

# Wavefront Reconstruction and Prediction with Convolutional Neural Networks

Robin Swanson<sup>a</sup>, Masen Lamb<sup>b</sup>, Carlos Correia<sup>c</sup>, Suresh Sivanandam<sup>b,d</sup>, and Kiriakos Kutulakos<sup>a</sup>

<sup>a</sup>Department of Computer Science, University of Toronto, 40 St. George Street, Toronto, Canada

<sup>b</sup>Dunlap Institute for Astronomy and Astrophysics, University of Toronto, 50 St. George Street, Toronto, Canada

<sup>c</sup>Aix Marseille Univ., CNRS, CNES, LAM, 38 rue F. Joliot-Curie, 13388 Marseille, France

<sup>d</sup>Department of Astronomy and Astrophysics, University of Toronto, 50 St. George Street, Toronto, Canada

## ABSTRACT

While deep learning has led to breakthroughs in many areas of computer science, its power has yet to be fully exploited in the area of adaptive optics (AO) and astronomy as a whole. In this paper we describe the first steps taken to apply deep, convolutional neural networks to the problem of wavefront reconstruction and prediction and demonstrate their feasibility of use in simulation. Our preliminary results show we are able to reconstruct wavefronts comparably well to current state of the art methods. We further demonstrate the ability to predict future wavefronts up to five simulation steps with under 1nm RMS wavefront error.

**Keywords:** Wavefront Reconstruction, Wavefront Prediction, Machine Learning, Deep Learning, Adaptive Optics

## 1. INTRODUCTION

Currently, adaptive optics relies heavily on the Shack-Hartmann (SH) wavefront sensor (WFS) and its traditional control sequence: recovering slopes from each of the SH lenslets and applying them to the deformable mirror via the command matrix. However, this system is inherently limited by the number of lenslets in the system and prone to certain error modes, e.g., waffle modes.<sup>1</sup> In the past few years, however, similar control sequences and imaging pipelines have seen dramatic improvements with the introduction of deep learning, and convolutional neural networks (CNN) in particular. It has been shown that individual components of an imaging pipeline can be successfully reproduced with a single, pre-trained, deep learning model which often outperforms classical methods.

Machine learning techniques have been previously applied successfully to adaptive optics (AO) systems. Osborn et al.,<sup>2</sup> for example, trained a single hidden layer multi-layer perceptron (MLP) to infer Zernike polynomial coefficients from wavefront slopes. This type of approach, however, falls into two pitfalls which we are now able to overcome. First, wavefront reconstruction through Zernike polynomials are limited in their ability to recover arbitrary signals, particularly high-frequency details, and can't be used with non-circular pupils. Second, simple models such as MLP are limited in their inference ability due to both their shallow structure (low number of learned features) and lack of spatial reasoning and spatial invariance enabled by learning convolutional features. CNNs, on the other hand, are able to reproduce very high frequency content and create output with any size or pupil shape. For these reasons, deep CNNs are a natural fit for approximating the AO pipeline.

---

Further author information: R.S.: E-mail: robin@cs.toronto.edu

## 2. SIMULATION ENVIRONMENT

### 2.1 Telescope and Atmosphere Model

The results presented in this paper are based on simulated data created with the Object-Oriented MATLAB Adaptive Optics Toolbox<sup>3</sup> (OOMAO). We base our simulated telescope model on the Gemini telescope. In all experiments our model is an 8 meter telescope with a  $20 \times 20$  lenslet array,  $120 \times 120$  pixel Shack-Hartmann wavefront sensor, and the Gemini pupil. We assume a three layer frozen flow atmosphere model with an  $r_0$  randomly distributed between 65 - 75cm (at H-Band), and a 0.75, 0.20, 0.05 fractional  $r_0$  and altitudes of 0, 4000, and 10000 meters for each of the layers respectively. In both training and testing simulations we randomly and independently choose a wind speed and direction for all three layers. In all of the following results, we trained with a magnitude 1, H-band natural guide star.

### 2.2 Data Generation

To acquire a wide variety of conditions for training our model, we performed 10,000 simulations each run for 500 time steps (each with the randomly chosen  $r_0$ , wind speed, and direction described above) for all atmospheric layers. Once complete, we take a number of sample wavefront and slope pairs evenly distributed throughout the simulation (ignoring the first 100 time steps to eliminate any similar initial conditions between simulations). This leaves us with approximately 150,000 data pairs with which to train our model. To test our models we simply generate additional data using these same settings. All of the reconstruction data is generated in closed loop, creating pairs of slopes and residual wavefronts which we attempt to reconstruct.

## 3. WAVEFRONT MODELLING

### 3.1 Wavefront Reconstruction

To learn a mapping from Shack-Hartmann slopes to the true wavefront, we follow current best practices and use a similar approach to the popular U-Net architecture.<sup>4</sup> Given a set of slopes, we learn and apply a series of convolutional weight layers to both the  $x$  and  $y$  maps independently. First, at each consecutive layer the number of convolutional features doubles as the spatial dimensions are halved via max pooling – which reduces spatial dimensions by discarding all but the highest values over a set area of pixels. After several layers, we concatenate the features from both the  $x$  and  $y$  branches before expanding the spatial dimensions by applying up-convolution (also known as transposed convolution) layers while halving the number of features at each layer. Finally, we include skip connections from both branches of the convolutional half to the deconvolution half between layers with matching spatial and feature dimensions. Each layer is activated with a leaky relu function<sup>5</sup> and we further include batch normalization at each convolutional layer during the spatial reduction phase, but not as the spatial size increases.

The skip connections have two intended benefits, first gradients can more easily pass from the output error to the beginning of the network which alleviates the issue of vanishing gradients. Furthermore, these skip connections exploit any spatial similarities in the data. To train the network we pass a pair of slopes into the network, evaluate its output, and compare it to the downsampled ground-truth wavefront. We use an  $\ell_1$  loss on the error, which is then propagated back through the network to update the model. Figure 1 shows a visualization of the reconstruction network architecture including the size and number of convolutions at each layer and the connections between different layers.

### 3.2 Wavefront Prediction

Unlike the reconstruction model which only takes spatial reasoning into account, in the case of wavefront prediction, we aim to exploit temporal structure in the data. To do so we create a model based on Long Short Term Memory (LSTM) networks. These networks retain state information between evaluations allowing for both short and longer term correlations to be found in the incoming data. They have proven very successful in many areas of deep learning, particularly natural language processing tasks. One extension of these methods is the convolutional LSTM which extends the basic LSTM module to learned spatial features and data. This extension was originally applied to tasks such as precipitation forecasting<sup>6</sup> but has been shown to generalize well to other types of data.

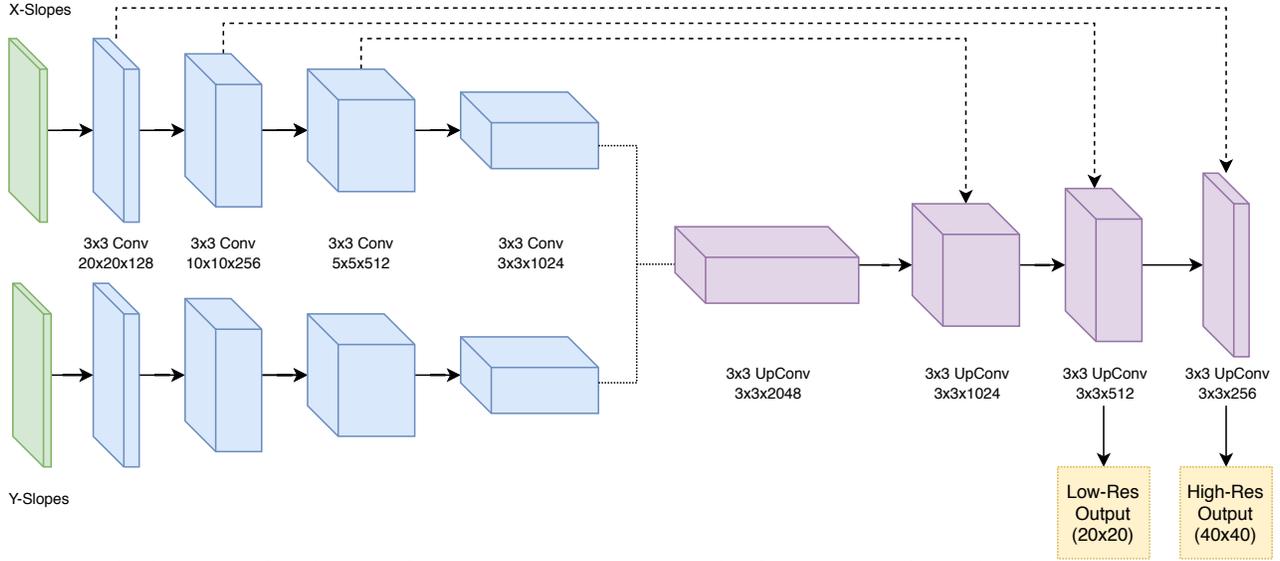


Figure 1: Architecture for the wavefront reconstruction network. Given a set of x and y slope maps, we apply a series of convolutions to extract features before building up to our resulting output wavefront.

In essence, our model takes  $n$  previous frames and tries to predict the next  $n$  frames into the future. To accomplish this our model has three LSTM modules: encoding, reconstruction, and prediction. First, each of the  $n$  frames of previous wavefront data is passed through the encoder. The purpose of this module is to extract pertinent information from the data and create a more useful intermediate representation. The output from the encoder is then passed to both the reconstruct and prediction modules. The purpose of the reconstruction module is to take the encoded data and attempt to perfectly reconstruct the input data – essentially acting as an autoencoder. This module is not used during evaluation but during training behaves as a regularizer, encouraging the encoder to find more general and useful representations for the data. Finally, the prediction module takes the encoded data and attempts to predict the next  $n$  wavefronts.

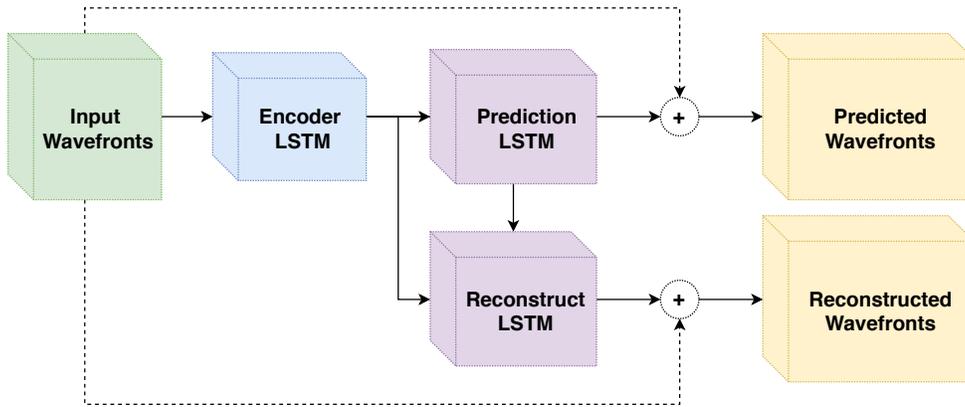


Figure 2: Architecture for the wavefront prediction network. Given a set of  $n$  wavefronts, we encode and then predict the next  $n$  wavefronts with a series of convolutional LSTMs.

To make the problem more tractable, instead of recreating and predicting the wavefronts from scratch, we predict a residual delta between the last input wavefront and what the next wavefront should be. We further make the assumption that these changes should be small, penalizing large changes during training. Our final loss function to train our model is an  $\ell_1$  error of the reconstruction, prediction, and an additional  $\ell_1$  loss on the magnitude of the residuals for both the reconstruction and prediction. The structure of our network can be seen

in Figure 2, as well as the connections between LSTM modules. For all three of our LSTMs we learn 20,  $3 \times 3$  convolutional features and use a leaky relu activation function.

## 4. RESULTS

### 4.1 Wavefront Reconstruction

After training our wavefront reconstruction network, we evaluated its performance in two different ways. First, we randomly generated 700 new wavefronts using the same parameters described above. Again, the direction and the velocity were randomly chosen for each simulation and this data was not used to train the original model. In Table 1 we show the average wavefront error in nm over all 700 of the new simulated data. Because our method outputs both low, and high resolution reconstructions we include the performance of both outputs. In Figure 3 we show two examples of the reconstruction methods output, and in Figure 4 we show the higher resolution reconstructions from our method for the same two samples.

To better evaluate how our model works in practice, we also compare our methods in closed loop simulation. Again, with the same parameters described above, we perform 100 simulations of a two minute exposure in closed loop. For comparison we include results using the slopes directly with the command matrix as well our method through recreating the wavefront and using the influence function matrix to apply deformations to the mirror. In Table 2 we show the averaged results for all three methods as measured in Strehl ratio for different NGS magnitudes and in Figure 6 we show two resulting PSF, each from a single simulation.

Our initial results show that we can well reconstruct wavefronts up to five NGS magnitudes outside of our training data. The cases in which our method does not perform well are typically wavefronts which contain large spikes near the edges of the pupil, as seen in Figure 5. This type of error can be reduced by ensuring these types of wavefronts are present in the training data, or moving to a pseudo-open loop simulation. Because reconstructing a higher resolution wavefront does not appear to provide better on average error, we may also achieve greater results by training a model with only the original slope-resolution reconstruction, allowing our model to focus exclusively on this resolution may enable better closed loop results as well.

NGS Magnitude	Ours (Low Res)	Ours (High Res)
0	138.2	187.3
5	140.5	181.0
10	146.4	216.2
15	464.4	502.9

Table 1: Average wavefront error in nm RMS from 700 randomly generated wavefronts reconstructed from slopes. Here Low-Res refers to the network output which corresponds to the size of the slope grid and High Res is two times the resolution.

NGS Magnitude	Command Matrix	Ours (High-Res)
0	56.98	48.65
5	57.69	48.66
10	58.32	27.89

Table 2: Average Strehl ratio from 100 closed loop AO simulations. The slopes column shows the result of using the slopes and command matrix directly while our method relies on first reconstructing the wavefront before using the influence function matrix to apply deformations to the mirror.

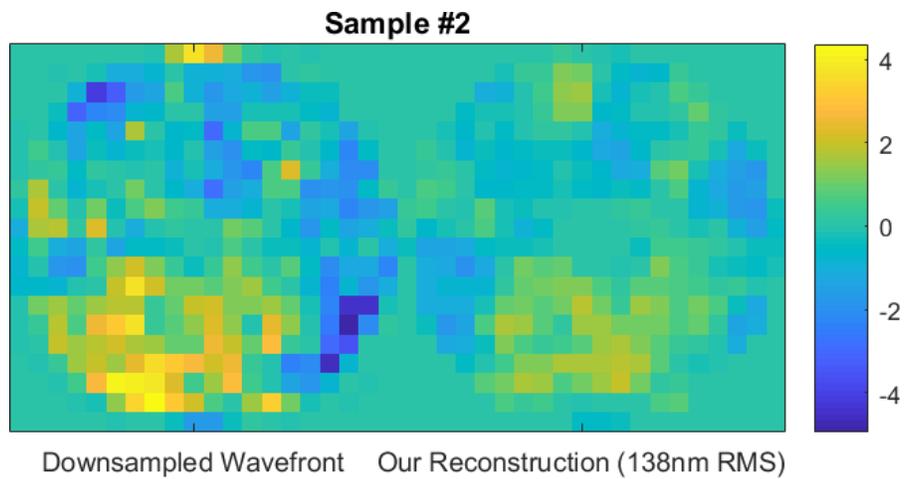
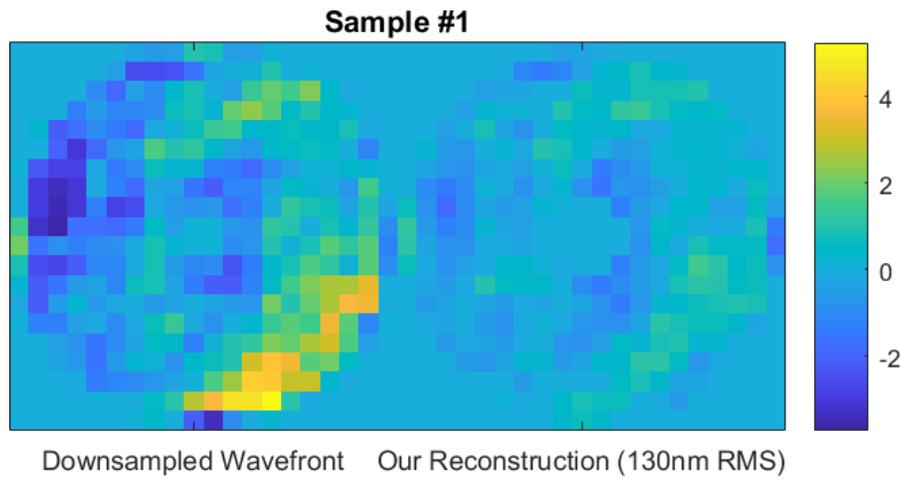


Figure 3: Wavefront reconstruction comparison between downsampled ground truth our learned model. Values displayed in nm.

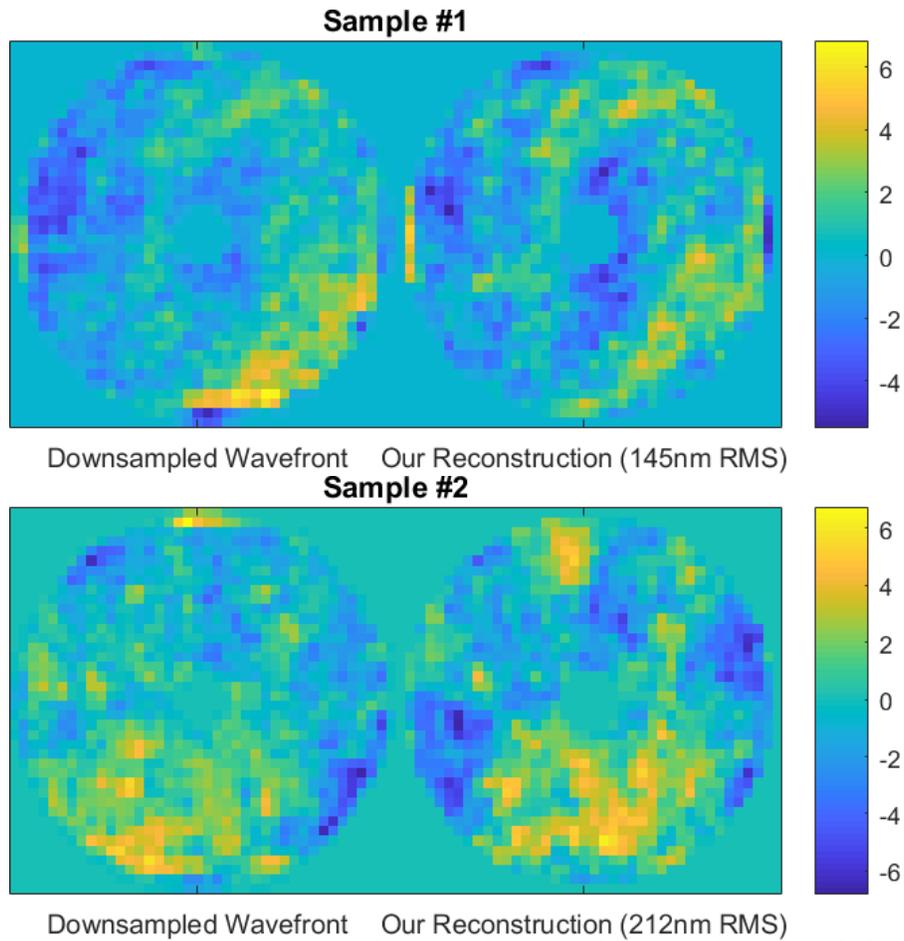


Figure 4: Wavefront reconstruction comparison between downsampled ground truth and the higher resolution output from our learned model. Values displayed in nm.

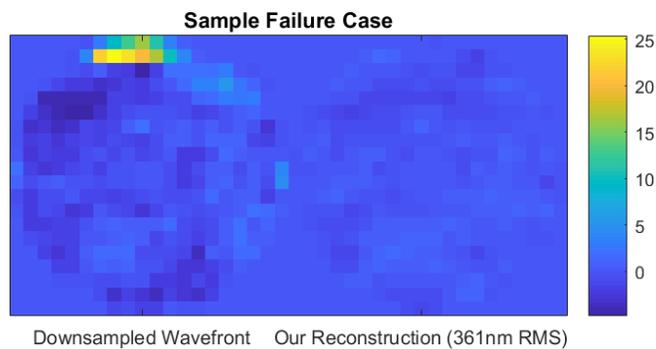


Figure 5: Example wavefront that our method was not able to well reconstruct. These types of errors can be mitigated in the future with improved training data or working in a pseudo-open loop simulation where these types of wavefronts would be very unlikely to occur.

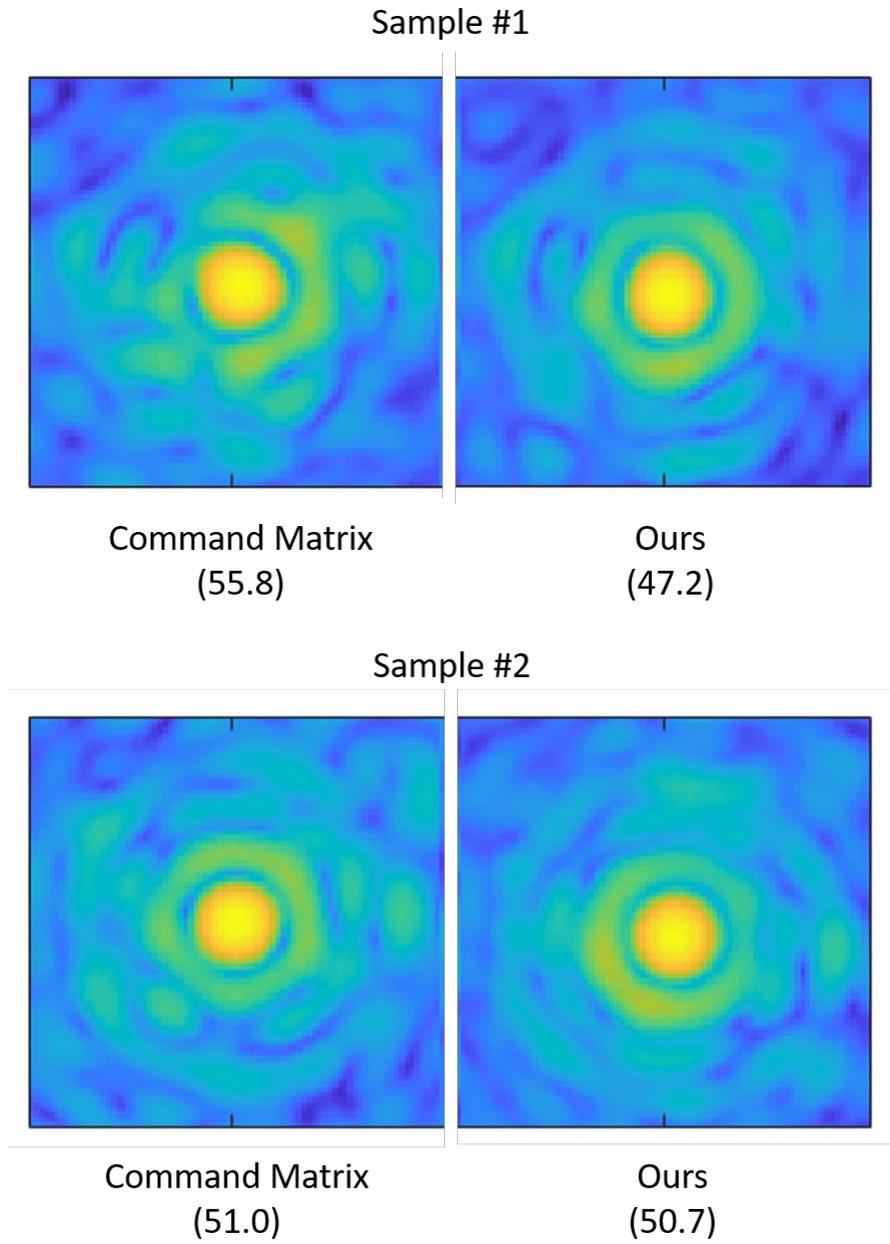


Figure 6: Resulting PSF from two closed loop simulations for both methods: slopes with command matrix and ours with influence function. PSF are presented in log scale and corresponding Strehl ratios are included in parenthesis.

Step	Predict	Reconstruct
0	—	0.260
1	0.423	0.442
2	0.586	0.597
3	0.691	0.811
4	0.779	1.012
5	0.919	—

Table 3: Average wavefront error in nm for both prediction and recreation of input wavefronts over 100 simulated wavefront series.

## 4.2 Wavefront Prediction

To evaluate our wavefront prediction model after training, we evaluate its predictive power for 100 new, randomly generated wavefront series. Here we show results when using the past 5 wavefronts. This number was chosen empirically as the minimum number of frames required to see improvements. More frames could give better results but will always require balancing the tradeoff between computation time and predictive power. However, our model is trained in a general way and can be used with any number of wavefronts without needing to retrain the network. In Table 3 we show the average wavefront error for each step of the predicted wavefront as well as the error for the recreated input wavefront. As can be seen in Figures 7 and 8, our predicted wavefront has a smaller wavefront error compared to holding the previous measurement constant. These results imply that the use of these predictive methods could reduce lag error in the control sequence leading to improved contrast and Strehl ratio.

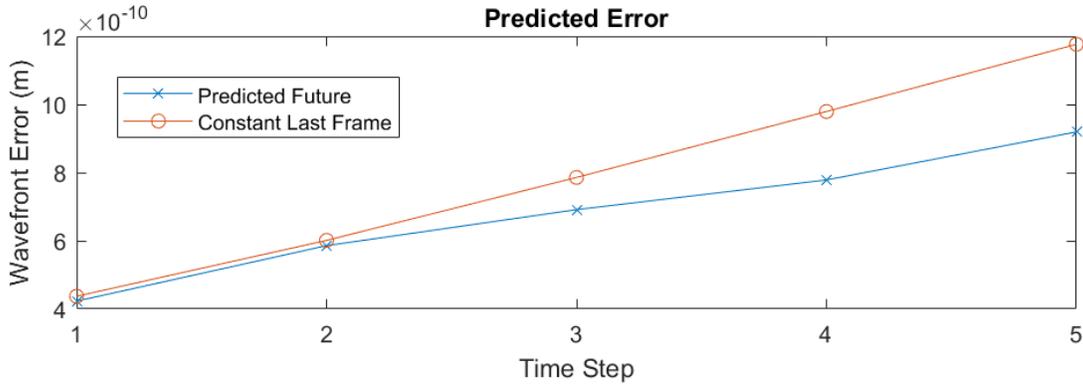


Figure 7: Average predicted wavefront error for each of the five simulation steps into the future. Also shown is the average wavefront error of holding the last known frame constant for each of these five steps.

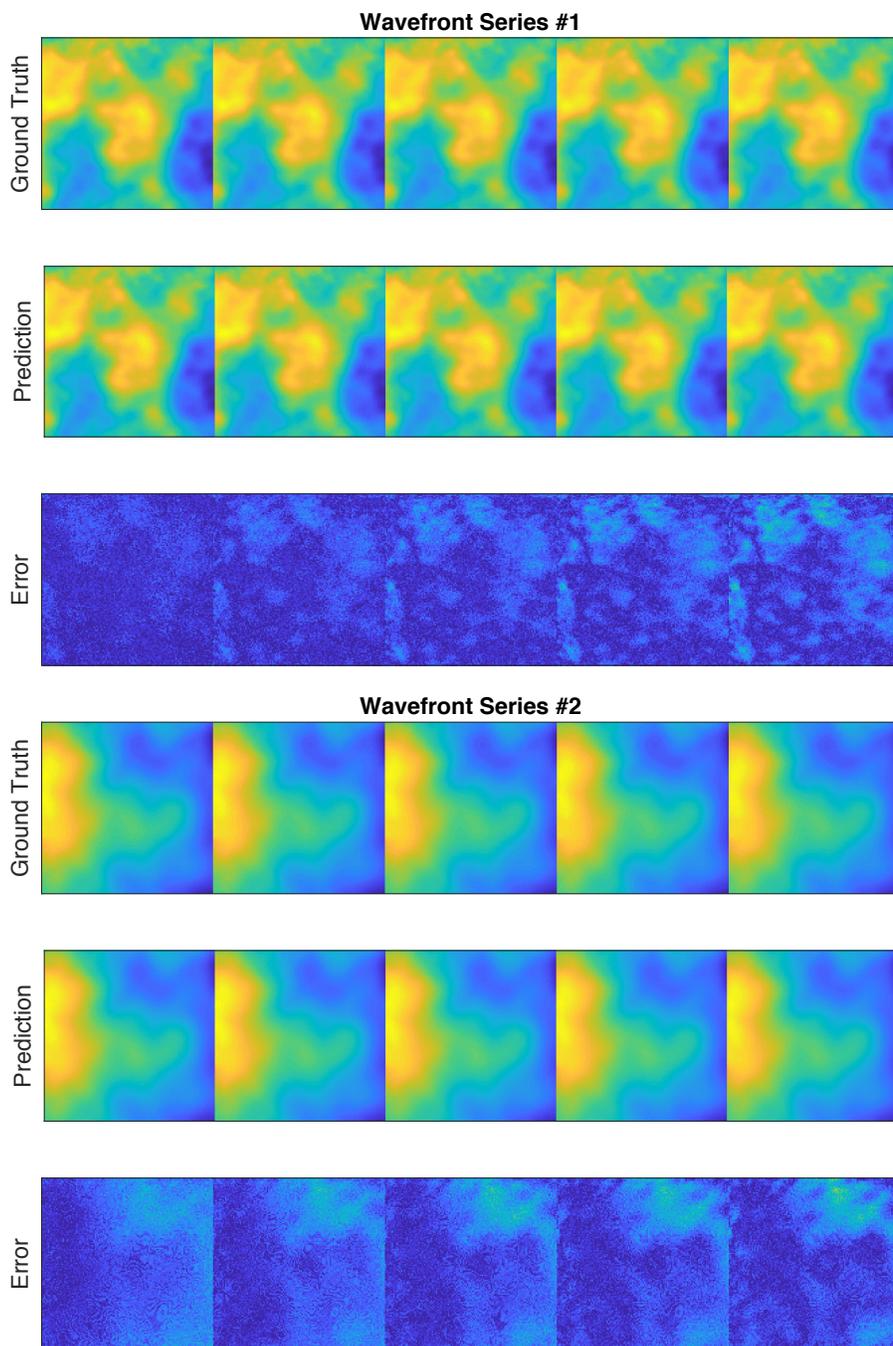


Figure 8: Two examples of predicted wavefronts. Given the five previous wavefronts, we predict the next five using our prediction network model. Here we show the five frames side by side for the ground truth wavefronts, our predicted wavefronts, and their absolute error below them.

## 5. CONCLUSION

### 5.1 Discussion

We have shown how modern, deep, convolutional neural networks can be applied to adaptive optics. We demonstrate a data-driven model for reconstructing the latent atmospheric wavefront from slope data that is comparable to other state of the art methods. We further show how convolutional LSTM networks can be applied to predicting future wavefronts with only five samples of previous data and in the future hope to both combine these two networks as well as apply similar methods to tomographic problems.

### 5.2 Future Work

Now that we have shown how these machine learning methods might be applied to AO and their capability there are many exciting future avenues for this work to continue. First, validating the predictive power of our model in closed loop simulation, comparing to previous methods, and assessing its contrast improvement. We then hope to combine the reconstruction and prediction networks in an end-to-end model. Finally, as a next step for our reconstruction model we hope to apply these methods to multi-sensor tomographic reconstruction.

Currently we are also in the process of assembling an AO bench to test our methods. We intend to train and test our model by applying a known phase map via a spatial light modulator.

## ACKNOWLEDGMENTS

R.S. was supported by the Vector Institute through the Vector Institute Affiliate Program.

## REFERENCES

- [1] Tyson, R. K. and Frazier, B. W., [*Field guide to adaptive optics*], vol. 2, SPIE Press (2004).
- [2] Osborn, J., Guzmán, D., de Cos Juez, F., Basden, A., Morris, T., Gendron, É., Butterley, T., Myers, R., Guesalaga, A., Lasheras, F. S., et al., “First on-sky results of a neural network based tomographic reconstructor: Carmen on canary,” in [*Adaptive Optics Systems IV*], **9148**, 91484M, International Society for Optics and Photonics (2014).
- [3] Conan, R. and Correia, C., “Object-oriented matlab adaptive optics toolbox,” in [*Adaptive Optics Systems IV*], **9148**, 91486C, International Society for Optics and Photonics (2014).
- [4] Ronneberger, O., Fischer, P., and Brox, T., “U-net: Convolutional networks for biomedical image segmentation,” in [*International Conference on Medical image computing and computer-assisted intervention*], 234–241, Springer (2015).
- [5] Maas, A. L., Hannun, A. Y., and Ng, A. Y., “Rectifier nonlinearities improve neural network acoustic models,” in [*Proc. icml*], **30**(1), 3 (2013).
- [6] Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c., “Convolutional lstm network: A machine learning approach for precipitation nowcasting,” in [*Advances in neural information processing systems*], 802–810 (2015).