

“Introduction To Next Generation Reservoir Computing ”

Semester Project Report

- EP20BTECH11019 (Rahhul J)

Overview:

In this report we will discuss about the things that were done in the project during the course of this semester. We would begin with the thesis review of – “Compensation Through Prediction For Atmospheric Turbulence Effect On Target Imaging And High Energy Laser Beam” by Jun H. Zhang and the paper review of “Adaptive Optics Based On Machine Learning:a Review” by Guo YM, Zhong LB, Min L, Wang JY, Wu Y et al. Then we would look at an introduction to Reservoir Computing and the development of the Next Generation of Reservoir Computing(NG-RC). In a sense, our objective of the project is to try implementing the NG-RC in place of CNN, that has been used in the paper, in order to reduce the requirement of training data and reduce the training times. In addition to this we would look at some simulation done during the course of the project.

1)Thesis Review:

High Energy Laser (HEL) systems have immense potential in military applications due to its unlimited rounds, precision strike and speed of attack. HEL systems had been extensively evaluated and can potentially be effectively against traditional threats like missile strikes and adversarial ships. However, HEL systems do not come without limitations. The presence of air molecules, water vapor and salt particles in the maritime environment significantly alters the beam focus and reduces the effective range of a laser. In addition, atmospheric turbulence can also be caused by variations in the index of refraction in the atmosphere, causing further degradation of the laser beam during its propagation through the atmosphere. To increase its effectiveness, atmospheric turbulence effects need to be compensated in the laser beam. Adaptive Optics (AO) systems with wavefront sensors and Deformable Mirrors (DM) are required to correct the atmospheric turbulence before the beam propagates through the atmosphere to reduce the effects of turbulence.

The concepts of optics and adaptive optics are required to better explain the study. Figure 2 shows a graphical representation of two cases with and without atmospheric turbulence in the atmosphere. The wavefront is represented as an imaginary surface representing points of a wave that vibrate in unison. When wavefronts of a point source pass through an ideal atmosphere with no turbulence, the resulting image observed at the telescope is a focused airy disk. However, with turbulence, the resulting telescope image is an aberrated image with distortion. The telescope images of these point sources are also referred to as PSF images.

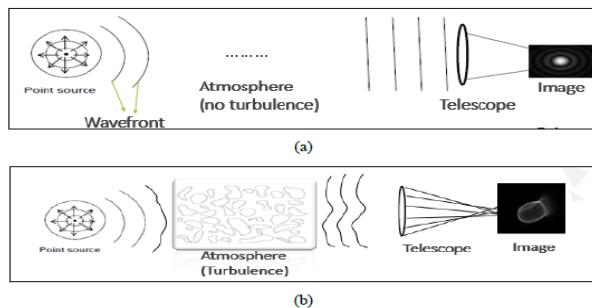


Figure 2. Graphical Representation of Atmospheric Turbulence and Telescope Image: No Turbulence (a), and Presence of Turbulence (b)

Figure 3 shows a conventional AO component in a laser system. To mitigate atmospheric turbulence effects, AO is used to correct laser beams before the beam propagates through the atmosphere. The conventional AO system uses a wavefront sensor to detect the atmospheric turbulence by measuring light captured at the system and formulating the wavefront that it received. For a Shack-Harmann wavefront sensor shown in Figure 3, an array of lens capture light from the target and interpolates the received wavefront. Coupled with a processing computer, the wavefront is re-constructed, the conjugate of the re-constructed wavefront is then fed to the DM to correct the laser beam before propagating it through the atmosphere to the target.

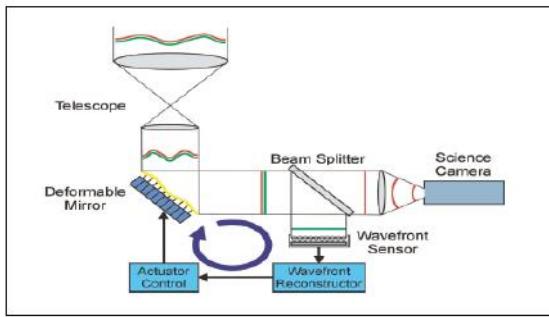


Figure 3. A Conventional AO System. Source: Murray (2006).

Current technologies for beam correction are limited by the wavefront sensor's resolution and the increasing capital cost to improve correction via the number of correction elements in the wavefront sensor. However, with the advent of Artificial Intelligence (AI) and Deep Learning (DL) for image correction, an opportunity to improve image quality and beam effectiveness is available. This thesis investigates the use of AI tools like Convolutional Neural Networks (CNN) and DL in predicting atmospheric turbulence for laser beam correction to increase the effectiveness of a HEL system.

First, a simplified optical model is conceived and used as a testbed to evaluate prediction performance of atmospheric turbulence. In particular, a point light source on a target and the corresponding Point Spread Function (PSF) image is generated for training the network. Zernike polynomials are used to represent the wavefront error of the light caused by the atmospheric turbulence. Second, the trained network is applied to simulated real-life Unmanned Aerial Vehicle (UAV) images dataset for Mongoose and Reaper UAVs. Third, a model is proposed for direct image correction of blur image caused by atmospheric turbulence to clear pristine image to improve target tracking in the HEL system. The computer simulations conducted in this thesis were based on MATLAB's DL toolbox and architecture.

An alternative to direct wavefront sensing is the use of state-of-the-art correction algorithms to apply aberration adjustments directly to the target image. With the advent of CNN and DL (Nishizaki et al. 2019; He et al. 2015), it is possible to predict atmospheric aberrations using trained CNN on a large dataset of pre-defined images. The training of CNNs is generally split into supervised and unsupervised

training. In this study, all training done are supervised training to guarantee performance of the model. First, the CNN can be fed images with known atmospheric turbulence induced aberrations (modelled with 33 term Zernike polynomials) and a prediction can be derived by the trained CNN. Second, the CNN can be used to deduced blurred images given correct clear pristine images as reference training material.

The Residual Network (ResNet) was introduced by (He et al. 2015) for image classification using deeper neural networks. While the ResNet was substantially deeper than other neural networks proposed, it was easier to train and optimize because of its lower complexity. The main feature of the ResNet is a residual learning block which bypasses certain number of layers such that the original data is still retained. These short-cut connections do not add additional learnable parameters or contribute to the complexity of the computation, which makes the network trainable with lesser time and resources.

The U-Net is a CNN used for biomedical image segmentation. It consists of an encoder and decoder subnetwork that are connected by a bridge section which enables strong data augmentation in training the network. This network could be used to train from start to end using very few images and yet still outperforms other networks in image segmentation of electron microscopic stacks (Ronneberger, Fischer, and Brox 2015). The main features of the U-Net are the ability to obtain good training results even with little image dataset. The simplicity of the network also means that the network is sufficiently fast for close to real time applications in the context of target tracking or HEL beam propagation.

The most common method to represent atmospheric turbulence effects on a beam is through the use of Zernike polynomials. Expressed conveniently in polar coordinates r and θ . Figure 9 shows the Zernike polynomials and the respective shape of wavefront.

$$\begin{aligned} Z_{even,j} &= \sqrt{n+1} R_n^m(r) \sqrt{2} \cos m\theta \\ Z_{odd,j} &= \sqrt{n+1} R_n^m(r) \sqrt{2} \sin m\theta \\ Z_j &= \sqrt{n+1} R_n^0(r), m=0 \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} m \neq 0$$

where, $R_n^m(r) = \sum_{s=0}^{(n-m)/2} \frac{(-1)^s (n-s)!}{s! [(n+m)/2-s]! [(n-m)/2-s]!} r^{n-2s}$

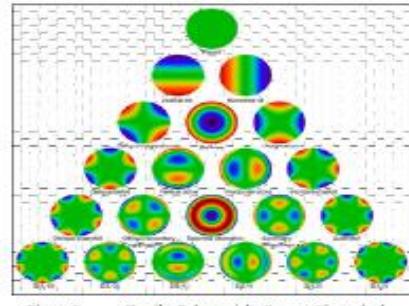


Figure 9. Zernike Polynomials. Source: Swena (n.d.). <https://cloudbugz.com/>.

This study uses 33 term Zernike polynomials to model atmospheric turbulence. The first three terms- piston, vertical tilt, horizontal tilt - are eventually omitted in the results since these three terms are usually associated with target tracking and if the target is not tracked, the model should not generate any predictions.

Using randomized coefficients of Zernike polynomials over a normal distribution with mean and variance determined experimentally (Noll 1976), a 33 term Zernike polynomial representation of atmospheric turbulence is generated. With this Zernike polynomial representation of atmospheric turbulence, the corresponding wavefront is constructed, and the Point Spread Function (PSF), h , obtained. The wavefront, w , is represented with the 33 term Zernike polynomial, z_i , with the normally distributed coefficients a_i .

$$w(r, \theta) = \sum_{i=1}^{\infty} a_i z_i(r, \theta) \quad (5)$$

Here, r is the radial distance and θ is the azimuthal angle in polar coordinates. The PSF, h is transformed using the Fourier transform, F

$$h = |F[w]|^2 \quad (6)$$

Finally, the target scene of interest, f , is convolved with the PSF, h , giving the blurred image g .

$$g = h \otimes f \quad (7)$$

First, the PSF is simulated using a distant point light source at diffraction limit. The PSF (input image) and the 30 term Zernike polynomial (output) are used to predict the correct Zernike polynomials coefficients using the modified ResNet-18 CNN architecture. Second, the PSF images from the first part is convolved with the UAV images dataset available at NPS to generate blurred extended point source images with known Zernike polynomials coefficients from the first part. Thirdly, the blurred UAV images are used as inputs to the U-Net CNN and the original clear UAV images as outputs for direct image to image

correction without using Zernike polynomials. Table 2 shows the training parameters that were used to generate the images used for training.

Table 2. Summary of Training Parameters Used

Parameter	Values	Additional Information (If Any)
Aperture Size / m	0.3	Size of Telescope
Grid Size	224x224x3	- Black and white images: Repeated in three channels - RGB images: Stacked images with different defocus levels 0.5 and 10
Wavelength, λ / μm	1.0	Laser beam wavelength
Target Distance, z / m	4,000	-
r_0 / m	0.01 to 0.1	-
Number of Images	20,000	-

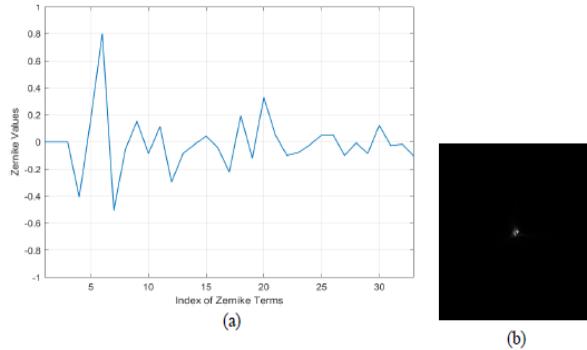


Figure 10. Example of PSF Images Generated for Simulation: Zernike Polynomials (a), and Corresponding PSF image (b)



Figure 11. Example of UAV Images Generated for Simulation:
Original clear UAV Images (a), Blurred UAV Images with Zernike Polynomial Recorded (b)

C. NUMERICAL MEASURES

1. Fried Parameter, r_0 , and Strength of Turbulence, C_n^2

The Fried parameter, r_0 , is a measure of the quality of light transmission through the atmosphere due to turbulence effects. Typical Fried parameter that gives good image quality is about 20 to 40cm (Axtell 2014). The value of 10cm was selected for initial investigation and smaller value (more turbulence) ranges were used to test the effective limit of the DL model in predicting and correcting images with r_0 up to 1cm. The corresponding refractive structure constant, C_n^2 , is related to the Fried parameter (Fussman 2014) by the equation

$$r_0 \approx 0.33 \frac{\lambda^{5/3}}{(d^{1/3} C_n^2)^{5/3}}, \quad (7)$$

where λ is the wavelength of the laser, d is the target distance. Typical values for the strength of turbulence, C_n^2 , is $10^{-17}\text{m}^{-2/3}$ and $10^{-13}\text{m}^{-2/3}$ for weak and strong turbulence, respectively (Fussman 2014). In this work, varying r_0 and C_n^2 conditions are simulated with the Zernike polynomial coefficients representation of atmospheric turbulence.

2. Strehl Ratio

The Strehl ratio is a “measure of the quality of optical image formation” originally proposed by Karl Strehl (Strehl 1902). It is used in imaging through atmospheric turbulence as a measure of the aberration and has a range between zero to one. With one representing a hypothetically perfect clear image. The ratio is used to compare against an ideal case where the laser beam is un-aberrated and estimate the effectiveness of the laser beam at the target. A simplified Strehl ratio can be calculated (Sacek 2006) as

$$S \approx \frac{1}{e^{(2\pi\omega)^2}}, \quad (8)$$

where e is the natural logarithm base and ω is the Root Mean Square wavefront error. The Strehl ratio is also used in the work to represent the wavefront estimation error from deep learning algorithms.

The PSF images generated were used to train the ResNet-18 CNN to output the expected 30 term Zernike polynomial. The prediction accuracy was found to be poor. Other than using a focused PSF image, defocus was also added to the PSF image to simulate phase diverse data for simulation. Figure 12 shows the PSF image from Figure 9 with an added defocus level of 5, 10 and 20 in the focus term of Zernike polynomials. Due to the nature of image recognition-related DL and the computer “seeing” these images, the prediction results obtained was best when defocus was added. This likely gave the network more information to predict and estimate the Zernike polynomials compared to the focused PSF image which has little information all concentrated in a few pixels in the middle of the image.

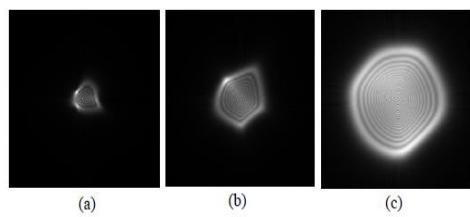


Figure 12. PSF Image with Defocus Added of 5 (a), 10 (b), and 20 (c)

Table 3. PSF Image Training Results for $r_0=10\text{cm}$ and Added Defocus

Level of Defocus	Training RMSE	Validation RMSE	RMS Wavefront Error / $\pi \cdot 0.4$	Corresponding Strehl Ratio
No Defocus	1.2576	1.2606	22	0.973
+ 5	0.4262	0.4167	6.79	0.992
+ 10	0.2732	0.2724	4.54	0.994
+ 20	0.2276	0.2115	3.46	0.996
Stacked image with three levels of defocus 0.5,10	0.3176	0.3137	5.08	0.994

A closer look at the training results indicates that without defocus added, the model is unable to predict well the Zernike polynomials and the results were unable to be used in HEL systems for wavefront correction. Figure 13 shows the overall scatter plot for the complete dataset of 20,000 training PSF images.

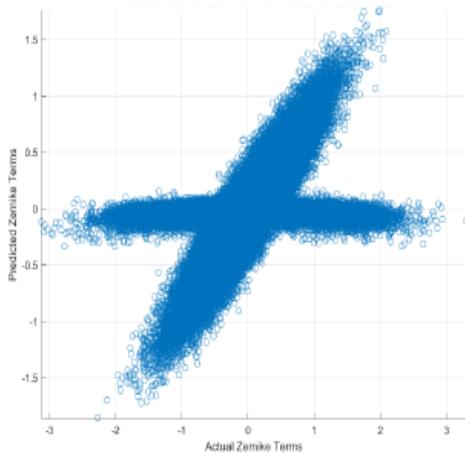


Figure 13. Scatter Plot of PSF Images Predicted versus Actual Zernike Polynomials—No Defocus Added

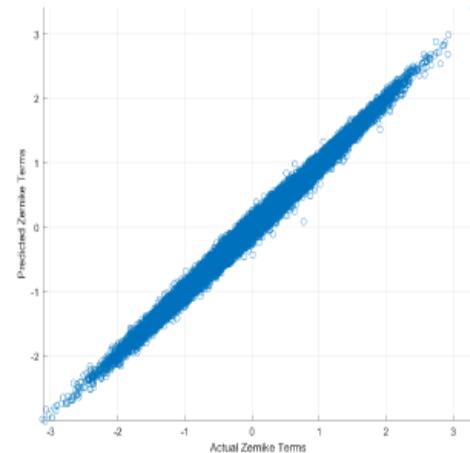


Figure 14. Scatter Plot of Predicted versus Actual Zernike Coefficients with +20 Defocus

From Figure 13, it is observed that the predictions for a focused PSF image causes false predictions along the entire spectrum of 30 Zernike Coefficients. Due to the likely lack of information in a focused PSF image, 8.6% of the predicted Zernike terms were near zero and failed in predicting a reasonable Zernike value. In contrast, the added defocus of 20 helped improved the estimation results and gave a much better prediction of the Zernike Coefficients to generate the wavefronts. Figure 14 shows this improvement where the prediction error standard deviation was 0.0337 compared to 0.163 in the case where no defocus was added. Using the best results for added defocus of 20, the CNN model was used to test against a new generated test set of up to 1,000 PSF images. A sample of the prediction results is shown in Figure 15. Prediction is highly accurate and predicting time was about 54ms1 per image, slightly more than 33ms for real-time processing. Given the close approximation of the actual Zernike Coefficients in the predicted values, the predicted wavefront also closely resembles the actual wavefront as illustrated in Figure 16.

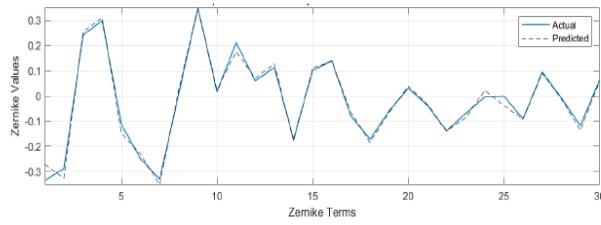


Figure 15. Sample of Predicted Zernike Coefficients for PSF Images with Added Defocus +20

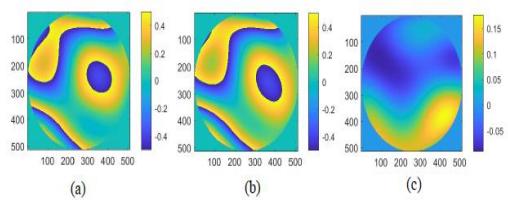


Figure 16. Generated PSF Images of Actual Wavefront (a), Predicted Wavefront (b), and Wavefront Error (c)

Table 4. PSF Image Training Results for $r_0=1$ to 5cm Stacked Images

r_0/cm	Training RMSE	Validation RMSE	RMS Wavefront Error / e-3	Corresponding Strehl Ratio
1	28.564	48.551	82.7	0.354
2	5.807	8.072	13.3	0.846
3	2.603	3.024	4.6	0.944
4	1.562	1.676	2.5	0.969
5	1.090	1.032	1.6	0.980

While PSF images are useful as a first approximation and proof-test of the simulation method, actual UAV images are more practical for use in HEL systems. Hence, the next dataset used is the simulated real-life UAV images from the Reaper and Mongoose UAV target images dataset. These images were convolved with the PSF images previously generated using randomized Zernike polynomials from the first part. A similar approach to the PSF images was adopted to train the ResNet-18 CNN to output the expected 30 term Zernike polynomial. The same parameters were also used as per listed in Table 2. The same levels of defocus were also added to the UAV images to test the response compared to PSF images. Figure 17 shows the original in-focus UAV images and the respective post-processed images that had defocus added. Images (c) through (f) has no visible differences since the only distinction between them is the level of defocus that was added to the image.

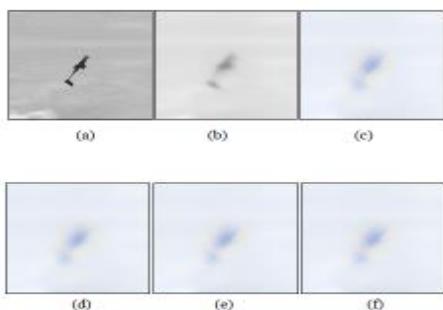


Figure 17. Types of UAV Images Used: Original Clear (a), Blurred No Defocus (b), Defocus +5 (c), Defocus +10 (d), Defocus +20 (e), and Stacked (f)

Table 5. Summary of UAV Image Training Results for Added Defocus

Level of Defocus	Training RMSE	Validation RMSE	RMS Wavefront Error / e-3	Corresponding Strehl Ratio
No Defocus	1.5007	1.4807	2.5	0.969
+ 5	1.1007	1.1113	2	0.975
+ 10	1.074	1.0966	1.9	0.976
+ 20	1.224	1.290	2.2	1.000
Stacked image with 3 levels of defocus 0,5,10	0.9774	0.991	1.8	1.000

A closer look at the training results indicates that unlike the outcome for PSF images, the UAV images dataset did not result in better prediction with added defocus. This fits well with the explanation previously since the UAV images already had sufficiently large amount of pixel information in each image unlike the few concentrated pixels in the PSF images with no defocus added. Consequently, the results for using UAV images as inputs was best with stacked images with three different defocus levels compared to the highest defocus level for PSF images.

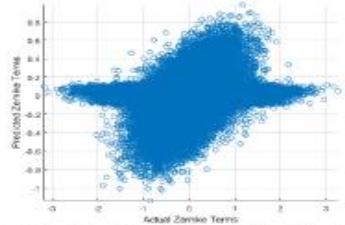


Figure 18. Scatter Plot of UAV Images Predicted Versus Actual Zernike Coefficients—No Defocus Added

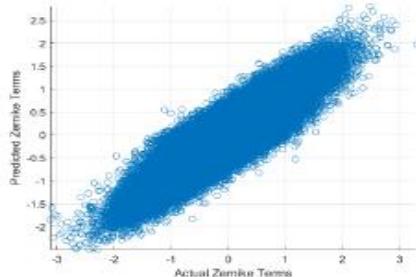


Figure 19. Scatter Plot of Predicted Versus Actual Zernike Coefficients with +20 Defocus

Using the best estimation results in Table 5, for stacked images, the model was used to test against a new generated test set of up to 1,000 PSF images. A sample of the prediction results is shown in Figure 20.

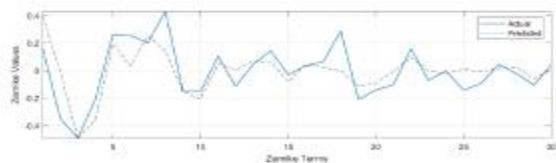


Figure 20. Predicted Zernike Coefficients for Stacked UAV Images

The corresponding predicted wavefronts are illustrated in Figure 21.

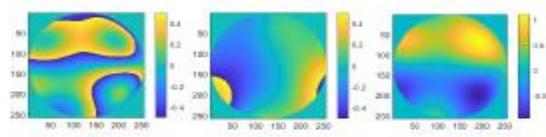


Figure 21. Results from UAV Images: Actual Wavefront (a), Predicted Wavefront (b), and Wavefront Error (c)

Possible Improvements:

One possible improvement to increase the accuracy of prediction is to include some form of pre-processing integral to the CNN architecture. This would help to filter out unwanted “noise” from the image. Real-world UAV images come with unwanted information like clouds and shadows which increase the complexity of the image and reduce the accuracy of prediction. Pre-processing the image would focus the information processing to the key features of the UAV and would likely result in better prediction accuracy.

The study was confined to dataset of two UAVs—the Mongoose and Reaper UAVs. While the trained networks are still usable for other UAVs, it may not give good results. Training data needs to be general enough to prevent overfitting. Hence, more training data UAV types may be required to improve the overall applicability of this model for other UAV types.

The use of direct image correction yields good results to give clearer UAV images and the spectrum of application is large. However, the applicability of this method is confined by the long processing time needed. Perhaps a customized CNN specific for this task would reduce the number of parameters in the CNN and reduce the processing time to a more palatable speed for deployment in naval ships.

2)Paper Review:

Adaptive optics (AO) is a dynamic wavefront compensation technique widely used in various applications such as ground-based telescopes, laser communication, and biological imaging etc.

Although AO has been already successfully implemented in many areas to improve the image resolution or peak energy of lasers, there are still some challenging problems. For example, how to get the wavefront of wavefront sensor less (WFS-less) AOS or array telescopes with high speed, how to estimate the wavefront along the line of sight to the scientific target in multi-object AO (MOAO) system and how to reduce the time delay error in extreme AOSs, etc. With the development of machine learning, especially the deep learning techniques, some of these complex or inverse problems can be solved. Machine learning is a concept that an algorithm can learn and adapt to new data without human intervention. It is usually divided into two kinds of methods, supervised learning and unsupervised learning. Supervised learning is provided with training data containing not only inputs but also outputs that are also named as labels. The most popular collection of supervised learning algorithms is deep learning which is composed by multiple layers of neural networks. In general, at least three important tasks in AO require such tools including determining the aberration from optically modulated images in WFS, predicting future wavefront with historical multi-source information and reconstructing the high-resolution images from the noisy and blurred ones. These problems are either ill-conditioned or highly nonlinear. Typical traditional algorithms such as least square fit (LSF), singular value decomposition (SVD), or Gauss-Seidel et al. either have weak fitting capability or require many iterations. On the contrary, deep learning algorithms

not only have strong fitting capabilities but also can contain the prior information in the network's structure and weights. These natures can help AO solve the above problems. Besides, the structure of AOS may be also simplified by the powerful algorithms and computation.

Traditional Wavefront Sensing: Taking the solar observation as an example, the AOS is basically composed of a deformable mirror (DM), a WFS, a real-time controller (RTC), and a post-processing program, as described in Fig. 1.

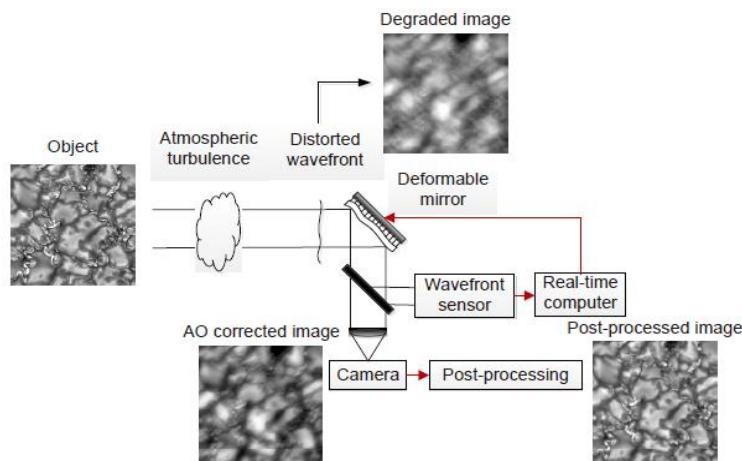


Fig. 1 | Overview of AOS for solar observation. The light waves emitted by the Sun suffer from wavefront distortion when pass through the atmospheric turbulence. The WFS detects the intensity distributions caused by the wavefront distortion and then transfers them to the RTC. The RTC reconstructs the wavefront and calculates the voltages sent to the DM to compensate the distorted wavefront. Meanwhile, the scientific camera records the corrected images and sends them for post-processing in order to get even higher resolution.

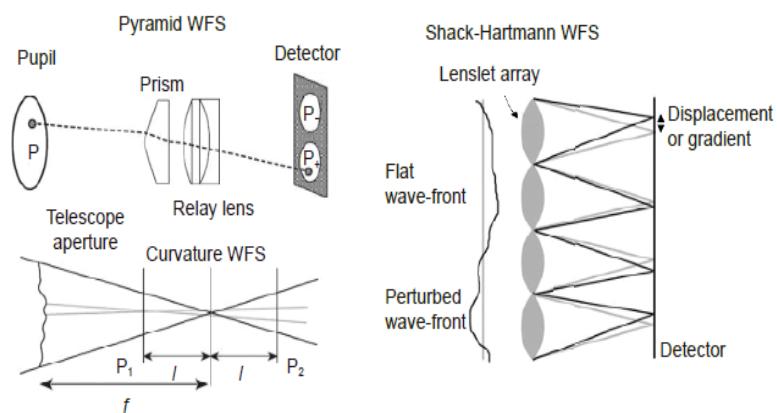


Fig. 2 | Principles of three kinds of WFS. Figure reproduced with permission from ref³², Annual Reviews Inc.

Intelligent Wavefront Sensing:

Shack-Hartmann wavefront sensor:

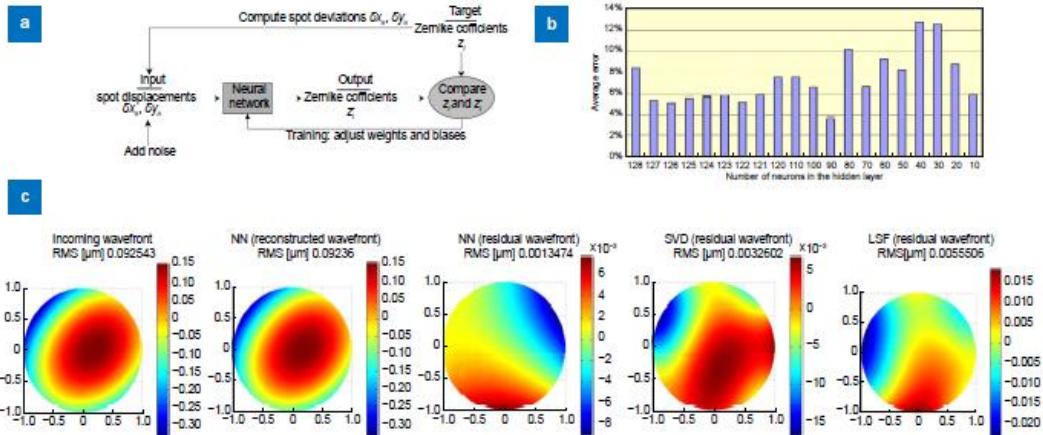


Fig. 3 | (a) The training algorithm of the ANN where displacements of the spots are taken as the inputs while the Zernike coefficients as outputs. To make the network more insensitive to the noise in the spot patterns, the training is also performed on noisy patterns. The noise added to the spot displacements follows Gaussian distribution. (b) The average reconstructed errors of ANNs with different number of neurons in the hidden layer shows that hidden layer with 90 neurons performs best. (c) Comparison of residual errors of LSF, SVD and ANN algorithms. Figure reproduced from ref.⁷¹, Optical Society of America.

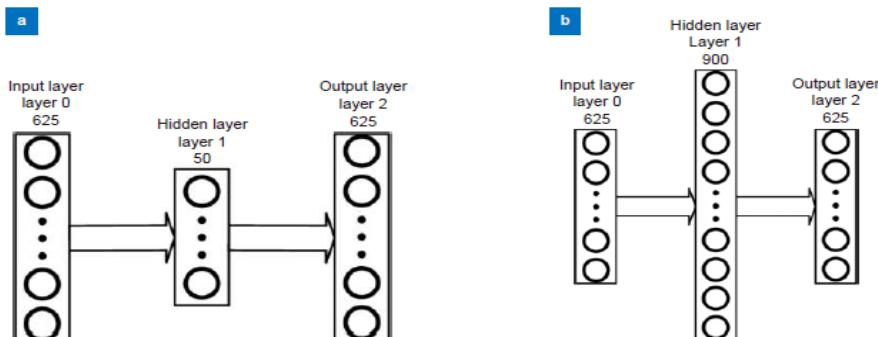


Fig. 4 | (a) The classification network similar to CoG method for spot detection with 50 hidden layer neurons named as SHNN-50. (b) The classification network with 900 hidden layer neurons named as SHNN-900. The input is the flattened subaperture image (25x25) and the output is a kind of classification in 625 classes, the same as the number of pixels indicating the potential center of the spot. Figure reproduced from ref.⁷³, Optical Society of America.

Table 1 | The false rate of different methods in low SNR situations, where CoG means Center of Gravity method and TmCoG means a modified CoG method using m % of the maximum intensity of spot as threshold. Table reproduced from ref.⁷³, Optical Society of America.

SNR _P	False Rate/%				
	CoG	Windowing	TmCoG	SHNN-50	SHNN-900
1	98	83	94	73	55
2	97	43	61	28	7
3	97	28	26	6	0
4	98	8	6	1	0
5	97	2	1	1	0

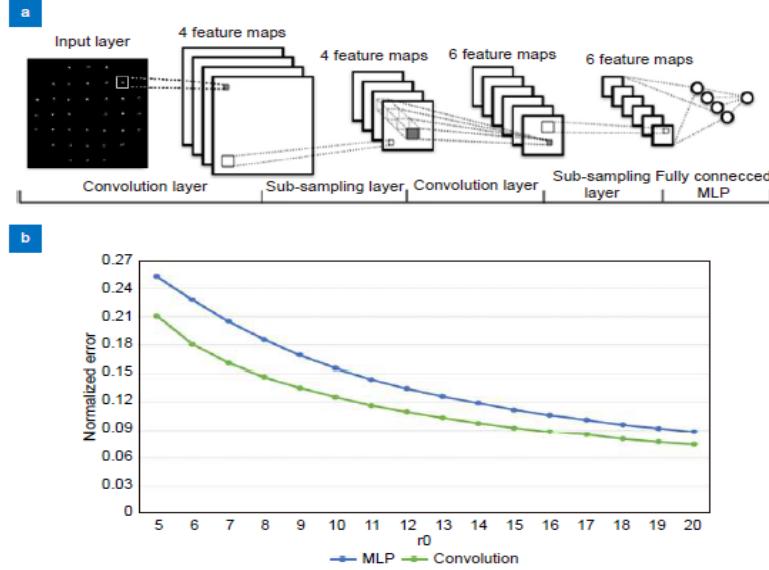


Fig. 5 | (a) The architecture of the CNN for SHWFS. The input is a SHWFS image and the output is a vector, which indicates centroids. (b) The performance of CNN compared with MLP where the error is calculated as the average of the absolute value of the difference between all the output network centroids and the simulated true centroids. Figure reproduced from: (a, b) ref.⁷⁴, International Conference on Hybrid Artificial Intelligence Systems.

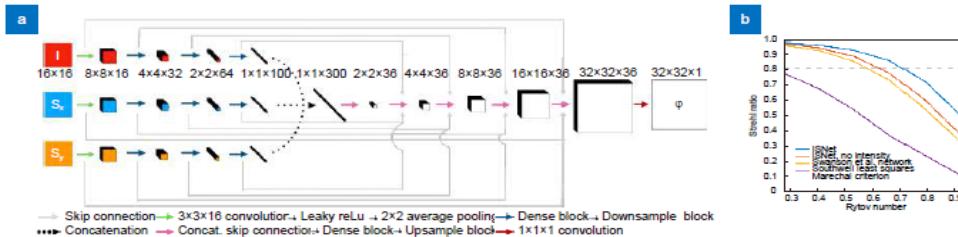


Fig. 6 | (a) Architecture of ISNet. This network takes three 16×16 inputs (x-slopes, y-slopes and intensities) and outputs a 32×32 unwrapped wavefront. (b) Plot of average Strehl ratio vs. Rytov number for different reconstruction algorithms. For comparison, the Strehl ratio of a Maréchal criterion-limited beam is shown in (b), which is 0.82. Figure reproduced from: (a, b) ref.⁷⁵, Optical Society of America.

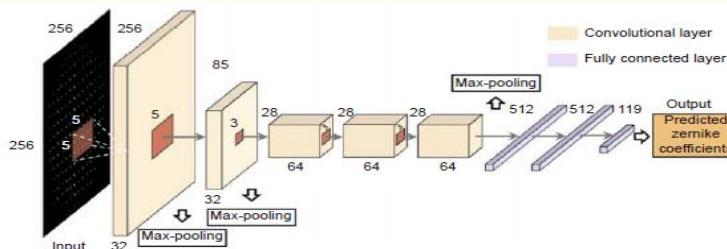


Fig. 7 | Architecture of LSHWS. The network contains five convolutional layers and three full connected layers. The input is a SHWFS image of size 256×256 . The output is a vector of size 119, which represents Zernike coefficients. Figure reproduced from ref.⁷⁶, Optical Society of America.

Table 2 | The correlation coefficient of SHWFS patterns and the Strehl ratio of PSFs. Table reproduced from ref.⁷⁶, Optical Society of America

Correction	Correlation coefficient		Strehl ratio	
	Mean	Std.Dev.	Mean	Std.Dev.
Distorted	0.5178	0.0245	0.1201	0.1702
TSHWS	0.8863	0.0093	0.7775	0.0707
LSHWS	0.9318	0.0039	0.9898	0.0238

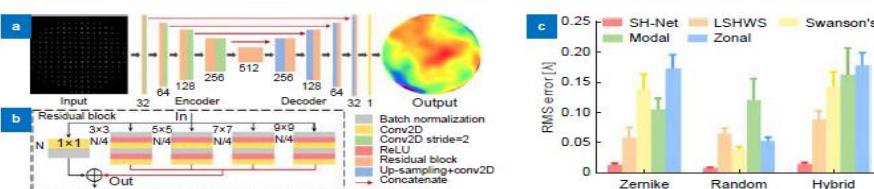


Fig. 8 | (a) Architecture of SH-Net. The input is a SHWFS image of size 256×256 , and the output is a phase map with the same size as the input. (b) The residual block. 'N' and 'N/4' indicate the number of channels. (c) Statistical results of RMS wavefront error of five methods in wavefront detection. Figure reproduced from: (a–c) ref.⁷⁷, Optical Society of America.

Table 3 | The detection speed of five methods. Table reproduced from ref.⁷⁷, Optical Society of America.

	SH-Net	LSHWS	Swanson's	Modal	Zonal
Detection speed	40.2 ms	0.1225 s	22.6 ms	0.1294 s	26.3 ms

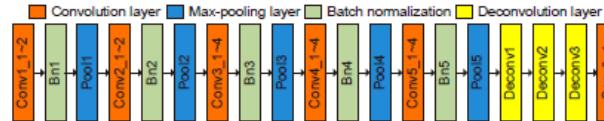


Fig. 9 | Architecture of the deconvolution VGG network (De-VGG). The De-VGG includes convolution layers, batch normalization layer filters, activation function ReLU, and deconvolution layers. The fully connected layers are removed, as the high output order slows down the calculation speed, but the deconvolution layer will not. Figure reproduced from ref.⁸⁰, under a Creative Commons Attribution License 4.0.

Table 4 | The inference time of De-VGG compared with SPGD. Table reproduced from ref.⁸⁰, MDPI.

	SPGD (ms)	De-VGG (ms)
D/r₀=6 (RMS=0.0703A)	304	11
D/r₀=10 (RMS=0.0703A)	448	11

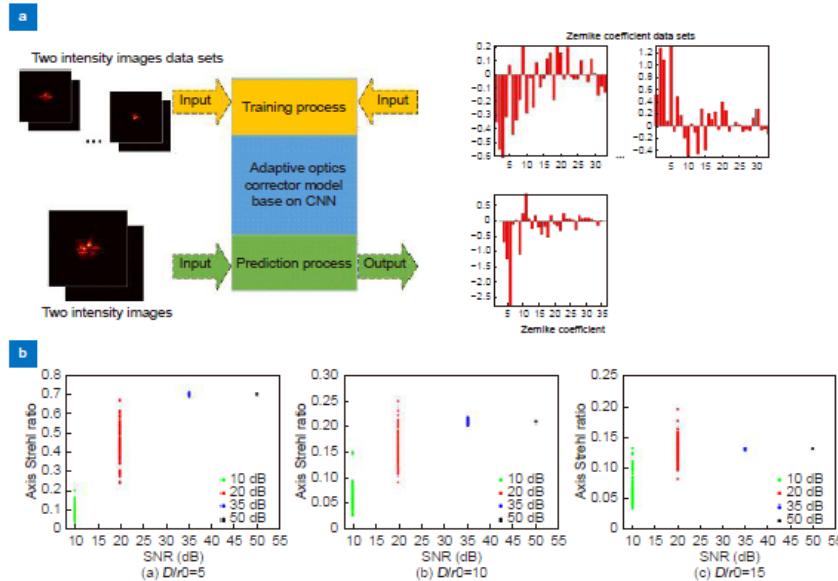


Fig. 10 | (a) The data flow of the training and inference processes. (b) Strehl ratio of CNN compensation under different SNR conditions. Figure reproduced with permission from: (a, b) ref.⁸¹, Elsevier.

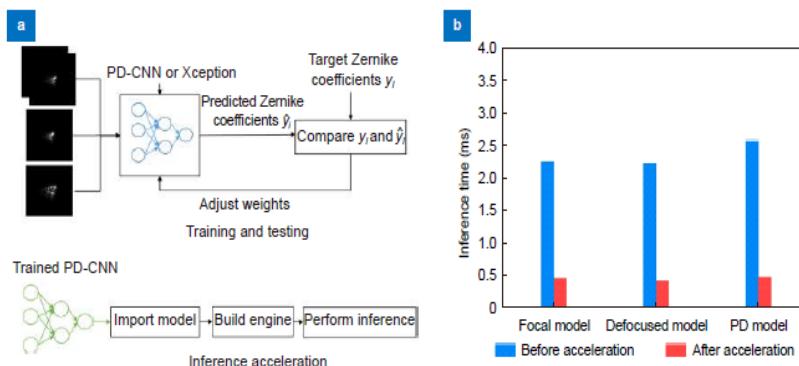


Fig. 11 | (a) The experiments of training and testing processes and inference acceleration. During training and testing processes, two networks were used, phase-diversity CNN (PD-CNN) and Xception. PD-CNN includes three convolution layers and two full connection layers. For each network, three sets of comparative experiments are set as following, inputting the focal and defocused intensity images separately, and inputting the focal and defocused intensity images at the same time. During the inference acceleration process, the trained model is optimized by TensorRT. (b) The inference time before and after acceleration with NVIDIA GTX 1080Ti. Figure reproduced from: (a, b) ref.⁸², under a Creative Commons Attribution License 4.0.

Table 5 | Summary of accuracies (Root Mean Square Error (RMSE): λ) of Zernike coefficients estimated by PD-CNN. The focal model and Defocused model mean the input of PD-CNN is a single focal intensity image or defocused intensity image. The PD model means that the input of PD-CNN includes both focal and defocused image. Table reproduced from: (a, b) ref.³², MDPI.

	Focal model	Defocused model	PD model
Before acceleration	0.1004 ± 0.0469	0.0823 ± 0.0492	0.0529 ± 0.0286
After acceleration	0.1142 ± 0.1119	0.0974 ± 0.1137	0.0696 ± 0.1092

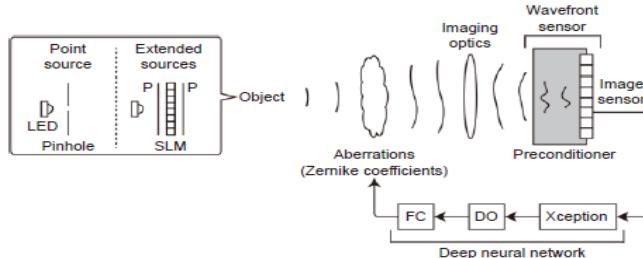


Fig. 12 | Schematic and experimental diagram of the deep learning wavefront sensor. LED: light emitting diode. P: Polarizer. SLM: Spatial light modulator. DO: Dropout layer. FC: full connection layer. Figure reproduced from ref.³², Optical Society of America.

200082-10

Table 6 | Summary of accuracies (RMSE: λ) of Zernike coefficients estimated by Xception. Table reproduced from ref.³³, Optical Society of America.

	In-focal	Overexposure	Defocus	Scatter
Point source	0.142 ± 0.032	0.036 ± 0.013	0.040 ± 0.016	0.057 ± 0.018
Extended source	0.288 ± 0.024	0.214 ± 0.051	0.099 ± 0.064	0.195 ± 0.064

Intelligent Wavefront Prediction:

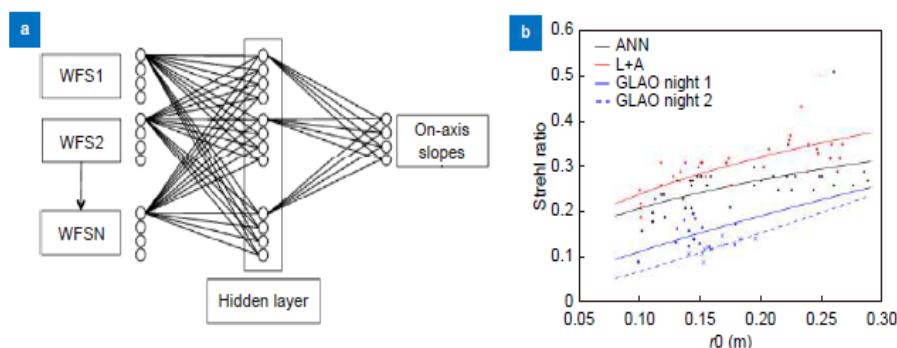


Fig. 13 | (a) Network diagram for CARMEN, where the inputs are the slopes of off-axis WFSs and the outputs are the on-axis slopes for the target direction. One hidden layer with the same number of neurons as the inputs is used to link the inputs and outputs and the sigmoid activation function is used. **(b)** On-sky Strehl ratio (in H-band) comparison with different methods. This on-sky experiment was carried out on the 4.2m William-Herschel Telescope. The Strehl ratios achieved by the ANN (CARMEN), the Learn and Apply (L&A) method and two GLAO night performances are compared. Figure reproduced from (a, b) ref.³⁴, Oxford University Press.

200082-11

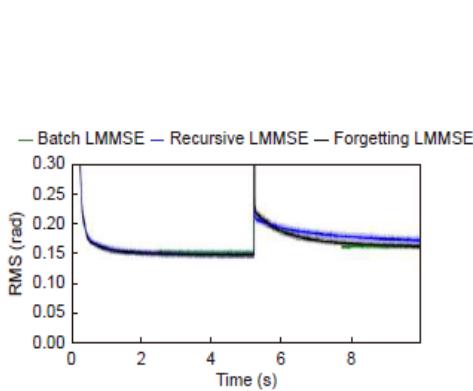


Fig. 14 | Wind jumps of 7 m/s showing the convergence of the recursive LMMSE (blue) and the forgetting LMMSE (black), as well as the resetting of the batch LMMSE (green) for a regressor with a 3-by-3 spatial grid and five previous measurements for each phase point. Figure reproduced from ref.³⁷, Optical Society of America.

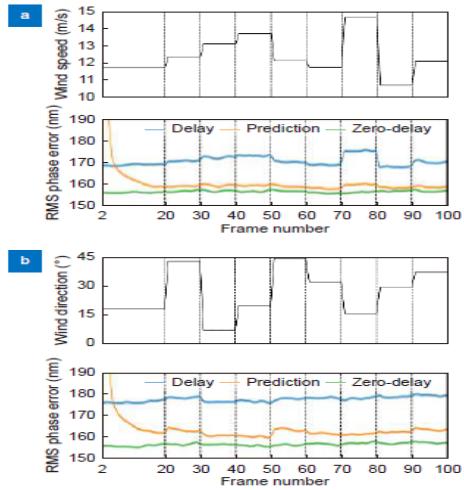


Fig. 15 | (a) Robustness of the predictor against wind speed fluctuations between 10 and 15 m/s every 10 frames. (b) Robustness of the predictor against wind direction fluctuations between 0 and 45 degrees every 10 frames. Figure reproduced from: (a, b) ref.³¹, Oxford University Press.

Intelligent Post-Processing:

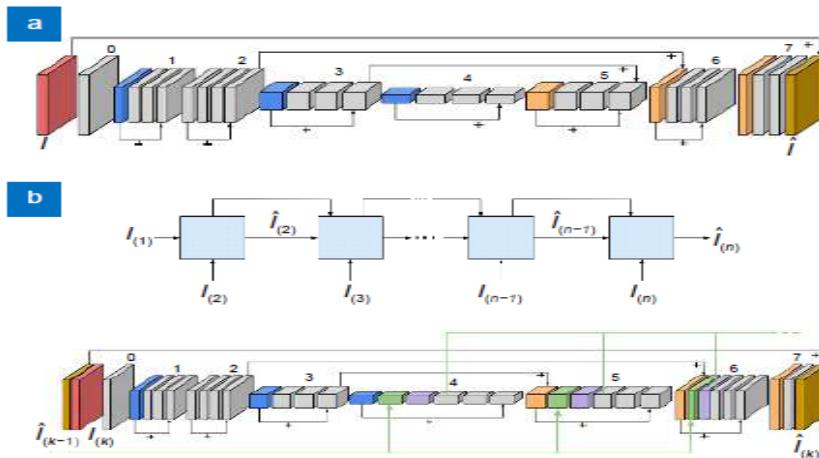


Fig. 16 | (a) Architecture of the encoder-decoder deconvolution neural network. The details of the architecture are described in Ref.⁹⁵. (b) Upper panel: end-to-end deconvolution process, where the grey blocks are the deconvolution blocks described in the lower panel. Lower panel: internal architecture of each deconvolution block. Colors for the blocks and the specific details for each block are described in the reference. Figure reproduced from: (a, b) Ref.⁹⁵, ESO.

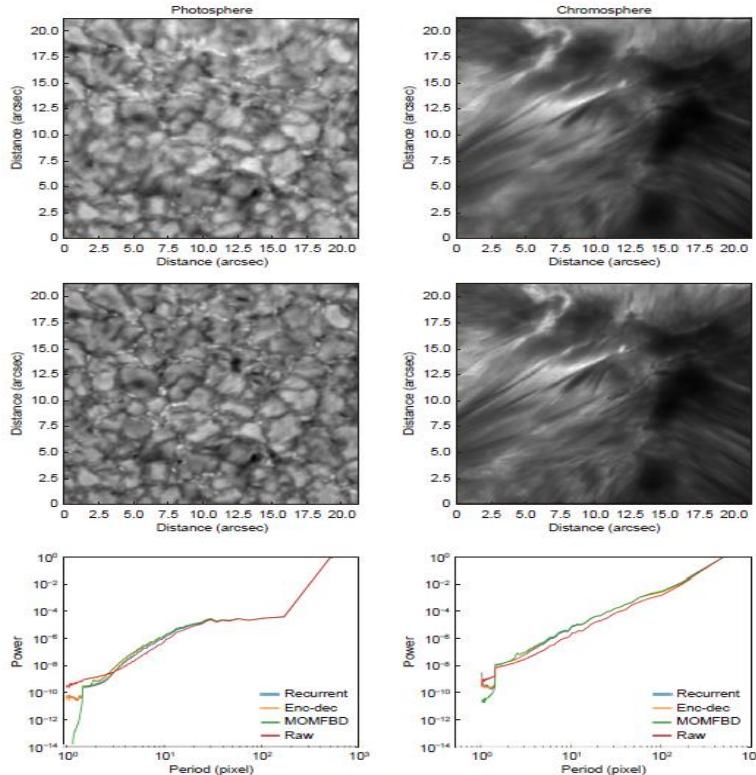


Fig. 17 | Top panels: a single raw image from the burst. Middle panels: reconstructed frames with the recurrent network. Lower panels: azimuthally averaged Fourier power spectra of the images. The left column shows results from the continuum image at 6302 Å while the right column shows the results at the core of the 8542 Å line. All power spectra have been normalized to the value at the largest period. Figure reproduced from ref.⁹⁵, ESO.

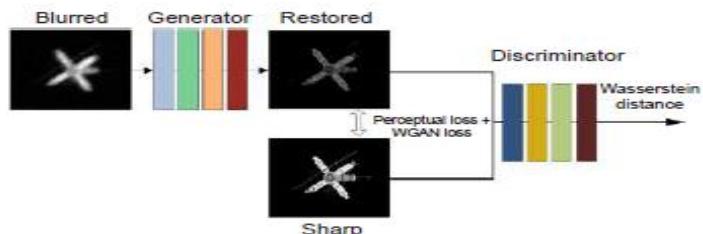


Fig. 18 | The flowchart of AO image restoration by cGAN. The whole network consists of two parts, generator network and discriminator network, which are used for learning the atmospheric degradation process and achieving the purpose of generating restored images. The loss function of the network is a combination of content loss for generator network and adversarial loss for discriminant network. Figure reproduced from ref.⁹⁶, SPIE.

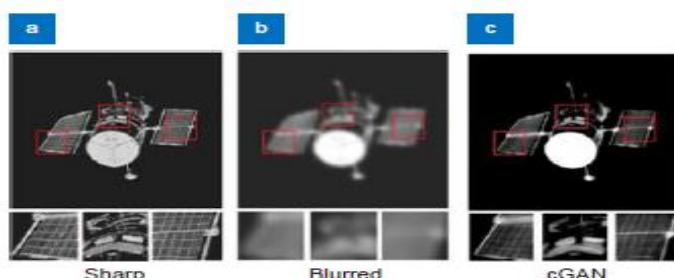


Fig. 19 | The results of blind restoration for the Hubble telescope. (a) the sharp image, (b) the blurred image by Zernike polynomial method in atmospheric turbulence strength $D/r_0 = 10$, and (c) the result of restoration by cGAN, respectively. Figure reproduced from ref.⁹⁶, SPIE.

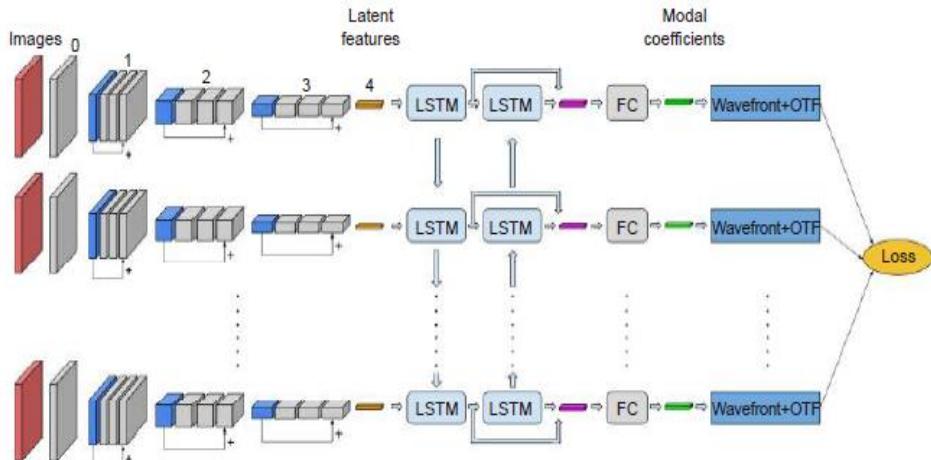


Fig. 20 | Block diagrams showing the architecture of the network and how it is trained by unsupervised training. Figure reproduced from ref.³⁷, arXiv.

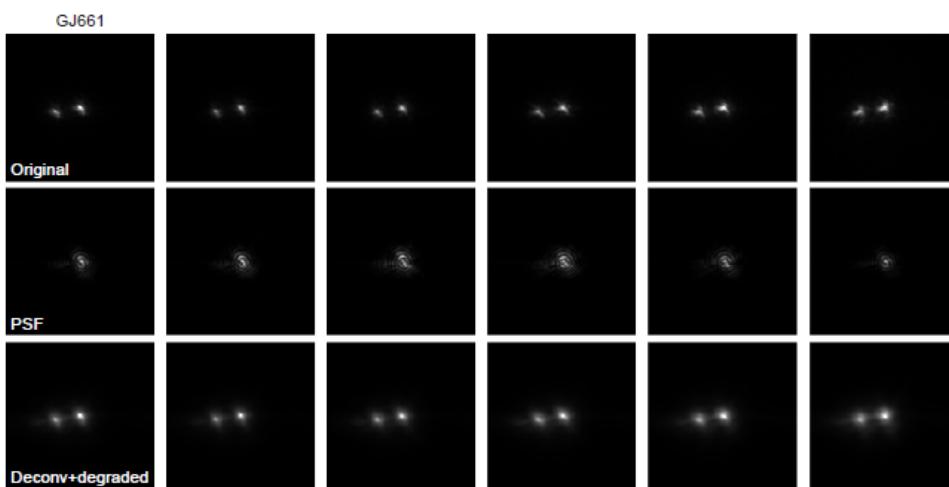


Fig. 21 | Original frames of the burst (upper row), estimated PSF (middle row) and for the GJ661. The upper row shows six raw frames of the burst. The second row displays the instantaneous PSF estimated by the neural network approach. The last row shows the results from reconvolve the deconvolved image with the estimated PSF. Figure reproduced from ref. ³⁷, arXiv.

Table 7 | The general information and differences of all the four networks

Neural network	Type	Training data	Label	Loss function	Output
Encoder-decoder deconvolution ³⁵	Supervised	80000 patches of degraded images	MOMFBD results	L2 metric	Deconvolved images
Recurrent encoder-decoder deconvolution ³⁵	Supervised	80000 patches of degraded images	MOMFBD results	L2 metric	Deconvolved images
Conditional generative adversarial ³⁶	Supervised	4800 frames of simulated AO images	Simulated object	Combination of content loss and adversarial loss	Deconvolved images
CNN-RNN ³⁷	Unsupervised	26000 frames of observed star images.	none	L2 metric	Deconvolved images and the wavefront

Machine Learning for AO Modelling:

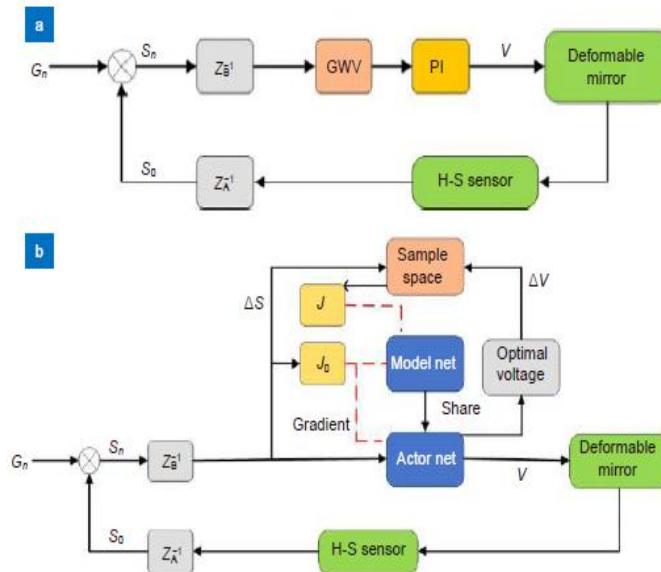


Fig. 22 | Controller for an AOS. (a) PI control model; (b) DLCM control model. Z_A^{-1} and Z_B^{-1} are the H-S sensor delay and control calculation delay, respectively. GWV (Generating wavefront to voltage). J and J_0 are the cost functions of the network and the red dotted line is a gradient data stream. Figure reproduced from ref.¹⁰⁴, Optical Society of America.

Recent advances in IAO are summarized including the intelligent wavefront sensing, wavefront reconstruction and control as well as post-processing. By using the machine learning, a lot of inverse or complex problems can be solved if large-scale datasets are available. Two main scenes seem to be particularly suitable to use machine learning at present. One is to use the ANN to build the relationship between the measurement (image) and the wavefront for wavefront reconstruction or the relationship between blurred image and the diffraction-limited image for post-processing. On one hand, features extracted from data may perform better than the manually selected linear ones. On the other hand, some iterative methods for phase retrieval may be replaced by deep learning for faster speed. The other is to use the ANN to do the nonlinear wavefront prediction for better accuracy and more importantly to adapt to the non-stationary turbulence. Besides of the algorithms, the large-scale and high quality data is also important which may be not easy to get. Some data can be generated by the computer for the

training but its adaptability to the real systems needs to be demonstrated. Although lots of simulation and laboratory results have been obtained, less have been used in real AOSs recently. The application of IAO in real system is of great importance for demonstrating the generalization ability and real-time performance. As long as an on-sky demonstrator succeeds, lots of great progresses can be expected. In the more distant future, with the development of unsupervised learning and reinforcement learning, we can imagine an IAO system that can keep learning the rules from on-line multisource data to improve itself.

3)Introduction to Next Generation Reservoir Computing:

Reservoir computing is a best-in-class machine learning algorithm for processing information generated by dynamical systems using observed time-series data. Importantly, it requires very small training data sets, uses linear optimization, and thus requires minimal computing resources. However, the algorithm uses randomly sampled matrices to define the underlying recurrent neural network and has a multitude of meta parameters that must be optimized. Recent results demonstrate the equivalence of reservoir computing to nonlinear vector autoregression, which requires no random matrices, fewer meta parameters, and provides interpretable results. Here, we demonstrate that nonlinear vector autoregression excels at reservoir computing benchmark tasks and requires even shorter training data sets and training time, heralding the next generation of reservoir computing.

Reservoir computing is an ML paradigm that is especially well-suited for learning dynamical systems. Even when systems display chaotic or complex spatiotemporal behaviors, which are considered the hardest-of-the-hard problems, an optimized reservoir computer (RC) can handle them with ease. Recent RC research has identified requirements for realizing a general, universal approximator of dynamical systems. A universal approximator can be realized using an RC with nonlinear activation at nodes in the recurrent network and an output layer

(known as the feature vector) that is a weighted linear sum of the network nodes under the weak assumptions that the dynamical system has bounded orbits.

Less appreciated is the fact that an RC with linear activation nodes combined with a feature vector that is a weighted sum of nonlinear functions of the reservoir node values is an equivalently powerful universal approximator. Furthermore, such an RC is mathematically identical to a nonlinear vector autoregression (NVAR) machine. Here, no reservoir is required: the feature vector of the NVAR consists of k time-delay observations of the dynamical system to be learned and nonlinear functions of these observations, as illustrated in Fig. 1, a surprising result given the apparent lack of a reservoir!

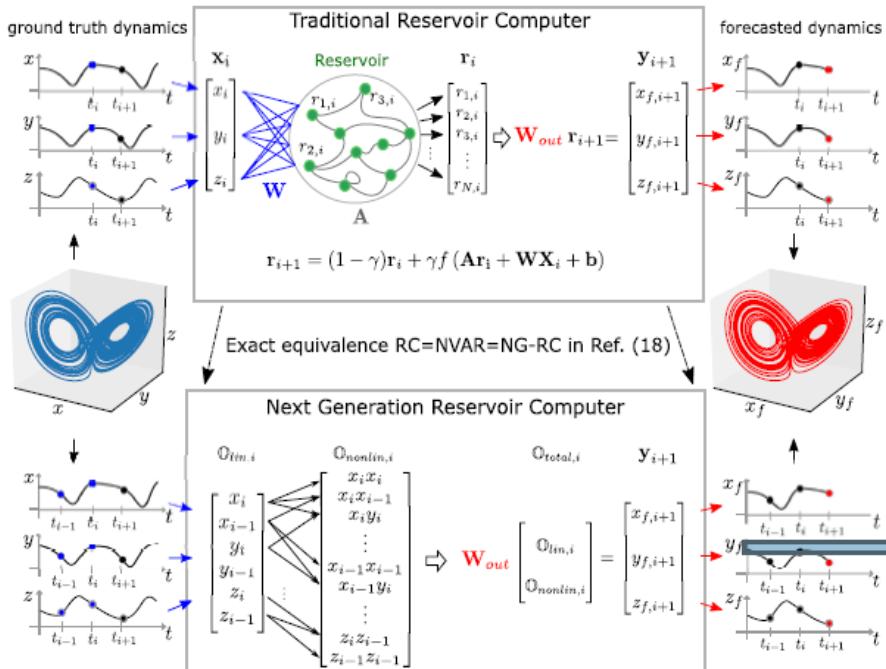


Fig. 1 A traditional RC is implicit in an NG-RC. (top) A traditional RC processes time-series data associated with a strange attractor (blue, middle left) using an artificial recurrent neural network. The forecasted strange attractor (red, middle right) is a linear weight of the reservoir states. (bottom) The NG-RC performs a forecast using a linear weight of time-delay states (two times shown here) of the time series data and nonlinear functionals of this data (quadratic functional shown here).

These results are in the form of an existence proof: There exists an NVAR that can perform equally well as an optimized RC and, in turn, the RC is implicit in an NVAR. Here, we demonstrate that it is easy to design a well-performing NVAR for three

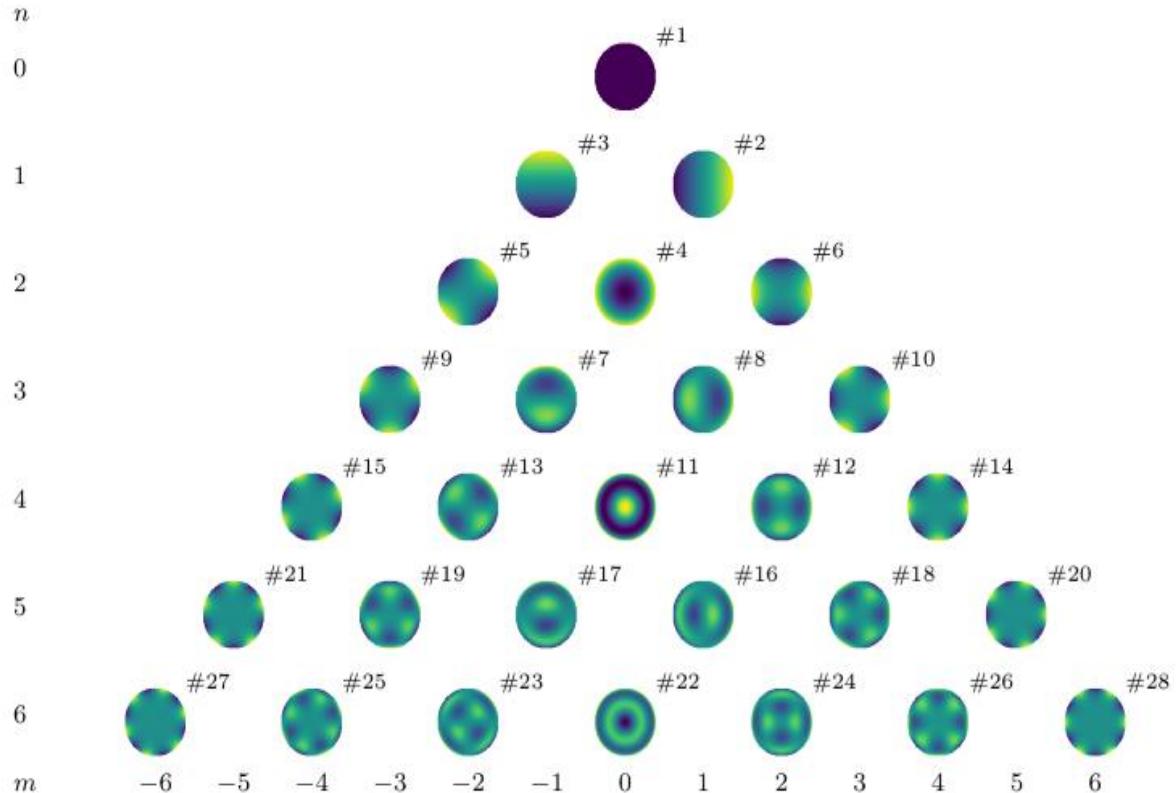
challenging RC benchmark problems:

- (1) forecasting the short term dynamics;
- (2) reproducing the long-term climate of a chaotic system (that is, reconstructing the attractors shown in Fig. 1);
- (3) inferring the behavior of unseen data of a dynamical system.

4) Simulations:

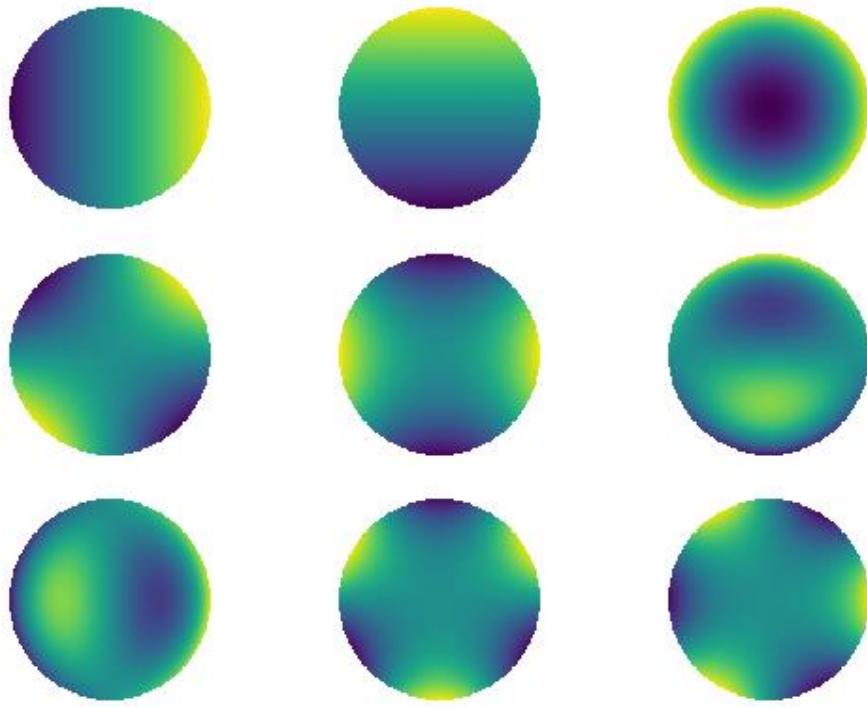
Plotting Zernike Pyramid,

```
python -m zernike.pyramid
```



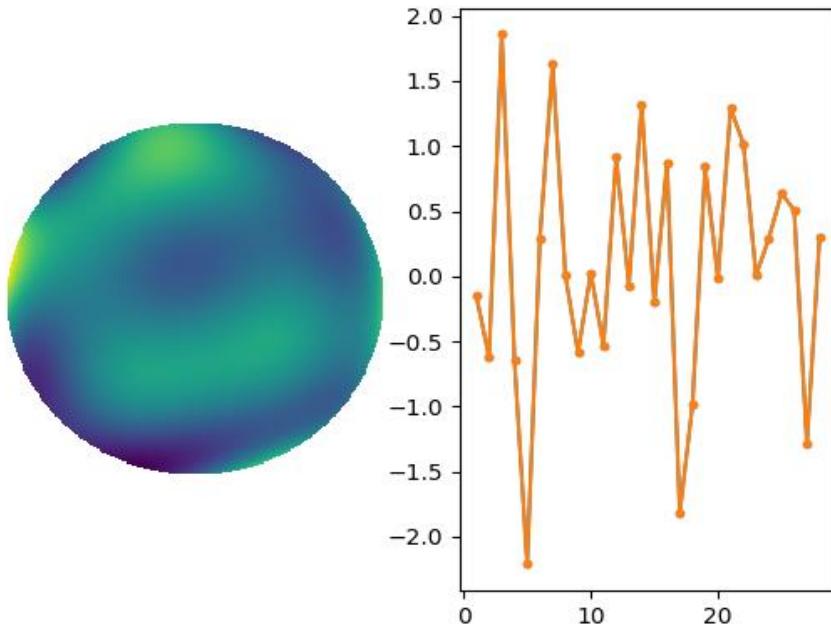
Plotting the first 9 Zernike Polynomials,

Code: https://github.com/rahhul17/ao_project/blob/main/9_zernike.py



Fit Zernike Polynomials in Cartesian coordinates,

Code: https://github.com/rahhul17/ao_project/blob/main/zernike_fit_cart.py

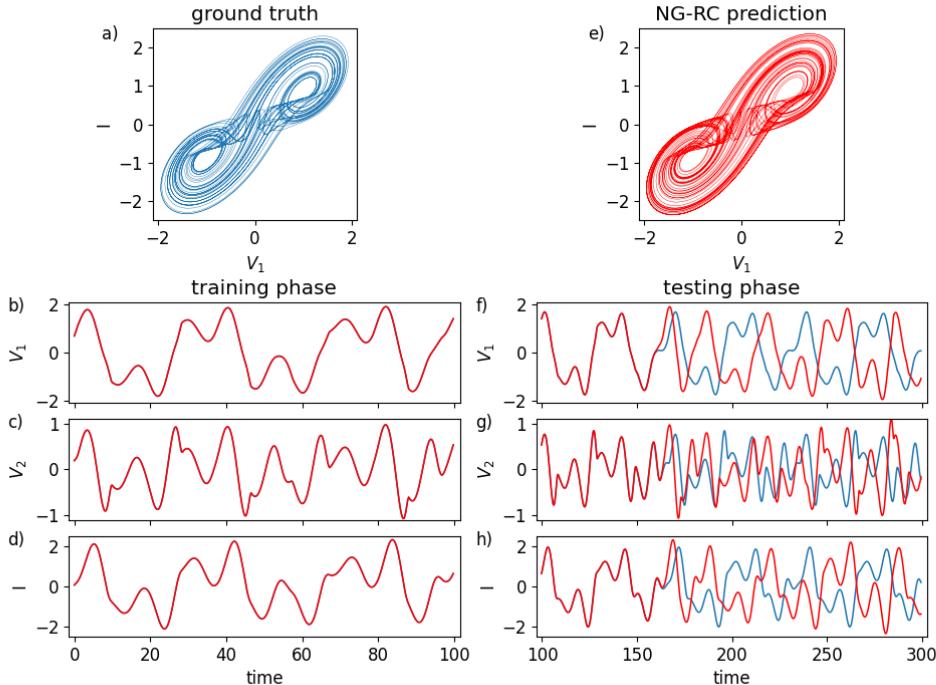


Fit Zernike in Polar Co-ordinates,

Codes: https://github.com/rahhul17/ao_project/blob/main/zernike_polar.py

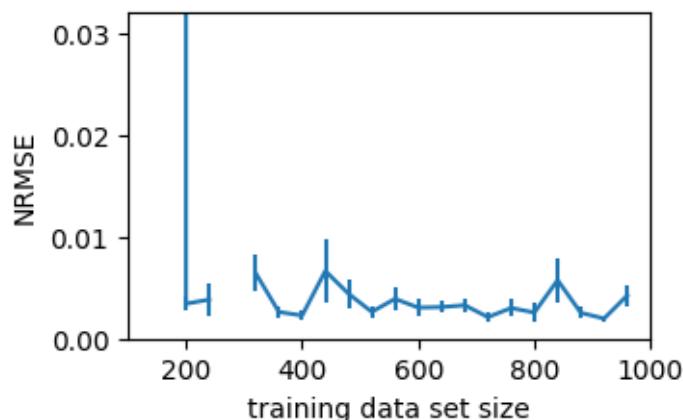
NVAR with time delays for double-scroll forecasting, NRMSE, and fixed points.

Codes: <https://github.com/quantinfo/ng-rc-paper-code/blob/master/DoubleScrollINVAR-RK23.py>



NVAR with time delays for Lorenz forecasting, NRMSE and fixed points.

Codes: https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzConstLinQuadraticNVAR-NRMSE-vsTraintime.py



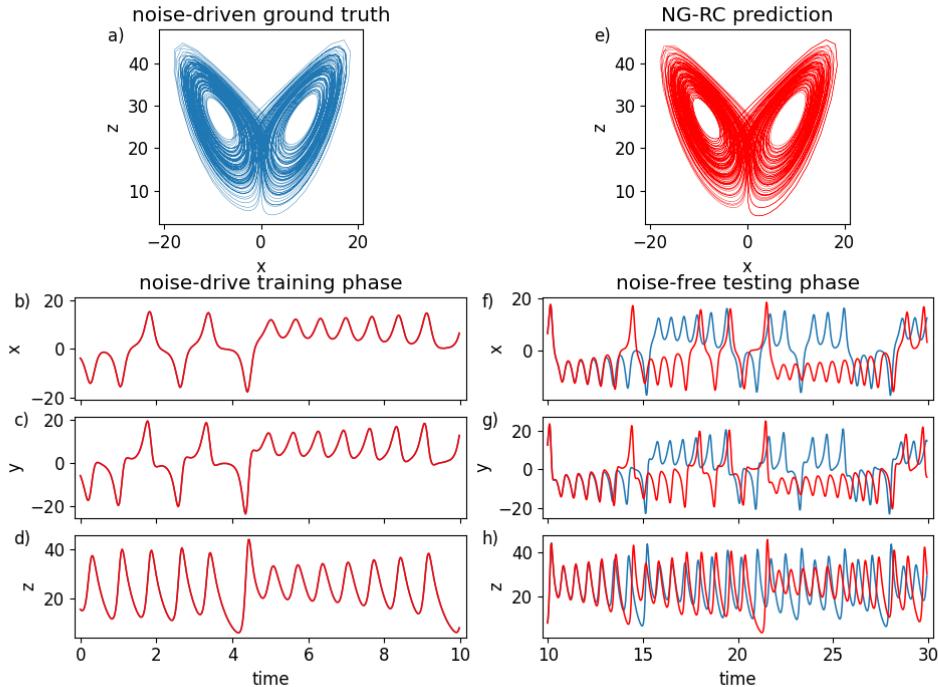
NVAR with time delays for Lorenz forecasting

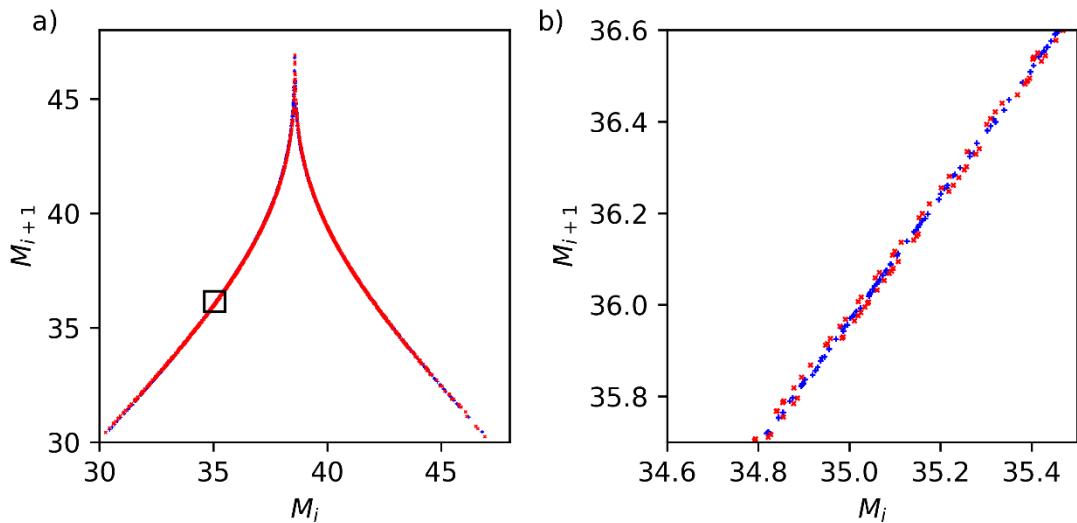
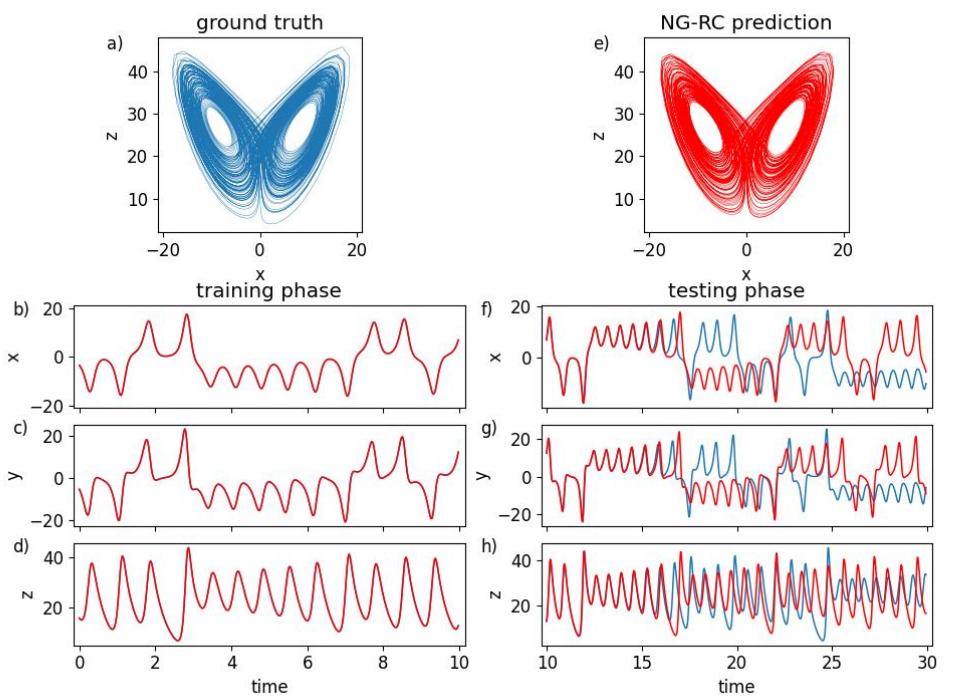
Codes: https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzConstLinQuadraticNVAR-Noise.py

https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzConstLinQuadraticNVARtimedelay-RK23.py

https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzConstLinQuadraticNVARtimedelayNRMSE-RK23.py

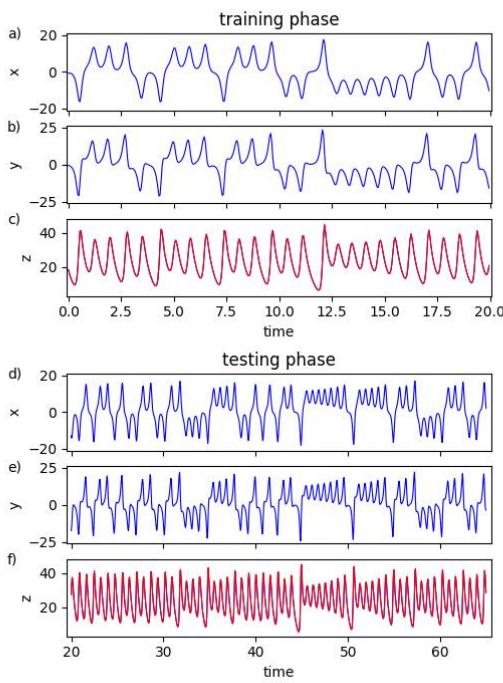
https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzConstLinQuadraticNVARtimedelayReturnMap-RK23.py





NVAR with time delays for Lorenz prediction. Measure x,y, predict z

Code: https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzPredictZQuadraticNVARtimedelay-RK23.py

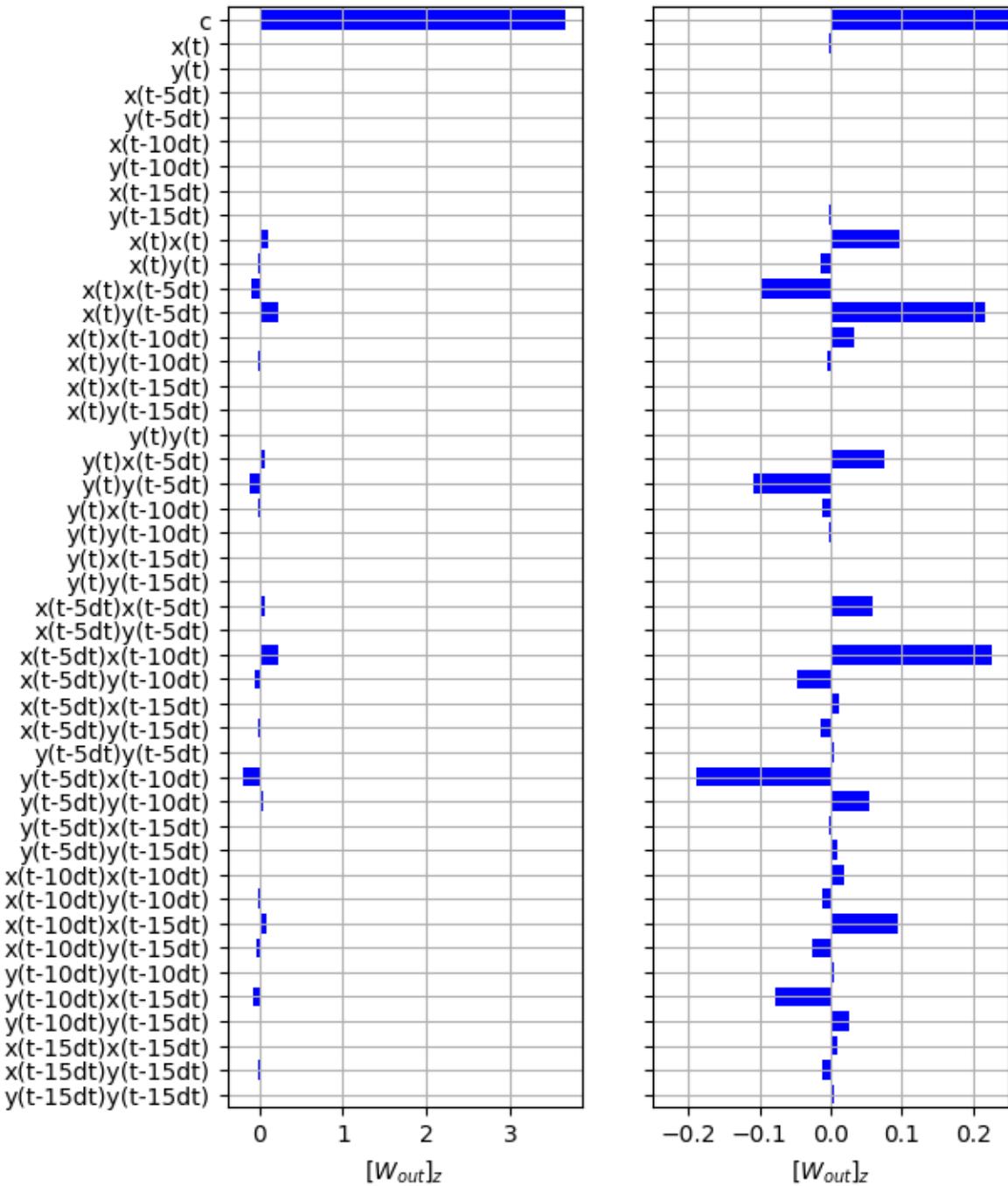


NVAR with time delays for Lorenz prediction, NRMSE.

Code :https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzPredictZQuadraticNVARtimedelayNRMSE-RK23.py

NVAR with time delays for Lorenz prediction, W_{out} plots.

Codes: https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzPredictZQuadraticNVARtimedelayWout-RK23.py



NVAR with time delays for Lorenz forecasting, W_{out} plots,

Codes: https://github.com/rahhul17/ao_project/blob/main/ng-rc-paper-code/LorenzQuadraticNVARtimedelayWout-RK23.py

