{\kappa})C_{NN,i}], \quad (8)$$

where $\tilde{\Theta}_{i-1}$ is the Fourier transform of the previous phase screen [21].

2.4. Convolutional Neural Networks

To analyse the outputs of the simulation, a CNN was chosen due to the widespread use and success it has had in the area of image recognition, specifically with OAM modes [8]. The structure of a CNN is designed to imitate a brain, comprising a network of interconnected layers shown in Fig.4. The basic components of this network are synthetic neurons called perceptrons. These take in a linear combination of the inputs z described by

$$z = w_0 + \boldsymbol{w_i} \cdot \boldsymbol{x_i}, \tag{9}$$

where w_0 is the bias, w_i is a vector of all the input weights, x_i is a vector containing the input values, and i enumerates over each connected neuron from the previous layer. This combination z is then passed through an activation function to get the output which can then be used for a feedforward step as an input for the next layer. For the case of multi-class classification presented by this problem, the activation function used for the final layer of perceptrons is the softmax function. This function is written as

$$s_i(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}, \tag{10}$$

where k is the number of classes. This outputs the probability that the input data belongs to that class, with the highest valued $s(z_i)$ being the prediction of the correct class by the network [22]. Other activation functions are used for different parts of the network such as sigmoid or Rectified Linear Unit (ReLU) [9] functions.

For a network to be useful it has to first be trained, enabling it to find values of weights and biases which produce correct results. This involves splitting the input data and its corresponding labels into two sets, one for training and another for validation. The training data is first put through the network which has random values for its parameters. These are refined by minimising a loss function J, which typically takes into account the difference between prediction

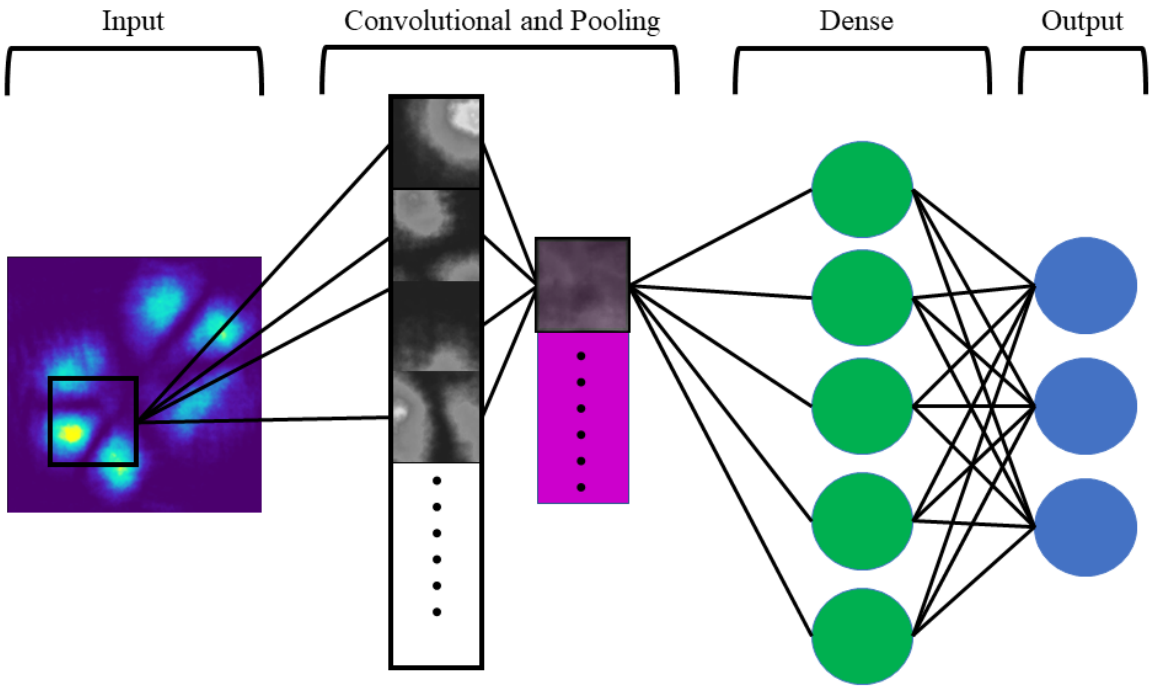

FIG. 4: Set up of a typical convolutional neural network structure used for image classification with convolutional and pooling layers leading into dense. Connected points represent a filter or neuron and the lines an input. Note only one layer of each type is shown and a small section of the convolutional/pooling layers for clarity.

and correct label of the input. Various methods can be used to minimise the loss function such as gradient descent, stochastic gradient descent or Newtonian second-order method. The loss function used for CNN's used in this paper is the cross-entropy, described by

$$J = -\sum_{c=1}^{M} y_{o,c} \log\left(p_{o,c}\right) \tag{11}$$

where M is the number of classes, y is a binary indicator for if predicted class c is the correct classification for true class o, and p is the CNN's output probability that the predicted class is the correct class [23]. The validation set of data is then used to observe how well the network has 'learned'; this is done by comparing the error rate of the two sets for each epoch of the network, where each epoch is the full application of the training set through this learning process to adjust the parameters of the network. From this comparison, adjustments can be made to the hyperparameters to improve the learning algorithm's efficiency. The rate at which the parameters are changed is determined by the learning rate η, a hyperparameter used to regulate the speed of learning. To facilitate the learning of the CNN, the ADAM algorithm will be used for optimization, chosen as it is "computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters" [24]. This is a stochastic gradient descent method, utilising individual adaptive learning rates for each parameter using first and second-order moments of the gradient. Another hyperparameter to be considered is the size of the training set, as too small a size can lead to overfitting to that particular set.

In order to get the images into the correct format to be processed by a neural network, they need to be flattened into a vector shape with a length of the number pixels in the image. For high-resolution images, this leads to a high quantity of input neurons. If a dense layer, where all the neurons from the previous layer are connected to every neuron in the dense layer, was connected to this high-resolution input, the number of parameters which would be required become computationally infeasible to apply learning algorithms to. To counter this, convolutional layers are utilised. These are layers in which a small kernel (typically 3×3 or 5×5 in OAM studies [12, 20]) is applied across the imputed image in order to detect features. Each convolutional layer has the parameter of the number of filters used in the kernel with individual filters acting as a different linear combination of the values input to the kernel. These produce a feature map summarising presence in input for each particular filter, which is then passed through an activation function. However, an issue with feature maps is their high sensitivity to small movements in feature location; to counter this, a pooling layer is commonly applied directly afterwards [25]. This works in a similar method to a convolutional layer; a small kernel is applied over each feature map in a non-overlapping method, calculating either the average or maximum value for each patch. This produces an overview of the features detected from the image, giving the network invariance to local transition. Another consideration for a neural network is to add dropout into the system between certain layers. This is a form of trainable parameter regularisation, it forces the weights to take smaller values and reduces overfitting when using a small number of training examples. This is realised by reducing a fraction of the outputs from a layer to 0 [26]. After application of these layers, the reduction in parameters is sufficient to allow for the use of dense layers to make the final label prediction.

3. SIMULATION PARAMETERS

TABLE I: The parameters used in the simulations for free space optical communication using Laguerre–Gauss beams with OAM modes.

Constant	Value	Constant	Value
Pixel Width (N)	512	Waist Radius (w_0)	0.13 m
Screen Width (L)	2 m	Inner Scale of Turbulence (l_0)	0.01 m
Wavelength (λ)	500 nm	Outer Scale of Turbulence (L_0)	50 m
C_n^2 for Weak Turbulence	$1.0 - 9.0 \times 10^{-15}$ m$^{-2/3}$	Total Propagation Distance	100 km
C_n^2 for Medium Turbulence	$0.9 - 1.6 \times 10^{-14}$ m$^{-2/3}$	Propagation Step Distance (Δz)	1 km
C_n^2 for Strong Turbulence	$1.6 - 2.0 \times 10^{-13}$ m$^{-2/3}$		

The constants used throughout the simulation are detailed in Table I. The total distance used was chosen due to the success of the 143 km experiment with a similar magnitude. The turbulence scales and parameters of the photons were selected in consideration of those physically measured and used [5] along with those utilised in previously designed simulations of this nature [9]. The range of structure constants of the refraction index, C_n^2, were chosen from the results they produced when used in the propagation. Weak, medium and strong turbulence were chosen to produce: minimal distortion to the intensity pattern, high distortion while remaining recognisable, and practical impossibility to distinguish modes, respectively. These were also compared to experimental values for validity [20].

Increasing petal modes from $l = 0$ to $l = \pm n$ were used, where n is 1 less than the number of modes the network is designed to classify. This was done to support the large number of modes this work aimed to have classified by networks and to minimise any loss in clarity caused by the limitation presented by the image resolution. Patterns up to $l = \pm 31$ were used, higher modes being beyond the practical scope of this paper.

Following the decision to have the total communication distance of 100 km, a suitable step size for the split-step Fourier propagation was required for the simulation. Previous works such as that by Chatterjee and Mohamed used a Δz of 10 m as they were simulating shorter total

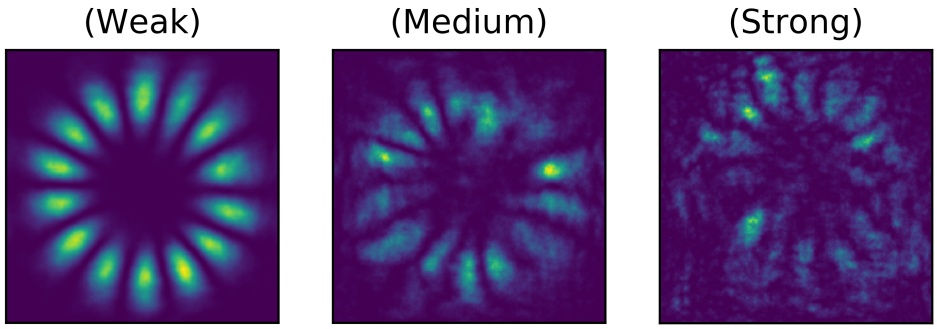

FIG. 5: Results of propagation on the $l = \pm 3$ mode for the C_n^2 ranges chosen, as detailed in Table I. All were propagated 100km with a dz of 1000m

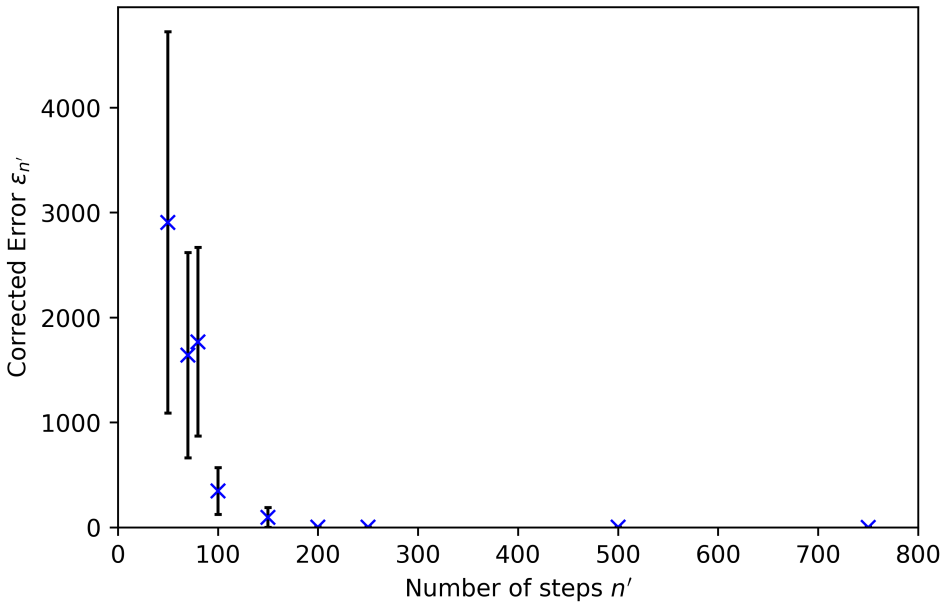

FIG. 6: Corrected error of propagation to number of steps used for the propagation.

distances [16]. However, this would require 10000 steps for each wavefront intensity pattern generated in this work. Computing power necessitated a more pragmatic approach; an investigation was made to determine an optimal number of propagation steps while minimising errors large steps cause in turbulence approximations. This was achieved using a method developed by Mitchell [27]. Beams are generated with constants in Table I to be propagated over 100 km using $n = 1000$ steps and phase screens, the maximum feasible quantity, to produce I_{1000}. The sum of all the phase screens used for this propagation Ψ is then calculated. Additional intensity patterns $I_{n'}$ are then computed, having propagated through $n' < n$ uniform phase screens of magnitude Φ/n' for each n' being tested. The corrected error, $\epsilon_{n'}$, for each intensity pattern can be calculated with

$$\epsilon_{n'} = \frac{I_{n'} - I_{1000}}{I_{1000}} - \epsilon_{1000} \tag{12}$$

to allow for comparison. ϵ_{1000} is the correction factor required to scale the error as recreating propagation with the same number of screens using Φ/n' produces errors in the intensity pattern. From the results of this investigation shown in Fig. 6, you can see that the corrected error $\epsilon_{n'}$ tends towards 0, reaching this at 200 and maintaining no major change in magnitude. Error bars for this figure are found using the standard error from $N = 10$ repeats of this investigation where the standard error, $\sigma_{corrected}$, is

$$\sigma_{corrected} = \frac{\sigma_\epsilon}{\sqrt{N}} \tag{13}$$

where σ_ϵ is the standard deviation of the corrected error. Combined with the time data required to generate, the optimal number of propagation steps was determined to be approximately 100, setting $\Delta z = 1$ km.

To facilitate sufficient quantities of training data to get an accurate CNN, multiple training images can be generated from one of the computationally expensive simulated results. This

is common practice for image recognition networks by use of the process of data augmentation. Most obvious is the use of horizontal and vertical flips or the combination of both, which produces the same result as if the random arrays used in the simulation were similarly flipped but with less computing. Other methods of data augmentation such as zoom and random noise transformations were avoided due to the dependence of the intensity pattern width on turbulence and the fact random noise should be added from the random phase screens [5]. Although an argument could be made for potential CCD errors occurring in the system during a practical experiment, this would be negligible, considering the size of most images, and of minimal impact on a CNN. It would only be a relevant consideration in the sensitive case of single-photon communication.

The structure of each CNN used is shown at the start of the relevant section. Structures were informed by standard models used for image recognition and those which have been used for the specific task of distorted OAM mode classification [10]. The lack of complex structure in most cases was decided due to the comparable success of shallow CNNs in this field compared to deeper neural networks [12]. Additionally, network design was informed by computational limitations: in generating sufficient quantities of training data and running the learning algorithms. The larger numbers of parameters found in complex networks, and the training of them, require more data. Previous complex architectures often utilised computer networks with a greater quantity of processing power, which was unavailable for this work. The construction of these CNN's was achieved with the use of Tensor Flows Keras API module in python, a library used to facilitate machine learning.

Each range of turbulence was used to train a unique network within the predefined structures. This was done to ensure the network adapted to the different patterns caused by distortion from each range of turbulence. To gain the maximum amount accuracy from the data, the networks were trained for 10 epochs as they stopped showing signs of improvement at this point. However, in the case of strong turbulence, the training of each networks was run for an additional 5 epochs as the accuracy still showed improvement after the 10 epochs of training used for the other turbulence levels. Due to limitations on computational power each network could only be trained once.

4. RESULTS

4.1. Neural Networks

1. CNN-1

Table III presents an overview of the structure for CNN-1. A combination of a single convolution, pooling, and dense layers was used as this is the form of a shallow convolutional neural network which has been successfully utilised to classify 100 modes [12]. The convolutional layer is chosen to help provide translational invariance and the dense layer to facilitate additional sorting of the key features detected. The network layers of CNN-1 were reduced to one of each type due to limitations in available computational time.

TABLE II: Full results of CNN-1 testing, showing the maximum accuracy the validation set achieved by the end of the networks training. All accurate to ± 1 on the last decimal place.

Turbulence	4	8	16	32
Weak	1.000	1.000	1.000	0.997
Medium	1.000	0.996	0.982	0.950
Strong	0.947	0.820	0.705	0.644

TABLE III: Details of CNN-1 structure and associated parameters used in descending order of operation. Y is the number of modes tested for by the network.

Layer	Neurons / Filters	Kernel	Activation Function	Trainable Parameters
Convolution	16	5×5	ReLU	416
Max Pooling		2×2		0
Dropout			0.2	0
Dense	32		ReLU	8388640
Dense	Y		Softmax	33Y

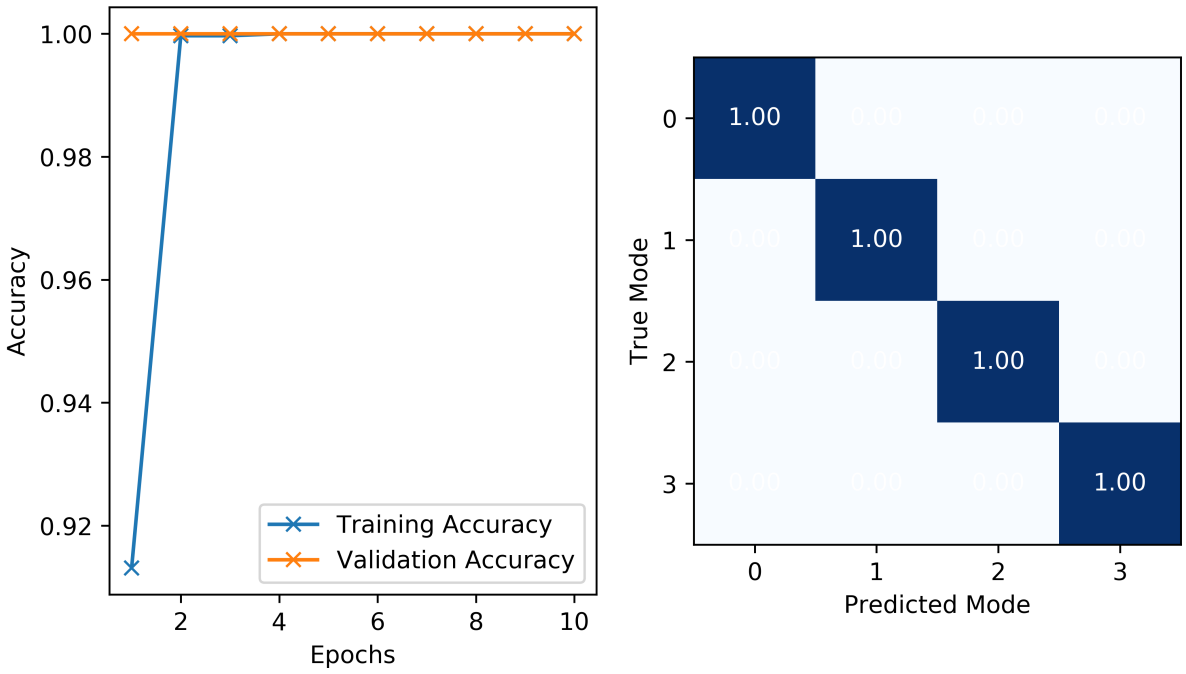

FIG. 7: Results for the training of CNN-1 for 4 modes in medium turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The confusion matrix on the right shows the proportion of predictions made for each mode using the validation data after training for 10 epochs. Zeros removed for clarity.

As expected for 4 modes in the case of medium turbulence, the CNN reaches 100% accuracy by the 4th epoch with minimal struggle for the training set; crucially 100% accuracy occurs within the first epoch of training for the validation data. The result, shown in Fig.7, is unsur-

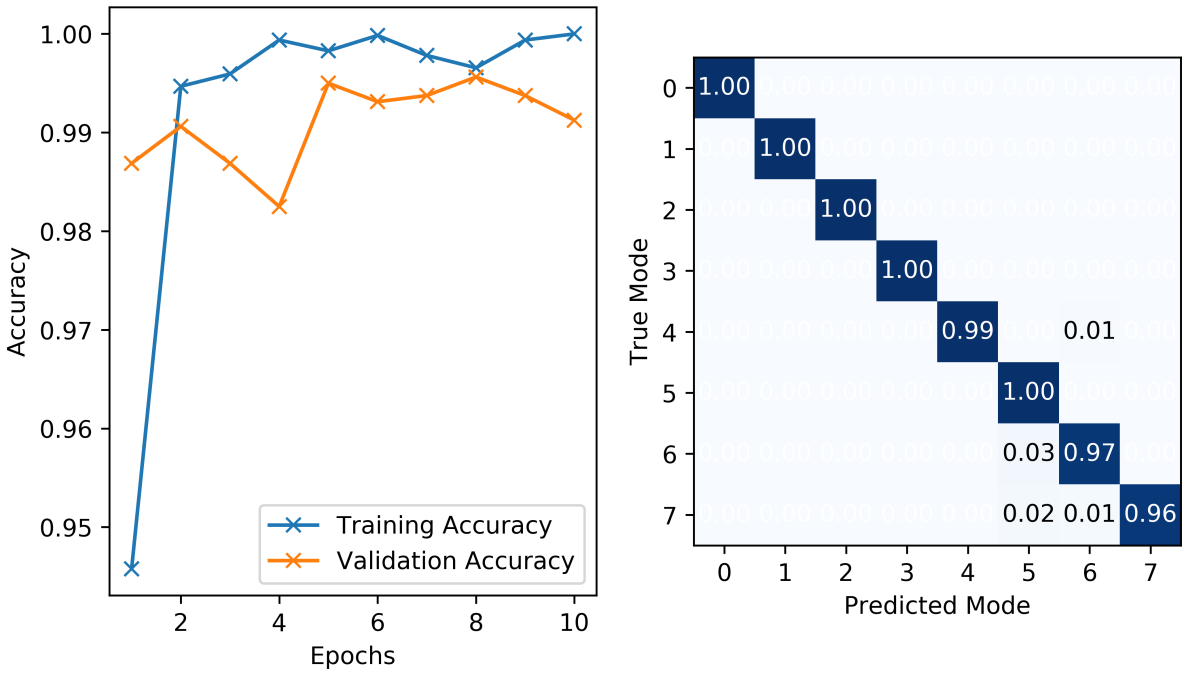

FIG. 8: Results for the training of CNN-1 for 8 modes in medium turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The confusion matrix on the right shows the proportion of predictions made for each mode using the validation data after training for 10 epochs. Zeros removed for clarity.

prising due to the low number of modes the network is required to classify, making each mode wavefront pattern distinctive. As the network is fully accurate in its predictions, the confusion matrix is in a purely diagonal form. Compared to SAM beams there is instant improvement within the medium turbulence regime in bits sent per signal.

However, when the network was trained using the strong simulations, a significantly lower accuracy of 0.947 ± 0.001 was achieved. As this effect occurs at the lowest number of modes trained on, the high significance of turbulence in reducing the effectiveness of a network can be inferred. Crosstalk in identification between modes was found in all cases except for modes $l = 0$ and $l = \pm 3$, which had the highest and lowest prediction accuracy respectively.

As would be expected, the increase in the number of modes the network was required to classify from 4 to 8 caused a reduction in accuracy, shown in Fig.8. This reduction in maximum accuracy of 0.996 ± 0.001 is expected; it illustrates the limiting effects of AT on OAM based communication, with the effects of AT impacting on signals carrying only 3 bits. However, this does not place limitations on its practical implementation; the quantity of error could be reduced with a larger training dataset or overcome with the use of information reconciliation algorithms. The training of the model shows a strong convergence between training and validation accuracy. The decrease in validation accuracy in contrast to the training accuracy during the final 2 epochs suggests overtraining. Clearly shown in the confusion matrix for this particular model, crosstalk between modes occurs only within the higher 4 modes. The lower 4 modes are recognised with full accuracy, as exhibited by the network trained solely on those modes.

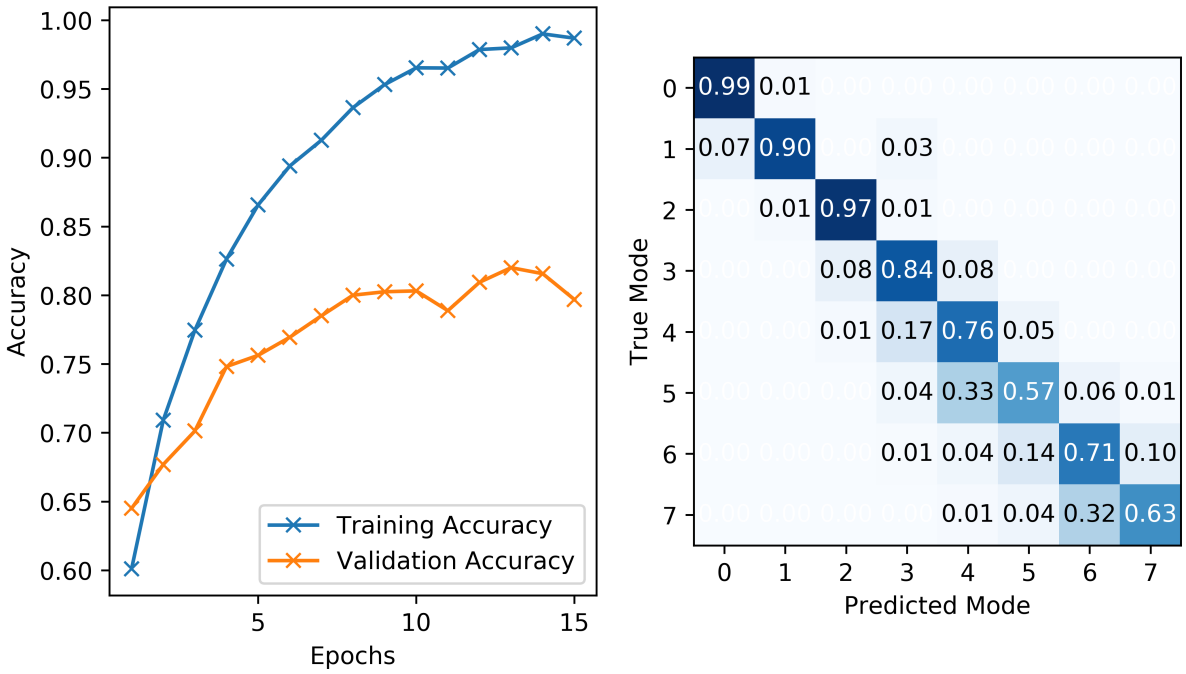

FIG. 9: Results for the training of CNN-1 for 8 modes in strong turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The confusion matrix on the right shows the proportion of predictions made for each mode using the validation data after training for 15 epochs. Zeros removed for clarity.

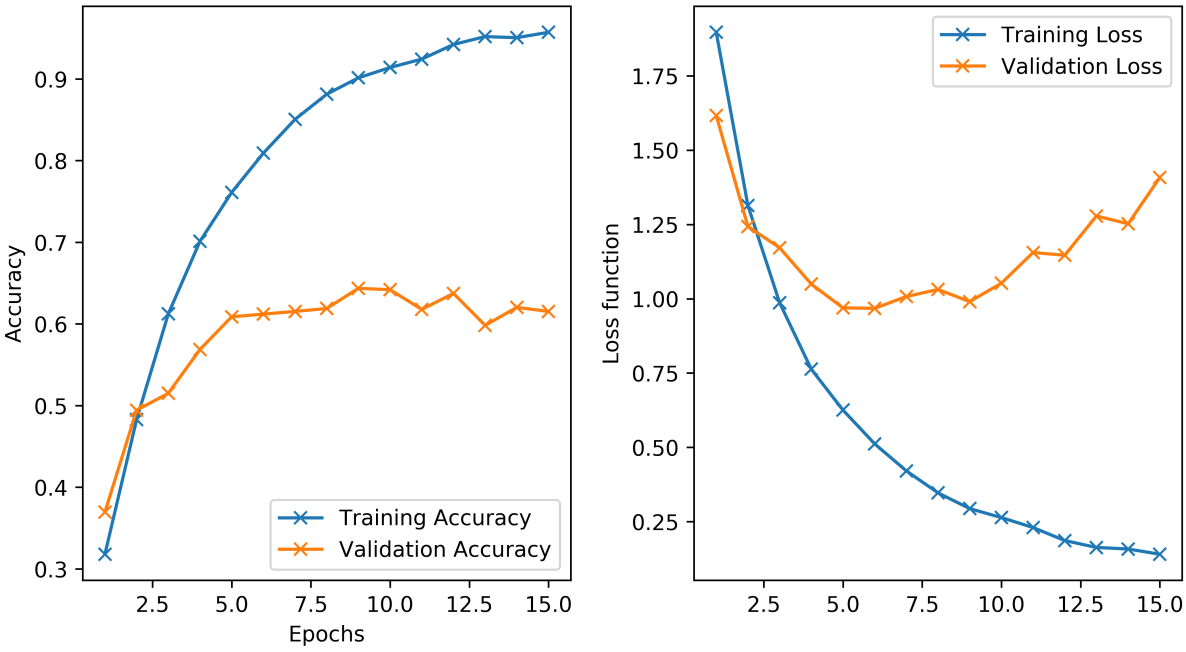

FIG. 10: Results for the training of CNN-1 for 32 modes in strong turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The graph on the left shows the corresponding loss for the training and validation data.

In direct contrast are the results presented in Fig.9 for 8 modes in strong turbulence, with the accuracy obtained in the validation set of 0.820 ± 0.001 being below that achieved in lower turbulence. The deviation between the accuracy for the training and validation data in training is significant with a difference in the accuracy of 0.265 ± 0.002 between them when the model is optimised within the training process. Improvement in the network could be made with access to greater resources. With access to more training samples, the separation between training and validation curves has the potential to decrease. As both types of accuracy show an increasing trend, this network could additionally improve with increased training time.

The confusion matrix in Fig.9 shows the extent of the misclassification of certain modes, with the mode of $l = \pm 5$ being correctly predicted 57.0% of the time. It is interesting to note that the accuracy for each mode is, approximately, inversely correlated to the modulus of the azimuthal mode index used. This trend indicates that the network struggles to identify larger numbers of highly distorted petals. In a majority of the cases for this set of modes, there is a lower proportion of correct predictions for odd-numbered modes compared to their surrounding even numbers. Another feature of the network revealed by the confusion matrix is its bias to predict a lower mode than the mode presented; in all modes for this network, the majority of the predicted modes were for a mode lower than the true mode. This bias isn't due to the order the network receives the training data, as it is randomised for each epoch. For all cases, an incorrect mode prediction occurs most frequently within 1 mode of the true mode, with the frequency of incorrect prediction decreasing with the difference between modes.

For the most extreme case tested – 32 modes in strong turbulence, the reduction in accuracy and increase of loss function, shown in Fig.10, shows that the limits of the network with these resources are being reached. The accuracy curve of this case has the same form as that for the 8 mode case, displayed in Fig.9, but with more exaggerated features, suggesting a similar limit. The validation accuracy stops showing an increase in accuracy and flattens out after 5 epochs; after this the network is over adapting to the training data. In the graph of the loss function, the increase in validation loss that occurs in the later epochs shows that the CNN was learning less about wavefront intensity patterns for each mode, instead – learning to recognise the individual training data images.

Contrastingly, for 32 modes in medium turbulence, a much higher validation accuracy 0.950 ± 0.001 is achieved compared to 0.644 ± 0.001 from strong turbulence. This combines with a much smaller divergence between training and validation accuracy of 0.040 ± 0.002 in medium turbulence compared to 0.314 ± 0.002 in strong turbulence. These differences highlight the strong effect that turbulence has in reducing the accuracy of communications, especially at higher mode numbers.

Overall from the results for this CNN, it can be seen that in low turbulence situations the network performs perfectly well, with 100% accuracy for 16 modes and below. With further tests these could be used successfully for high-quality data communication. The higher turbulence networks can be seen to overtrain with a large divergence between the training and validation accuracies, suggesting the quantity of training data is the limiting factor. Therefore more data is required for networks used for communication in higher turbulence.

2. CNN-2

TABLE IV: Full results of CNN-2 testing, showing the maximum accuracy the validation set achieved by the end of the networks training. All accurate to ± 1 on the last decimal place.

Turbulence	4	8	16	32
Weak	1.000	1.000	1.000	0.999
Medium	1.000	0.994	0.992	0.982
Strong	0.963	0.874	0.813	0.714

TABLE V: Details of CNN-2 structure and associated parameters used in descending order of operation. Y is the number of modes tested for by the network.

Layer	Neurons / Filters	Kernel	Activation Function	Trainable Parameters
Convolution	8	5×5	ReLU	208
Max Pooling		2×2		
Dropout			0.2	
Convolution	16	3×3	ReLU	1168
Max Pooling		2×2		
Dropout			0.2	
Convolution	32	3×3	ReLU	4640
Max Pooling		2×2		
Dropout			0.2	
Dense	32		ReLU	1048608
Dense	32		Softmax	33Y

For the structure of this particular CNN, presented in Table V, a multitude of convolutional layers were used, with each layer having more filters than the previous. This was chosen as a result of previous successful studies utilizing this structure with dense layers at the end [3, 8, 9, 20]. It was designed to have a large number of convolutional layers to better counter the effects of any translation of the image and better recognise features. The dense layer had the same size as that used in CNN-1 due to the reduced inputs and thus fewer trainable parameters, retaining the focus on the feature recognition provided by convolutional layers.

In all cases for the training of CNN-2, the training and validation accuracy curves maintain a higher level of convergence, leading to higher accuracies than CNN-1 overall. The case of 4 modes in strong turbulence is a clear example, shown in Fig.11, as it improves in accuracy by $1.6 \pm 0.2\%$. Despite the abnormal early section in the first 4 epochs of training, the validation accuracy maintains values close to the accuracy of the training data, further converging once the learning has stabilised into a more typical structure. Interestingly, the confusion matrix for this network has a feature wherein the lower two modes of LGB patterns have a smaller proportion of correctly predicted modes compared to the higher two. This is striking as in all the cases with more modes for the same turbulence, the $l = 0$ and $l = \pm 1$ modes are predicted with higher accuracy, highlighting the variability caused by the limitations of this report.

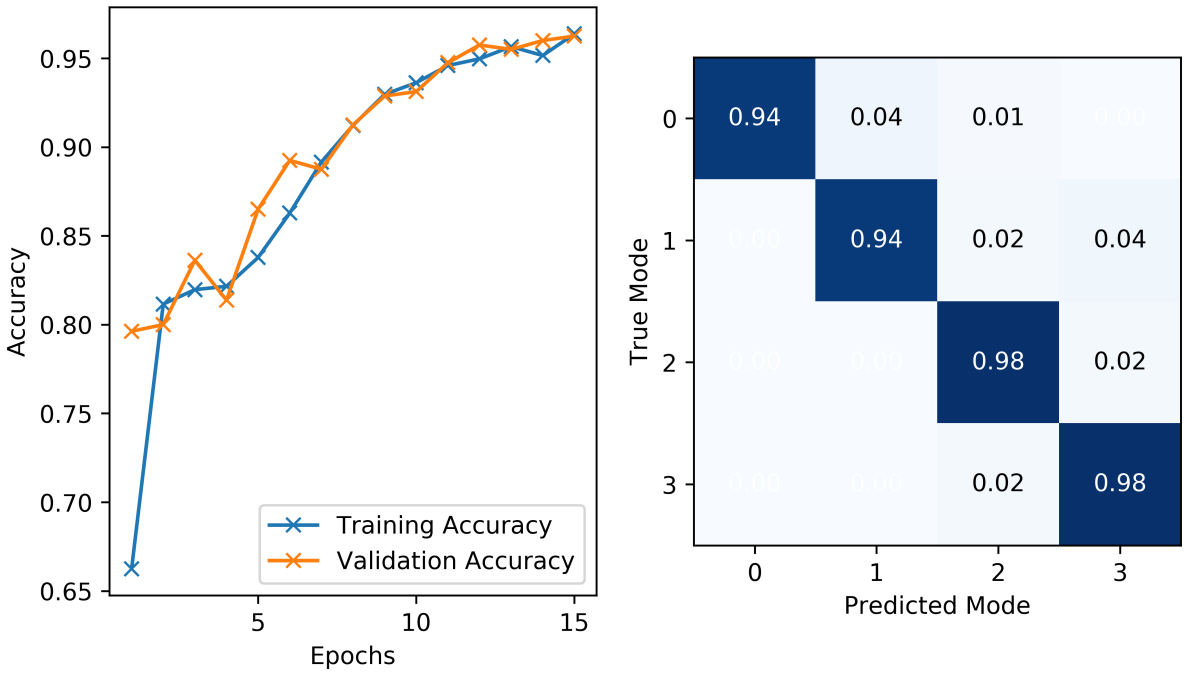

FIG. 11: Results for the training of CNN-2 for 4 modes in strong turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The confusion matrix on the right shows the proportion of predictions made for each mode using the validation data after training for 15 epochs. Zeros removed for clarity.

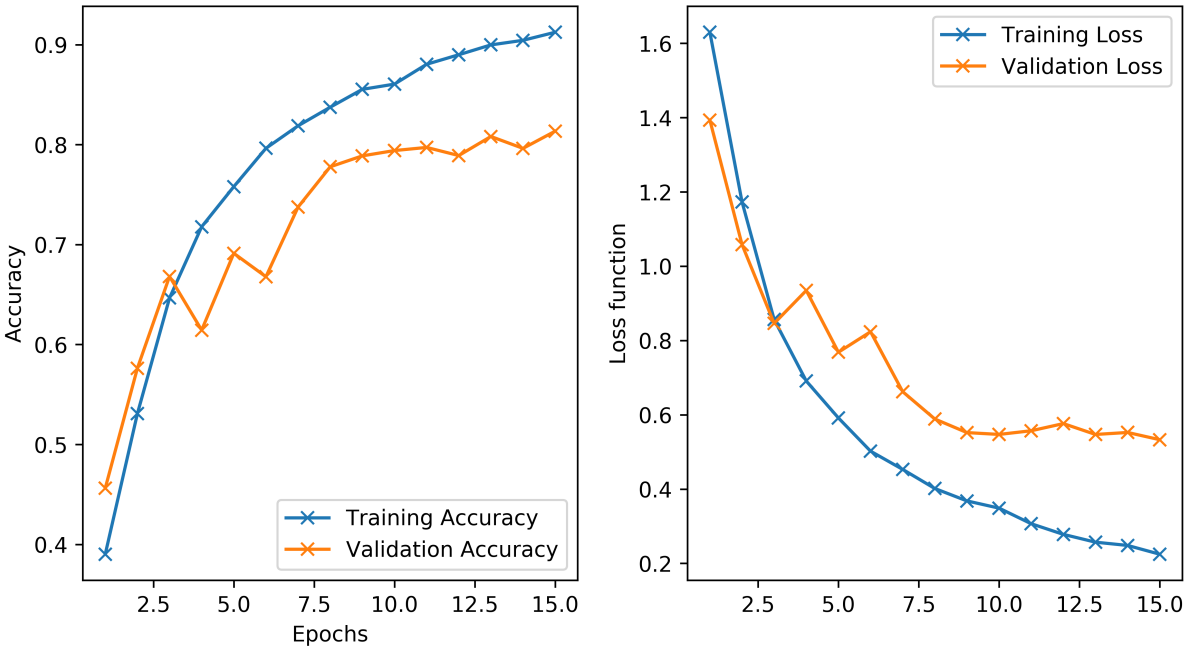

FIG. 12: Results for the training of CNN-2 for 16 modes in medium turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The graph on the left shows the corresponding loss for the training and validation data.

The training curves of CNN-2 for 16 modes in strong turbulence in Fig.12 show another clear example of how a higher accuracy was achieved. The resulting validation accuracy identifies $10.8 \pm 0.1\%$ more of the patterns correctl,y compared to the same set-up with CNN-1. The peak training data accuracy for CNN-2 of 0.912 ± 0.001 is lower than that of CNN-1 at 0.970 ± 0.001. Critically, this suggests that the higher number of convolutional layers enables the recognition of more key features of the LGB patterns during learning, and substantially reduces overtraining to recognise an individual piece of data. Further evidence of this can be identified in the loss function curves of the data, which plateau off and show a minimal divergence in higher epochs. This indicates overtraining only occurs in the final epochs, not to the detriment of validation accuracy.

On comparison of the performance of CNN-2 at classifying 32 modes in medium and strong turbulence, presented in Fig.13, the hurdle that highly distorted LGB patterns present to pattern recognition is evident. In the medium turbulence case, a steeper curve for both datasets can be observed as the network learns at a rapid rate. This directly contrasts the network structure learning for the case of strong turbulence, which shows a shallower curve with a slower rate, as the key patterns are more distorted and difficult for the algorithm to identify. However, the validation accuracy achieved by the curves is higher in both cases compared to those produced by CNN-1, despite having a lower peak training accuracy. The most surprising aspect of the curves is that the validation accuracy in medium turbulence is consistently greater than the training accuracy. This is suggests that the increase in the number of convolutional layers helped in correctly identifying key features as this does not occur in the CNN-1 data.

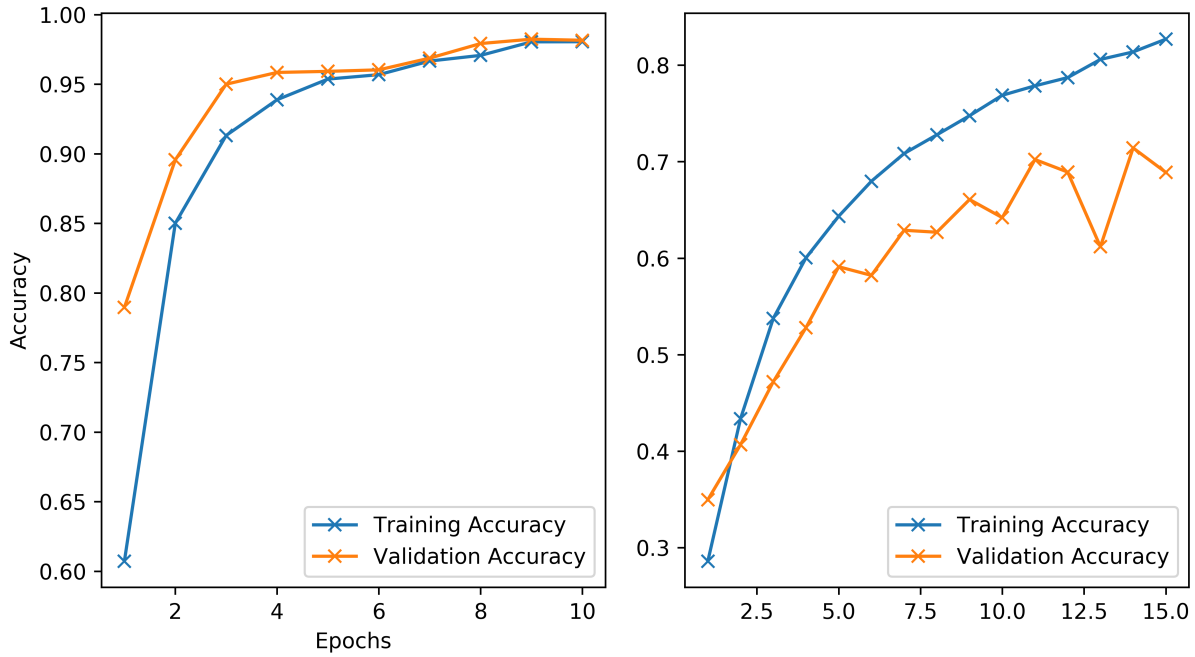

FIG. 13: Results for the training of CNN-2 for 32 modes. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch in medium turbulence. The graph on the right shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch in strong turbulence.

In summary, from looking at the results of this network structure in Table IV, it is clear that CNN-2 outperforms CNN-1 in the large majority of cases, despite having fewer parameters overall. Examining this result reflects the power that convolutional layers provide in the area of pattern recognition. These convolutional layers require fewer epochs to maximise results and are less susceptible to overtraining, as shown in Fig.12 and Fig.13 which exhibit a higher convergence between training and validation data compared to the results for CNN-1. However, these networks would still benefit from additional data to reduce disparity between the accuracy by providing a more diverse range of wavefront patterns to be analysed. These would provide data for additional key features which could be used by the network in identification.

3. Shallow Neural Network

TABLE VI: Full results of shallow neural network testing, showing the maximum accuracy the validation set achieved by the end of the networks training. All accurate to ± 1 on the last decimal place.

Turbulence	4	8	16	32
Weak	1.000	1.000	1.000	0.997
Medium	1.000	0.994	0.982	0.932
Strong	0.921	0.765	0.539	0.361

TABLE VII: Details of shallow neural network structure and associated parameters used in descending order of operation. Y is the number of modes tested for by the network.

Layer	Neurons / Filters	Kernel	Activation Function	Trainable Parameters
Dense	128		ReLU	8388736
Dense	Y		Softmax	129Y

The structure for this network, outlined in Table VII was designed to test the pattern recognition capabilities of a shallow neural network: a network consisting of only a single dense hidden layer. This network does not have the feature recognition capabilities provided by convolutional layers. However, it was expected to achieve high accuracies on the lower two turbulence ranges due to the lack of major distortion at these levels. The number of neurons used for the single dense layer was 128; this resulted in the quantity of trainable parameters being within the same order of magnitude as CNN-1. Due to the continued increases in accuracy, the training of these networks was left to run for 15 epochs in all cases to allow for their full potential within the limitations of the computational resources available.

As would be assumed, this less complex neural network did not perform as well in the strong turbulence regime. On examination of the confusion matrix in Fig.14, the extent can be observed for the case of 16 modes in strong turbulence. There are several key features to note; firstly the confusion matrix shows a low point in accurate prediction for the case of $l = \pm 11$ of 0.210, with comparable proportions being incorrectly predicted for the surrounding modes, this level of low accuracy is highly prevalent in the upper modes. Secondly, this mode is also

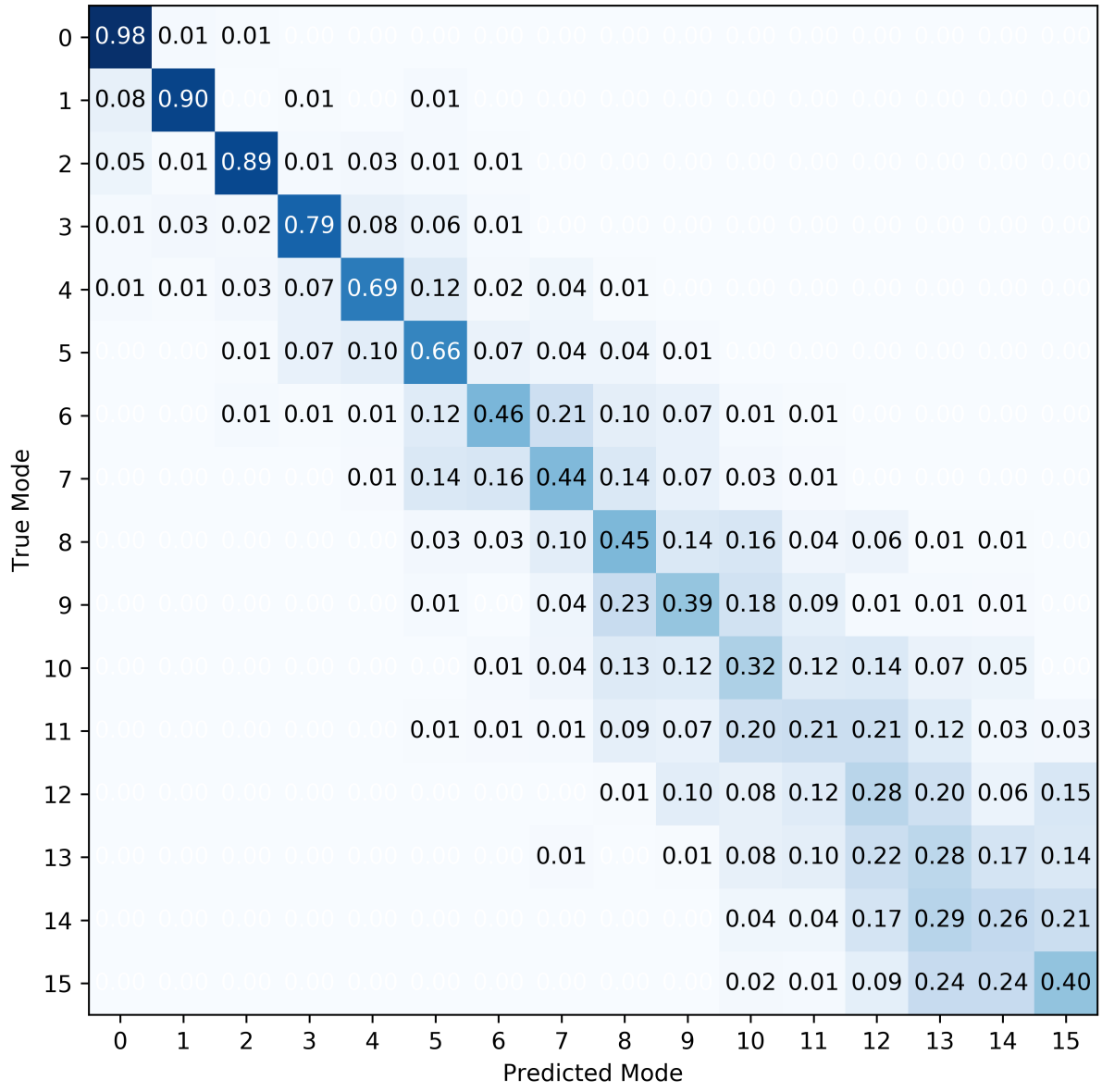

FIG. 14: Results for the training of shallow neural network for 16 modes in strong turbulence. The confusion matrix shows the proportion of predictions made for each mode using the validation data after training for 15 epochs. Zeros removed for clarity.

one of two modes that have predictions for it up to 6 modes distant, showing the large spread of misclassification that can occur in this type of network. Finally, it is critical to draw attention to the case of $l = \pm 14$ for which a higher proportion of data was predicted to be mode $l = \pm 13$ than the correct mode, highlighting the limitations associated with a shallow neural network. This may be caused by the limited resolution of the images used; a higher resolution would provide more distinctive key nodes.

Another example which shows the limitations of the shallow neural network is the case of 32 modes in medium turbulence, as shown in Fig.15. This has a significant divergence of training and validation accuracy curves of $5.3 \pm 0.2\%$, large in comparison to values of $4.0 \pm 0.2\%$ and $0.1 \pm 0.2\%$ for CNN-1 and CNN-2 respectively. This is a result of the network adapting to

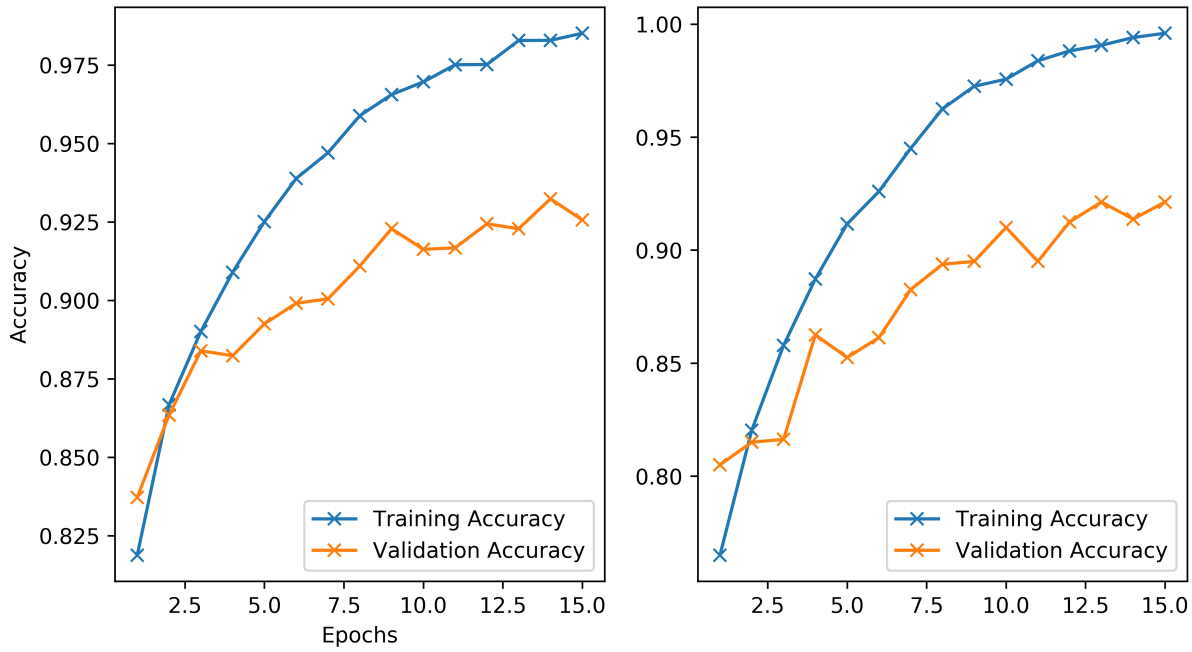

FIG. 15: Results for the training of shallow neural network. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch in medium turbulence for 32 modes. The graph on the right shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch in strong turbulence for 4 modes.

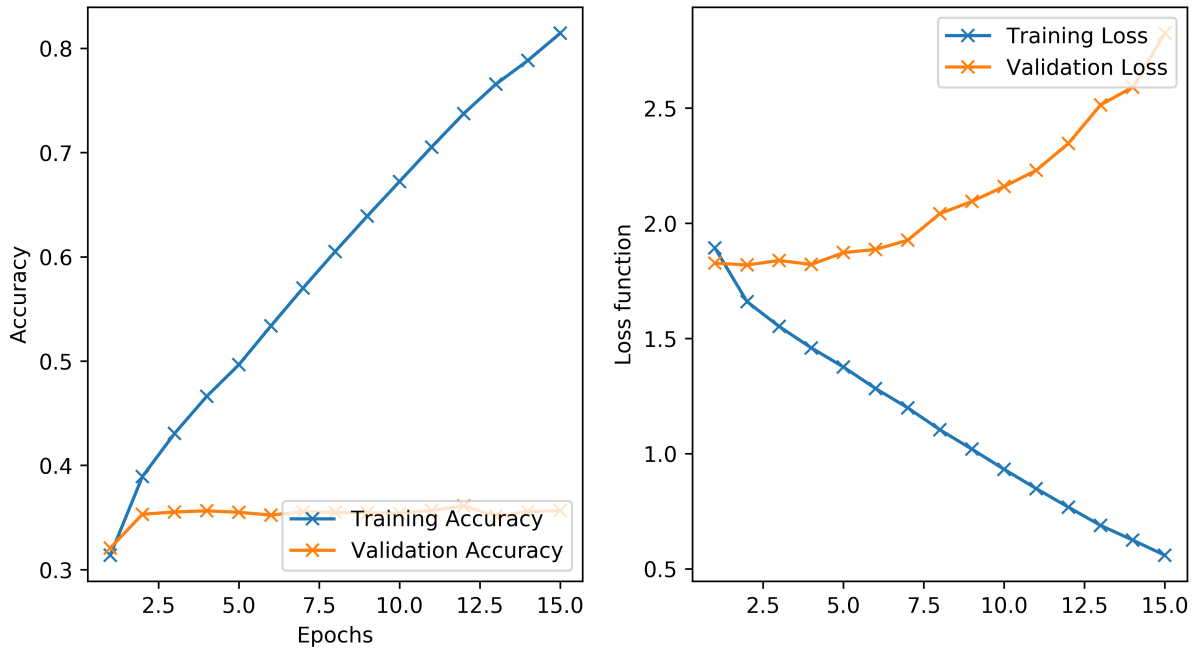

FIG. 16: Results for the training of shallow neural network for 32 modes in strong turbulence. The graph on the left shows the decimalised accuracy of the network for the training (blue) and validation (orange) data at each epoch. The graph on the left shows the corresponding loss for the training and validation data.

specific cases too much, additional training data would reduce this overtraining. The shape of the accuracy curves over the training process indicate that additional learning time is required to optimise the validation accuracy. The SNN training curve is shallow in comparison to the training curve for CNN-2 as displayed in Fig.13; the SNN curve instead resembles that of the strong turbulence case for the same number of modes. Similarly, the case of 4 modes in strong turbulence also displays a large divergence between the training and validation accuracy curves with $7.5 \pm 0.2\%$, in comparison to values of $5.1 \pm 0.2\%$ and $0.1 \pm 0.2\%$ for CNN-1 and CNN-2 respectively. In both cases for the shallow neural network, the curves have a strikingly similar shape, indicating that they both suffer from the same issue.

At the far end of the spectrum of training curves, the case for 32 modes in strong turbulence, depicted in Fig.16, conveys the limitations of this network structure, with a resulting validation accuracy of less than 50%. This failure can be observed in the graph depicting accuracy; the nearly linear increase in training accuracy as the network overtrains to the training data set, as key class defining features are failed to be identified, should be noted. Crucially, the validation accuracy shows a decrease as a result of the overtraining; this is best shown in the increasing validation loss whilst the training loss is monotonically decreasing. However, it should be noted that this network is not a complete failure as the validation accuracy still maintains a value of correct classification of $36.1 \pm 0.1\%$ noticeably better than random at 3.1%.

Overall, this particular network does not produce any higher accuracy networks. The structure manages to produce comparable results in the low and medium turbulences, especially for networks trained on a lower number of modes. However, for the more complex scenarios that were trained for this network was significantly worse, and were vastly outperformed by the networks with convolutional layers. The shallow neural network suffered from a large amount of overtraining due to its inflexibility resulting in failure to identify the shifting features present in the high turbulent situation. This network structure could be improved with the addition of a larger number of neurons in the single dense layer. However, it would be impractical considering the computational requirements compared to the addition of a convolutional layer.

4.2. Noise

TABLE VIII: Complete results of random noise testing on each of the neural network structures and the relevant turbulence and modes used. Results are of the format of the predicted mode for the noise input and the associated softmax function output for that mode below with an uncertainty of ± 0.01.

Turbulence	Weak				Medium				Strong			
Modes Used	4	8	16	32	4	8	16	32	4	8	16	32
CNN-1	0	0	0	0	0	1	2	12	0	0	1	26
	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.90	1.00	1.00	0.85	0.27
CNN-2	0	0	0	0	0	0	0	0	0	0	0	28
	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.52	0.67
SNN	3	7	15	29	3	7	11	19	1	3	4	28
	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00

To get a better understanding of the neural networks constructed and how they function, images containing random noise were passed through them 200 times. The random noise images will reveal the mode that the network defaults to and provide insights into how. Additionally by passing these random images through all networks it can be observed how turbulence is handled by each of the structure designs along with the high quantities of modes to classify. On examination of the data presented in Table VIII, the data has clear characteristics unique to network structure and turbulence trained on.

Firstly, on examining the weak data, there is a clear separation between the networks containing convolutional layers and the network without; a pattern which can be observed throughout this data. The networks with convolutional layers for weak turbulence all default to have the LGB mode $l = 0$ as the predicted mode with full confidence. The shape of the $l = 0$ being unique in the set of LGB modes used due to its lack of petal structure, having a Gaussian form, may be closer in features to random noise compared to petal displaying modes as an explanation for this outcome. In contrast, the shallow neural network, having no convolutional layers, tends towards picking the highest number of mode possible. This may be a result of the complexity of pattern increasing with mode number. Taking the path of least resistance and due to the simple single-layer structure, the network may find it more efficient to only focus on identifying the lower mode patterns. This would cause categorisation of higher modes in an else subcategory thus explaining this result. In the case of the 32 mode case the $l = \pm 29$ mode is defaulted to instead; this is likely due to the random nature of the order of the data fed into the learning process combined with the complexity of the pattern causing this mode to be the default instead of the highest order.

In the case of networks trained in medium turbulence, the results of this investigation become more varied. CNN-1 predicts different modes for the noise as the number of modes the network is trained for increases. The network maintains a full level of certainty in the case of 8 and 16 modes, predicting mode of $l = \pm 1$ and $l = \pm 2$ respectively, which form the least distinctive patterns for this network. The case for 32 modes predicts, with the least certainty of any network in this category, of 0.90 ± 0.01, the mode $l = \pm 12$; showing the network has an increasing spread in possibilities in the convolutional layer before making a prediction. Maintaining the trend from the previous turbulence levels, CNN-2 predicts the noise to be the mode $l = 0$ with maximum certainty up until 32 modes. This is theorised to be due to the increased pattern recognition facilitated by the additional convolutional layers. The decrease in certainty for the 32 modes is likely for the same reason as CNN-1 but with a reduced impact from the additional convolutional layers. For the shallow neural network, the confidence remained at 100% with the same modes being predicted as for weak turbulence for 4 and 8 modes. A change in prediction for the networks trained on a higher number of modes occurs for reasons discussed in the previous paragraph, with these modes being ones the network struggles to classify.

Finally, the networks trained on the strong turbulence data had the most striking results and the lowest confidence levels for all models investigated. Maintaining the trend for convolutional neural networks, $l = 0$ was predicted for the noise for the two networks with the least modes with full confidence. Networks trained with a higher quantity of modes made their predictions with lower confidence. The mode $l = \pm 26$ was predicted in the case for the 32 mode network, with the lowest confidence of 0.27 ± 0.01, still a significant margin. However, this shows bias towards one of the least accurately predicted modes, indicating the network is struggling

to classify a large volume of modes at the more complex end of turbulence, near the limit of the functionality of the network. Displaying the functionality of multiple convolutional layers until the most turbulent situation, CNN-2 maintains its trend of predicting the mode $l = 0$. This is until the network trained on 32 modes; significant because it follows the trend of the other networks and predicts a high mode with a more complex pattern, showing the limit this network possesses and breaking the trend it maintained. This is also signalled by a decrease in certainty for the networks trained in higher two number of modes, caused by the model having no definitive other classification. The shallow neural network maintains its own trend of full certainty for all of the predictions it made for the noise inputs. It only has a drop in confidence to 0.99 ± 0.01 for the 4 modes trained network. This is striking as it is the only case where it occurs for both a 4 mode network and the shallow neural network structure, although this is reasoned to be minor and is within error. Additionally, the network structure in this turbulence regime no longer predicts the maximum possible mode number in these cases, suggesting the network is struggling with the high distortion which occurs in the pattern of larger petals. However, like all other networks in strong turbulence, and this particular network structure for the 32 mode case, a high mode is predicted. This is thought to occur for a combination of reasons, as previously discussed, given for these other cases.

In consideration of all of these results, it can be observed that no single network is directly learning all of the possible patterns and their distinctive feature maps. All neural networks trained have a clear distinct default state with significant confidence in random noise being in that state. In the cases where there is not 100% confidence in the prediction of the network, the confidence value given by the softmax function in the final layer in all cases is still significantly above that which you would expect if the network had no bias. This having a value of $1/N$, where N is the number of modes which the network has to classify.

4.3. Rotational Data Augmentation

Another technique, which will allow for better understanding of the neural networks that have been developed, and how they perform, is testing how they react to the validation data set

TABLE IX: Complete results of rotational data augmentation noise testing on each of the neural network structures and the relevant turbulence and modes used. Results are of the format of the minimum percentage of the maximum data accuracy to the nearest $\pm 0.1\%$ and angle at which this occurred.

Turbulence	Weak				Medium				Strong			
Modes Used	4	8	16	32	4	8	16	32	4	8	16	32
CNN-1	0.480	0.376	0.374	0.354	0.404	0.432	0.389	0.496	0.507	0.621	0.726	0.833
	43	16	8	60	40	60	60	60	60	43	47	60
CNN-2	0.500	0.401	0.379	0.348	0.500	0.434	0.439	0.684	0.532	0.526	0.761	0.877
	58	39	60	4	56	40	8	61	53	54	40	62
SNN	0.388	0.375	0.375	0.348	0.400	0.442	0.392	0.400	0.465	0.468	0.552	0.596
	42	17	8	60	40	60	9	4	36	64	56	60

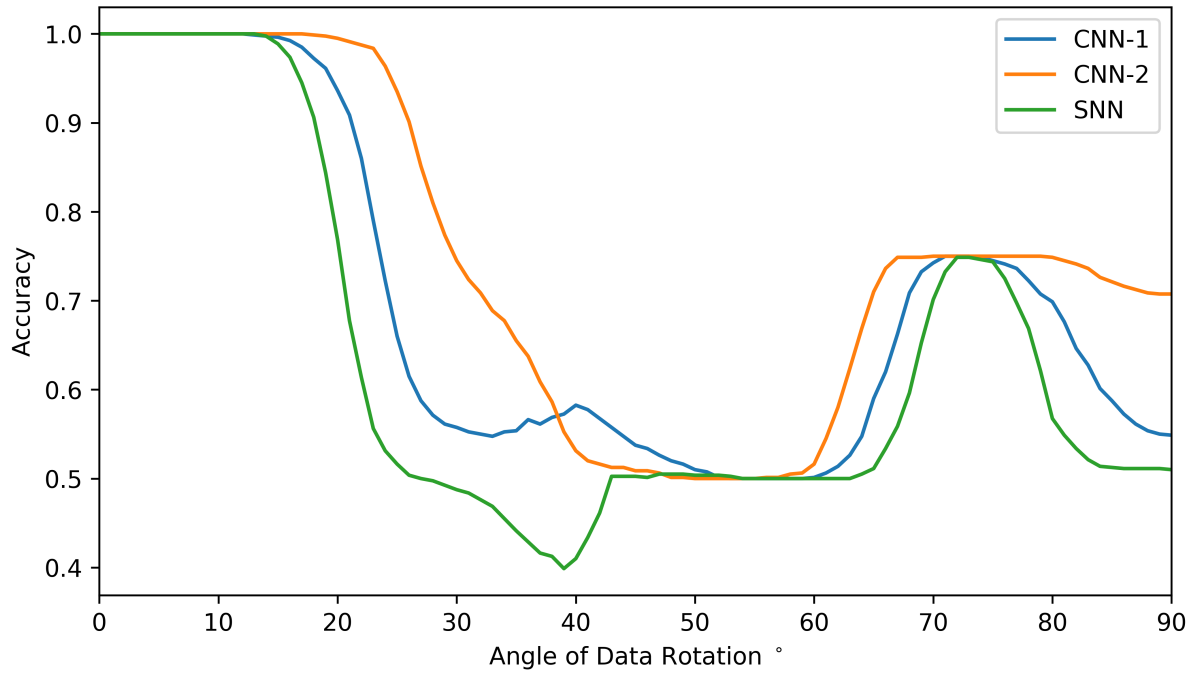

FIG. 17: Results for the accuracy after rotation of validation data for networks with 4 modes in medium turbulence. With convolutional neural network 1 in blue, convolutional neural network 2 in orange, and shallow neural network in green. Vertical error bars removed for clarity.

after the images have been rotated a fixed angle. These rotated images will test the networks' potential practical suitability, as this is an effect which could occur in practical use. The convolutional layers are intended to be translationally invariant to recognise features in a variety of locations. These tests on the networks will help show their effectiveness in a rotational capacity. Tests were completed by rotating the validation images every integer angle between $0°$and $90°$ and evaluating the model's accuracy on these augmented images. The full results of the investigation that occurred are detailed in Table IX, including the minimum percentage of the unaltered data accuracy and the angle at which this occurred.

In the case of networks which had obtained a 100% accuracy on the validation set, this level of accuracy was maintained for rotations close to a certain threshold. An example of this is shown in Fig.17 wherein all three network structures maintain their accuracy for at least $10°$; this suggests that some of the key features each network detects for these modes are unaltered by small rotations around the centre of the beam. Another feature of this figure is that the accuracy curves all remain smooth. This implies that in medium turbulence the networks are focusing on larger features that are gradually altered rather than drastically changing within a degree of rotation. At multiple points throughout the graph, all three networks can be seen to have the same accuracies, implying that all of the networks were using the same features for their classification which was affected by the rotation in the same way. The peaks can be seen to occur around the values of rotational symmetry for the LGB petal modes, with some having more effect on the accuracy of the networks than others. In the lower half of the angles tested, CNN-2 is less affected by the augmentation of the data; this is attributed to the multiple convolutional layers providing better rotationally invariant feature recognition. The SNN has

sharper peaks, suggesting that the network is more affected by the rotations, having a smaller range of angle where they can be detected. Additionally, the SNN has a different minimum than the other networks, occurring at $39°$, directly contrasting where CNN-1 has a local maximum. This shows that the lack of convolutional layers causes different features to be used.

Following a similar initial form, increasing the number of modes breaks down the distinctive shared peaks. In a more turbulent case with an increase in the number of modes, each network has a unique location of maxima and minima, shown in Fig.18. However, some trends remain the same: the initial increases in the angle the validation set was rotated lead to a large reduction in accuracy before a second peak. The largest noticeable difference between medium and strong turbulence is that the smoothness of the accuracy curve decreases with turbulence, suggesting the networks are using smaller features more affected by rotation in comparison. Additionally, it should be noted that there is a reduction in the magnitude of maxima and minima; this is attributed to the increased number of modes being classified, reducing the contribution of the individual accuracy of each mode to the overall network accuracy. The most surprising thing about this result is CNN-1 starting off with lower accuracy than CNN-2, then at larger angles of rotation, maintaining a consistently higher accuracy. The minimum accuracy that CNN-1 reaches is $62.1 \pm 0.01\%$ which is noticeably larger than CNN-2 with a minimum accuracy of $52.6 \pm 0.1\%$. This is the one case where this occurs and it can only be explained by CNN-1 learning a larger quantity of rotationally invariant cases compared to CNN-2 in this situation. This is further supported by CNN-2 breaking the trend for strong turbulence of an increasing minimum accuracy with modes used in this specific case, implying this network is anomalous.

For the case of networks with a high number of modes in strong turbulence, the initial de-

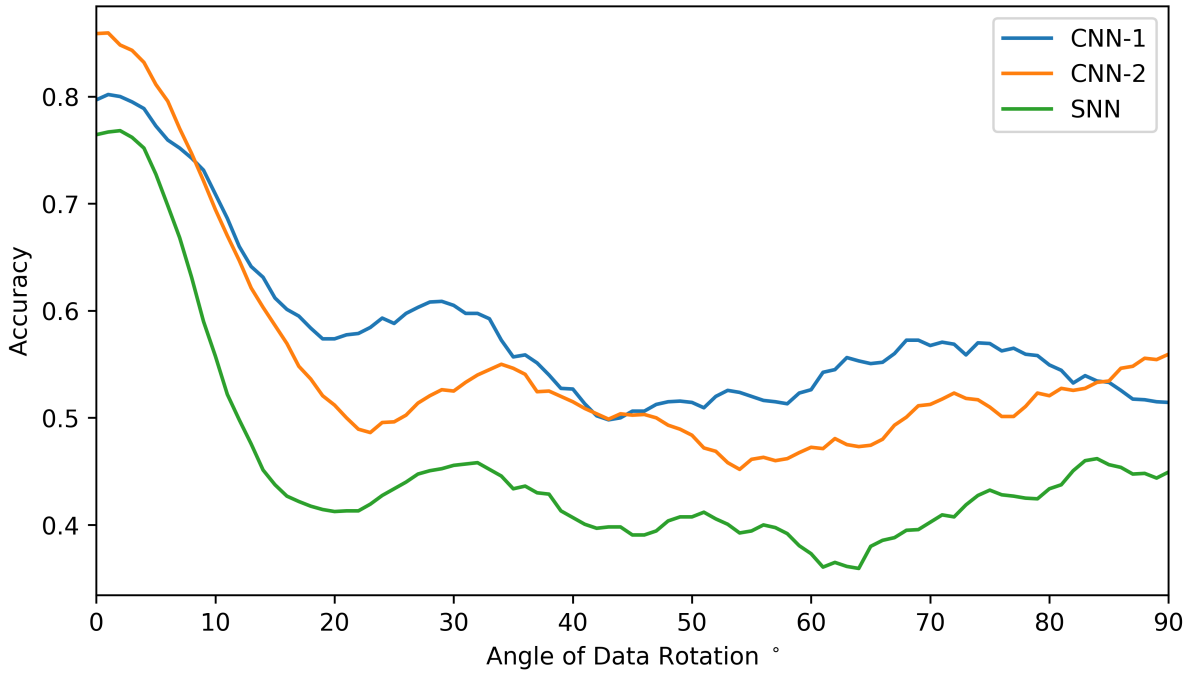

FIG. 18: Results for the accuracy after rotation of validation data for networks with 8 modes in strong turbulence. With convolutional neural network 1 in blue, convolutional neural network 2 in orange, and shallow neural network in green. Vertical error bars removed for clarity.

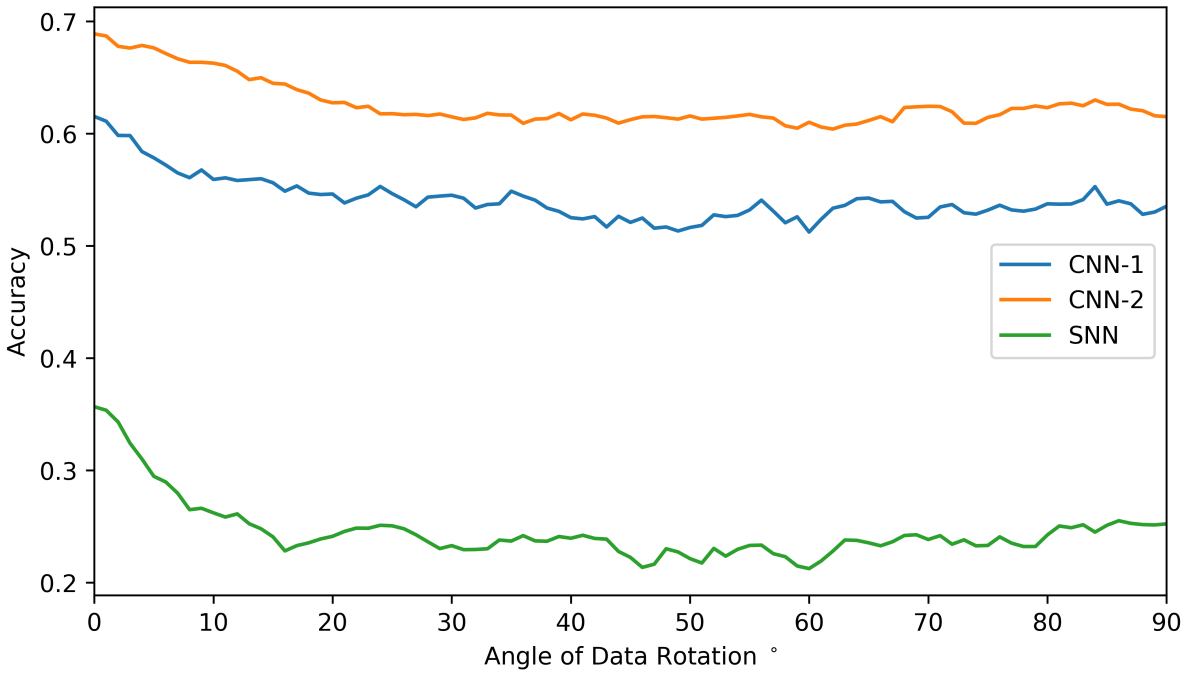

FIG. 19: Results for the accuracy after rotation of validation data for networks with 32 modes in strong turbulence. With convolutional neural network 1 in blue, convolutional neural network 2 in orange, and shallow neural network in green. Vertical error bars removed for clarity.

crease is reduced in scale with all networks having their smallest decreases in accuracy, as shown in Fig.19. The smaller reduction in the accuracy with a higher number of modes can be attributed to an increase in modes classified which have a smaller angle of rotational symmetry, leaving them in a more recognisable form to the network for a wider range of angles. Due to the quantity of higher l modes used, the reduction in accuracy caused by lower l values are minimised. Following the trend for an increase in the number of modes, the distinctive maxima seen are reduced in magnitude and increased in number. This makes distinguishing the effects of individual modes on the accuracy impossible. In a comparison of the networks, the SNN takes the largest decrease in the accuracy; this is attributed to the majority of accurately classified modes having a low l, which posses larger angles of rotational symmetry.

This analysis of the data has revealed the dependence that most of the networks possess in the orientation of the incoming LGB wavefront patterns. This could be reduced by training networks using a variety of orientations or alternatively identifying and correcting for any rotation using a guide beam. For the networks trained in low and medium turbulence, a higher number of modes resulted in a lower accuracy in most cases. Networks that were trained on a higher number of modes in strong turbulence retained a higher level of accuracy as modes had smaller angles of rotational symmetry and networks used more rotationally invariant features. Turbulence also had an effect on the accuracy as more turbulent data had fewer smooth curves, as smaller features which are more affected by the rotation have a more important role in these networks. In general, the relative performance of each network remained the same. The SNN had larger decreases in the accuracy which are attributed to a lack of convolutional layer, resulting in a greater rigidity in its classifications and leading to larger accuracy reductions.

5. CONCLUSIONS

Through this investigation, multiple neural network structures have been tested in their suitability to distinguish different quantities of LGB modes in a variety of turbulent regimes. Each of these neural network structures had varying levels of success in correctly predicting the correct mode of the simulated data, depending on the number of modes it was required to classify and the turbulence level of the simulated data passed through the network. The first network structure tested was CNN-1, a shallow convolutional network. This structure performed with high accuracy on low and medium turbulence data, maintaining the accuracy above 95%, with its training and validation accuracies showing that the network was learning at a good rate. However, when trained on the high turbulence data, the network showed signs of overtraining and validation accuracy significantly decreased. The second network tested, CNN-2, was the most advanced, utilising multiple convolutional layers for increased feature detection. For the low and medium data, accuracies stayed above 98% showing improvement compared to CNN-1. Despite the drop in accuracy for the strong turbulence of this network, CNN-2 performed better in comparison to CNN-1, with over 6% greater accuracy. Finally, SNN which does not use any convolutional layers in its structure was tested. For the low and medium turbulence, this structure worked well, achieving higher accuracies than CNN-1 in some cases. However, even in these cases, the learning curves showed the network was beginning to overtrain. In strong turbulence, this network had the lowest accuracies, with accuracy and loss curves showing the limitations a dense layer provides. From this, it is clear that the optimal network structure would use multiple convolutional layers which remained competitively functional in higher turbulence.

The investigations that were done on the trained networks allowed for a more detailed exploration of the constructed neural networks. This helped provide an understanding of how the networks function and illustrate the possible limitations that these contain. The use of multiple test images containing random noise showed that each network had a default mode it would classify data as. Which mode was the default classification depended primarily on the networks constituent layers. Networks with convolutional layers had a default mode of $l = 0$ in almost all cases of 16 modes and less. For 32 modes with strong turbulence, the convolutional networks had higher mode used as default, as recognising more complex LGB petal mode patterns became more difficult. This lead to a reduction in certainty as the default became less distinct. Conversely, the SNN predicted the mode with most petals for the noise with a 100% certainty in every range of atmospheric turbulence, showing the classification parameters were rigidly defined in all situations. Tests that were done on the developed neural networks' accuracy when the validation dataset had been rotated a fixed angle gave insights into the scale of features detected and their location dependence. All networks showed a decrease in the accuracy from the initial value, after a certain angle for networks which had obtained 100% accuracy and instantly for the others. From the 4 and 8 mode networks, it is clear that the orientation of the pattern plays a key role in mode classification due to the presence of distinctive maxima, which occur around angles of rotational symmetry for those modes. To remove this effect it is suggested that a network would be trained on data that has petal mode wavefront patterns in a range of orientations, making the networks more robust.

This paper also provided further validation of the use of this particular method of generating data. The full details of the simulation of the propagation of these beams have been fully

discussed and considered to produce distorted wavefront patterns that are comparable to those physically measured. For these networks, it was decided to use Laguerre-Gauss beams but it would be valid to use other physically realisable beams such as Bessel-Gauss beams and to simulate their turbulent propagation with this method. However, this method does produce an issue when it comes to the volume of data which would be required to determine practical use in telecommunications. This is due to the low bit error rates these communications require as a standard [28]. Using this method requires a large quantity of computational power due to the volume of fast Fourier transforms which are critical to the process for even a single image. To train a network to this standard would require a data set of the size beyond practical feasibility with this method. This, combined with the computational power it would take to train this network, which would be more complex in structure than those trained in this study thus implying that this is not currently practically realisable.

The use of machine learning to facilitate the use of twisted photons in free space optical communication shows great potential. With an increased resolution and improved methods of accounting for turbulent distortion, this technique with its potentially infinite basis has the ability to revolutionise communication methods. Simplistic neural networks have been used in this investigation and have shown that high accuracies are obtainable, a more complex network could be used to achieve higher accuracy. Further investigations into this area could improve from the use of other methods in mitigating the effects of atmospheric turbulence such as the use of a guide laser which are currently used in telescopes.

References

[1] L. Allen et al. Orbital angular momentum of light and the transformation of laguerre-gaussian laser modes. Phys. Rev. A, 45:8185, 1992.

[2] A. Trichili. Communicating using spatial mode multiplexing. arXiv:1808.02462v3, 2019.

[3] A. Watnik T. Doster. Machine learning approach to oam beam demultiplexing via convolutional neural networks. Appl. Opt, 56:3386, 2017.

[4] R. Ionicioiu. Sorting quantum systems efficiently. Sci. Rep, 6:25356, 2016.

[5] M. Krenn et al. Twisted light transmission over 143 km. Proc. Natl. Acad. Sci. U.S.A, 113:13648, 2016.

[6] M. Mirhosseini et al. Efficient separation of the orbital angular momentum eigenstates of light. Nat. Commun, 4:2781, 2013.

[7] M. Krenn et al. Communication with spatially modulated light through turbulent air across vienna. New J. Phys, 16:113028, 2014.

[8] M. Zhang J. Li and D. Wang. Adaptive demodulator using machine learning for orbital angular momentum shift keying. IEEE Photonic Tech L, 29:1041, 2017.

[9] T. Wang et al. urbo-coded 16-ary oam shift keying fso communication system combining the cnn based adaptive demodulator. Opt. Express, 26:27849, 2018.

[10] S. Lohani and R. Glasser. Turbulence correction with artificial neural networks. Opt. Lett, 43:2611, 2018.

[11] E. Knutson et al. Deep learning as a tool to distinguish between high orbital angular momentum optical modes. Proc. SPIE, 9970:997013, 2016.

[12] S. Park et al. De-multiplexing vortex modes in optical communications using transport-based pattern recognition. Opt. Express, 26:4004, 2018.

[13] T. Doster and A. Watnik. Laguerregauss and besselgauss beams propagation through turbulence: analysis of channel efficiency. Appl. Opt, 55:10239, 2016.

[14] X z. Cui et al. Analysis of an adaptive orbital angular momentum shift keying decoder based on machine learning under oceanic turbulence channels. Opt. Commun, 429:138, 2018.

[15] M. Chen et al. Simulating non-kolmogorov turbulence phase screens based on equivalent structure constant and its influence on simulations of beam propagation. Results Phys, 7:3596, 2017.

[16] M. Chatterjee and F. Mohamed. Split-step approach to electromagnetic propagation through atmospheric turbulence using the modified von karman spectrum and planar apertures. Opt. Eng, 535:126107, 2014.

[17] M. Neifeld J. Anguita and B. Vasic. urbulence-induced channel crosstalk in an orbital angular momentum-multiplexed free-space optical link. Appl. Opt, 47:2414, 2008.

[18] J. Li et al. Mitigation of atmospheric turbulence with random light carrying oam. Opt. Commun, 446:178, 2019.

[19] L. Andrews. An analytical model for the refractive index power spectrum and its application to optical scintillations in the atmosphere. J. Mod. Opt, 39:1849, 1992.

[20] J.Li et al. Joint atmospheric turbulence detection and adaptive demodulation technique using the cnn for the oam-fso communication. Opt. Express, 26:10494, 2018.

[21] S. Srinath et al. Computationally efficient autoregressive method for generating phase screens with frozen flow and turbulence in optical simulations. Opt. Express, 23:33335, 2015.

[22] Neural Networks and Deep Learning. Determination Press, 2015.

[23] L. Ding Y. Li and X. Gao. On the decision boundary of deep neural networks. arXiv:1808.05385, 2019.

[24] D. Kingma and J. Lei Ba. Adam: a method for stochastic optimization. ICLR, 2015.

[25] Deep Learning. MIT Press, Cambridge, MA, 2016.

[26] Image classification. https://www.tensorflow.org/tutorials/images/classification, as of the 16^{th} January 2020.

[27] F. Mitchell. Twisted photons and machine learning. 2019.

[28] Douglas et al Walsh. Practical bit error rate measurements on fibre optic communications links in student teaching laboratories. 2005.

Appendix

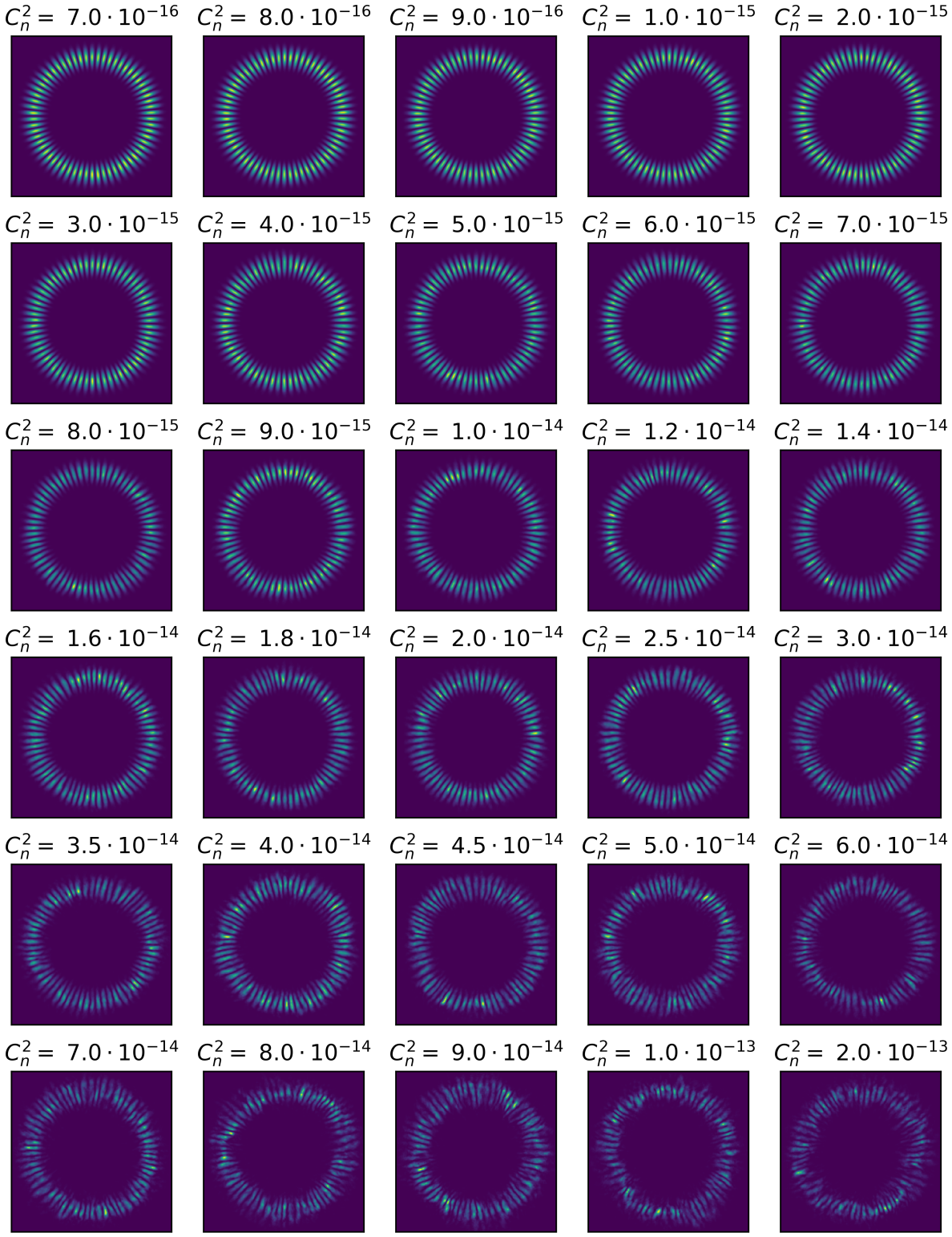

FIG. 20: Images of the $l = \pm 31$ mode after propagating through different strength turbulences. The strength parameter for the relevant turbulence is stated above each image.