Predictive control for adaptive optics using neural networks

Alison P. Wong,[a,b,*] Barnaby R. M. Norris,[a,b,c] Peter G. Tuthill,[a] Richard Scalzo,[d] Julien Lozi,[e] Sébastien Vievard,[e,f] and Olivier Guyon[e,f,g]

[a]University of Sydney, Sydney Institute for Astronomy, School of Physics, New South Wales, Australia
[b]University of Sydney, Sydney Astrophotonic Instrumentation Laboratories, New South Wales, Australia
[c]University of Sydney, AAO-USyd, School of Physics, Australia
[d]University of Sydney, Centre for Translational Data Science, Darlington, Australia
[e]National Astronomical Observatory of Japan, National Institutes of Natural Sciences, Subaru Telescope, Hilo, Hawaii, United States
[f]Astrobiology Center of NINS, Mitaka, Tokyo, Japan
[g]University of Arizona, College of Optical Sciences, Tucson, Arizona, United States

Abstract. Adaptive optics (AO) has become an indispensable tool for ground-based telescopes to mitigate atmospheric seeing and obtain high angular resolution observations. Predictive control aims to overcome latency in AO systems: the inevitable time delay between wavefront measurement and correction. A current method of predictive control uses the empirical orthogonal functions (EOFs) framework borrowed from weather prediction, but the advent of modern machine learning and the rise of neural networks (NNs) offer scope for further improvement. Here, we evaluate the potential application of NNs to predictive control and highlight the advantages that they offer. We first show their superior regularization over the standard truncation regularization used by the linear EOF method with on-sky data before demonstrating the NNs' capacity to model nonlinearities on simulated data. This is highly relevant to the operation of pyramid wavefront sensors (PyWFSs), as the handling of nonlinearities would enable a PyWFS to be used with low modulation and deliver extremely sensitive wavefront measurements. © *2021 Society of Photo-Optical Instrumentation Engineers (SPIE)* [DOI: 10.1117/1.JATIS.7.1.019001]

Keywords: adaptive optics; neural networks; wavefront sensors.

Paper 20145 received Sep. 24, 2020; accepted for publication Jan. 12, 2021; published online Feb. 6, 2021.

1 Introduction

One of the biggest challenges faced by ground-based telescopes is overcoming the effects of atmospheric turbulence, which limits angular resolution irrespective of telescope size. The main remedy for this is adaptive optics (AO).[1] In an AO system, the phase of an incoming wavefront is detected by a wavefront sensor, and the opposite phase profile is applied to a deformable mirror. The abberrated wavefront is then corrected upon reflection. AO systems are now an essential component of ground-based telescopes and have allowed us to reap the intrinsic resolution improvements offered by larger telescopes.

The success of an AO system depends on how well the deformable mirror can replicate the opposite phase profile of the incoming wavefront. This in turn is dependent on the accuracy of the wavefront sensor and the speed at which wavefront corrections can be applied to the deformable mirror. Because the latter is not immediate, errors arise from the outdated measured phase

*Address all correspondence to Alison P. Wong, a.wong@sydey.edu.au

Based in part on data collected at Subaru Telescope, which is operated by the National Astronomical Observatory of Japan.

profile, which lags behind the evolving wavefronts. Although standard AO (without prediction) performs well in conditions of good seeing, it performs poorly when the phase of the incoming wavefronts varies on short timescales (shorter than the AO loop correction speed), and the system is unable to keep up with the rapid seeing. Although the processing and execution times within the AO control loop can be reduced with powerful computers and well-optimized code, they will never be instantaneous. Similarly, the time needed to acquire sufficient photons to achieve a good signal-to-noise ratio in the wavefront sensor measurements (WFSMs) is nonzero. This is particularly problematic in the case of faint targets and motivates the desire to predict wavefronts in advance. Consequently, predictive control has become a pressing problem, and requires new methods of wavefront prediction to compensate for latencies in the AO control loop.

To put this into perspective, the wavefront error due to latency is comparable to errors from WFS photon noise. This is also related to latency as the AO control loop gain is adjusted to balance temporal lag error and WFS photon noise error: a slower turbulence evolution (or good prediction) would allow for WFS light to be accumulated over a longer time span to reduce photon noise. Other sources of wavefront error (non-common path errors, wavefront chromaticity, and scintillation) are less significant. Solving for temporal lag by wavefront prediction and optimizing the control law accordingly therefore address the two dominant terms in the error budget: temporal lag and WFS photon noise.

The most recently established method for predictive control adapts empirical orthogonal functions (EOFs),[2] historically used for weather and climate forecasting. This method relies on a linear algebra framework to construct a predictive filter (a matrix) that is multiplied by observed data to make a prediction. Details are given in Sec. 1.1. EOF predictive control is currently in operation at the Subaru telescope and has also been successfully demonstrated on Keck-II.[3]

Recent years have also witnessed a remarkable rise in machine learning techniques, specifically deep learning,[4] which have matured to the stage at which they are capable of solving an extensive range of problem types spanning a vast range of disciplines. The growth of neural networks (NNs) makes the present an opportune time to apply NNs to astronomical AO systems. There are a number of potential advantages to using NNs over the EOF predictive filter.

- **Extend complexity**. Changing the network size allows the NN to be of arbitrary complexity for fixed input/output size.
- **Model nonlinearities**. Although it has been shown the EOF converges to the global linear optimal solution, i.e., the best possible linear solution,[2] the NN can learn nonlinearities in the data that may occur in the time domain or relationships within the data itself.
- **Not memory limited**. Due to the singular value decomposition (SVD) that must be computed to obtain a predictive filter using EOFs, the standard EOF algorithm is memory limited (typically allows a few minutes of training data on machines with terabytes of memory) and must be frequently recalculated during an observing run so that the training data tracks the observing conditions. An NN can be extensively pre-trained on an arbitrarily large amount of training data and can therefore be exposed to a huge range of observing conditions. It could also be continuously trained during an observing run to adjust to immediate conditions.
- **More powerful regularization**. As we demonstrate in this paper, the NN has more powerful regularization than those typically used to generate EOF predictive filters.
- **Specialized architectures**. Although not addressed in this paper, more advanced architectures that can easily be adapted into the NN framework exist. Recurrent neural network (RNN) architectures such as the long-term short memory (LSTM) were specifically designed to handle time-series data and would naturally lend themselves to predictive control. Convolutional neural networks (CNNs), which specialize in image processing, would allow for alternate wavefront representations and for more versatile input formats that could improve prediction.
- **Sensor fusion**. Information from a range of different sensors and sensor types (e.g., different wavefront sensors and accelerometers to measure vibrations of the telescope's support structure and thermometers) will provide a more complete understanding of the

> environment at any given time and is expected to enhance prediction. NNs, when extended to specialized architectures (e.g., an encoder–decoder structure[5] or combining dense networks with CNNs[6]), are more amenable to sensor fusion than the EOF framework. NN sensor fusion has already been successfully demonstrated in the control of autonomous vehicles[5] and hand gesture recognition.[6]

The first attempts to harness the power of NNs for AO began in the early 1990s,[7] with in-focus and out-of-focus images being used to predict tip–tilt[8,9] or Zernike coefficients[10] applied to data from the Multiple Mirror Telescope. Later, a European collaboration worked on a feedforward NN called CARMEN,[11,12] which has seen on-sky extraction of Zernike polynomial coefficients from off-axis WFSMs.[13]

Recent work has revolved around RNN and CNN architectures after their emergence. Wavefront prediction and reconstruction have been demonstrated with simulated data for the Gemini telescope with an NN that combines both LSTMs and CNNs[14] and there has been some success in applying Google's Inception v. 3 CNN to predict Zernike coefficients from in-focus and out-of-focus artificially generated point spread functions (PSFs).[15] CNNs have also been used to reconstruct phase maps of simulated data from the corresponding PSF,[16] and an LSTM has been used to extract wavefront aberrations from in-focus and out-of-focus images with simulated data.[17] In an indirectly related area, feedforward NNs have been used to control a deformable mirror using Shack–Hartmann WFSMs.[18]

Sensor nonlinearities are a major source of error for predictive control. Wavefront sensor nonlinearities are of particular interest as wavefront correction is limited by the precision with which the wavefronts can be measured. An increasingly popular wavefront sensor is the pyramid wavefront sensor (PyWFS),[19] which is employed by first-class AO systems such as the Large Binocular Telescope Adaptive Optics system,[20] the Magellan Adaptive Optics system,[21] and the Subaru Coronagraphic Extreme Adaptive Optics system (SCExAO).[22] There are also plans for the PyWFS to be used in the Giant Magellan Telescope,[23] the European Extremely Large Telescope,[24] and the Thirty Meter Telescope,[25] which are ∼4 times larger in diameter than today's large telescopes.

The PyWFS's competitive edge comes from its ability to maintain sensitivity at all spatial frequencies.[26] Because it requires a diffraction limited PSF to achieve maximum sensitivity, it is typically used in a closed-loop AO system or in conjunction with some initial upstream wavefront correction. Ideally, the PyWFS would be used in a fixed position where the apex of the pyramid lies stationary in the focal plane and delivers extremely sensitive measurements. However, in this mode of operation, the PyWFS is quick to saturate and does so even in the presence of small wavefront aberrations, which cause the PyWFS intensities to become highly nonlinear with respect to the phase. Alternatively, the PyWFS can be modulated to increase the range over which it is linear, but it then operates at a greatly reduced sensitivity. Hence, there is much interest in finding methods that allow the PyWFS to be used with low amplitude modulation.

Here, we present our findings on the performance of a simple feedforward NN with an architecture that is directly comparable to the EOF predictive filter and can trivially be applied to problems using the same framework. In the first half of this paper, we compare the regularization of the linear EOF predictive filter and a linear NN on real data obtained from the SCExAO system at the Subaru telescope. Our findings show that the regularization of the NN is superior to the EOF predictive filter. We then extend to nonlinear NNs, expecting to better predict nonlinearities in the temporal domain. Due to the reconstruction process of our pseudo open-loop telemetry, which involved a linear translation of the wavefront measurements to a modal representation, other nonlinearities were lost. Finally, we generated simulated data with an artificial nonlinear saturation to emulate the saturation of a PyWFS, which we use to demonstrate the advantage of a nonlinear NN over the EOF predictive filter on nonlinear data.

1.1 *Empirical Orthogonal Functions for Wavefront Prediction*

The EOF method is a brute force least-squares minimization currently employed at SCExAO and will serve as a baseline method for comparison. A very brief summary is given here, but further details can be found in Ref. 12.

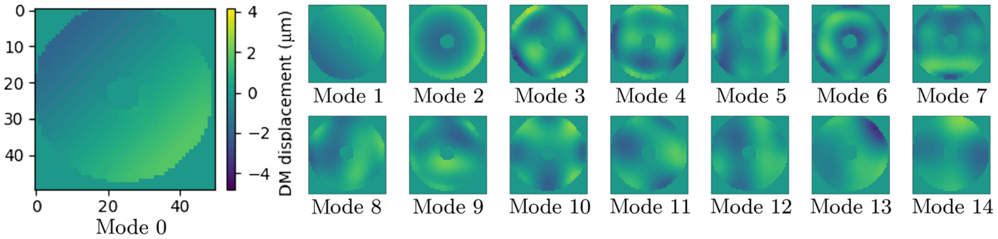


Fig. 1 The first 15 of 1111 wavefront modes used in Eq. (1). The colorbar and pixel scale of modes 1 to 15 are the same as those shown for mode 0. These are 50×50 pixel images that can be directly applied to the SCExAO deformable mirror.

A wavefront seen through a telescope's circular aperture can be represented by a summation of modes, in which each mode is a function across a unit disk representing the telescope aperture. A selection of modes from our mode basis is shown in Fig. 1. An WFSM is described by a vector of mode coefficients where WFSM $= (c_1, c_2, \ldots, c_n)$. The equation to reconstruct the wavefront is then

$$\text{wavefront} = \sum_{i=1}^{n} c_i \text{mode}_i. \tag{1}$$

Predictions are made in the WFSM mode coefficient space, which significantly reduces dimensionality compared with any image representation. When predicting a future WFSM, the number of previous WFSM time samples used is called the order. Due to operations in the AO control loop (prediction calculations, deformable mirror control processes, and video readout), in a single iteration of the loop, there are unseen WFSMs between the last WFSM used in prediction and the WFSM that is predicted. The length of the unseen time interval is lag $\times$ dt, where dt is the time between consecutive measurements. An example is shown in Fig. 2, where the lag renders the $t = 3$ WFSM unavailable.

To make WFSM predictions, the EOF predictive filter is given a set of observed WFSMs called a history vector. This is constructed by stacking WFSM column vectors to create an even taller column vector. The EOF predictive filter (F) is multiplied by the history vector to output a single WFSM prediction.

To derive F, we construct a data matrix (D) comprising horizontally concatenated history vectors and a response matrix R of the corresponding WFSMs to be predicted. The goal is to find F such that

$$F \times D = R. \tag{2}$$

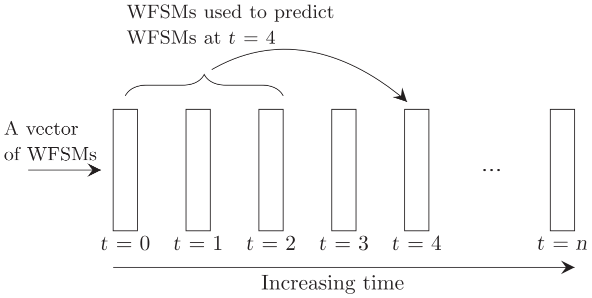


Fig. 2 Example of predicting the WFSM at $t = 4$ where order = 3 and lag = 2. Each WFSM is a column vector of mode coefficients. When predicting, the input to the model is given a history vector, which is constructed from the stacked WFSMs that will be used to make the prediction. In this illustration, the history vector is generated by vertically stacking the WFSMs for $t = 0$,1, and 2 and will be used to predict the WFSM at $t = 4$.

To obtain F, we perform the calculation

$$F = R \times D^{+}. \tag{3}$$

where D^{+} is the Moore–Penrose pseudoinverse of D and is calculated with an SVD as shown in Sec. 2.2.1.

1.2 *Neural Network Basics*

We use the classic feedforward NN (also called a multilayer perceptron), which is composed of layers of neurons with neurons fully connected between adjacent layers (Fig. 3). Each connection is associated with a weight (w), and each neuron is associated with a bias (b). The activation of the neurons in layer i is given as

$$\mathbf{a}_i = \mathbf{W}_{i-1,i}\mathbf{o}_{i-1} + \mathbf{b}_i, \tag{4}$$

with final output

$$\mathbf{o}_i = f(\mathbf{a}_i), \tag{5}$$

where f is the activation function. Common activation functions are the hyperbolic tan (tanh) and ReLU, which are the source of the network's nonlinearities. Note that the activation function is not applied to the final layer; otherwise it would restrict the network's output to the range of the activation function.

$$\text{linear:}\ f(x) = x, \tag{6}$$

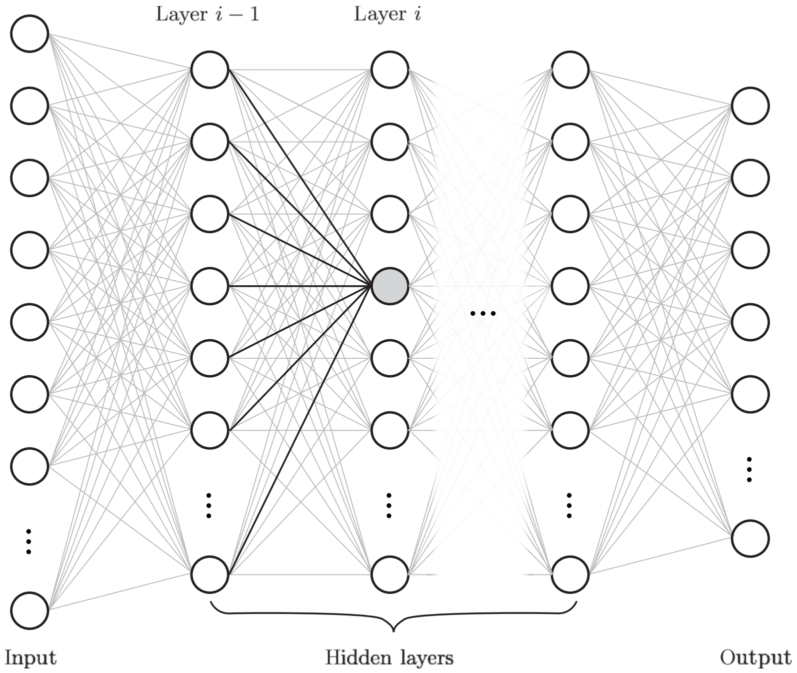


Fig. 3 Example schematic of a feedforward NN. Neurons in adjacent layers are fully connected, and information flows through the network in the figure from left to right, meaning that a single neuron can be influenced by all of the neurons in the previous layer. The activation of a neuron is given in Eq. (4). For example, the activation of the shaded neuron can by calculated by taking the outputs of the previous layer and multiplying these by the strength of the connection (solid black lines) given by the weights before adding the bias. Although this is illustrated graphically, the computation can be performed by a simple matrix multiplication. The final output is the activation function applied to the activation as shown in Eq. (5). The number of layers and the number of neurons in each layer are hyperparameters that must be selected.

$$\tanh: \; f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \tag{7}$$

$$\text{ReLU}: \; f(x) = \max(0, x). \tag{8}$$

Using a linear activation function results in a purely linear network as it reduces the forward propagation through the network down to a series of matrix operations.

For our particular task, the inputs to the NN are the history vectors [i.e., $\mathbf{o}_0$ in Eq. (4) is a history vector], and the desired outputs are the corresponding predicted WFSMs.

An NN is trained over a number of epochs in which the network sees each training sample once. The weights and biases are updated in a process known as backpropagation, which is a gradient descent on all of the model parameters to minimize the objective function, usually a mean-square error, chi-square, or log posterior probability. Gradient descent requires a hyperparmeter known as a learning rate (η), which controls the step size used in the gradient descent. A number of adaptive gradient descent techniques have been developed to optimize the learning process. We used a popular technique known as Adam,[27] which scales η for each parameter and constantly updates η from an exponentially weighted average of accumulated gradients.

We implemented our networks in Keras[28] using Tensorflow[29] as a backend for automatic differentiation of the objective function and Talos[30] as a hyperparameter optimizer.

2 Evaluation of EOF Predictive Filter and NNS on Real Data

2.1 *Dataset*

The wavefront data used here were obtained from the Subaru telescope using the SCExAO instrument.[31] The SCExAO instrument is a high-performance AO system designed for extreme AO that operates faster and with more actuators (2000-actuator deformable mirror) than standard AO systems. It uses a phase-induced amplitude apodization coronagraph, optimal for a small inner working angle,[31] and a highly sensitive PyWFS. It also benefits from upstream AO correction by AO188, named for its 188 actuator deformable mirror.

We used pseudo open-loop telemetry, which consists of approximations of the true aberrated wavefront but reconstructed from corrected wavefronts acquired while the SCExAO control loop is closed and correcting. The final dataset comprises wavefronts obtained from the SCExAO deformable mirror (DM) correction. These are the residual wavefronts from AO188.

Our primary reason for using pseudo open-loop telemetry rather than closed-loop telemetry is that our predictive control methods did not depend on the performance of the AO control loop. It also allowed us to make a direct comparison of the NNs with the currently used EOF method.

The data were obtained on May 9, 2018, under poor and rapidly evolving seeing conditions (seeing $\sim 0.9''$), which is representative of conditions in which nonlinearities diminish the performance of the AO system. The observation target was Altair, an unresolved star with an apparent visual magnitude of 0.76 in the V band.[32]

The data are represented in mode coefficient space, with the mode basis (Fig. 1) constructed from an SVD applied to wavefront data. The pseudo open-loop telemetry was derived from a real-time summation of wavefront sensor residauls and DM correction on a common modals basis. The modal WFS residuals coefficients were computed by linear (matrix-vector multiplication) processing of the PyWFS images. Although this is a reasonable approximation in closed-loop operation in which WFS residuals are small, the pseudo open-loop computation ignores DM and WFS calibration errors, which contain both linear and nonlinear terms. A significant contributor to WFS calibration errors is the pyramid WFS optical gain, which weakens the sensor response when the wavefront quality is poor. By comparing multiple response matrices acquired on-sky, we have measured optical gain variations of $\sim 15\%$, which accounts for $\sim 1\%$ of the overall error. The amplitude of other errors is largely unknown, and their effect on predictive control performance remains an open question.

Ultimately, the pseudo open-loop telemetry is the best linear reconstruction of the aberrated wavefronts. Although this representation is sufficient for the current linearly optimal EOF framework, the loss of nonlinear wavefront information impairs predictive capabilities. The data does however retain temporal nonlinearities, but these are expected to be small, with the major source of nonlinearities arising from within the telescope. In our case, the primary source of nonlinearities arise from the DM and WFS calibration errors as the calibration is performed linearly. The AO control loop is operated at 3 kHz with each WFSM comprising 1111 mode coefficients. Our data suffer from periodically missing values due to internal processes overloading the computer's resources. These occur regularly at ~1 s intervals, each time dropping on the order of 5 WFSMs. This is remedied by linearly interpolating the missing values.

The training set was composed of 2 min of consecutive data. In experiments in which only a subsection of training data was used, the samples were taken from the start of the training set. The validation data is composed of 14 sets of 10 s samples taken at 2-min intervals, which is representative of atmospheric conditions over a 30-min period. The test data had 12 sets of 10 s samples taken over the same period but with samples offset from the validation data by 1 min.

The goal of the models was to predict future wavefronts from the current wavefront in the mode coefficient basis from the on-sky data. This was done offline in the absence of time constraints.

2.2 *Regularization Comparison*

When model fitting, there is always a concern that the model may become overly complex as the model begins to memorize the noise in the data instead of learning the underlying trends. As a result, the model will perform extremely well on the training data but poorly on unseen data. This is called overfitting, and it can be prevented by regularization.

Regularization is used to reduce the complexity of a model. The amount of regularization is controlled by hyperparameters, selected in the process of hyperparameter tuning. We used holdout, a simple validation method in which models are trained with different amounts of regularization before being evaluated on the validation data. The model that performs best on the validation data is then expected to be the model that generalizes best to unseen data and should be complex enough to learn the trends of the data but not so complex that it also learns the noise.

2.2.1 *Regularization methods*

Generation of the EOF predictive filter F requires us to calculate D^+ [Eq. (3)]. To do so, we perform an SVD on D, which is an $m \times n$ matrix where $m = \text{order} \times 1111$ modes and n = number of training samples. D is decomposed into three matrices: U, Σ, and V^* such that,

$$D = U\Sigma V^*, \tag{9}$$

where U and V are square unitary matrices with dimensions $m \times m$ and $n \times n$, respectively, Σ is an $m \times n$ diagonal matrix, and V^* is the conjugate transpose of V. (It is worth noting that, for a unitary matrix M, $M^* = M^{-1}$).

The pseudoinverse of D is then,

$$D^+ = V\Sigma^+ U^*, \tag{10}$$

where Σ^+ is the pseudoinverse of Σ.

To regularize the EOF predictive filters, the values in Σ are truncated. Σ is a diagonal matrix with non-negative real values (singular values) decreasing in size down the diagonal. To regularize, a threshold value is selected and every singular value below the threshold is set to 0 (Fig. 4). Small singular values signify lower importance and result in small contributions from

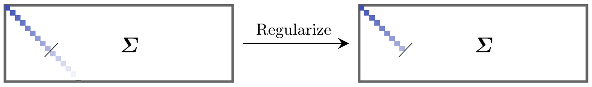


Fig. 4 Σ is a diagonal matrix with values decreasing down the diagonal and is used to calculate the EOF predictive filter. To regularize, we pick a threshold value, and every value below that threshold value is set to 0. In general, these small values encode noise.

the corresponding elements in V and U when generating $D+$. Such values largely correspond to noise, which is why eliminating them is an effective means of regularization.

Other forms of SVD regularization, such as the Tikhonov method,[33] exist but are not explored here. No SVD regularization technique has been established as the gold standard, and the regularization method is chosen through methods such as cross-validation. Although the exploration of other SVD regularization techniques is interesting and important, it is beyond the scope of this work.

NNs are commonly regularized by L1 and L2 regularization, originally developed for regression. Although these can be applied to both weights and biases, typically the number of weight parameters is significantly larger than the number of bias parameters, so it is common practice to only regularize the weights, as we do here.

L1 regularization (Lasso regression) encourages sparsity, which used here will effectively reduce the number of connections in the network by driving the weights to 0, and L2 regularization (ridge regression) will force the weights to be small. This is done by adding penalty terms p_1 and p_2 to the objective function weighted by hyperparameters λ_1 and λ_2, respectively.

$$p_1 = \lambda_1 \sum |w_i|, \tag{11}$$

$$p_2 = \lambda_2 \sum w_i^2. \tag{12}$$

The full objective function is then

$$\text{objective function} = \text{error metric} + p_1 + p_2 \quad = MSE + \lambda_1 \sum |w_i| + \lambda_2 \sum w_i^2. \tag{13}$$

Another popular method of regularization is dropout.[34] During the training process, different sets of neurons are randomly removed (or “dropped out”). The fraction of neurons that remain is called the dropout rate. Dropout is effective because it prevents the network from becoming over reliant on a small subset of inputs and neurons. It also encourages the network to behave like an ensemble of smaller networks—these networks being a reduced form of the original network as neurons are dropped out.

2.2.2 *Model comparison*

In this first section, we compare the regularization capabilities of the EOF predictive filter and the NN on the same on-sky data. Experiments were conducted using the same 5, 10, 15, 20, 30, and 40 s training samples for each model. Due to memory constraints (arising from the SVD computations), the maximum training size was 40 s of data, and we required order $= 10$ to be small. We set lag $= 2$, realistic for SCExAO. For a fair comparison, the NN was linear and had no hidden layers, so the complexity of the NN and the EOF predictive filter were approximately the same. The NN weight matrix (W) and the EOF predictive filter (F) were both of dimensions $(1111 \times 11,110)$. There were 11,110 input values (for 10 stacked WFSMs of 1111 mode coefficients) and 1111 output values (for the predicted WFSM). The complete equations for these models are given below, where x is an input history vector.

$$\text{output NN} = \mathbf{W}x + b, \tag{14}$$

$$\text{output EOF} = Fx. \tag{15}$$

The key difference between these two models is that the NN has biases (b), which is standard for an NN. Although the EOF framework allows for the EOF predictive filter to have biases, these are not used in current predictive control implementations and have been excluded here.

For each experiment (using 5, 10, 15, 20, 30, or 40 s of training data), we identified the optimal regularization hyperparameters for our models. We scanned through different threshold values when calculating the EOF predictive filters and evaluated their performance with hold-out

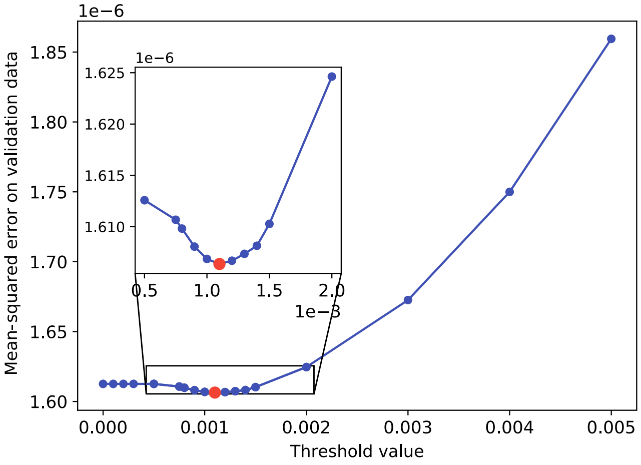


Fig. 5 A typical threshold sweep to determine the optimal regularization for the EOF predictive filter. For this example, the threshold value of 0.0011 corresponded to the lowest MSE on the validation data (marked as the large red point).

validation. It is worth noting that the validation and test sets are larger than the training set. This is atypical and is due to the training size being limited by memory requirements for constructing the EOF predictive filter rather than the availability of the data. An example of one of these threshold sweeps is given in Fig. 5. In our experiments, the threshold values ranged from 2×10^{-2} to 4×10^{-4}. We used holdout to determine the optimum threshold value for each of the EOF filters.

The NNs were regularized using elastic net regularization (i.e., using L1 and L2 regularization) on the weights and trained for 1000 epochs. The hyperparameters for these networks are shown in Table 1 and were found by performing a hyperparameter scan using Talos to identify a good learning rate (η) and regularization parameters (λ_1, λ_2). We also benefited by reducing the learning rate of our models when the objective function plateaued, using the learning rate reduction hyperparameter f_η provided by Keras. By default, if the loss function plateaus over five epochs, the learning rate is reduced so that the new learning rate is given by $\eta' = f_\eta \eta$. The learning rate was reduced under this condition until $\eta_{\text{min}} = 10^{-7}$ was reached. Each NN was trained on the same training sets as the EOF predictive filter. We used the mean squared error (MSE) in mode coefficient space as our objective function.

A comparison of the performance of both the EOF predictive filter and the NN was evaluated on the test data and is shown in Fig. 6.

Table 1 Optimal hyperparameters defining the NNs used to generate the results in Fig. 6. η is the learning rate, f_η is the learning rate reduction factor, and λ_1 and λ_2 are the L1 and L2 regularization parameters, respectively.

Training data (s)	η	f_η	λ_1	λ_2
5	0.1	0.5	4×10^{-9}	8×10^{-11}
10	0.01	0.7	2×10^{-9}	5×10^{-10}
15	0.01	0.5	8×10^{-10}	8×10^{-11}
20	0.01	0.6	8×10^{-10}	8×10^{-11}
30	0.1	0.6	8×10^{-10}	0
40	0.1	0.5	8×10^{-10}	1×10^{-11}

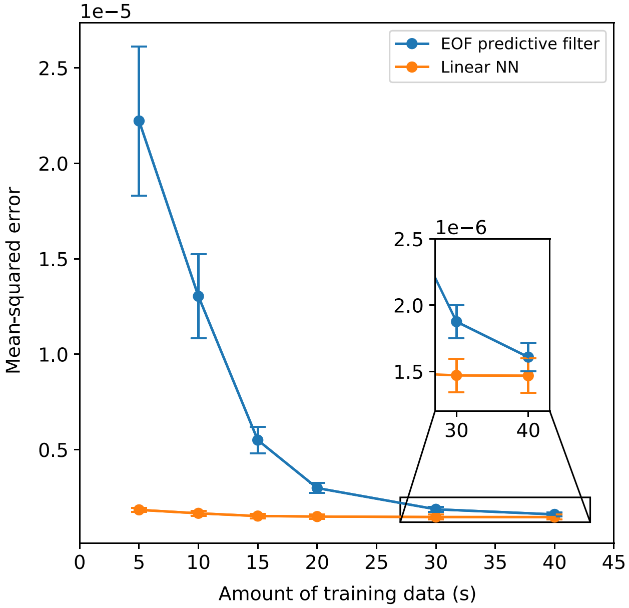


Fig. 6 Performance of the EOF predictive filters and the linear NNs on test data when trained on varying amounts of data. The MSE is given in mode coefficient space and is unitless. The error bars show the standard deviation of the MSE over the test set. Interpretable visualizations of the MSE values are shown in Fig. 7.

2.2.3 *Discussion and Results*

Observation of Fig. 6 shows that with limited training data the NN predicts significantly better than the EOF predictive filter. The models differ only on the presence of biases (in the NNs) and in the regularization methods. The 1111 biases do not significantly increase the complexity of the NNs, which already have $111 \times 11,1110$ weight parameters. Further, setting the biases of a trained NN to 0 resulted in negligible change in the MSE, which proves that the superior performance of the NN was due to better regularization. The NN regularization techniques are already more complex than those for the EOF predictive filter purely by the number of regularization parameters.

Attempts to apply dropout were unsuccessful. This is likely due to the NNs being shallow and relatively small. Dropout typically requires wider network architectures to compensate for missing neurons.

Figure 7 helps to interpret the MSE values given in Fig. 6, which are in mode coefficient space. Figure 7 shows the predictions of a typical wavefront using the EOF predictive filters and the NNs for 5 and 40 s of training data. We can see that the reconstruction by the EOF predictive filter trained on 5 s of training data only vaguely captures the phase features across the wavefront, whereas the 5 s NN does almost as well as the 40 s models.

Although not illustrated here, investigation into the residuals on a mode-by-mode basis for the 5 s training data models shows that the NN outperforms the EOF predictive filter for all modes. The magnitude of these residuals are roughly proportional to the magnitude of the mode.

The strong regularization capabilities exhibited by the NN point toward a promising future for the application of NNs in predictive control. This suggests that an NN may better generalize to unseen atmospheric conditions than the EOF predictive filter. Whereas our experiments only used up to 40 s of training data (due to memory required for the SVD computation), in practice, 1 to 2 min of training data is used to compute the EOF predictive filter. This has to be regularly updated ($\sim$ every minute) during an observing night if seeing conditions diverge. The predictive filter adjusts slowly to temporal non-stationarity with a response time of ~ 5 min.

It has also been observed that the EOF does not gain from training on large amounts of training data (>2 min). This is likely due to the EOF predictive filter being unable to handle large changes in the underlying statistics describing the atmospheric turbulence. It is possible that the EOF method may match the NN's performance with ~ 1.5 min of training data, but its

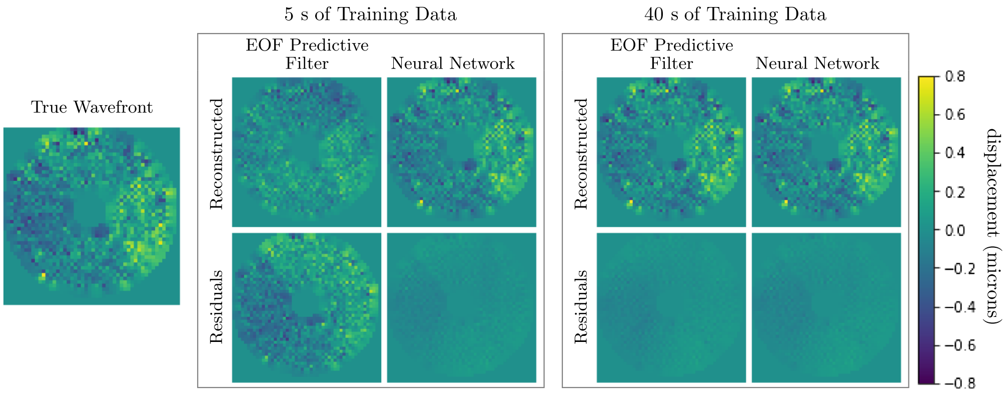


Fig. 7 This shows an example of a wavefront reconstructed by EOF predictive filters and linear NNs (with no hidden layers) for 5 and 40 s of training data. Here, we use on-sky pseudo open-loop data accumulated on tMay 9, 2018, observing the bright unresolved source Altair during poor seeing conditions (seeing ∼0.9″). These reconstructions provide a tangible interpretation of the performance gap between the EOF predictive filters and linear NNs observed in Fig. 6.

performance is expected to diminish with more training data. By contrast, the NN should improve with more training data.

It is unknown how well the NN will perform on new seeing in which the turbulence statistics are significantly different than those on which it has trained. This will likely depend on the architecture of the network. It is expected that the NN will be more adaptable to new seeing conditions than EOF, which is known to generalize poorly. It is also conceivable that an NN could be extensively pre-trained and exposed to many observing conditions. Theoretically, it should generalize well and offer good performance under a diverse range of conditions. Additionally, the network could be continued to be trained with on-sky data during the observing run to help it learn the current conditions. Given that the standard deviation of NN's MSE was also significantly smaller than that of the EOF predictive filter, we also expect the performance of the NN to be more reliable.

2.3 *Comparison of Linear and Non-Linear Networks*

2.3.1 *Model comparison*

One major advantage of the NNs over the EOF predictive filter is their ability to model nonlinearities. Our data contain the best linear approximation of the aberrated wavefronts, so we only expect to find nonlinearities in the temporal domain. Thus we increase order (=200) to be 20 times longer than before but keep the lag (=2) the same. This corresponds to the models seeing 0.2 s of data for a prediction 2 ms into the future (Fig. 8). Due to the memory required for constructing the EOF predictive filter, we reduced the number of modes from 1111 down to 2 (mode 0, tip and mode 1, tilt). This made the input length size 400 and the output size 2. Our models were trained on 40 s of training data.

We performed a threshold sweep for the EOF predictive filter but found that it did not benefit from regularization. This is consistent with what is observed in practice when the data matrix is overconstrained.

We trained NNs with three hidden layers of 5000 neurons each. We expanded the architecture of our previous networks to make them wider and deeper to benefit from the added complexities that the NN could offer over the EOF predictive filter. This included a much larger number of model parameters and capacity to handle nonlinearities. We experimented with the popular linear, tanh, and ReLU activation functions.

The linear activation function allowed for direct comparison with the EOF predictive filter, and the tanh activation function (which is linear near the origin) allowed us to compare with the linear activation function. By default, we initialized with Keras' default initialization (weights initialized using Glorot uniform initializer[35] and biases initialized to 0). As a result, the state of

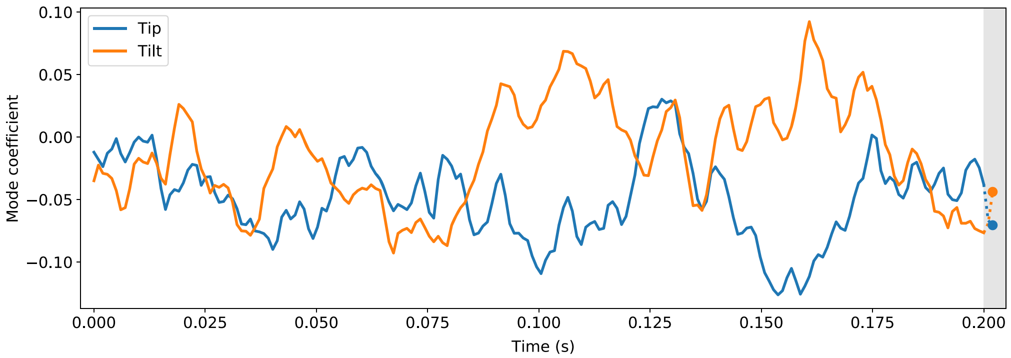


Fig. 8 Typical sample of tip–tilt data for order = 200 and lag = 2. This corresponds to a 0.2 s memory length of data (solid line in white region), which would be used by the algorithm to predict the true value (shown as a point in the shaded region) 2 ms into the future. The dotted lines in the shaded region show data that is unseen by the model when making a prediction.

the initial untrained network has activations close to 0. This causes the tanh network to begin in the linear regime and makes it difficult for the network to learn nonlinearities. Hence, we initialized the weights and biases of the tanh networks to follow a normal distribution with a standard deviation of 5, which takes the initial activations well outside the linear region.

Like the EOF predictive filter, we found that the linear NN did not benefit from regularization. By contrast, the ReLU and tanh models did benefit from regularization, with optimal hyperparameters identified using Talos and tabulated in Table 2. Each network was optimized with Adam for 1000 epochs with $f_\eta = 0.7$ and $\eta_{\min} = 10^{-7}$.

2.3.2 *Discussion and results*

We found that the EOF predictive filter (model 1) and the linear NN (model 2) performed equally well on our tip–tilt data. We did not observe the regularization gap between the EOF predictive filter and the linear NN that is noticeable in the inset of Fig. 6, which is expected given that these models did not benefit from regularization. It is worth noting that the MSE values from our tip–tilt analysis are not directly comparable to those in Fig. 6. This is because most of the power is in the tip–tilt modes and the magnitude of the tip–tilt coefficients are on average 10 times larger than the other modes, so the MSE values we report are similarly larger.

Interestingly, the EOF predictive filter has only 800 parameters compared with the ≈50 million of the linear NN. Because these models are not benefitting from regularization (models have ample training data) or increased complexity (EOF predictive filter has enough

Table 2 Performance and hyperparameters of the EOF predictive filter and various NNs on tip–tilt data predicting with order = 200 (corresponding to 0.2 s) and lag = 2.

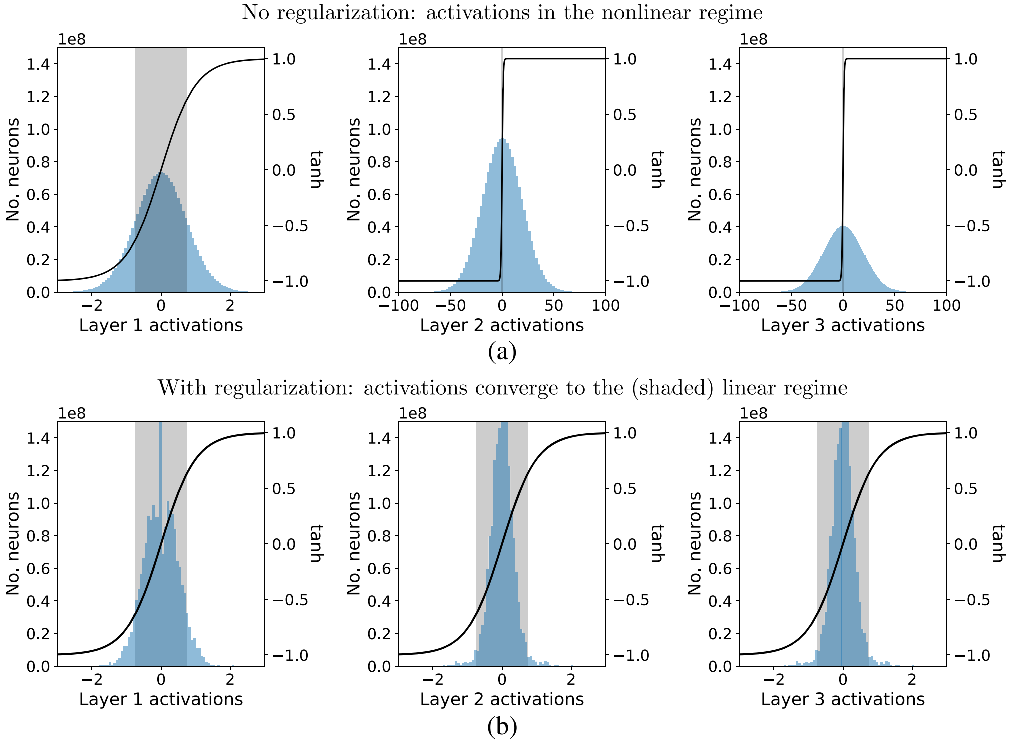
			Hidden layers		Output layer		MSE ($\times10^{-4}$)	
No.	Model description	η	λ_1	λ_2	λ_1	λ_2	Training data	Test data
1	EOF predictive filter (no reg)	—	—	—	—	—	1.426	2.4882
2	NN linear (no reg)	10^{-4}	—	—	—	—	1.436	2.4886
3	NN tanh (no reg)	10^{-5}	—	—	—	—	2.395×10^{-5}	14.4636
4	NN Relu (no reg)	10^{-4}	—	—	—	—	1.2474×10^{-4}	6.8841
5	NN tanh	10^{-5}	0	$\times10^{-5}$	0	0	1.4407	2.5389
6	NN Relu	10^{-3}	5×10^{-8}	10^{-9}	10^{-8}	0	1.4101	2.5482

parameters), it appears that these linear models have reached the limit of the predictive capabilities on this particular dataset.

We turn now to our nonlinear NNs with no regularization (models 3 and 4). These achieve an extremely small MSE on the training data but a relatively large MSE on the test data, which means that these models have enough complexity to overfit the data. Unlike our linear models, these would benefit from regularization. Because these networks have the same architectures as the linear NN, the added complexity is due to the nonlinear tanh and ReLU activation functions.

Using elastic net regularization reduced overfitting in our nonlinear models (models 5 and 6), however, these models only matched the performance of our linear models but did not surpass them. The fact that their MSEs coincide with the linear models suggest that regularization drove the nonlinear models toward the linear solution.

Proof of the tanh model moving into the linear regime can be shown by examining the activations. Figure 9 shows the activations of our tanh models (models 3 and 5). The activations for the unregularized tanh network are well outside the linear regime for the deeper layers. By contrast, the activations for the regularized tanh are close to zero and populate the region where tanh is approximately linear. Replacing the tanh function with a linear function of the form $f(x) = mx$, where m matches the gradient of the tanh central region, should only slightly perturb the model. This approximation would allow the forward propagation through the network to be written as a series of matrix operations. The conclusion drawn is that regularization of the tanh network recovers the same optimal linear solution located by our linear models. Note that, if modeling nonlinearities benefitted our tanh network, regularization should not have driven it to the linear regime.



(a)

(b)

Fig. 9 Activations [Eq. (4)] of our tanh models for the three hidden layers with the tanh function plotted over the top. $|x| < 0.75$ shades the region where tanh is approximately linear. (a) Model 3, no regularization. The activations of the second and third hidden layers stretch well beyond the region in which tanh is linear. The activations of the first layer show only moderate use of the tanh nonlinear region, likely due to small inputs (on the order of 10^{-3}), which keeps the activations small. (b) Model 5, with regularization. Most of the activations are roughly within the linear regime of the tanh function with the activations of the deeper layers lying almost exclusively within the shaded region. A close linear approximation of this network would be to replace the tanh activation function with a linear function of the form $f(x) = mx$, where m matches the gradient of the tanh central slope.

It could be argued that, because L2 regularization encourages small weights, the linearization of our tanh model is a consequence of our regularization choice rather than a global optimum. Our counter argument is that we see the same performance from our regularized ReLU model which is not explicitly forced into the linear regime with L1 or L2 regularization.

Attempts were made to apply dropout, but dropout alone was unable to replicate the low MSE achieved by elastic net regularization.

There is a small but notable difference between the performance of our linear models and our regularized nonlinear networks. Under our belief that the nonlinear models are regularized toward the linear solution, we should expect the MSEs to be the same. However, we observe that the purely linear models offer better performances. This is likely because it is a lot easier for the linear models to locate the linear global solution than it is for our nonlinear models as the error space of the nonlinear models would be more complex. We also expect that the linear models are initially closer to the global solution.

Our nonlinear models locating a linear optimal solution is evidence that the predictive information in the data is linear and that only the noise is nonlinear. It appears that our models are not able to benefit from temporal nonlinearities. It is possible that at these timescales our data is predominantly linear and that any nonlinearities are masked by noise. Although we may benefit more from temporal nonlinearities at an increased lag, the improvements would be inconsequential to our application as the lag we have used is reasonable for an ExAO loop, so there is no need to predict further into the future.

As discussed, we did not witness any benefit from modeling nonlinearities. This is unsurprising as the major sources of nonlinearities in the AO control loop are spatial (arising from the instrumentation). These were not present in the data, which only retained temporal nonlinearities. Whether we may benefit from the temporal nonlinearities across different observing conditions is unknown and a subject for future work.

3 Simulated Nonlinear Data

3.1 *PyWFS: Source of Nonlinearity*

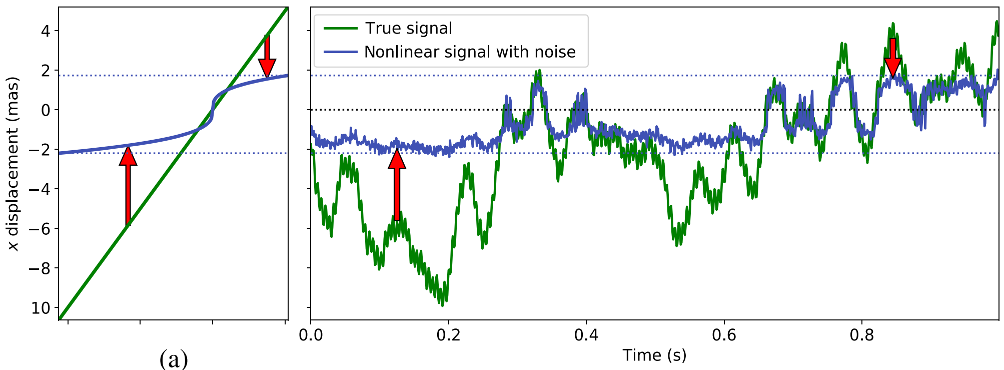
The pseudo open-loop telemetry that SCExAO produces is calculated by its existing linear reconstruction algorithm. Consequently, only linear information is retained when interpreting the output of sensors (such as the PyWFS), which during bad seeing conditions, is far from the true nonlinear response.

Because nonlinearities were not preserved in the SCExAO pseudo open-loop data, simulated data were created to probe the NNs handing of nonlinearities. In a true open-loop system at SCExAO, a major contributor to the nonlinearities is known to arise from the saturation of the PyWFS as predictions of the true wavefront must be made from the saturated measurements. As a proof of concept, we demonstrate the performance of the EOF predictive filter and nonlinear NNs on simulated data with artificial PyWFS saturation to show that the NN is capable of learning the nonlinear relationship.

3.2 *Data Generation*

Our simulated tip–tilt data were created as a summed spectrum of 2D sinusoidal functions to mimic telescope vibrations. These are a primary error source for SCExAO in closed-loop operation[36] and need to be corrected by the AO system. These vibrations are predominantly mechanical in nature and have various sources stemming from environment conditions or internal mechanisms of the telescope and predominantly affect low order modes.[37]

The locus of each vibration is a straight line centered on the origin, at an angle $\theta \sim \mathrm{U}[-\frac{\pi}{2}, \frac{\pi}{2})$ and a maximum displacement $A \sim \mathrm{U}[0.2, 1.5)$ mas. The displacement along the line was described by a sine function, with phase $\phi \sim \mathrm{U}[0,2\pi)$. The frequencies of the vibrations were produced from a power law distribution $f^{-1.5}$ with bounds [0, 199] Hz. The selected frequencies ranged from 0.09 to 173 Hz. With a sampling rate of 1 kHz, the maximum observed frequency



(a)

(b)

Fig. 10 Visualization of how the cube root function in Eq. (16) saturates our simulated data. (a) The green line represents a series of x displacements. Applying Eq. (16) maps the x displacements to the blue line as per the red arrows. (b) In green is a sample of our simulated data. This is then saturated using the mapping on the left to obtain the blue nonlinear signal that then has Gaussian noise added to it.

was well below the Nyquist frequency. This produced correlated displacements in the x and y directions, simulating tip–tilt.

Although it is important that we are able to predict these vibrations, our principle interest is whether our nonlinear NNs are able to also predict the true signal from the saturated signal, which is a nonlinear transformation. To introduce a nonlinear saturation into our data, we use the cube root function given in Eq. (16), which is independently applied to the x and y components of our data. Figure 10 shows how the cube root function is applied to the data, simulating a saturated wavefront sensor. It is true that, for $|x| < 1$, the cube root function magnifies the signal, but this is small compared with the saturation for $|x| > 1$ and can be considered approximately linear. Our key interest is in how well our models handle large saturations.

$$f(x) = \begin{cases} \sqrt[3]{x}, & \text{for } x \geq 0 \\ -\sqrt[3]{-x} & \text{for } x < 0 \end{cases}. \tag{16}$$

We also add Gaussian noise ($\sigma = 0.16$ mas) along each axis of the nonlinear signal to simulate photon noise. This is a reasonable amount of noise for an observation in the H-band at 1 kHz of a $m_H = 9.05$ star using an 8-m telescope.[2]

In these experiments order $= 800$ (corresponding to a memory length of 0.8 s) and lag $= 3$ as used in Ref. 2. A vibration with $f = 1.25$ Hz would contribute exactly one period of oscillation in a single input sample to the EOF filter or the NN. Of the 50 frequencies, 25 were below 1.25 Hz, which means half of the oscillations do not complete a full cycle in the 0.8 s of data used for a prediction.

3.3 *Model Training*

The EOF predictive filter had 2×1600 parameters for order $= 800$. Again, we regularized with a threshold sweep and found the best threshold value to be 0.014. We used an NN with a single hidden layer of 1000 neurons and a ReLU activation function. The NN had 1 603 002 parameters, significantly more than our EOF predictive filter.

We used L1 ($\lambda_1 = 0.0008$) and L2 ($\lambda_2 = 0.0005$) regularization on the weights as this resulted in better performance than dropout. Making the network deeper and using tanh activations offered approximately equal performance to the ReLU model, so they were not pursued.

The models were trained on 60 s of training data. We trained the EOF predictive filter on different amounts of training data to verify that 60 s was enough to ensure that the EOF predictive filter did not suffer from the effects shown earlier in Fig. 6. This ensures that any difference in the performance of the EOF predictive filter and the nonlinear NNs is due to the NNs nonlinear capacity rather than the models being poorly constrained.

Table 3 Performance of the EOF predictive filter and the ReLU NN on the simulated data with artificial saturation.

Model	MSE on test data (mas^2)
EOF predictive filter	0.9268
NN	0.1909

3.4 *Discussion and Results*

Overall, the NN makes more accurate predictions than the EOF predictive filter. The MSEs that each model achieves on the test data are given in Table 3. A visualization of the performance of these models is given in Fig. 11. This figure shows a 2D histogram (on a log color scale) of the models' residuals (for both tip–tilt modes) spread over the x (and y) displacements of the true signal. Overplotted in black is the saturation function, which can be derived by subtracting the blue line from the green line in Fig. 10 and shows the amount of saturation the signal would have experienced for a given amplitude. The residuals for a model that predicts perfectly but does not address saturation effects would follow the black line. We observe that the rotational symmetry of Figs. 10(a) and 10(b) is a result of the rotational symmetry of the cube root function about the origin.

The most striking contrast between Figs. 11(a) and 11(b) is that the residuals of the EOF predictive filter resembles the saturation function in shape, which suggests that the EOF predictive filter has failed to model the nonlinear cube root function. Further, at large saturations, the EOF predictive filter tends to greatly underestimate the true signal. By contrast, the shape of the NNs residuals does not resemble the saturation function. In fact, the long horizontally-shaped cluster illustrates that the NN has successfully learned the saturation function with the remnants likely due to noise.

Figure 12 shows a typical example of the predictive capabilities of the two models on samples of test data. Although the data was constructed with 2D oscillations, we show only the displacement projected onto the x axis. Figure 12(a) shows a region where the mapping from the true signal to the nonlinear signal is close to linear. When the input signal is approximately linear ($|x\,\text{displacement}| < 1$), there appears to be little difference between the performance of the EOF predictive filter and the NN. However, the structures in the residuals of the EOF predictive filter shown in Fig. 11 and the comparatively larger residuals identify that, even at these small displacements, the NN is superior to the EOF predictive filter.

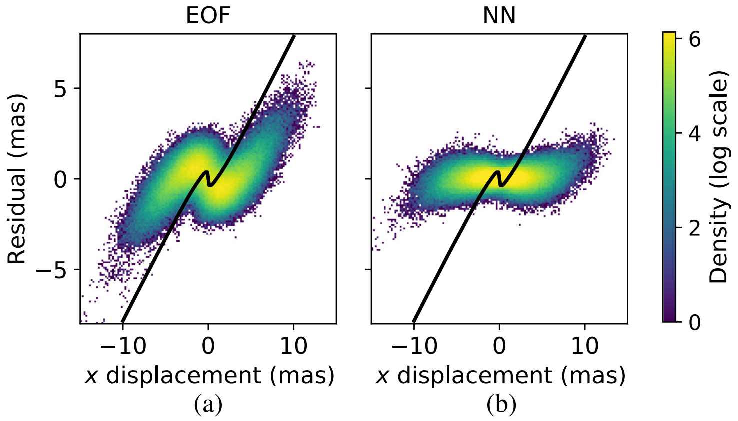


Fig. 11 2D histograms showing the tip–tilt residuals of our models when presented with different amounts of saturation. The x axis corresponds to x (or y) displacements of the true signal as shown in Fig. 10. The black line shows the amount of saturation for a point at that amplitude. The residuals of a model that predicts perfectly but does not account for saturation would lie on this line.

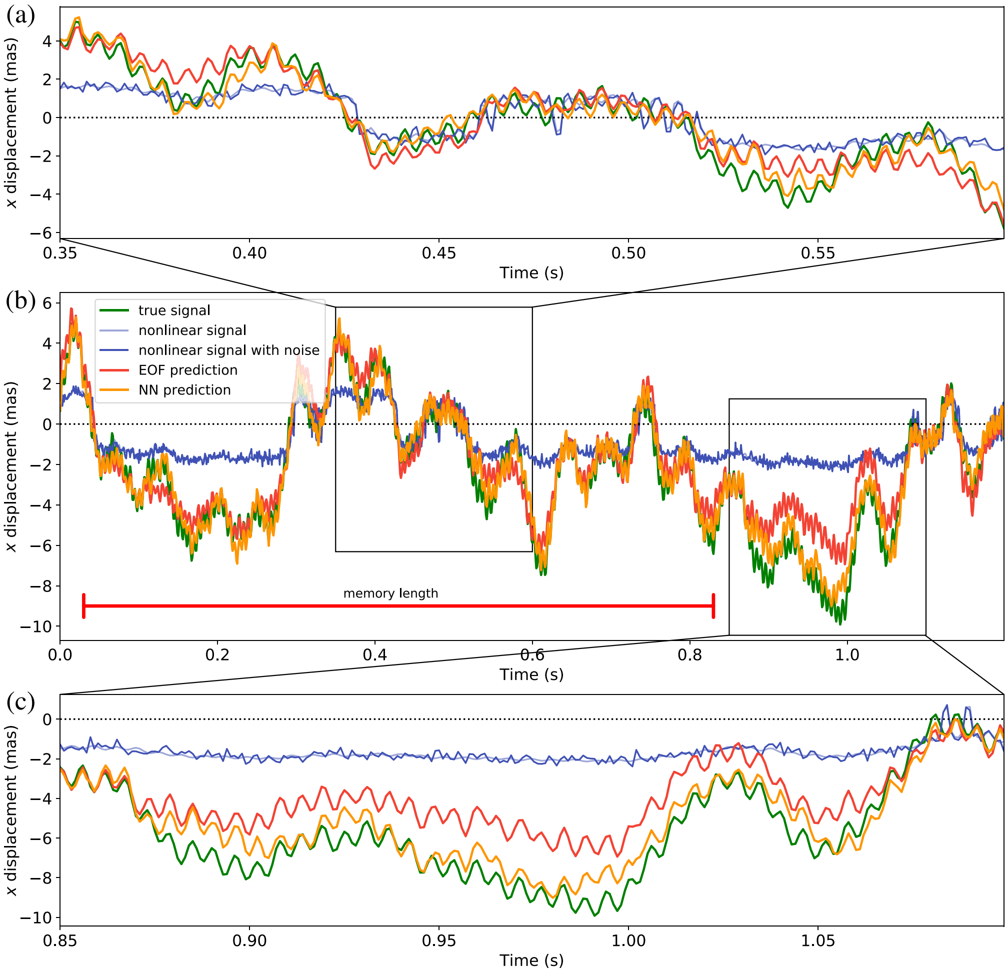


Fig. 12 Predicting a nonlinear signal with noise. The true signal is shown in green. A cube root function [Eq. (16)] is applied to the true signal to produce the nonlinear signal (light blue). The input into the models is the nonlinear signal with Gaussian noise ($\sigma = 0.16$) (blue). The predictions of the EOF predictive filter and the NN are shown in red and yellow, respectively. The nonlinear signals are shifted with respect to the true signal due to lag. The memory length shows the timescale covered by a single input to the models. (a) and (c) highlight regions where the true signal is close to and far from the origin, respecitvely. (a) The mapping from the true signal to the nonlinear signal is approximately linear, and by eye, it is not obvious which model is the better predictor. However, inspection of Fig. 11 (which shows the residuals) tells us that the NN performs better than the EOF predictive filter even with low saturation. (c) Highlights a period of large excursion from the origin, where it is obvious that the EOF underestimates the amplitude of the true signal whereas the NN makes a more accurate prediction.

Figure 12(c) shows that, in the highly nonlinear case, at high excursions from the origin ($|x\,\text{displacement}| > 6$), the NN performs significantly better than the EOF predictive filter, a result clearly observable in Fig. 11; this is also observed at moderate displacements ($1 < |x\,\text{displacement}| < 5$) but to a smaller degree.

Figure 13 shows the behavior of our models at longer time scales, with subsequent panels showing 10× more data. The curves are smoothed with a Gaussian filter ($\sigma = 5$, 10, 700 for the three panels), and the unfiltered data are kept in the background for reference. Even at these timescales, we observe that the NN is a much more successful predictor than the EOF predictive filter.

The data modeled here only simulates tip–tilt, but similar results are expected when extending to higher order modes. Both the EOF predictive filter and the NN are able to learn cross-coupling relationships between these modes. These cross-coupling coefficients manifest in the EOF predictive filter as the off-diagonal terms, whereas the NN learns these relationship

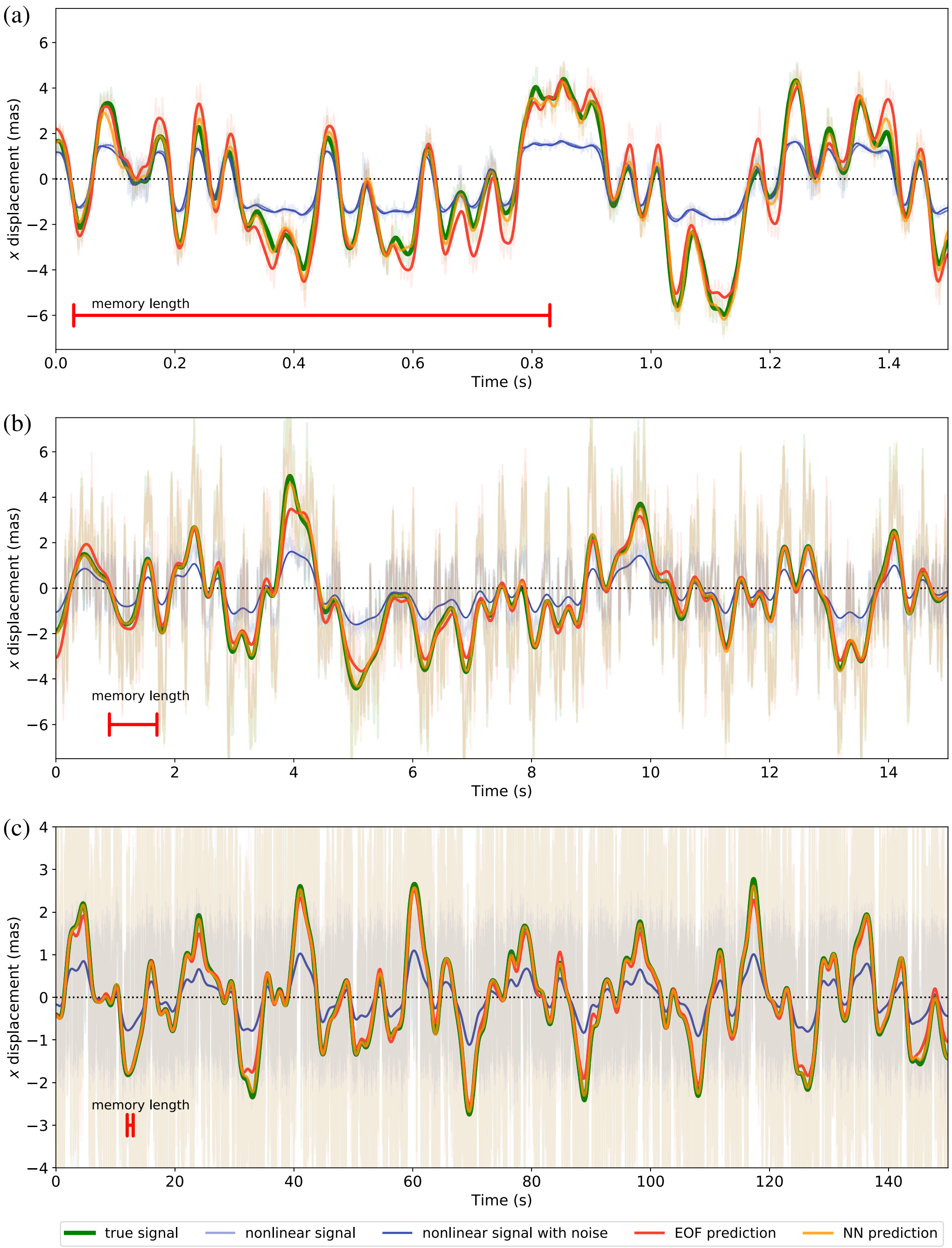


Fig. 13 As Fig. 12 but on different timescales. A Gaussian filter is applied to each panel with $\sigma = 5{,}100{,}700$ from (a) to (c). The unfiltered data are shown in the background for reference.

automatically because the network layers are fully connected. A key distinction is that the EOF is still limited to learning linear relationships, whereas the NN is not.

4 Conclusions

Here, we have demonstrated the application of NNs to predictive wavefront control, with results comparing favourably with the existing EOF approach and offering several major advantages. The NN framework allows us to easily extend model complexity in the number of model parameters and the inclusion of nonlinearities. We also have the capacity to increase the volume of training data, benefit from more powerful regularization, and integrate inputs from multiple sensors, which will allow for more accurate predictions. Additionally, there is the potential to adopt more specialized architectures offered by RNNs and CNNs. These architectures present themselves as a more powerful tool as they should seamlessly expand to integrate temporal information, allow for alternate wavefront representations, and incorporate data from a diverse range of sensors.

Our results show that elastic net regularization for an NN is superior to the truncated regularization of the EOF predictive filter, which allows the NN to perform well even when trained on a limited amount of data. These experiments were conducted with on-sky pseudo open-loop telemetry obtained by the SCExAO system with wavefront data represented by mode coefficients. For a small training size of 5 s of data, the NN achieves an MSE of 1.1×10^{-7}, which is less than a tenth of that achieved by the EOF predictive filter ($\mathrm{MSE} = 1.8 \times 10^{-6}$) and with a much smaller standard deviation ($\sigma = 1.1 \times 10^{-7}$ and $\sigma = 3.9 \times 10^{-6}$, respectively). This suggests that the NN may generalize better and perform more reliably than the EOF predictive filter under diverse seeing conditions.

As a proof of concept, we have shown that an NN is capable of modeling nonlinearities that are unseen by the EOF predictive filter. Here we generated tip–tilt data with artificial saturation to mimic the saturation of a PyWFS. We demonstrate that the NN can model the saturating cube root function, which is beyond the capabilities of the EOF predictive filter, and provides a more accurate reconstruction of the true signal from the saturated signal. It is hoped that this will further encourage conventional use of the PyWFS with reduced modulation so as to profit from more highly sensitive wavefront measurements: a scenario so far impractical due to the PyWFS saturating at low modulation and producing further nonlinear effects.

A topic of our current research is to directly find a nonlinear mapping of the PyWFS outputs to the wavefront using an NN. This would allow us to bypass the need for the linear pseudo open-loop telemetry and work directly with on-sky data. At present, this mapping is achieved by a linear matrix-vector multiplication. Key challenges of this task include working with high dimensional data, designing an appropriate NN architecture (potentially extending those used here to use CNNs for image processing), and selecting the optimum basis for wavefront representation.

A further avenue of investigation would be to employ more complex NN architectures and exploit their more advanced processing capabilities, which would allow for the use of multiple inputs in a variety of formats and from a range of sensors. Combining information, for example, from wavefront sensors, accelerometers and thermometers, will give a fuller picture of the nonlinearities arising within the telescope and should improve wavefront predictions.

Eventually, the logistics of predicting with these NNs in real time need to be considered. The current assumption is that an NN will be extensively pre-trained (with optional contemporaneous training to tune to observation conditions) and the model parameters extracted out into matrices, so prediction calculations can be executed much like they are now with the EOF predictive filter. Such a feat should be feasible for large scale AO systems. Larger and more complex networks (e.g., ResNet-50 V1.5[38]) achieve millisecond latencies when predicting on a single GPU.[39] Once this has been achieved, we anticipate on-sky tests.

Acknowledgments

We would 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 Nos. 23340051, 26220704, 23103002, 19H00703, and 19H00695), 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 that we are most fortunate to have the opportunity to conduct observations from this mountain.

References

1. H. W. Babcock, "The possibility of compensating astronomical seeing," *Publ. Astron. Soc. Pac.* **65**, 229 (1953).
2. O. Guyon and J. Males, "Adaptive optics predictive control with empirical orthogonal functions (EOFS)" (2017).
3. R. Jensen-Clem et al., "Demonstrating predictive wavefront control with the Keck II near-infrared pyramid wavefront sensor," *Proc. SPIE* **11117**, 275–284 (2019).
4. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* **521**, 436–444 (2015).
5. Z. Huang et al., "Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding," *IEEE Sens. J.* (2020).
6. S. F. Chevtchenko et al., "A convolutional neural network with feature fusion for real-time hand posture recognition," *Appl. Soft Comput.* **73**, 748–766 (2018).
7. P. McGuire et al., "Adaptive optics: neural network wavefront sensing, reconstruction, and prediction," *Lect. Notes Phys.* **522**, 97–138 (2007).
8. J. R. Angel et al., "Adaptive optics for array telescopes using neural-network techniques," *Nature* **348**(6298), 221–224 (1990).
9. M. Lloyd-Hart et al., "First results of an on-line adaptive optics system with atmospheric wavefront sensing by an artificial neural network," *Astrophys. J.* **390**, L41–L44 (1992).
10. D. G. Sandler et al., "Use of a neural network to control an adaptive optics system for an astronomical telescope," *Nature* **351**(6324), 300–302 (1991).
11. J. Osborn et al., "Using artificial neural networks for open-loop tomography," *Opt. Express* **20**, 2420–2434 (2012).
12. C. Gonzalez et al., "Comparative study of neural network frameworks for the next generation of adaptive optics systems," *Sensors* **17**, 1263 (2017).
13. J. Osborn et al., "First on-sky results of a neural network based tomographic reconstructor: Carmen on Canary," *Proc. SPIE* **9148**, 91484M (2014).
14. R. Swanson et al., "Wavefront reconstruction and prediction with convolutional neural networks," *Proc. SPIE* **10703**, 107031F (2018).
15. T. Andersen, M. Owner-Petersen, and A. Enmark, "Neural networks for image-based wavefront sensing for astronomy," *Opt. Lett.* **44**, 4618–4621 (2019).
16. H. Guo et al., "Improved machine learning approach for wavefront sensing," *Sensors* **19**, 3533 (2019).
17. Q. Xin et al., "Object-independent image-based wavefront sensing approach using phase diversity images and deep learning," *Opt. Express* **27**, 26102–26119 (2019).
18. Z. Xu et al., "Deep learning control model for adaptive optics systems," *Appl. Opt.* **58**, 1998–2009 (2019).
19. R. Ragazzoni, "Pupil plane wavefront sensing with an oscillating prism," *J. Mod. Opt.* **43**(2), 289–293 (1996).
20. S. Esposito et al., "Large binocular telescope adaptive optics system: new achievements and perspectives in adaptive optics," *Proc. SPIE* **8149**, 814902 (2011).
21. L. M. Close et al., "Diffraction-limited visible light images of orion trapezium cluster with the magellan adaptive secondary adaptive optics system (MagAO)," *Astrophys. J.* **774**, 94 (2013).
22. N. Jovanovic et al., "The Subaru coronagraphic extreme adaptive optics system: enabling high-contrast imaging on solar-system scales," *Publ. Astron. Soc. Pac.* **127**, 890 (2015).
23. S. Esposito et al., "Wavefront sensor design for the GMT natural guide star AO system," *Proc. SPIE* **8447**, 84471L (2012).
24. K. El Hadi, M. Vignaux, and T. Fusco, "Development of a pyramid wave-front sensor," in *Proc. Third AO4ELT Conf.*, S. Esposito and L. Fini, Eds., p. 99 (2013).
25. C. Boyer, "Adaptive optics program at TMT," *Proc. SPIE* **10703**, 107030Y (2018).
26. O. Guyon, "Limits of adaptive optics for high-contrast imaging," *Astrophys. J.* **629**, 592–614 (2005).
27. D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Int. Conf. Learn. Represent.* (2014).

28. F. Chollet et al., “Keras,” 2015, https://keras.io.
29. M. Abadi et al., “TensorFlow: large-scale machine learning on heterogeneous systems,” 2015, tensorflow.org
30. Talos, “Autonomio talos [computer software],” 2019, http://github.com/autonomio/talos.
31. O. Guyon et al., “Wavefront control with the Subaru coronagraphic extreme adaptive optics (SCExAO) system,” *Proc. SPIE* **8149**, 814908 (2011).
32. F. Ochsenbein, P. Bauer, and J. Marcout, “The VizieR database of astronomical catalogues,” *Astron. Astrophys. Suppl. Ser.* **143**(1), 23–32 (2000).
33. A. N. Tikhonov, “Solution of incorrectly formulated problems and the regularization method,” *Dokl. Akad. Nauk SSSR* **151**, 501–504 (1963).
34. N. Srivastava et al., “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014).
35. X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. Int. Conf. Artif. Intell. and Stat.*, Soc. Artif. Intell. and Stat. (2010).
36. J. Lozi et al., “Characterizing vibrations at the subaru telescope for the subaru coronagraphic extreme adaptive optics instrument,” *J. Astron. Telesc. Instrum. Syst.* **4**, 049001 (2018).
37. C. Kulcsár et al., “Vibrations in AO control: a short analysis of on-sky data around the world,” *Proc. SPIE* **8447**, 84471C (2012).
38. K. He et al., “Deep residual learning for image recognition,” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).
39. Nvidia, “Data center deep learning product performance,” 2020, https://developer.nvidia.com/deep-learning-performance-training-inference.

Alison P. Wong is a PhD student at the University of Sydney. Her work examines applying machine learning and deep learning methods in optics and astrophysics. Her current research focuses on applying NNs to problems faced in AO.

Barnaby R. M. Norris is a research fellow at the University of Sydney and an instrument scientist for AAO-USyd. His work focuses on astrophotonics, instrumentation design, implementation, and observations. He specialises in high-angular resolution imaging, including direct imaging, AO (especially machine learning), and interferometry. Recent project leads include the VAMPIRES polarimetric imager and the GLINT photonic nulling interferometer at the Subaru Telescope. Astronomical research interests focus on planet formation and mass-loss from evolved stars.

Peter G. Tuthill received his degrees in physics from the University of Queensland and Australian National University and his PhD from Cambridge University. His research has focused on stellar astrophysics, astronomical imaging, and stellar interferometry. His academic career has taken him from the University of California at Berkeley to the University of Sydney, where he is now a professor of astrophysics.

Richard Scalzo is a senior research fellow at the ARC Industrial Transformation Training Centre in Data Analytics for Resources and the Environment. His research focuses on the fusion of probabilistic and deterministic models in applications across the physical sciences, including astrophysics, geophysics, geology, and hydrology.

Julien Lozi is an optical scientist at the Subaru Telescope. He joined the Subaru Coronagraphic Extreme Adaptive Optics project in 2014. He is in charge of maintaining and upgrading the hardware inside the instrument and Python interfaces aimed at optimizing observations and collaborations. His areas of interest are wavefront sensing, vibration corrections, and instrumental characterization.

Sébastien Vievard is an instrument scientist at the Subaru Telescope. He works on instrumentation dedicated to the study of circumstellar environments, including interferometry, coronagraphy, and spectroscopy. He also develops new methods to improve the image quality on ground-based and/or future space-based telescopes

Olivier Guyon is an astronomer and optical scientist at the Subaru Telescope and at the University of Arizona. His work focuses on the development of exoplanet imaging instrumentation, including coronagraphy and AO. He leads the Subaru Coronagraphic Extreme Adaptive Optics instrument, which serves as both a science instrument and a development platform for high contrast imaging techniques.