



# Nonlinear Wave Front Reconstruction from a Pyramid Sensor using Neural Networks

Alison P. Wong<sup>1,2,3</sup> , Barnaby R. M. Norris<sup>1,2,4</sup> , Vincent Deo<sup>5</sup> , Peter G. Tuthill<sup>1</sup> , Richard Scalzo<sup>6</sup> , David Sweeney<sup>1,2</sup> , Kyoho Ahn<sup>5</sup> , Julien Lozi<sup>5</sup> , Sébastien Vievard<sup>5,7</sup> , and Olivier Guyon<sup>5,7,8,9</sup>

<sup>1</sup> Sydney Institute for Astronomy, School of Physics, Physics Road, University of Sydney, NSW 2006, Australia; [a.wong@sydney.edu.au](mailto:a.wong@sydney.edu.au)

<sup>2</sup> Sydney Astrophotonic Instrumentation Laboratories, Physics Road, University of Sydney, NSW 2006, Australia

<sup>3</sup> Discipline of Business Analytics, University of Sydney, NSW 2006, Australia

<sup>4</sup> Astralis, School of Physics, University of Sydney 2006, Australia

<sup>5</sup> National Astronomical Observatory of Japan, National Institutes of Natural Sciences, Subaru Telescope, 650 North A'ohōkū Place, Hilo, HI 96720, USA

<sup>6</sup> Centre for Translational Data Science, University of Sydney, Darlington NSW 2008, Australia

<sup>7</sup> Astrobiology Center of NINS, 2-21-1, Osawa, Mitaka, Tokyo, 181-8588, Japan

<sup>8</sup> College of Optical Sciences, University of Arizona, Tucson, AZ 85721, USA

<sup>9</sup> Steward Observatory, University of Arizona, Tucson, AZ 85721, USA

Received 2023 June 29; accepted 2023 September 27; published 2023 November 2

## Abstract

The pyramid wave front sensor (PyWFS) has become increasingly popular to use in adaptive optics (AO) systems due to its high sensitivity. The main drawback of the PyWFS is that it is inherently nonlinear, which means that classic linear wave front reconstruction techniques face a significant reduction in performance at high wave front errors, particularly when the pyramid is unmodulated. In this paper, we consider the potential use of neural networks (NNs) to replace the widely used matrix vector multiplication (MVM) control. We aim to test the hypothesis that the NN's ability to model nonlinearities will give it a distinct advantage over MVM control. We compare the performance of a MVM linear reconstructor against a dense NN, using daytime data acquired on the Subaru Coronagraphic Extreme Adaptive Optics system (SCExAO) instrument. In a first set of experiments, we produce wavefronts generated from 14 Zernike modes and the PyWFS responses at different modulation radii (25, 50, 75, and 100 mas). We find that the NN allows for a far more precise wave front reconstruction at all modulations, with differences in performance increasing in the regime where the PyWFS nonlinearity becomes significant. In a second set of experiments, we generate a data set of atmosphere-like wavefronts, and confirm that the NN outperforms the linear reconstructor. The SCExAO real-time computer software is used as baseline for the latter. These results suggest that NNs are well positioned to improve upon linear reconstructors and stand to bring about a leap forward in AO performance in the near future.

Unified Astronomy Thesaurus concepts: [Neural networks \(1933\)](#); [Astronomical optics \(88\)](#)

## 1. Introduction

Adaptive optics (AO) systems are essential to the scientific productivity of many ground-based optical/IR telescopes, as they enable diffraction-limited imaging performance despite the angular smearing caused by seeing, which would otherwise limit resolution to approximately 0''.5–2''. The main components that make up an AO system are: a wave front sensor (WFS) that measures instantaneous phase aberrations, a deformable mirror (DM) which rapidly corrects aberrations upon reflection, and a control system that determines the DM shape from the wave front sensor. The post-AO image quality relies on the accuracy to which aberrated wavefronts can be measured, reconstructed, and compensated; it is therefore ideal to employ highly sensitive wave front sensors and effective reconstruction techniques.

Employment of pyramid wave front sensors (PyWFSs; Ragazzoni 1996) has increased as AO systems have begun to trade the more conventionally used Shack–Hartmann wave front sensor (SHWFS; Esposito et al. 2000; Platt & Shack 2001; Verinaud et al. 2004; Guyon 2005) for the PyWFS, which offers greater sensitivity. The PyWFS also offers users flexibility as its sensitivity and dynamic range can be tuned

through modulation and, when used with a charged-coupled device (CCD) camera, the resolution of the PyWFS image can be changed to accommodate various noise levels. PyWFSs are in use at some of the world's leading general and high-contrast AO facilities (Esposito et al. 2011; Close et al. 2013; Wang et al. 2021), including Subaru Coronagraphic Extreme Adaptive Optics system (SCExAO; Jovanovic et al. 2015, Guyon et al. 2010, Vogt et al. 2010, Garrel et al. 2010). PyWFSs are also the WFS of choice for most instruments of upcoming extremely large telescopes such as the Giant Magellan Telescope (GMT; Esposito et al. 2012; Bouchez et al. 2018), the Thirty Meter Telescope (TMT; Boyer 2018; Crane et al. 2018) and the European Extremely Large Telescope (ELT; El Hadi et al. 2013; Brandl et al. 2021; Davies et al. 2021). These telescopes are ∼4 times larger in diameter than the current largest optical telescopes and will benefit greatly from the improved resolution enabled by the AO systems of tomorrow. The advent of the ELTs with their AO systems will enable diffraction-limited imaging with visible-wavelength angular resolutions better than ∼10 mas.

One major challenge that comes with using a PyWFS is that it is an inherently nonlinear sensor (approximately linear only for small wave front aberrations) while most of the commonly used wave front reconstruction techniques are linear. Modulation—steering the beam using a rapid scanning tip-tilt mirror synchronized with the acquisition camera—increases the linear regime of the PyWFS, but at the cost of sensitivity. The



Original content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

modulation amplitude can be chosen based on seeing conditions to maximize sensitivity while maintaining favorable conditions for stable closed-loop control with a linear reconstructor. No facility PyWFS AO system, to date, has achieved a consistent, reliable operational performance while using a PyWFS at its maximum, unmodulated sensitivity. Yet, there is strong interest in achieving such a design, particularly to meet the requirements of extreme adaptive optics (ExAO) systems.

This has motivated the development of various nonlinear wave front reconstruction techniques (Hutterer & Ramlau 2018). Iterative phase retrieval algorithms (Gerchberg & Saxton 1972) are not amenable to fast real-time use, and approaches have been proposed to address this via pre-computation of optical systems and solving with quasi-Newton methods (Frazin 2018). Another approach consists in dynamically adapting a linear reconstruction law to accommodate for nonlinear changes in the PyWFS first-order response (Chamboleyron et al. 2020; Deo et al. 2021). Predictive control algorithms (Guyon & Males 2017; Haffert et al. 2021) can also help tackle the nonlinearity problem.

Reconstruction techniques have more recently turned to applying methods that have emerged from deep learning (DL). Early examples where DL has been applied to AO include the design of NNs to predict Zernike coefficients from in-focus and out-of-focus images (Angel et al. 1990; Sandler et al. 1991; Lloyd-Hart et al. 1992) or off-axis WFS measurements (Osborn et al. 2014). Dense NNs have also been applied to wave front reconstruction from SHWFSS (Guo et al. 2006; Xu et al. 2019, 2020), along with newer approaches combining dense NNs with reinforcement learning (Nousiainen et al. 2021, 2022; Pou et al. 2022). More recently, NNs with more complex architectures have been applied to wave front prediction and reconstruction. This includes the use of recurrent neural network (RNN)s and convolutional neural network (CNN)s on e.g., SHWFS (Swanson et al. 2018; Escobar & Vera 2021; He et al. 2021) data; the PSF (Nishizaki et al. 2019); or pairs of in-focus and out-of-focus images (Xin et al. 2019). Work is also being done on the recovery wavefronts of from a single in-focus PSF (Paine & Fienup 2018; Andersen et al. 2019; Guo et al. 2019).

Previous research has already demonstrated wave front reconstruction from PyWFS measurements using NNs, first using CNNs to recover wavefronts from a three-sided PyWFS (Diez et al. 2008) for ophthalmologic applications. More recently, a CNN has been used in combination with a matrix vector multiplication (MVM) prediction and has shown that a model combining both improves the reconstruction for large wave front aberrations where the PyWFS response is nonlinear (Landman & Haffert 2020) in closed-loop operation. Additionally, a comparative study was conducted that investigated the performance of a number of NN architectures in their ability to reconstruct wavefronts in the Zernike basis from both a SHWFS and PyWFS (Escobar & Vera 2021).

In a tangentially related and rapidly growing area of research, automatic differentiation has proven to be a powerful tool for the optimization of optical systems (Pope et al. 2021; Wong et al. 2021). This has since been used to jointly optimize the sensitivity of Fourier-filtering wave front sensors and their reconstructor and are able to achieve greater sensitivity compared to the PyWFS (Landman et al. 2022). Reinforcement learning is also surfacing as a promising tool in AO control,

which could be used in conjunction with a PyWFS (Landman et al. 2021).

In this paper we compare the performance of various NNs against linear reconstructors, using data obtained using the SCExAO system at the Subaru Telescope, in off-sky testbed mode. Test wavefronts were applied to the system's DM and measured using the system's PyWFS for a range of modulation amplitudes. We then measured the accuracy to which these wavefronts could be reconstructed.

In Section 3 we study the reconstruction of low-order wave front errors comprised of Zernike polynomials and compare the reconstruction accuracy between methods and for different pyramid modulation amplitudes. We also compare the performance of a NN predicting modal coefficients, and one directly predicting actuator values. In Section 4 we repeat this analysis using realistic, high-order wave front error data created by applying von Kármán phase screens to the SCExAO DM. We show that the NN greatly out-performs linear reconstructors at low modulation amplitudes, and at high wave front error.

## 2. Methodology

The data used for our experiments were obtained using the SCExAO system at the Subaru Telescope. Using this system, we acquired data consisting of wavefronts applied by the system's DM, together with corresponding PyWFS signals recorded at a wavelength of 750 nm and a 50 nm bandwidth.

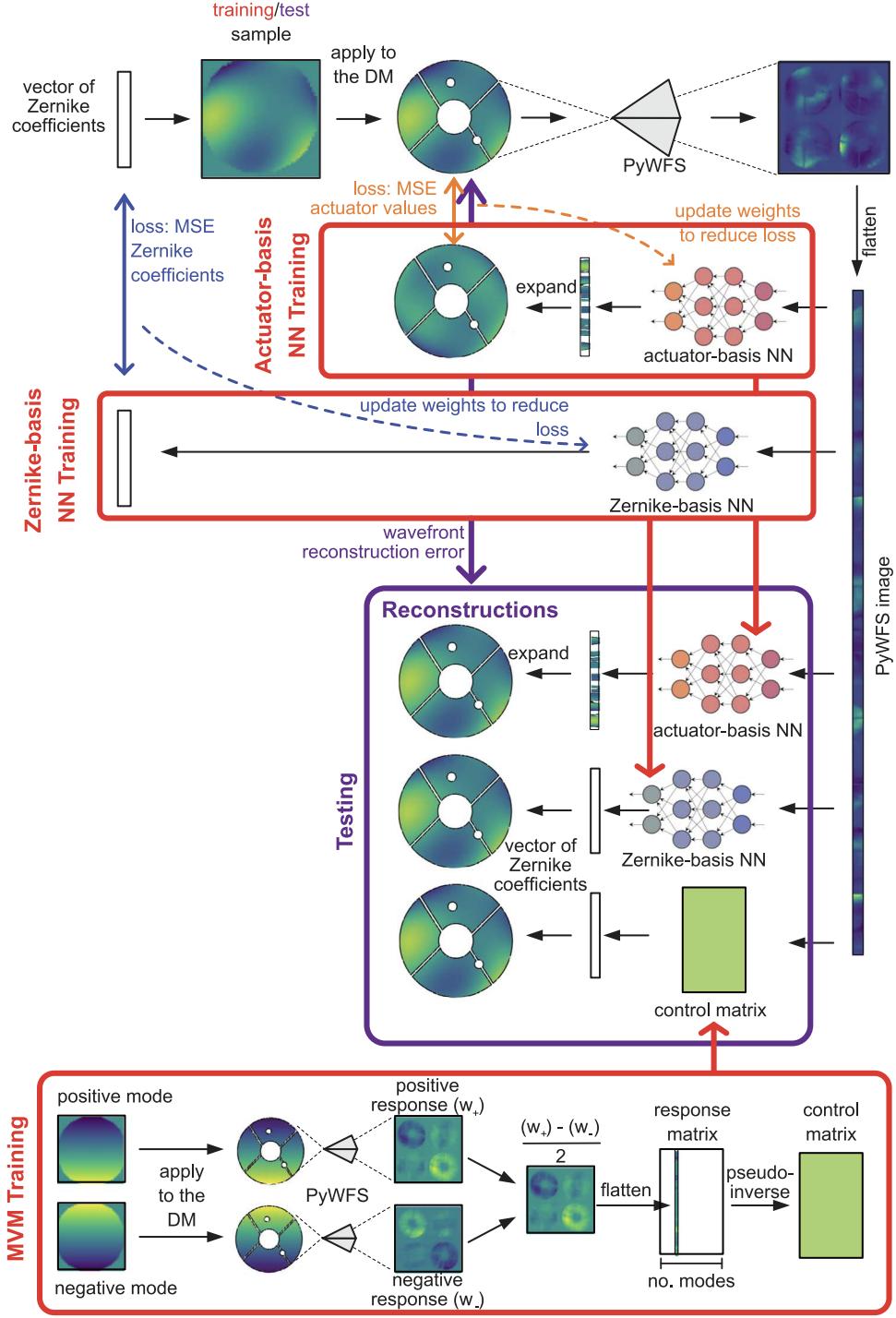
For clarity, we provide a visualisation (Figure 1) of the training and testing process described in Sections 2.1 and 2.

SCExAO is equipped with a 2040-actuator Boston Micro-machines DM, with a  $50 \times 50$  square pitch grid layout. Of these, 1365 actuators lie within the illuminated area of the pupil and are considered active. The root mean squared (rms) reconstruction errors and the wave front rmss are calculated using only these active elements. The PyWFS modulates the beam via a 2-axis piezo steering mirror and uses an OCAM<sup>2</sup>K EMCCD camera from First Light Imaging as its detector. The frame rate (and modulation speed) used was 1 kHz. For further details on the SCExAO system, see Ahn et al. (2021).

The SCExAO bench was used to produce two kinds of wave front condition: a low-order Zernike data set and a data set intended to mimic atmospheric turbulence. For each data set, we built linear reconstructor models (described in Section 2.1) and NNs (described in Section 2.2).

### 2.1. Linear Reconstructors

For each of the experiments described in this paper, we produced a simulated data set using the SCExAO bench as well as measuring a conventional response matrix for the linear reconstructor. The data set was used to train the NN, and to evaluate the success of both the NN and linear methods. This allows us to benchmark the NN performance against linear reconstructors. For the low-order Zernike set, a response matrix was acquired by a classical push-pull method around the best-effort flat wave front used as reference on SCExAO (which provides  $\gtrsim 97\%$  *H*-band Strehl). The 14 first non-piston Zernike terms were probed, as shown on Figure 2. These also formed the basis for wave front generation for our data set. The system control matrix, which lets us retrieve DM-space wave front maps from normalized PyWFS measurements, was obtained by direct inversion of the response matrix. The restriction to 14

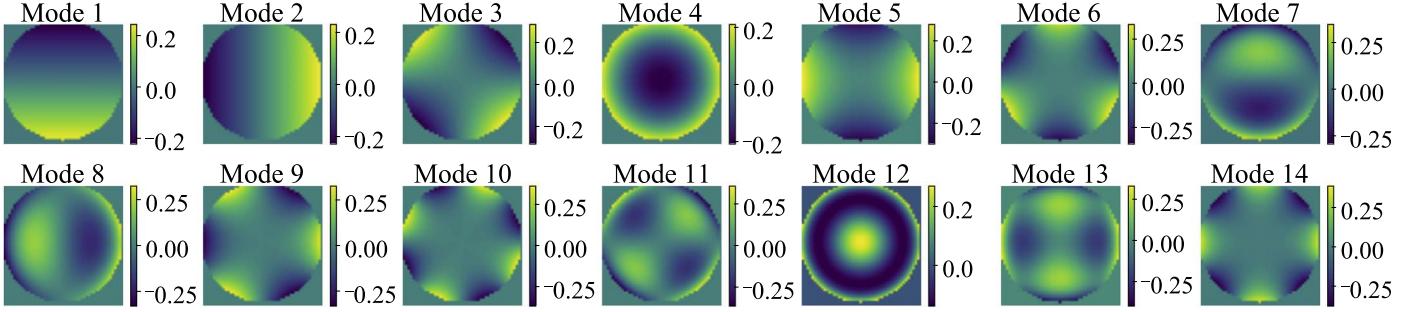


**Figure 1.** Visualization of the training and testing process on the low order Zernike wavefronts as described in Sections 2 and 3. For each data set, we built 3 models: the MVM, Zernike-basis NN and actuator-basis NN. The MVM was “trained” by calculating a response matrix as described in Section 2.1. The NN models were trained via backpropagation on a data set consisting of 70,000 pairs of wavefronts and their corresponding PyWFS measurements. The Zernike-basis NN was trained on wavefronts represented by Zernike mode coefficients while the actuator-basis NN was trained using the pixel values of the wavefronts. At test time, all three models were used to reconstruct wavefronts from PyWFS images, which were then compared to the true wavefronts. The results of these models for each modulation are shown in Figure 5.

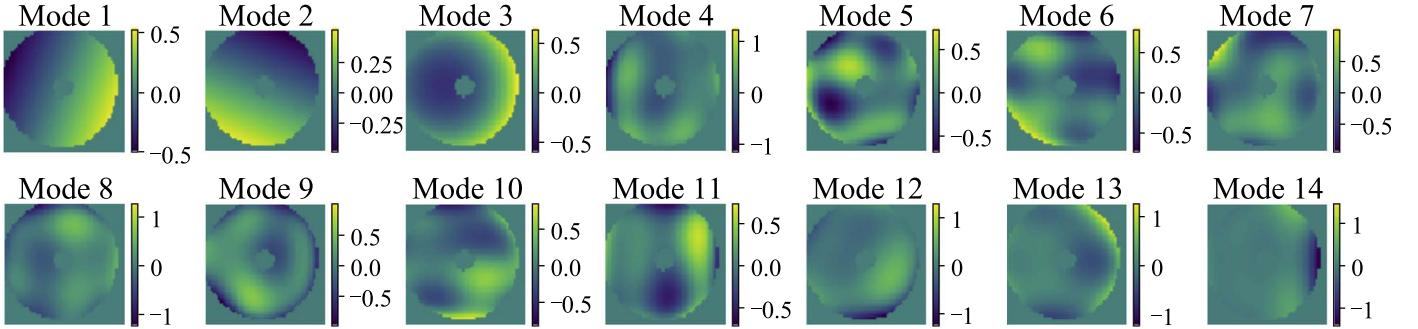
modes against a high-order PyWFS guaranteed proper conditioning of the matrix.

For the turbulence-like data set, we used the existing AO control software used for SCExAO operations, the Compute-and-control for Adaptive Optics (CACAO (Guyon et al. 2018)) software. A more complete description of the AO calibration

process using CACAO can be found in Guyon et al. (2018). A response matrix is measured from push-pull sequences of a basis of Hadamard modes; the control matrix is obtained through a block-wise double diagonalization of the response matrix, applying singular value decomposition (SVD) truncations on each of 15 blocks spanning  $\sim 1500$  modes. For



**Figure 2.** The first 14 Zernike modes (excluding piston) as used on the SCExAO  $50 \times 50$  actuator grid. The color scale is in units of microns. Each mode is normalized to 1 radian wave front rms at  $\lambda = 750$  nm.



**Figure 3.** The first 14 modes obtained through the CACAO response matrix calibration method, which is used as a benchmark for the neural network method. The color scale is in units of microns. These modes are normalized to 1 rad rms wave front at  $\lambda = 750$  nm.

illustrative purposes, we plot in Figure 3 the 14 lowest order modes obtained through this double-diagonalization inversion process.

## 2.2. Neural Network Implementations

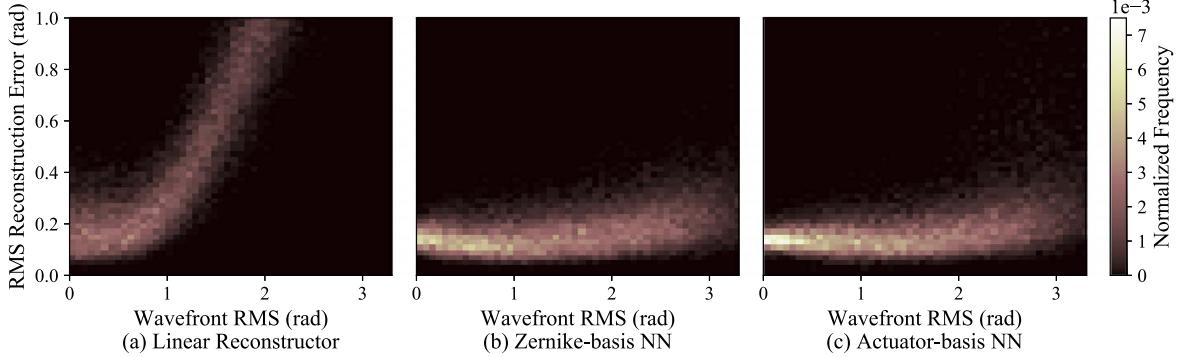
For comparison with the MVM linear reconstructors, we implemented and trained dense NNs. Two models were evaluated—one which predicted Zernike mode coefficients, and one which directly predicted the  $50 \times 50$  DM actuator map. Our NNs followed a simple architecture and comprised of fully connected layers of neurons. The input to each network was 14,400 flattened pixel values from a  $120 \times 120$  PyWFS image and the output was either the set of mode coefficients or the set of 2500 DM actuator values.

Reconstruction in the Zernike mode basis was performed for the low order experiments using the same basis that was used to generate the wavefronts (*low-order Zernike-basis NN* hereafter). This is advantageous for basic testing as it provided a complete reconstruction, with no loss of modal subspace between the introduced modes and the reconstructed modes. We later switched to perform wave front reconstruction in the actuator basis where the network directly outputs stroke commands for each actuator (*low-order actuator-basis NN*). This is more realistic, because typically the optimal mode basis is unknown. Additionally, the actuator basis is complete and can be easily extended to real seeing data, which we demonstrate in Section 4.

The number of hidden layers were adjusted per experiment and their exact architectures are detailed in the respective sections of this paper (Sections 3 and 4). The NNs were implemented with a ReLU (Nair & Hinton 2010) activation function, which gave them the advantage over the linear

reconstructors as it allowed the NNs to model nonlinearities. To improve model performance, batch normalization (Ioffe & Szegedy 2015) was used between the hidden layers. All networks were implemented using Keras (Chollet et al. 2015), with the Tensorflow (Abadi et al. 2015) backend. Training and predictions were run on an Nvidia 2080Ti Graphics Processing Unit (GPU).

A key advantage of NNs over linear reconstructors is that they have much more powerful regularization capabilities (Wong et al. 2021). Regularization of the MVM models was achieved by truncating singular values in the SVD computation as the small singular values tend to encode noise. NNs can represent more complex relationships, and so regularization is key to good performance by preventing overfitting. Methods of regularization used in regression L1 (lasso; Tibshirani 1996), L2 (ridge) or a combination (elastic net) are often applied to NNs. L1 regularization penalizes the norm of the weights in the NN, which encourages sparsity and L2 regularization penalize the square of the weights in the NN, which encourages edges in the NN graph to be close to 0. Elastic net is a weighted combination of the L1 and L2 regularization. A more popular method of NN regularization is dropout (Srivastava et al. 2014), in which neurons (or nodes) in the NN are randomly removed i.e., “dropped out” during training so that the NN does not become heavily reliant over a subset of the neurons, and effectively “trains” sub-networks within the NN. As a result, the NN behaves like an ensemble of smaller networks at test time. To determine the optimal method of regularization and for hyperparameter tuning we performed holdout validation. This is where we put aside a small portion of the data during model training so that we could evaluate our models on unseen data and select the model with the best generalization



**Figure 4.** Heatmaps illustrating the rms reconstruction error for different wave front rmss on the test data for a PyWFS modulated at 25 mas. (a) Linear reconstructor (b) low-order Zernike-basis NN and (c) low-order actuator-basis NN. The counts in these heatmaps are normalized by dividing the raw counts by the total number of samples.

**Table 1**  
L1 Regularization Hyperparameters for the NNs Trained in Section 3

Modulation	Zernike-basis NN	Actuator-basis NN
25 mas	$10^{-9}$	$10^{-9}$
50 mas	$10^{-9}$	$10^{-9}$
75 mas	$10^{-10}$	$10^{-9}$
100 mas	$10^{-10}$	$10^{-10}$

**Note.** These values were selected using holdout validation.

capabilities. We found that in all experiments L1 regularization was the most effective.

NNs work well in practice, but often with little insight into how or why. To gain an understanding of what a NN might learn from our data, we also implemented a *bottleneck-NN*. This is a NN where the network has its smallest layer (least number of neurons) at the middle of the network, hence creating a “bottleneck” through which information passes through. To gain insight into how a NN “thinks” the bottleneck can be examined to reveal learned representations of the network, because the bottleneck forces the NN to learn an efficient representation of the information it carries. This network, and its results, are described in Section 3.4.

### 3. Reconstruction of Low Order Zernike Wave Fronts

#### 3.1. Data Set

100,000 wavefronts were generated from a linear combination of the first 14 Zernike modes where the coefficient,  $c$ , for each mode was generated from a uniform random distribution ( $c \sim \mathcal{U}[-1, 1]$ ). Wave front maps were then scaled such that the wave front rms of the samples was approximately uniformly distributed between 0 and 3.25 radians. These samples were broken up into 70,000 training samples, 10,000 validation samples and 20,000 test samples.

Data acquisition was performed with the PyWFS at four modulation radii: 25, 50, 75 and 100 mas (typical on-sky modulation at SCExAO is 75 mas), which corresponds to 1.29, 2.59, 3.87 and 5.17  $\lambda/D$  for  $\lambda = 750$  nm. Each data set used the same 100,000 wave front samples.

#### 3.2. Model Hyperparameters

Reconstructing in the Zernike mode basis only required small NNs as the model only had to output 14 mode coefficients. These low-order Zernike-basis NNs had three hidden layers with 3000, 2000 and 1000 neurons respectively and had a learning rate of  $10^{-5}$ . Reconstructing in the actuator mode required larger networks due to the increased complexity of the basis. These low-order actuator-basis NNs had four hidden layers with 10,000, 7000, 5000 and 5000 neurons respectively and had a learning rate of  $10^{-3}$ . We applied L1 regularization to the weights of these models and have tabulated the corresponding hyperparameter in Table 1. These networks were trained for 2500 epochs.

The bottleneck-NN was trained on the 25 mas modulation data set. It had six hidden layers with 10,000, 5000, 2000, 14, 500 and 1500 neurons, respectively. We used a learning rate of  $10^{-3}$ , L1 regularization hyperparameter  $10^{-9}$  and trained the network for 500 epochs.

#### 3.3. Discussion and Results

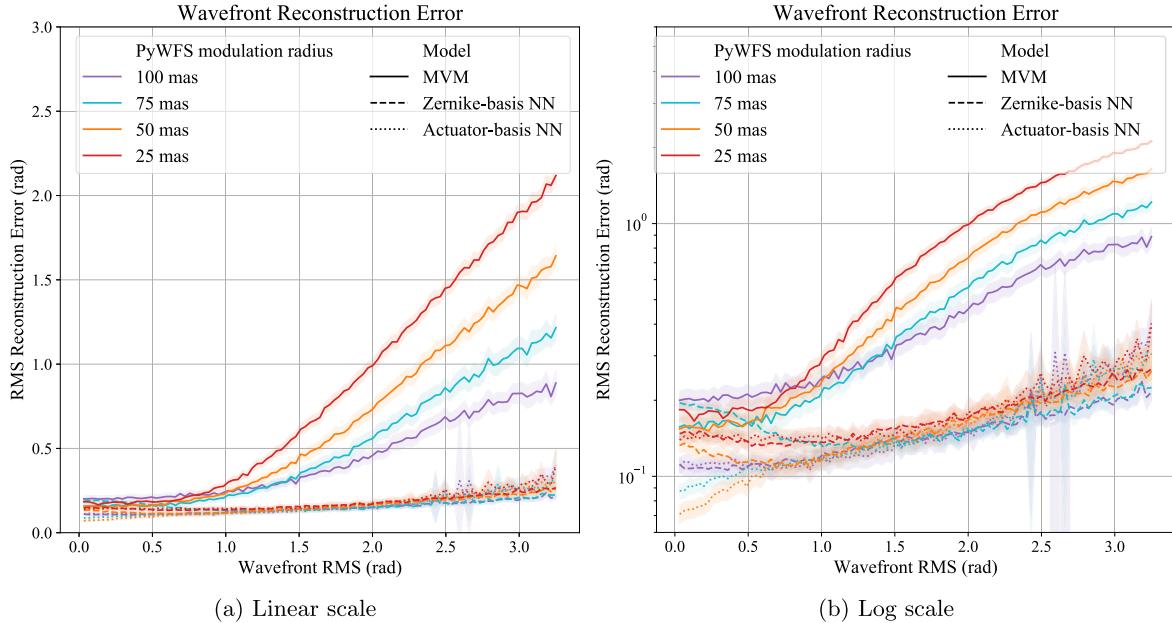
Results showing the reconstruction accuracy for the 25 mas pyramid-modulation data set are shown in Figure 4. These heatmaps visually show the rms reconstruction error against wave front across the 20,000 test samples. Here we define rms reconstruction error as

$$\sigma_{\text{reconstr.}} = \sqrt{\frac{1}{N} \sum_{i=0}^N (\text{pred}_i - \text{true}_i)^2} \quad (1)$$

where  $\text{pred}_i$  and  $\text{true}_i$  are the wave front value (in radians) for the  $i$ th actuator number, for a total of  $N$  active (i.e., unobstructed) actuators.

Figure 5 shows a summarized view of these reconstruction errors alongside those for the 50, 75 and 100 mas modulation data sets. In Figure 5, the point at which the PyWFS response becomes nonlinear—and the reconstruction accuracy for linear models rapidly decreases—is clearly visible as the “knee” of the plot. As expected, the onset of nonlinearity occurs at a larger wave front rms as the modulation radius is increased.

An important result that can be observed from Figure 5 is that the NNs are able to handle the PyWFS nonlinearities induced by introducing a wave front whose gradient reaches into a saturated regime. Further adding to this result is that the NN offers better performance over the MVM at all modulations



**Figure 5.** Average rms reconstruction error for the MVM and NNs for different pyramid modulations, for the low-order experiments. The MVM models are plotted with a solid line, the Zernike-basis NNs are plotted with a dashed line and the actuator-basis NNs are plotted with a dotted line. The width of the shaded region shows  $1\sigma$  uncertainties and the colors indicate the PyWFS modulation. The MVM only performs well at a high modulation where the PyWFS response is the most linear. However, the NNs are seen to work successfully even at large wave front rmss, despite the PyWFS' nonlinearities. All NNs have very similar performance, which implies that the actuator basis is as effective as the optimal Zernike mode basis.

and all wave front errors. This implies the NN may be suitable for wave front reconstruction from a pyramid with no modulation and, therefore maximum sensitivity. We considered this to be beyond the scope of this paper as the operation of SCEExAO at 0 PyWFS modulation is technically more involved as it is not typically used. Additionally, at the time the data set was obtained, the tip-tilt drift was too large to allow for consistent unmodulated data set, but we hope that it will be the subject of future work.

Figure 5 shows a clear offset in the rms reconstruction errors, with the best predictions plateauing at  $\sim 0.1$  radians rms reconstruction error, even as the applied wave front rms approaches zero. This is consistent across all model types. Inspection of the MVM model trained on the 25 mas data set showed that the average reconstruction residual map for samples with wave front rms  $< 0.5$  radians was found to have an overall wave front rms of  $4.5 \times 10^{-2}$  rad, with a maximum standard deviation for a given pixel to be 0.125 rad, which is relatively small and indicates high confidence in our mean residual maps. We obtained similar results when inspecting the other data sets and implies that there is not a systematic bias in the reconstructions and that the offset is consistent with measurement noise.

It should also be noted that there is a small difference in offset between the MVM models and the NNs. While it is expected that both models should offer the same performance at low wave front rms where the PyWFS response is linear, we must remember that the MVM has been globally optimized to perform well at all wave front rms. It is likely that the MVM sacrifices some performance at low wave front rms to achieve better performance at high wave front rms.

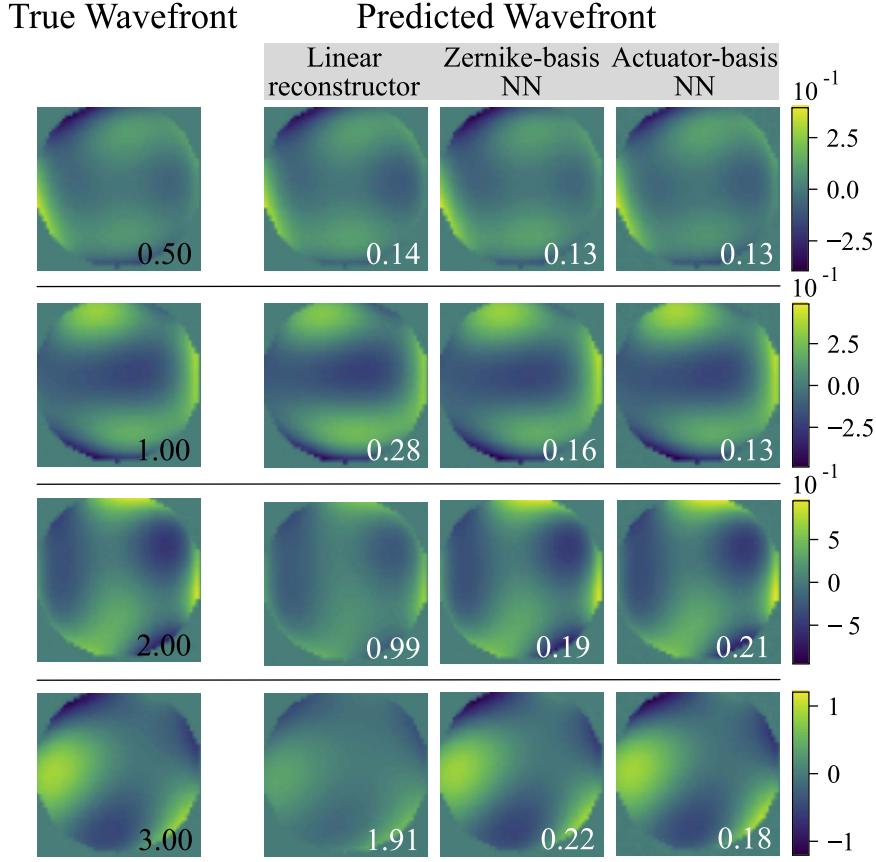
A selection of representative wave front samples are shown in Figure 6, and shows reconstructions for wavefronts with various rmss.

Another important outcome of this demonstration is that the actuator-basis NNs performed equally as well as the Zernike-basis NNs. As the Zernike-basis NN was, by design, optimal on the Zernike data set, this result implies that the NN models are agnostic to the mode basis. This means an actuator basis should be suitable for reconstructing arbitrarily shaped wavefronts and additional efforts into choosing an optimal mode basis may not be necessary.

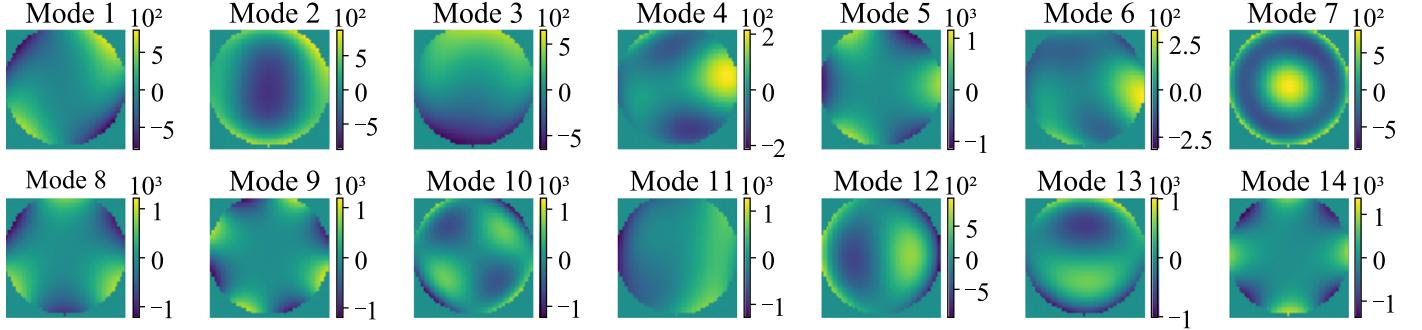
### 3.4. Bottleneck Network

As mentioned in Section 2.2, to gain an insight into how our NNs “think,” we produced a bottleneck network, which we trained on one of the low-order data sets used in Section 3. This NN performed reconstruction in the actuator basis but was designed with a central bottleneck layer with just 14 neurons, one for each mode used to generate the data. This encoder-decoder structure allowed for a mode “basis” to be recovered by breaking the network at the bottleneck and then activating each of the 14 neurons, thus emulating the two-step structure of a dual-stage modal MVM reconstructor, from wave front measurements into modal space and then into actuator space. The set of wavefronts represented within the bottleneck-NN were probed by maximally activating a neuron (inputting a large value) in the bottleneck layer while setting the remaining 13 neurons to 0. For each excited neuron in the bottleneck there was a corresponding wave front mode produced by the network. The result is shown in Figure 7. This is analogous to the modal basis of a linear system, except in this case these “modes” are created and combined in a nonlinear fashion to produce the output wave front. Thus, these are not modes in the usual sense, but they do permit qualitative insight into the features the network is most strongly discriminating on.

These bottleneck modes bear a striking resemblance to the Zernike modes (Figure 2), with difference due to the network’s



**Figure 6.** Sample reconstructions from the MVM, the Zernike-basis NN and the actuator-basis NN on the 25 mas modulation data set with the rms reconstruction error in radians shown in white. The first column shows the true wavefronts that were applied to the DM with the corresponding rms amplitude in radians shown in black. Note the differing color scales for each row. The color scale is in units of microns.



**Figure 7.** The mode “basis” recovered from the bottleneck-NN, which was trained on the 25 mas modulation data set. The color scale is in units of microns.

ability to generate and combine these modes nonlinearly. It is interesting to observe that for many modes, the NN naturally trained to learn direct-Zernike terms rather than linear combinations thereof, which in a truly linear regime would be equivalent. We performed this experiment a number of times, and the resultant mode basis always resembled the Zernike basis, but with slight variations to the mode shape and with random changes to the mode order.

#### 4. Reconstruction of High-order Turbulence-like Wave Fronts

The efficacy of the NN compared to the linear method was also tested on high-order wave front error data, with data

generated via a simulated atmospheric phase screen applied to the SCExAO DM. While for the linear model the standard response matrix was used, the choice of training basis for a NN is arbitrary. Any set of known input wavefronts can be used to train the model, as long as they broadly span a similar volume of wave front space similar to that of the real, on-sky wavefronts. To that end, we simply used other randomly generated sets of Kolmogorov phase screens as the training data.

##### 4.1. Data Set

The high-order data set was produced with wavefronts generated using HCIPy (Por et al. 2018) to simulate real seeing

conditions. We simulated a telescope with an 8.2 m diameter aperture (to match the Subaru telescope) and used a wave front sensing wavelength of 750 nm. One data set was produced with outer scale  $L_0 = 20$  m to simulate general atmospheric conditions, and another data set was produced with outer scale  $L_0 = 1$  m to simulate residual seeing after initial low spatial frequency correction from the facility AO system (AO188; Hayano et al. 2008, Hayano et al. 2010), after which SCExAO is located. For simplicity and to ensure the absence of statistical correlation across the training set, we generated independent wave front maps that were not temporally related. The amplitudes of these wave front maps were then multiplied by a scaling factor to produce the desired distribution of wave front rms.

The data sets were taken at a PyWFS modulation of 75 mas, which is typical for SCExAO observations. Each data set contained 220,000 samples, which were broken into 170,000 training samples, 20,000 validation samples and 30,000 test samples. The test and validation sets were constructed to contain a uniform distribution of rms wavefronts between 0 and 3 radians, which was achieved by multiplying wavefronts by appropriate constants. In a similar fashion, the training set was constructed to follow a cubic distribution so that there were more training samples available for wavefronts with large aberrations. This was based on an educated guess as it is expected that more training samples were required for large wave front rms, and the optimal distribution was not known to us. When the full  $50 \times 50$  wave front images are analyzed the wave front rms follows a true cubic distribution, however we re-examined these distributions after masking out the inactive pixels and retaining only the active pixels (corresponding to actuators in the pupil), and results in the adjusted cubic distributions shown in (Figure 8).

#### 4.2. Model Hyperparameters

The network trained on the  $L_0 = 20$  m data set had three hidden layers with 5000, 4000 and 3000 neurons and used L1 regularization with hyperparameter  $\lambda_{L1} = 10^{-9}$ . It was trained for 2500 epochs with a learning rate of  $10^{-4}$ . The network trained on the  $L_0 = 1$  m data set had four hidden layers with 3000, 2000, 2000 and 2000 neurons. This network was trained for 2500 epochs with a learning rate of  $10^{-3}$  and found that it performed best with no regularization. Batch normalization was used for both models. These networks were trained using the raw, un-normalized PyWFS images.

We chose NNs that were overly large and complex for the problem at hand so that regularization could be used to automatically reduce model complexity for us and select a model of appropriate complexity. It was important that we did not start with a model that was too simple, because a simpler model cannot become more complex, but a more complex model can become simpler through regularization.

For the linear reconstructor benchmark, the control matrix was produced as per Section 2.1. The modes are roughly sorted such that as the mode number increases, so does the spatial frequency. The first 14 modes are shown in Figure 3. To prevent overfitting, the MVM was regularized by selecting the number of modes kept for reconstruction which was chosen using holdout validation. This has the effect of reducing the complexity of the MVM model and while it may perform worse on the training data, it should generalize better to unseen data. In this process, the reconstruction accuracy was measured

for the same data but containing a varying number of modes. The number of modes which produced the most accurate reconstructions was then used for the subsequent experiments. For the  $L_0 = 20$  m data set this kept the first 136 modes and for the  $L_0 = 1$  m data set it kept the first 379 modes. This process is summarised in Figure 9.

#### 4.3. Discussion and Results

Heatmaps showing the rms reconstruction error versus the wave front rms are illustrated in Figures 10 and 11, and tabulated summaries are provided in Tables 2 and 3. The results clearly show that the NN outperforms the MVM method at all wave front rms for both data sets. The performance improvement of the NN over the MVM is much more pronounced for the  $L_0 = 20$  m data set than for the  $L_0 = 1$  m data set. This may be because the MVM algorithm is better optimized for the  $L_0 = 1$  m data set (i.e., for the wave front properties seen following correction by the facility low-order AO system), whereas the NNs have been equally optimized for both data sets.

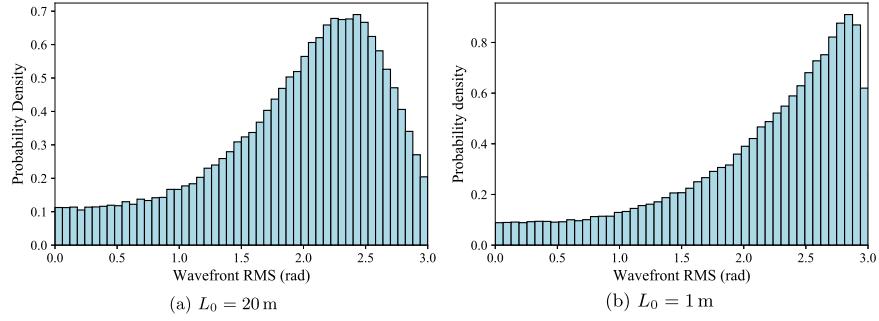
An obvious observation is that in both data sets the NN outperforms the MVM even when the wave front rms is small, i.e., when the PyWFS is operating in the linear regime. Initially, one would expect that both models should offer similar performance here, however it is important to recognize that the regularization for the MVM was optimized on the *entire* data set and most likely some performance on low wave front errors was sacrificed in order to obtain a better performance on higher wave front errors.

Additionally, reconstruction errors for the  $L_0 = 20$  m data set are smaller than the  $L_0 = 1$  m data sets for both the MVM and NN. This indicates that the  $L_0 = 1$  m data set is more difficult to model than the  $L_0 = 20$  m data set owing to the higher spatial frequencies in the data set. This is supported by the fact that the MVM required more modes to reconstruct the  $L_0 = 1$  m than the  $L_0 = 20$  m wavefronts. Additionally, the NN did not benefit from regularization, which implies that it may not have been complex enough to model the trends in the data.

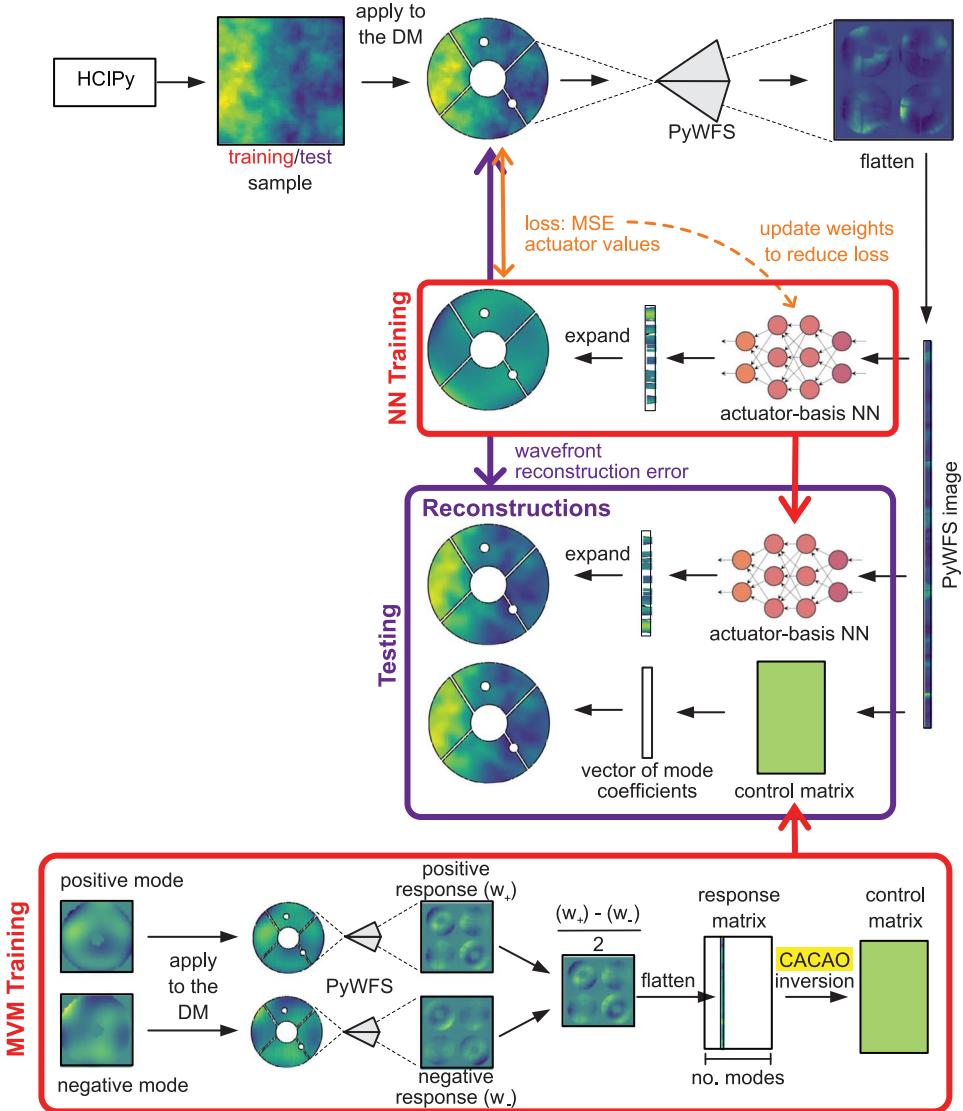
One potential reason that the NN performs better on the  $L_0 = 20$  m data set rather than the  $L_0 = 1$  m data set is that they both have data sets of the same size but each wave front in the  $L_0 = 1$  m data set contains significantly more information and is harder for the model to learn. More training data is expected to improve the performance of the model, so the results presented do not necessarily demonstrate the best performance a NN could theoretically achieve.

To further understand the performance of the MVM and NN models, we show some representative reconstruction samples in Figure 12. Like in Figure 6, we see that while the MVM does well to reconstruct the wave front profile, it tends to underestimate the magnitude of the wave front, particularly when the wave front errors become large. The NN does not appear to suffer the same problem. Additionally, the NN does appear to more accurately reconstruct the wave front profile than the MVM, most clearly seen in the  $L_0 = 1$  m examples in Figure 12.

Figure 13 shows sample NN reconstructions without the masking of inactive actuators. Superimposed over the top is an outline of the SCExAO pupil. It is interesting to note that the NN somewhat sensibly extrapolates the wave front outside of the pupil, a region where it has no information about the wave front. In order for the network to inpaint outside the pupil, it



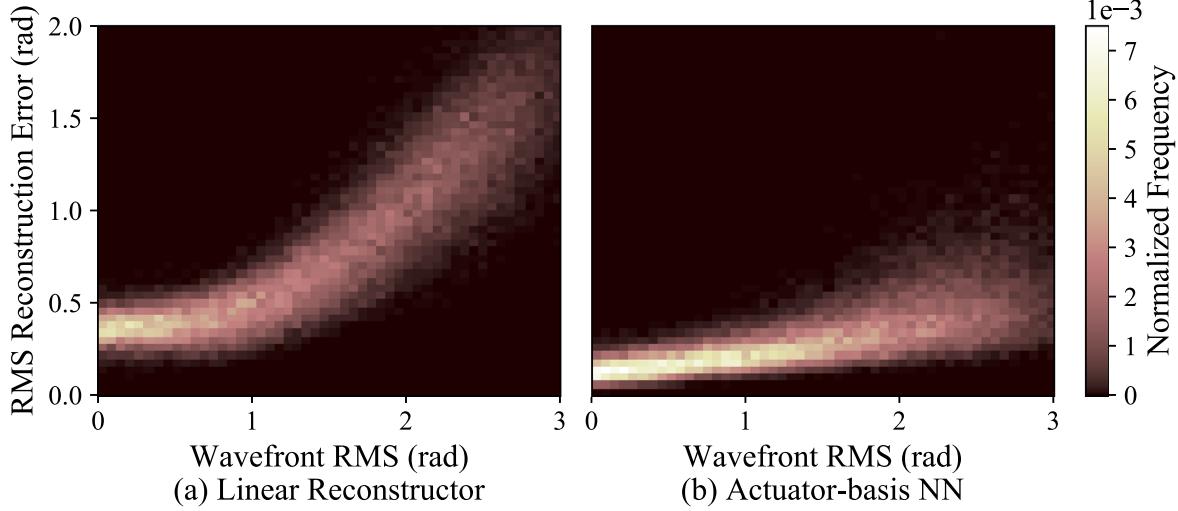
**Figure 8.** Distribution of the wave front rmss in the training data for the two outer scales tested, normalized so that the area under the curve is 1. There are more training samples at larger wave front rms where the mapping from PyWFS measurements to wave front is more complex.



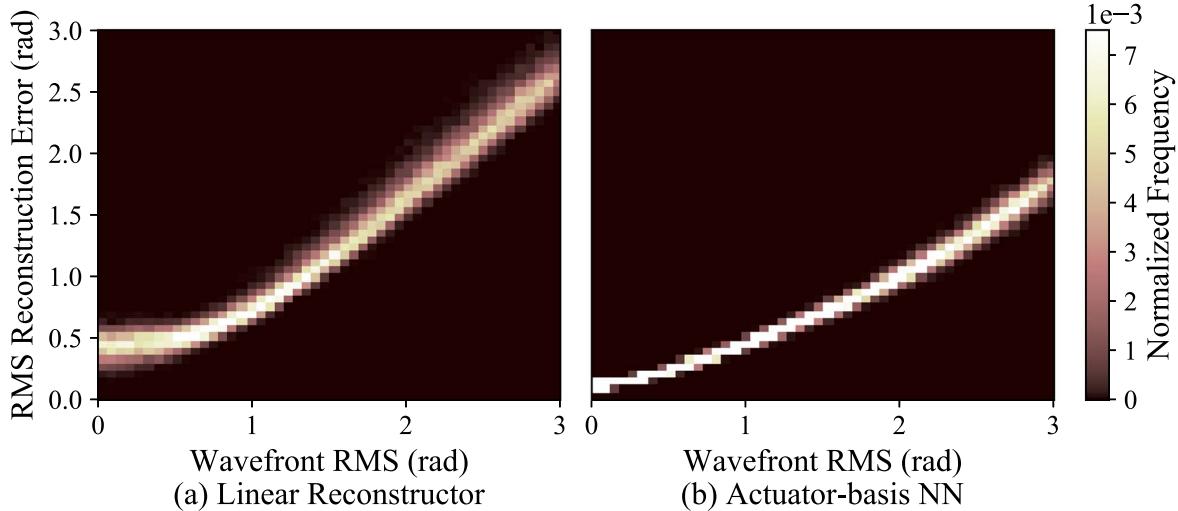
**Figure 9.** Visualization of the training and testing process on the turbulence-like data set as described in Sections 2 and 4. For each data set, we built 2 models: the MVM and actuator-basis NN. The MVM was “trained” by calculating a response matrix as described in Section 2.1. The NN was trained via backpropagation on a data set consisting of 170,000 pairs of wavefronts and their corresponding PyWFS measurements. At test time, both models were used to reconstruct wavefronts from PyWFS images, which were then compared to the true wavefronts. The results of these models for each modulation are shown in Figures 10 and 11.

has presumably learnt about the statistics behind the seeing, which it then uses to extrapolate the missing wave front regions. That is, in the cases of spatial frequencies of a similar

spatial extent to the missing regions, the set of wave front solutions that maintain spatial continuity and obey the learned distribution is well constrained.



**Figure 10.** Heatmaps showing the mean rms reconstruction error for samples of different wave front rmss, for the  $L_0 = 20$  m data set, which simulates general seeing conditions.



**Figure 11.** Heatmaps showing the rms reconstruction error for samples of different wave front rmss, for the  $L_0 = 1$  m data set, which simulates wavefronts already corrected by a low-order AO system.

**Table 2**

A Summary of the Mean Rms Reconstruction Error (rad) for Different Wave Front Rmss for the  $L_0 = 20$  m Data Set Comparing the MVM (Linear Reconstructor) with the NN

Model		Wave Front Rms (rad)		
		0–1	1–2	2–3
MVM	MVM	0.40	0.73	1.32
NN	NN	0.18	0.31	0.48

**Table 3**

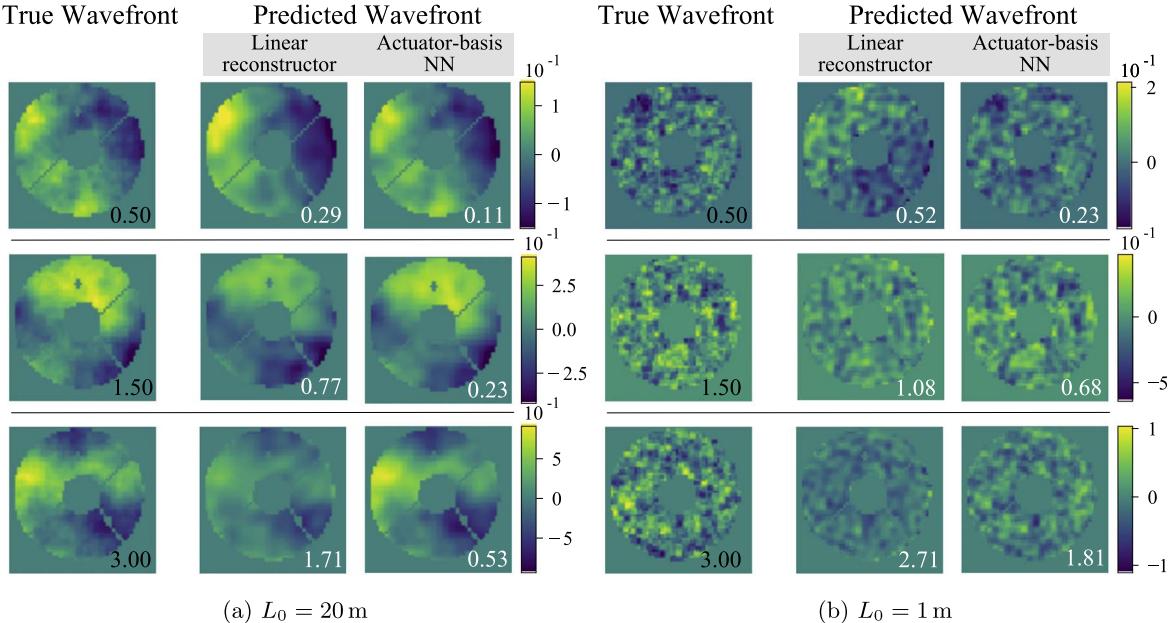
A Summary of the Rms Reconstruction Error (Rad) for Different Wave Front Rmss for the  $L_0 = 1$  m Data Set Comparing the MVM (Linear Reconstructor) with the NN

Model		Wave Front Rms (rad)		
		0–1	1–2	2–3
MVM	MVM	0.52	1.17	2.15
NN	NN	0.25	0.71	1.36

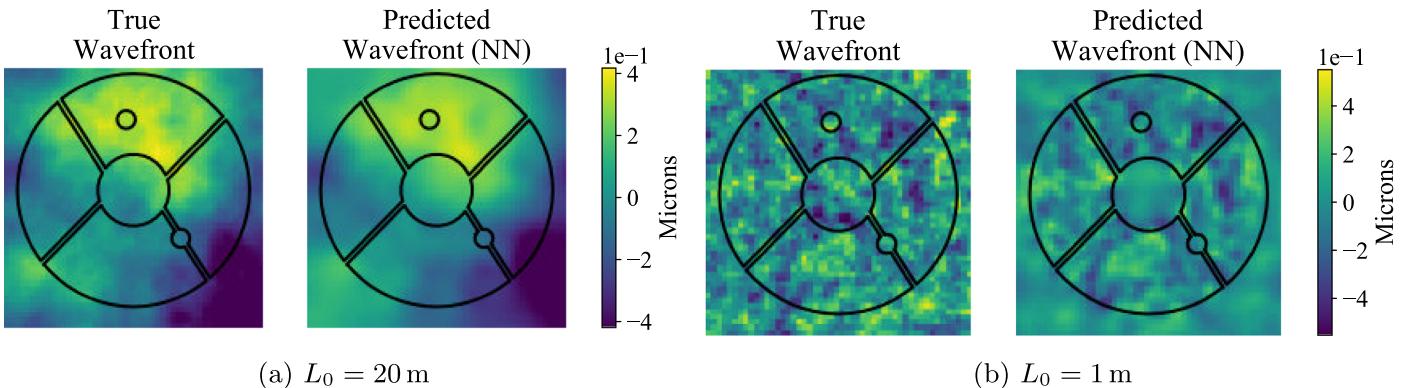
## 5. Conclusion

Here we have presented a number of experiments which compare the performance of the traditional linear, MVM-based method of wave front reconstruction with a dense NN reconstructor. Test data was acquired using the SCExAO system's PyWFS, for a range of test wavefronts applied to the system's DM. A key finding is that the NN was much better

able to handle nonlinearities that arose from the PyWFS in the presence of large wave front aberrations. The NN performed well on all PyWFS modulation amplitudes, while the performance of the MVM control matrix quickly decreased as the modulation decreased. It is possible then that a NN could be used in conjunction with an unmodulated PyWFS, thus taking full advantage of its high sensitivity.



**Figure 12.** Sample reconstructions from the MVM and NNs models with the rms reconstruction error in radians shown in white. The first column of each subfigure shows the true wavefronts that were applied to the DM with the corresponding wave front rms amplitude in radians shown in white. For display purposes, these figures were cropped to show only the active pixels. The color scale is in units of microns. We also see a similar effect to that in Figure 6 where the MVM method underestimates the wave front amplitude as the aberrations increase.



**Figure 13.** Typical samples from the  $L_0 = 20$  m and  $L_0 = 1$  m data sets. Left: True wave front. The  $50 \times 50$  array of actuator commands to apply a wave front to the DM. Right: NN reconstruction. The NNs produce a  $50 \times 50$  pixel map. Superimposed over these images is the SCEExAO pupil. It is interesting to note that the PyWFS is blind to anything outside of the pupil, however the NN can make a sensible estimate of the wave front in these regions based on the statistics learned from the training data set.

The MVM method and the NN were also compared using simulated seeing data, with and without simulated low-order pre-correction. It was found that the NN method would consistently perform more accurate wave front reconstructions than the MVM method at all wave front rmss, suggesting it may offer improved performance under poor seeing conditions.

While the results presented in this paper largely address an open loop AO system, where the WFS errors are large and more likely to push the PyWFS into the linear regime, these results still have important implications to a closed loop system. First, nonlinear reconstruction at large wave front errors makes it easier to close the AO loop during bad conditions. While in theory you can close the loop with any large wave front error, convergence can be slow. Thus, it can be impractical (or impossible) to close the loop during bad conditions when seeing is varying extremely rapidly, but this

can be alleviated by being able to handle large wave front errors. Second, in closed loop operation, there are many modes that cannot be sensed by the PyWFS. These interact nonlinearly with the lower order modes that the PyWFS can “see,” which means that nonlinear reconstruction is still beneficial.

A clear next step is to test a NN reconstruction in closed loop, on sky. One requirement is for the NN to reconstruct on sub-millisecond timescales. Current benchmarks demonstrate latencies well below 1 ms on far more complicated NNs with 100s of layers (NVIDIA 2022a), but work needs to be done to implement this within existing AO software stacks. Either an existing low-latency machine-learning framework could be used (such as TensorRT (NVIDIA 2022b)), or since operations for a NN can be broken down into a series of matrix operations, each followed by an evaluation of the activation function for the neurons, current AO software could be adapted to

implement specific models. An ideal test configuration would be to quickly switch between NN and MVM based algorithms while observing a given target, repeated for range of seeing conditions.

In addition to the absolute wave front correction performance, the robustness of a NN model over different seeing conditions and across multiple observing epochs needs to be evaluated. NN models generally take some time to train (the models in this paper taking  $\sim 10$  hr to train on a single consumer-grade GPU) compared to an MVM control matrix calculation. Having one pre-trained model, but with a small additional amount of “fine-tuning” training applied on sky, should be investigated.

The fully connected networks used in this paper are a very simple architecture. Further work in improving the NN model is required, including experimenting with different network architectures, such as a convolutional neural network, which is an ideal choice as it is designed for processing image data. The translation-invariance of these network’s inputs may also make them robust against PyWFS misalignment. These also have fewer free parameters, though may take longer to train. Additionally, a time domain component can be included in the model to allow for predictive control and overcome latencies in the AO loop. Sensor fusion is also an important area of interest, but it is difficult for the MVM method to handle a variety of different data types from multiple sources. A NN however, should be much more amenable to this problem.

### Acknowledgments

We would also like to acknowledge the Gadigal people of the Eora Nation as the traditional owners and custodians of the land on which the University of Sydney is built and on which some of this work was carried out. We pay our respects to their knowledge and to their Elders past, present, and emerging.

The development of SCExAO was supported by the Japan Society for the Promotion of Science (Grant-in-Aid for Research #23340051, #26220704, #23103002, #19H00703, #19H00695 and #21H04998), the Astrobiology Center of the National Institutes of Natural Sciences, Japan, the Mt Cuba Foundation and the director’s contingency fund at Subaru Telescope. The authors wish to recognize and acknowledge the very significant cultural role and reverence that the summit of Maunakea has always had within the indigenous Hawaiian community, and are most fortunate to have the opportunity to conduct observations from this mountain.

### ORCID iDs

- Alison P. Wong [ID](https://orcid.org/0000-0003-1236-7283) <https://orcid.org/0000-0003-1236-7283>
- Barnaby R. M. Norris [ID](https://orcid.org/0000-0002-8352-7515) <https://orcid.org/0000-0002-8352-7515>
- Vincent Deo [ID](https://orcid.org/0000-0003-4514-7906) <https://orcid.org/0000-0003-4514-7906>
- Peter G. Tuthill [ID](https://orcid.org/0000-0001-7026-6291) <https://orcid.org/0000-0001-7026-6291>
- Richard Scalzo [ID](https://orcid.org/0000-0003-3740-1214) <https://orcid.org/0000-0003-3740-1214>
- David Sweeney [ID](https://orcid.org/0000-0002-7528-1463) <https://orcid.org/0000-0002-7528-1463>
- Kyohoon Ahn [ID](https://orcid.org/0000-0002-1094-852X) <https://orcid.org/0000-0002-1094-852X>
- Julien Lozi [ID](https://orcid.org/0000-0002-3047-1845) <https://orcid.org/0000-0002-3047-1845>
- Sébastien Vieuard [ID](https://orcid.org/0000-0003-4018-2569) <https://orcid.org/0000-0003-4018-2569>
- Olivier Guyon [ID](https://orcid.org/0000-0002-1097-9908) <https://orcid.org/0000-0002-1097-9908>

### References

- Abadi, M., Agarwal, A., Barham, P., et al. 2015, Software available from [tensorflow.org](https://tensorflow.org)
- Ahn, K., Guyon, O., Lozi, J., et al. 2021, *Proc. SPIE*, 11823, 1182303
- Andersen, T., Owner-Petersen, M., & Enmark, A. 2019, *OptL*, 44, 4618
- Angel, J. R., Wizinowich, P., Lloyd-Hart, M., & Sandler, D. 1990, *Natur*, 348, 221
- Bouchez, A. H., Angel, G. Z., Ashby, D. S., et al. 2018, *Proc. SPIE*, 10703, 107030W
- Boyer, C. 2018, *Proc. SPIE*, 10703, 10703Y
- Brandl, B., Bettonvil, F., van Boekel, R., et al. 2021, *Msngr*, 182, 22
- Chambouleyron, V., Fauvarque, O., Janin-Potiron, P., et al. 2020, *A&A*, 644, A6
- Chollet, F., et al., 2015 Keras, <https://keras.io>
- Close, L. M., Males, J. R., Morzinski, K., et al. 2013, *ApJ*, 774, 94
- Crane, J., Herriot, G., Andersen, D., et al. 2018, *Proc. SPIE*, 10703, 107033V
- Davies, R., Hörmann, V., Rabien, S., et al. 2021, *Msngr*, 182, 17
- Deo, V., Gendron, É., Vidal, F., et al. 2021, *A&A*, 650, A41
- Diez, C. A., Shao, F., & Bille, J. 2008, *JMOp*, 55, 683
- El Hadi, K., Vignaux, M., & Fusco, T. 2013, in Proc. Third AO4ELT Conf., ed. S. Esposito & L. Fini, 99
- Escobar, D., & Vera, E. 2021, in CHILEAN Conf. on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON) (Piscataway, NJ: IEEE)
- Esposito, S., Feeney, O., & Riccardi, A. 2000, *Proc. SPIE*, 4007, 416
- Esposito, S., Pinna, E., Quirós-Pacheco, F., et al. 2012, *Proc. SPIE*, 8447, 84471L
- Esposito, S., Riccardi, A., Pinna, E., et al. 2011, *Proc. SPIE*, 8149, 814902
- Frazin, R. A. 2018, *JOSAA*, 35, 594
- Garrel, V., Guyon, O., Baudoz, P., et al. 2010, *Proc. SPIE*, 7736, 77365V
- Gerbberg, R. W., & Saxton, W. O. 1972, *Optik*, 35, 237
- Guo, H., Korablinova, N., Ren, Q., & Bille, J. 2006, *OExpr*, 14, 6456
- Guo, H., Xu, Y., Li, Q., et al. 2019, *Senso*, 19, 3533
- Guyon, O. 2005, *ApJ*, 629, 592
- Guyon, O., & Males, J. 2017, arXiv:[1707.00570](https://arxiv.org/abs/1707.00570)
- Guyon, O., Martinache, F., Garrel, V., et al. 2010, *Proc. SPIE*, 7736, 773624
- Guyon, O., Sevin, A., Gratadour, D., et al. 2018, *Proc. SPIE*, 10703, 107031E
- Haffert, S. Y., Males, J. R., Close, L., et al. 2021, *Proc. SPIE*, 11823, 1182306
- Hayano, Y., Takami, H., Guyon, O., et al. 2008, *Proc. SPIE*, 7015, 701510
- Hayano, Y., Takami, H., Oya, S., et al. 2010, *Proc. SPIE*, 7736, 77360N
- He, Y., Liu, Z., Ning, Y., et al. 2021, *OExpr*, 29, 17669
- Hutterer, V., & Ramlaun, R. 2018, *ApOpt*, 57, 8790
- Ioffe, S., & Szegedy, C. 2015, in Proc. 32nd Int. Conf. on Machine Learning, Proc. Machine Learning Research 37, PMLR, ed. F. Bach & D. Blei (Lille: PMLR), 448
- Jovanovic, N., Martinache, F., Guyon, O., et al. 2015, *PASP*, 127, 890
- Landman, R., Haffert, S. Y., Radhakrishnan, V. M., & Keller, C. U. 2021, *JATIS*, 7, 039002
- Landman, R., & Haffert, S. Y. 2020, *OExpr*, 28, 16644
- Landman, R., Keller, C., Por, E. H., et al. 2022, *Proc. SPIE*, 12185, 1218589
- Lloyd-Hart, M., Wizinowich, P., McLeod, B., et al. 1992, *ApJL*, 390, L41
- Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Int. Conf. on Machine Learning, ICML’10 (Madison, WI: Omnipress), 807
- Nishizaki, Y., Valdivia, M., Horisaki, R., et al. 2019, *OExpr*, 27, 240
- Noisiainen, J., Rajani, C., Kasper, M., & Heilin, T. 2021, *OExpr*, 29, 15327
- Noisiainen, J., Rajani, C., Kasper, M., et al. 2022, *A&A*, 664, A71
- NVIDIA 2022a, <https://developer.nvidia.com/deep-learning-performance-training-inference>
- NVIDIA 2022b, <https://developer.nvidia.com/tensorrt>
- Osborn, J., Guzman, D., de Cos Juez, F. J., et al. 2014, *Proc. SPIE*, 9148, 91484M
- Paine, S. W., & Fienup, J. R. 2018, *OptL*, 43, 1235
- Platt, B., & Shack, R. V. 2001, *Journal of Refractive Surgery*, 17, S575
- Pope, B. J. S., Pueyo, L., Xin, Y., & Tuthill, P. G. 2021, *ApJ*, 907, 40
- Por, E. H., Haffert, S. Y., Radhakrishnan, V. M., et al. 2018, *Proc. SPIE*, 10703, 1070342
- Pou, B., Ferreira, F., Quinones, E., Gratadour, D., & Martin, M. 2022, *OExpr*, 30, 2991
- Ragazzoni, R. 1996, *JMOp*, 43, 289
- Sandler, D. G., Barrett, T. K., Palmer, D. A., Fugate, R. Q., & Wild, W. J. 1991, *Natur*, 351, 300
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *Journal of Machine Learning Research*, 15, 1929

- Swanson, R., Lamb, M., Correia, C., Sivanandam, S., & Kutulakos, K. 2018, [Proc. SPIE, 10703, 107031F](#)
- Tibshirani, R. 1996, *Journal of the Royal Statistical Society. Series B (Methodological)*, 58, 267
- Verinaud, C., Le Louarn, M., Korkiakoski, V., & Braud, J. 2004, [Proc. SPIE, 5490, 1177](#)
- Vogt, F., Martinache, F., Guyon, O., et al. 2010, [Proc. SPIE, 7736, 773612](#)
- Wang, J. J., Delorme, J.-R., Ruffio, J.-B., et al. 2021, [Proc. SPIE, 11823, 1182302](#)
- Wong, A., Pope, B., Desdoigts, L., et al. 2021, [JOSAB, 38, 2465](#)
- Wong, A. P., Norris, B. R. M., Tuthill, P. G., et al. 2021, [JATIS, 7, 019001](#)
- Xin, Q., Ju, G., Zhang, C., & Xu, S. 2019, [OE Expr, 27, 26102](#)
- Xu, Z., Wang, S., Zhao, M., et al. 2020, [ApOpt, 59, 4768](#)
- Xu, Z., Yang, P., Hu, K., Xu, B., & Li, H. 2019, [ApOpt, 58, 1998](#)