Predictive AO Control with Convolutional Neural Networks

Robin Swanson, Masen Lamb, Carlos Correia, Suresh Sivanandam, Kyros Kutulakos
robin@cs.toronto.edu

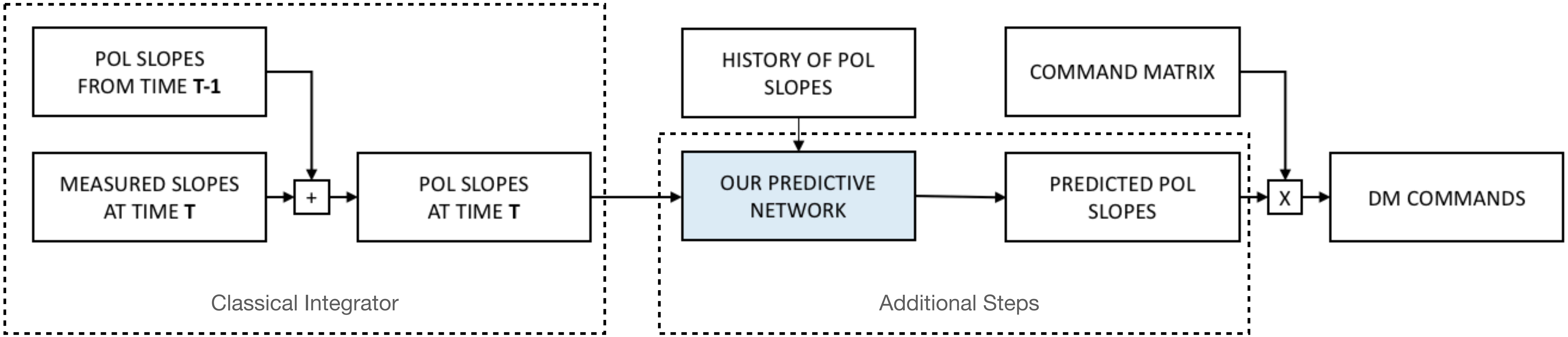
Project Goals and Outcomes

- **Goal:**
  - Create a general purpose, predictive adaptive optics (AO) integrator based on convolutional neural networks (CNN)
- **Results:**
  - A novel way to train robust, closed-loop integrators
  - Two CNN models that outperform classical methods under all conditions
  - One greatly improves servo-lag at low magnitudes
  - Another which performs better at removing noise at high magnitudes

Motivation for Using CNNs

- **Servo Lag:**
  - Delay between slope measurement and commands sent to the DM
  - Very noticeable for lower magnitude guide stars
  - Reduces contrast — important for exoplanet imaging
- **Noise:**
  - Dominates at high magnitude guide stars
  - Degrades AO performance (Strehl Ratio)

Predictive Pseudo Closed Loop Operation

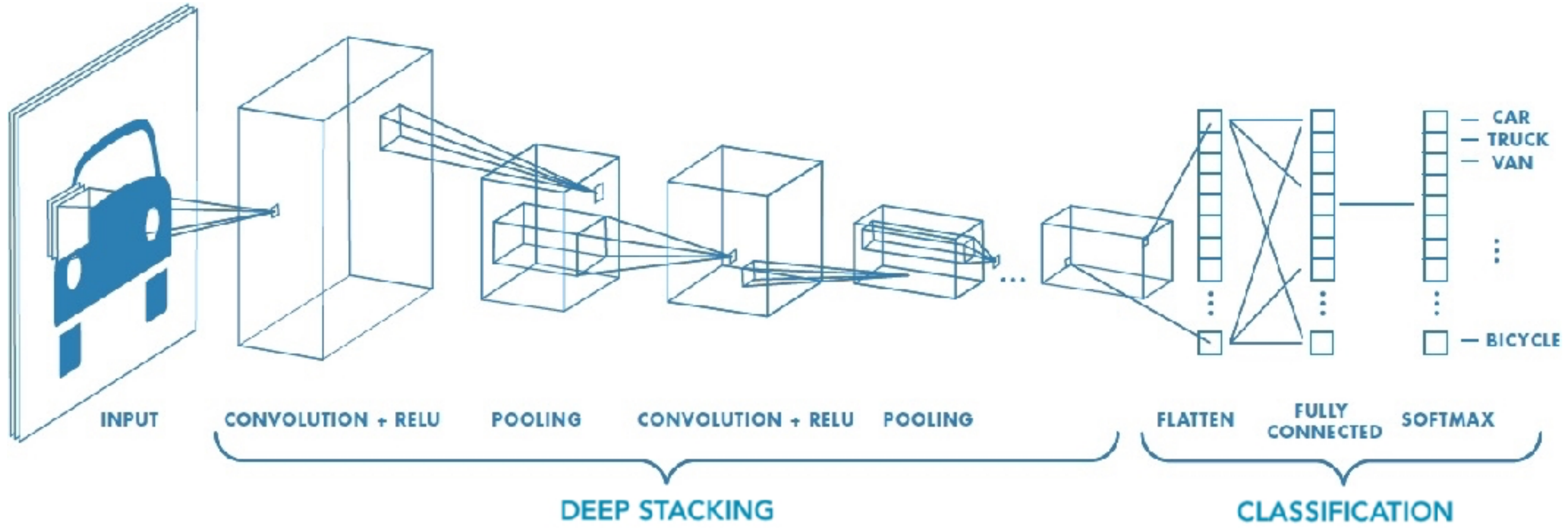


Brief History of Convolutional Neural Networks

CNNs: Unreasonably Effective

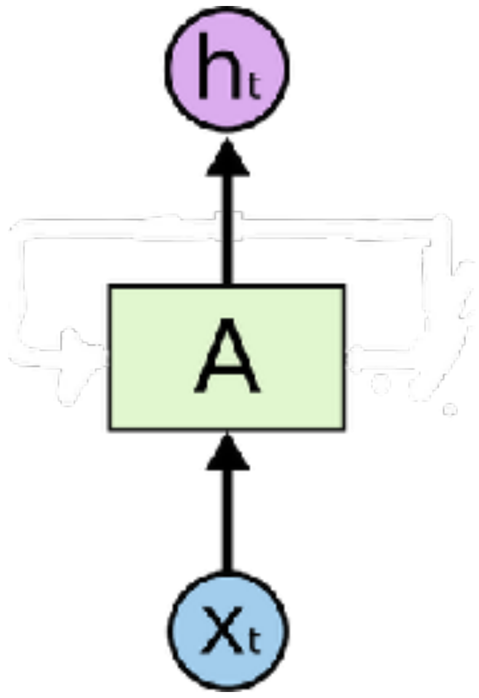
(Convolutional Neural Networks)

- Good at solving "difficult" visual problems
- Learned feature extractors at each convolutional layer
- Fully differentiable to train parameters via gradient descent
- Key enablers: GPU hardware, Millions of parameters, Lots of data



LSTMs: Making Time Matter

(Long Short Term Memory)

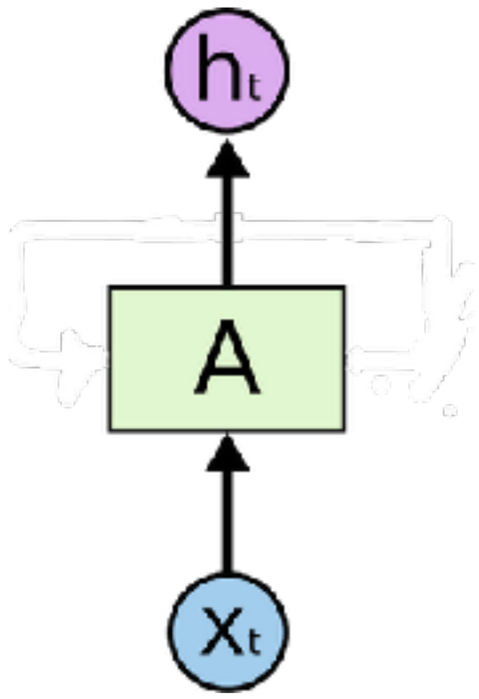


Feed Forward Neural Network

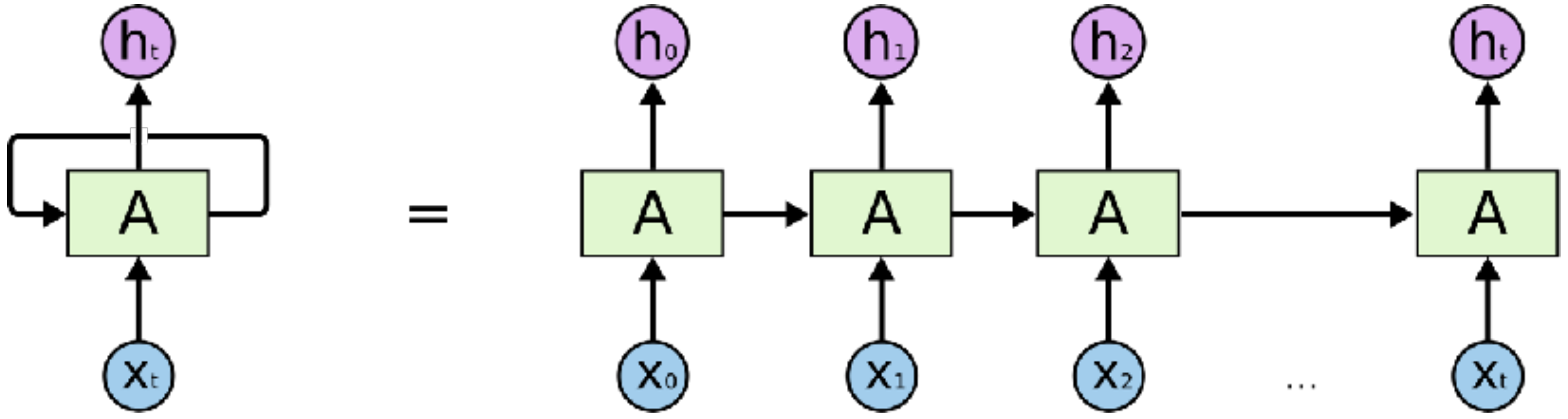
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTMs: Making Time Matter

(Long Short Term Memory)



Feed Forward Neural Network

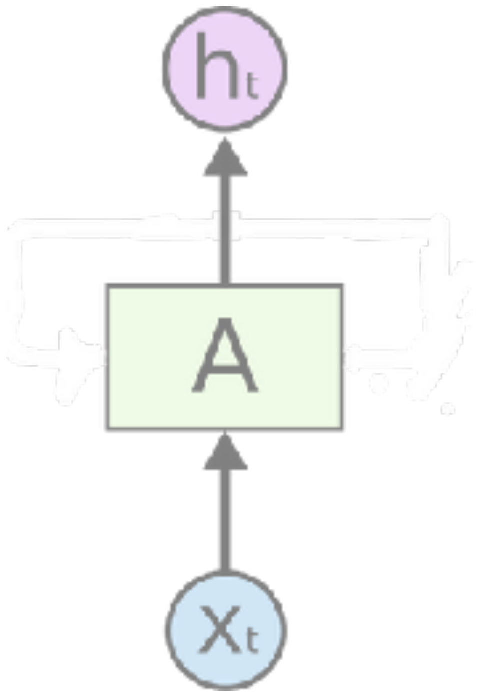


Unrolled Recurrent Neural Network

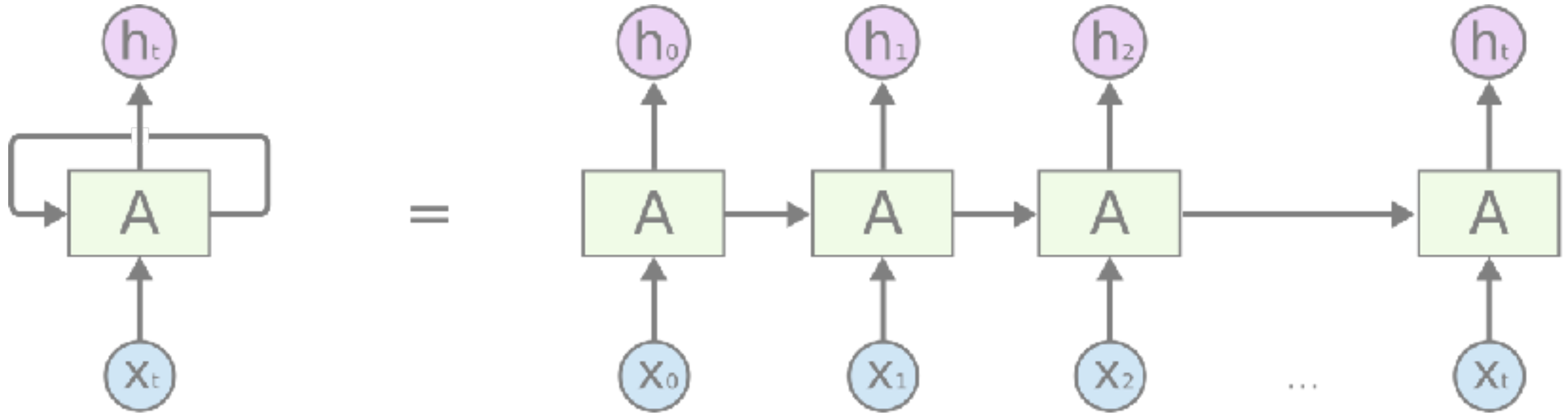
https://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTMs: Making Time Matter

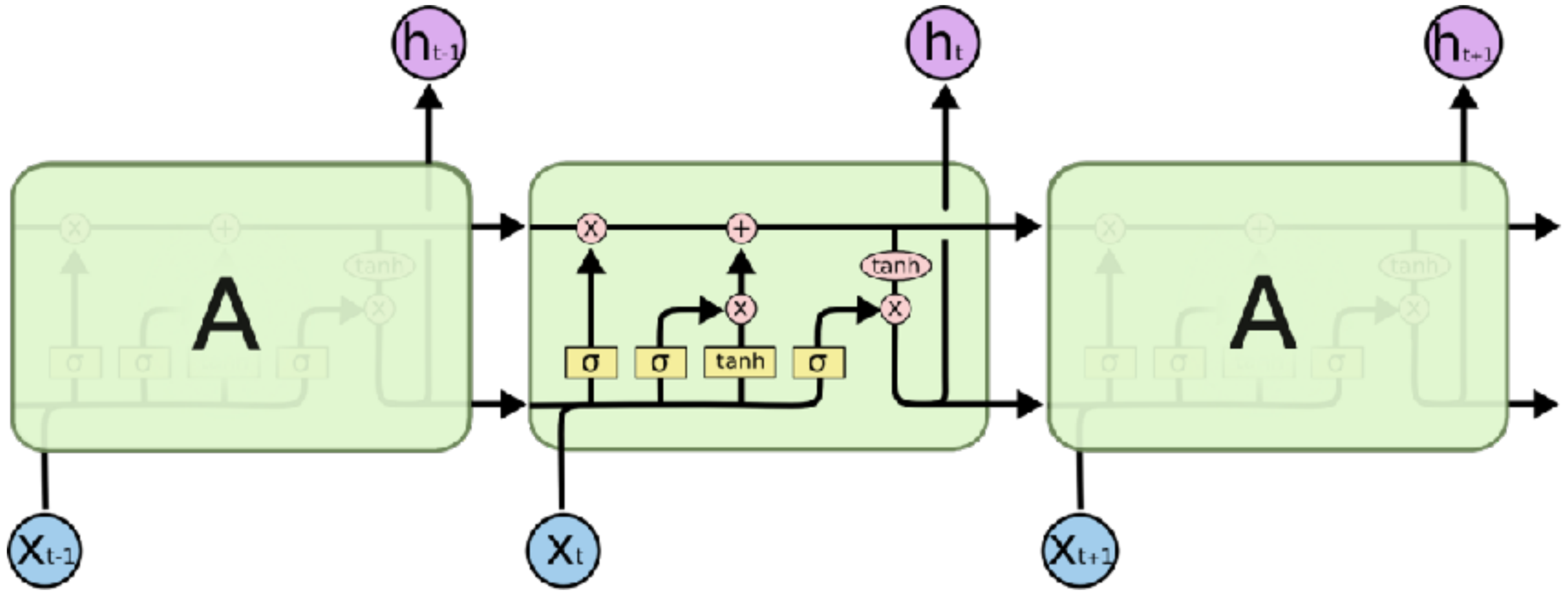
(Long Short Term Memory)



Feed Forward Neural Network



Unrolled Recurrent Neural Network



Long Short Term Memory (LSTMs)

https://colah.github.io/posts/2015-08-Understanding-LSTMs/

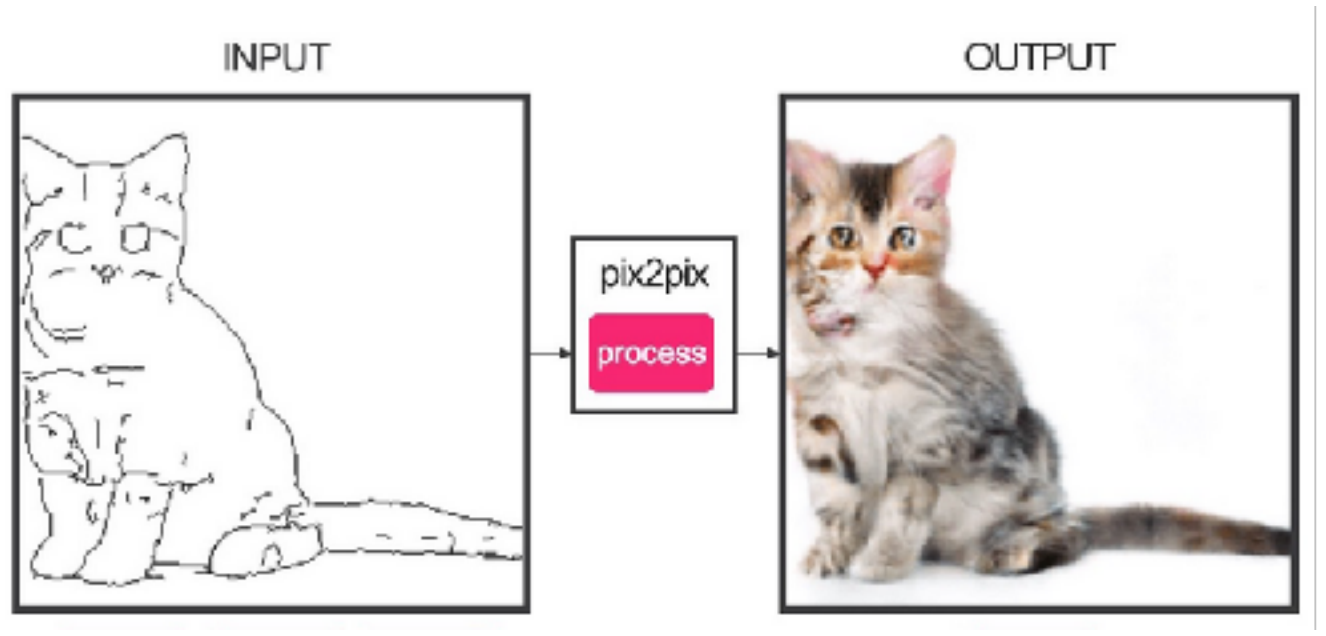
GANs: Solving Indescribable Problems

(Generative Adversarial Networks)

- How do we train a generative network to create new (original) images?
- Train a second CNN that judges the output of the generative CNN
- The generator must learn to "trick" the judge into thinking its output is from the true data
- Can be used to make simulated data look like experimental data

CycleGAN

thispersondoesnotexist.com



pix2pix

Motivation

- Use recent convolutional neural network and deep learning techniques to create a predictive AO controller
- Goals:
  - **Predictive** — Mitigate servo lag by predicting future slopes
  - **Denoising** — CNNs naturally produce images with low noise
  - **General Purpose** — Should work under a wide range of seeing conditions and not be restricted to any particular type of telescope
  - Compare CNN architectures to determine which models work best

Our Simulation Settings

- **8m** Telescope Diameter
- **800 Hz** Sampling Frequency
- **2 Frames** of Delay
- **16 x 16** Order SH WFS
- **17 x 17** Order DM
- **R Band** NGS
- **K Band** Science Camera
- **Pseudo Open Loop** (POL) Control

- **3 Layer** Atmosphere
- **[0, 4, 10] km** Altitude
- Frozen Flow

Data Generation

- Run thousands of independent OOMAO simulations to generate data

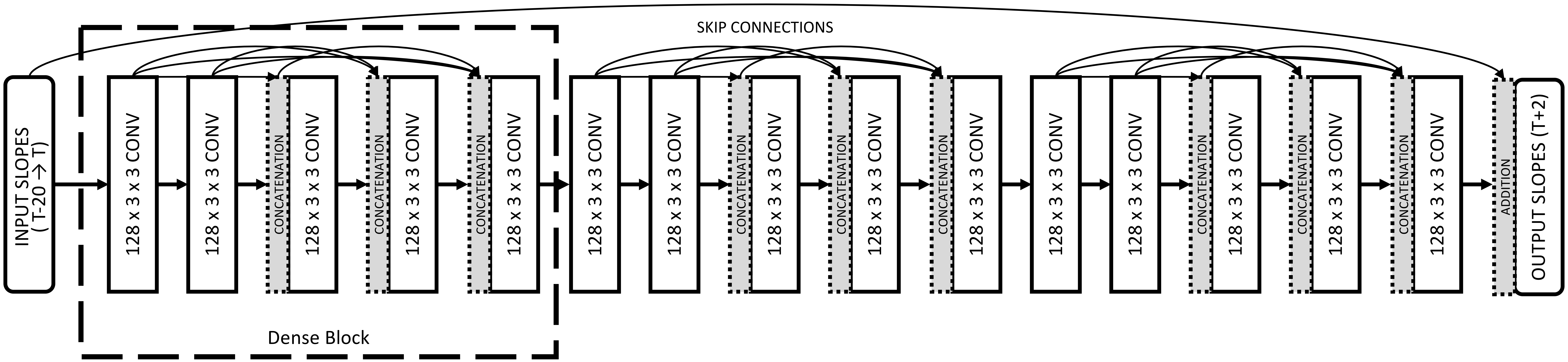
- **Data to Save:**
  - Classic Integrator Slopes
  - Ground Truth Slopes

- **Variables to Randomize**
  - $\mathbf{r_0}$: 0.15cm +/- 0.02
  - **Wind Direction**: $[0, 2\pi)$
  - **Wind Speed**: [5, 10, 15] km/s +/- [2.5, 5, 10]
  - **NGS Magnitude**: 8-16

Network Models and Training

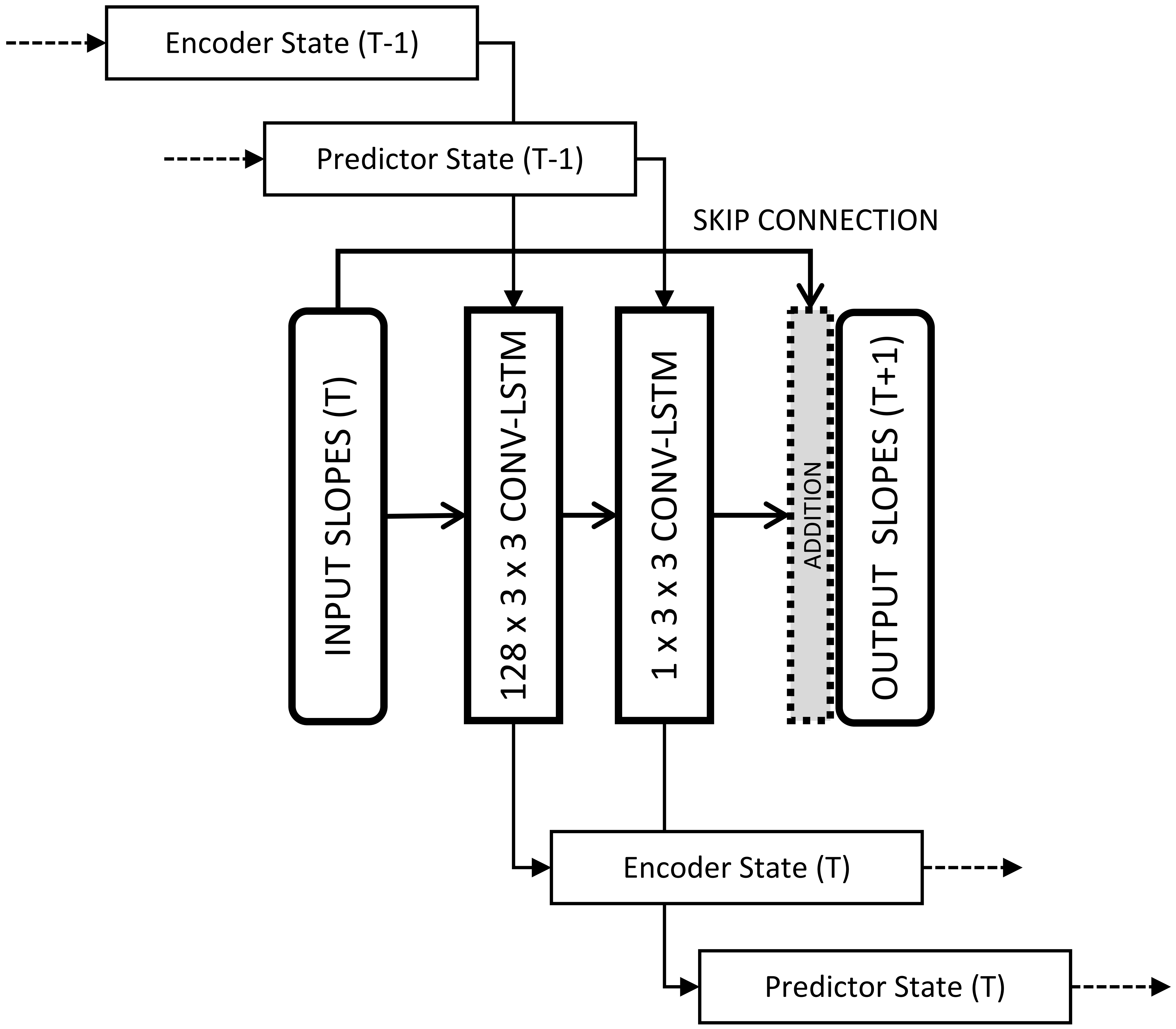
Our Models: Dense Network

- Takes in a 3D matrix of the slope maps (X, Y, Time)
- Uses information from the current slopes and 20 past loop steps
- Outputs a single set of slopes for time T+2 to mitigate the servo lag

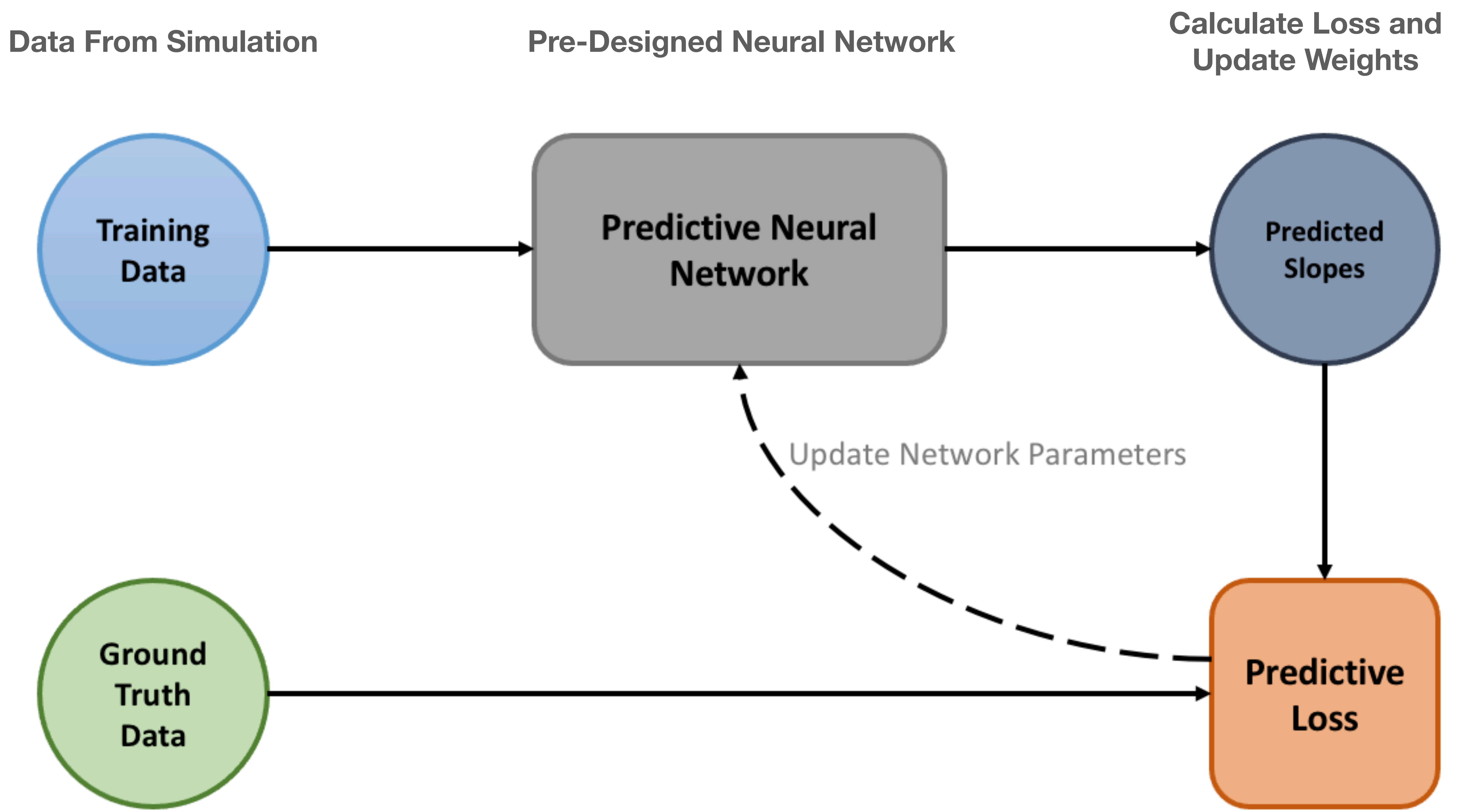


Our Models: LSTM Network

- Takes in a single set of slopes at a time
- Extracts relevant information and saves it in its state for the next time step
- Outputs a single set of slopes for time T+2



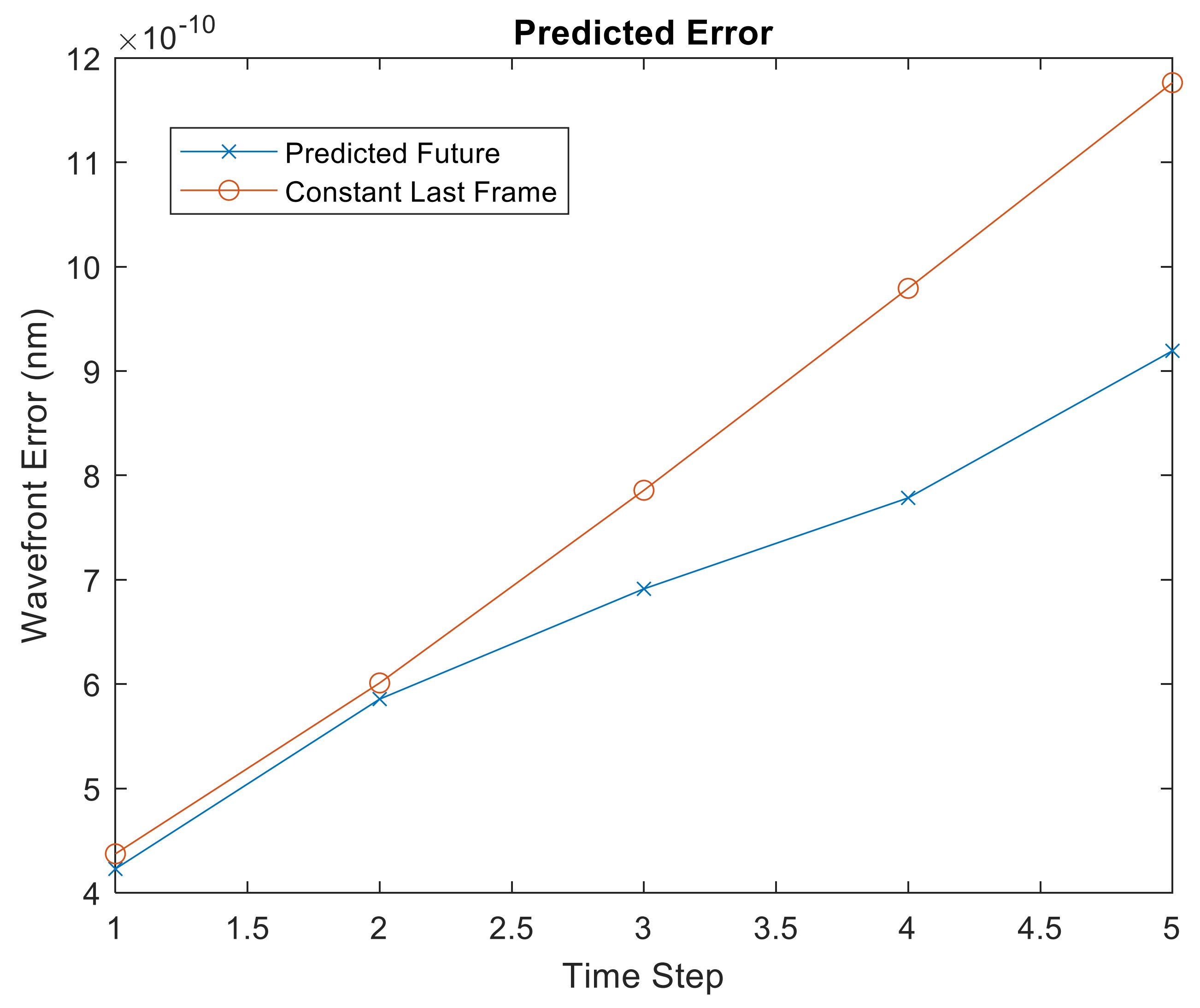
Predictive Network Training

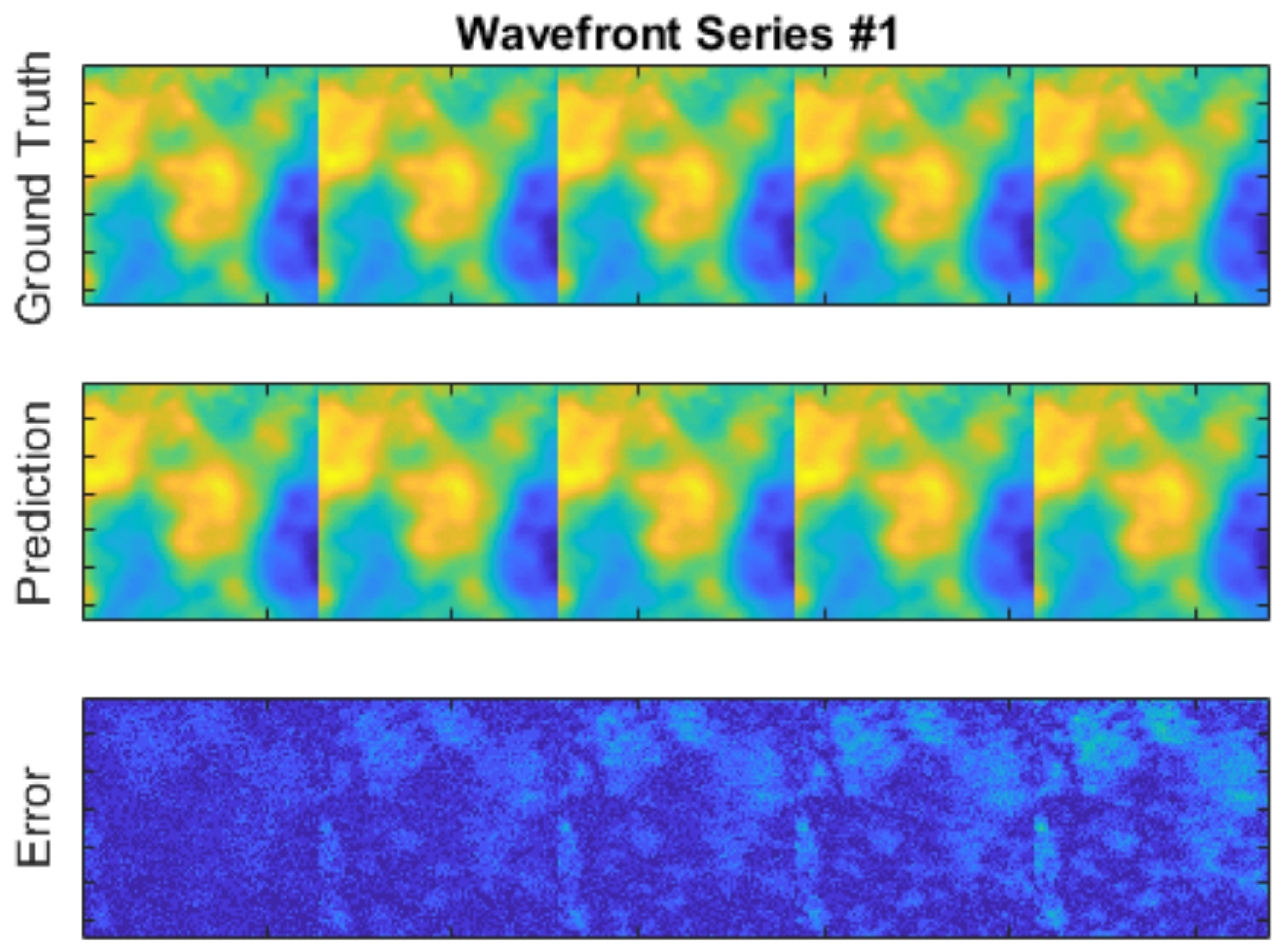


Open Loop Results

- Given a series of slopes (or wavefronts) our networks can very accurately predict several frames into the future

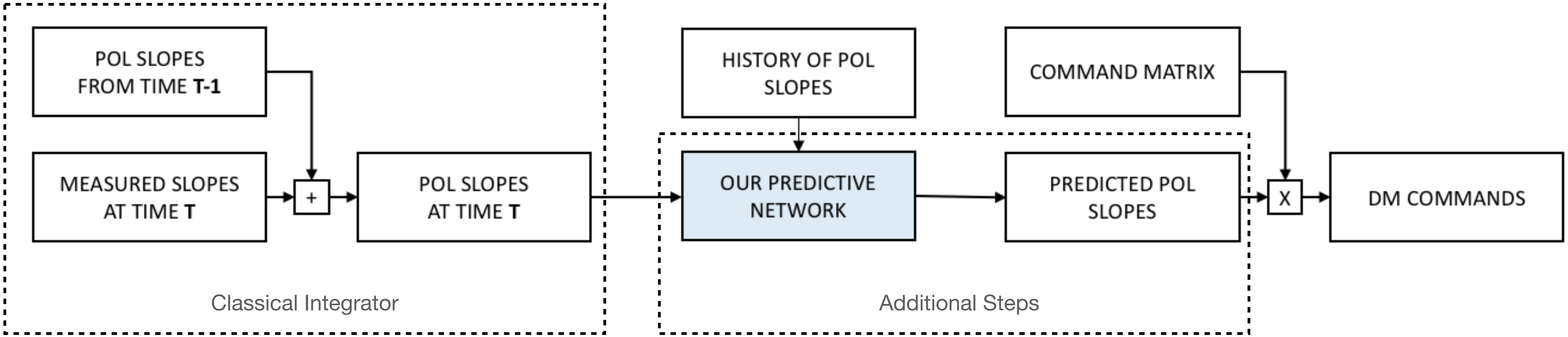
- However, this isn't how the integrator would be used in a closed loop system



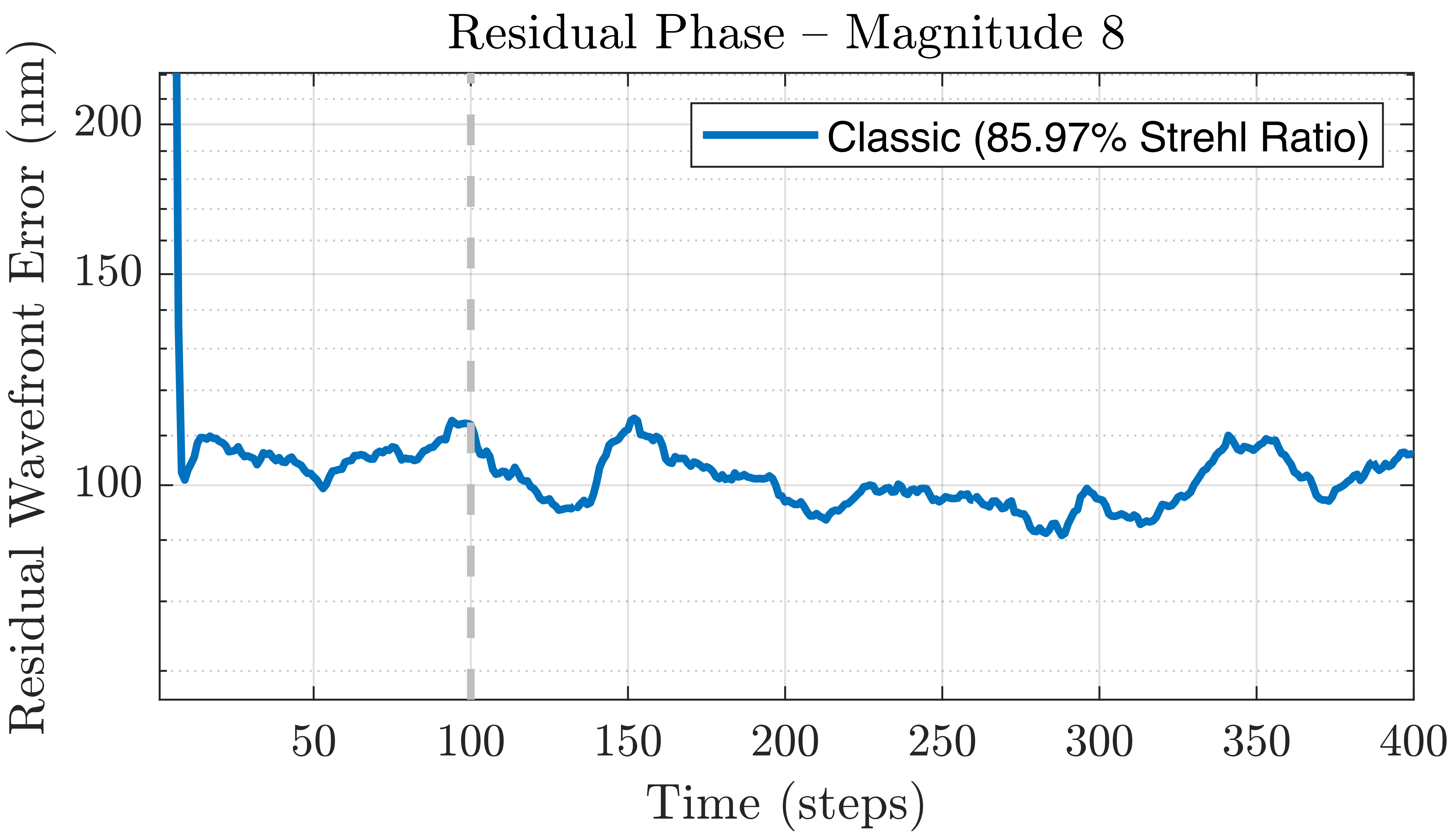


Swanson, Robin, et al. "Wavefront reconstruction and prediction with convolutional neural networks." *Adaptive Optics Systems VI*. Vol. 10703. International Society for Optics and Photonics, 2018.

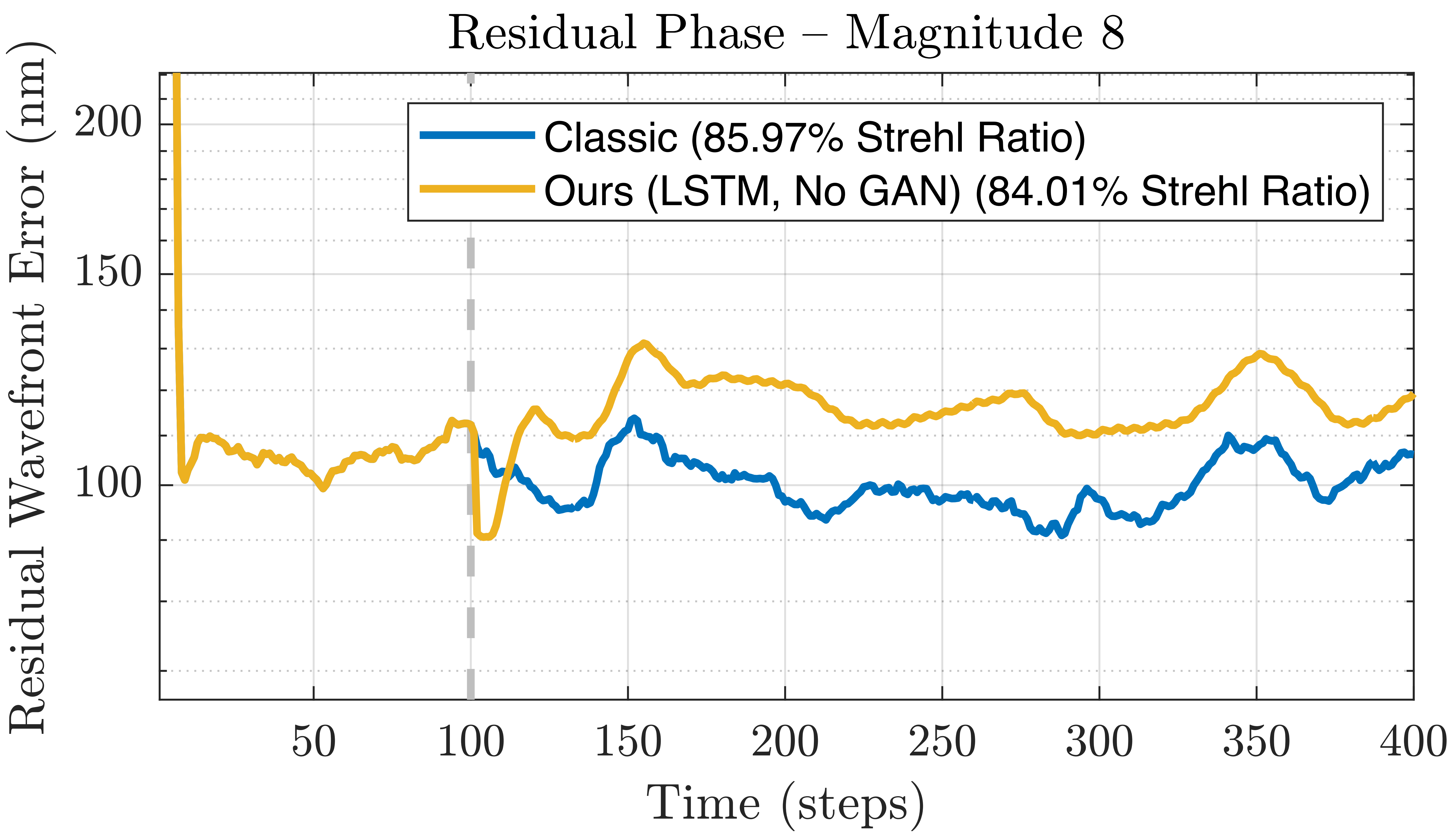
Predictive Pseudo Closed Loop Operation



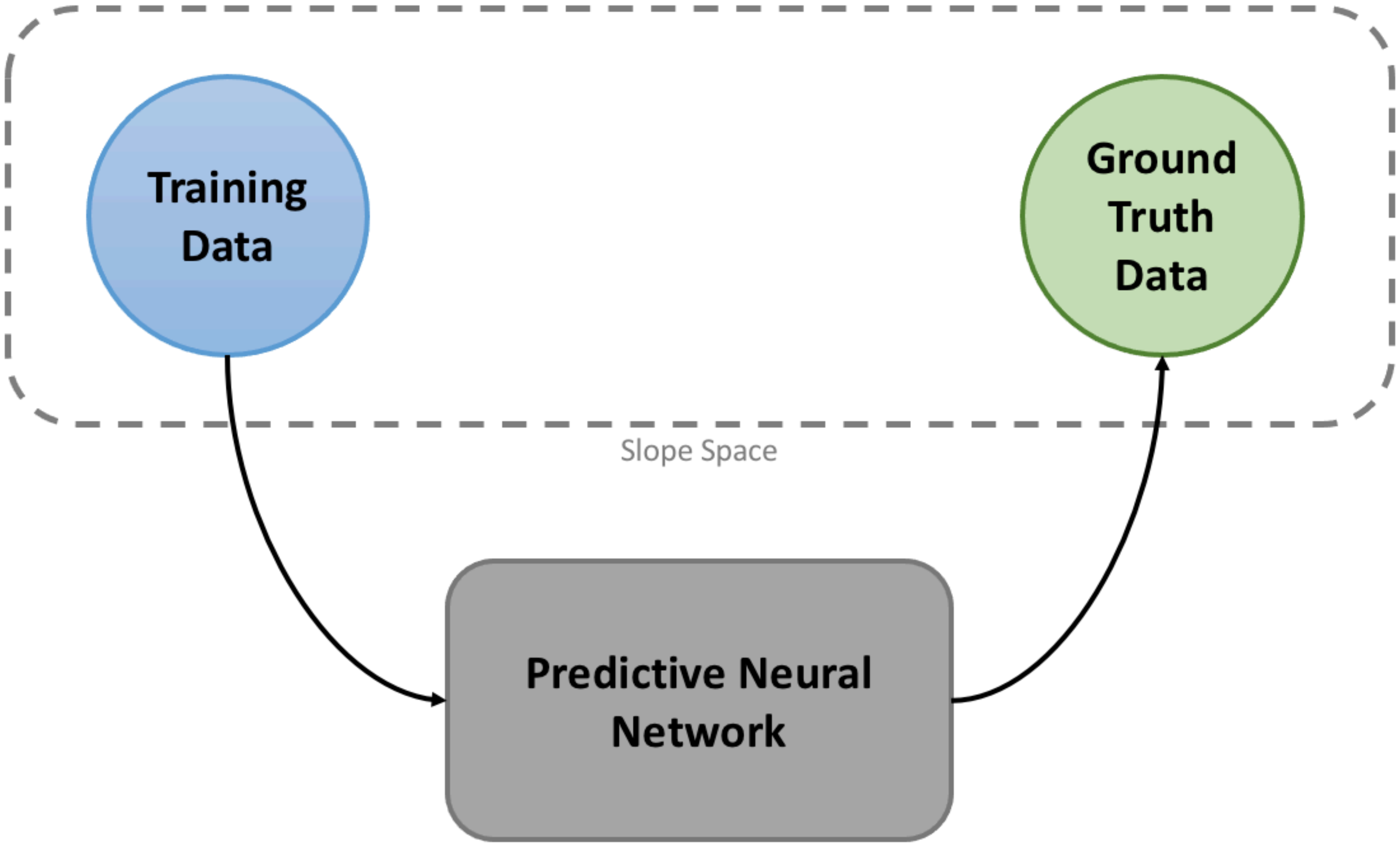
Closed Loop Results



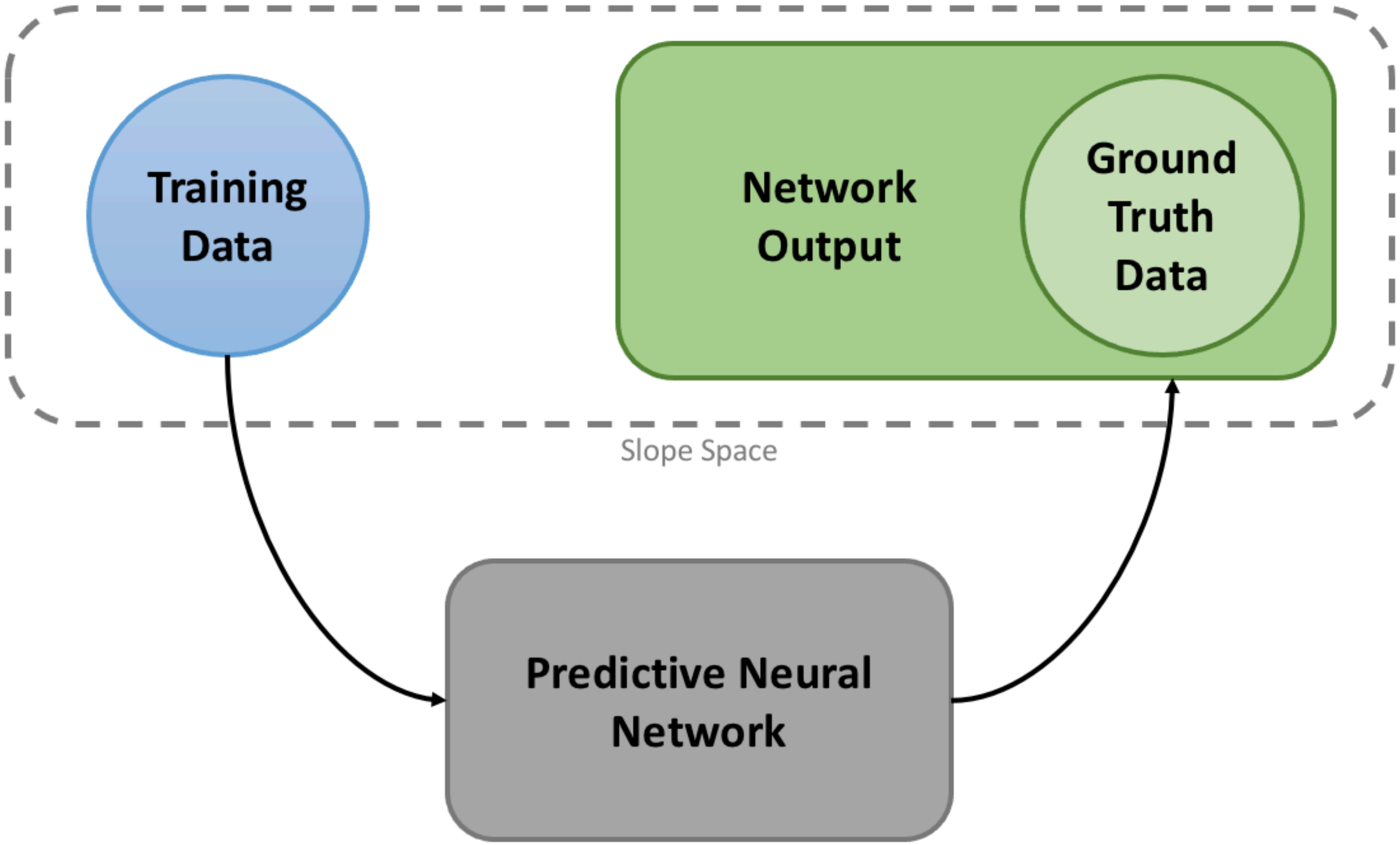
Closed Loop Results



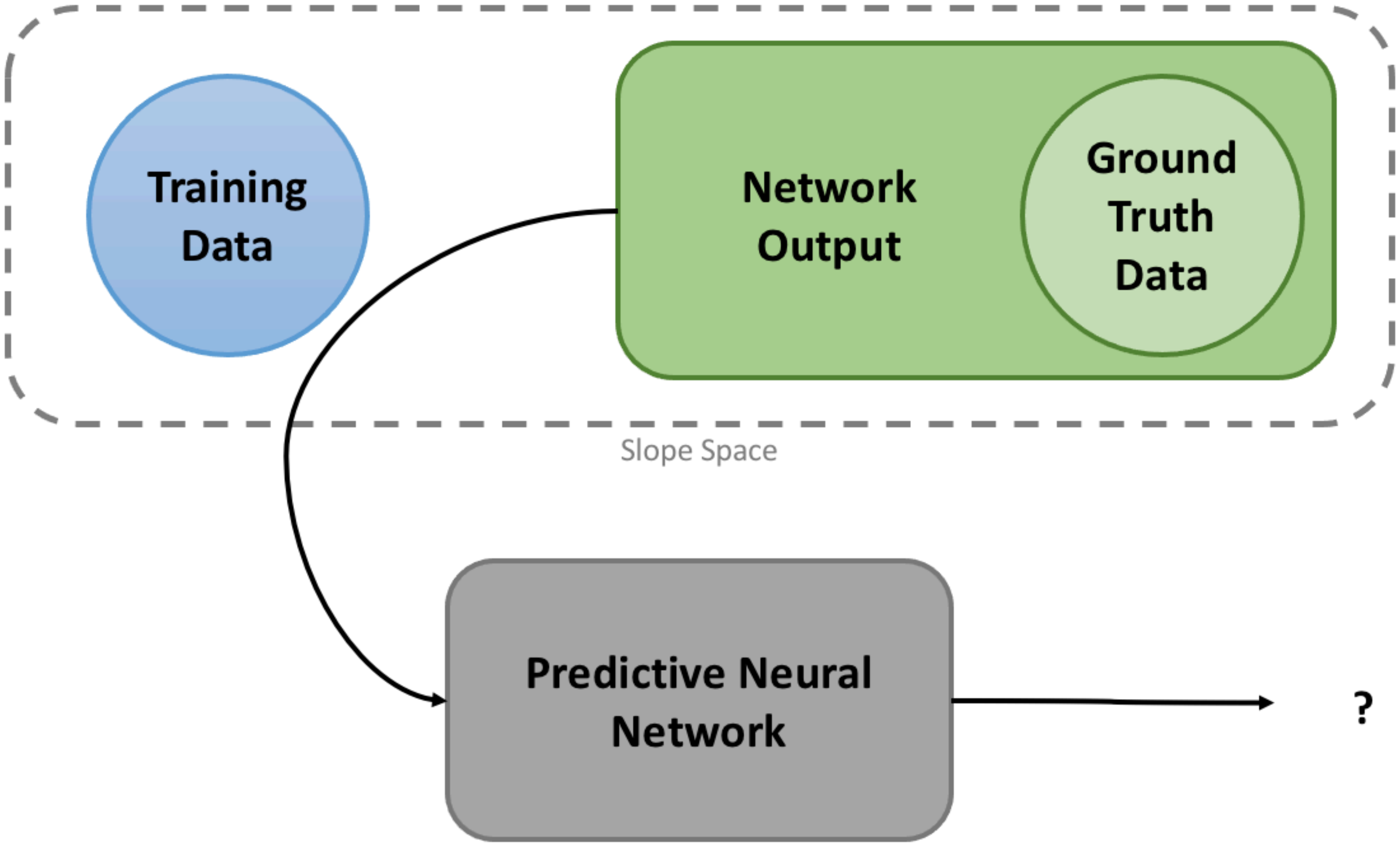
Closed Loop Divergence



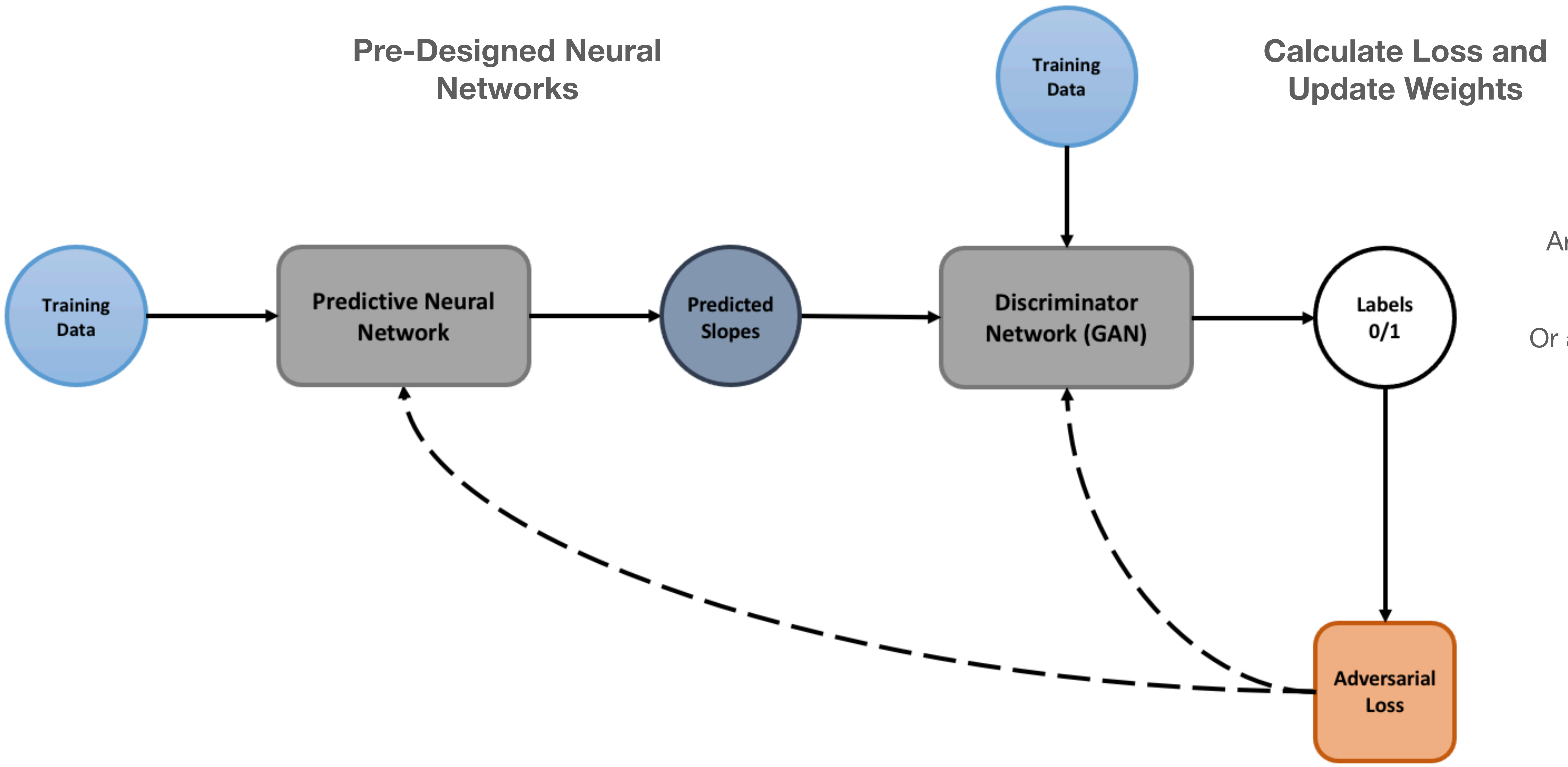
Closed Loop Divergence



Closed Loop Divergence



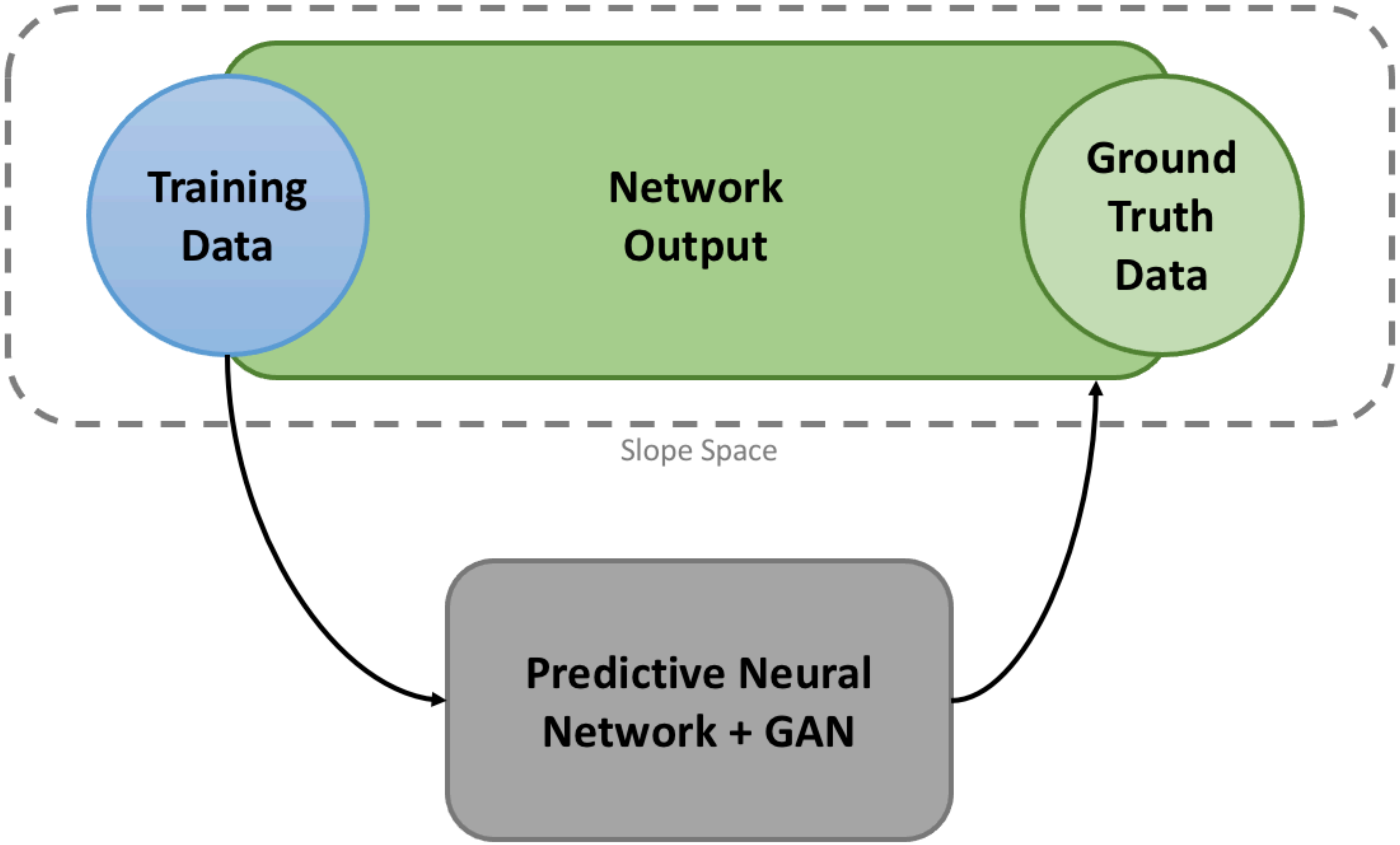
Adversarial Network Training



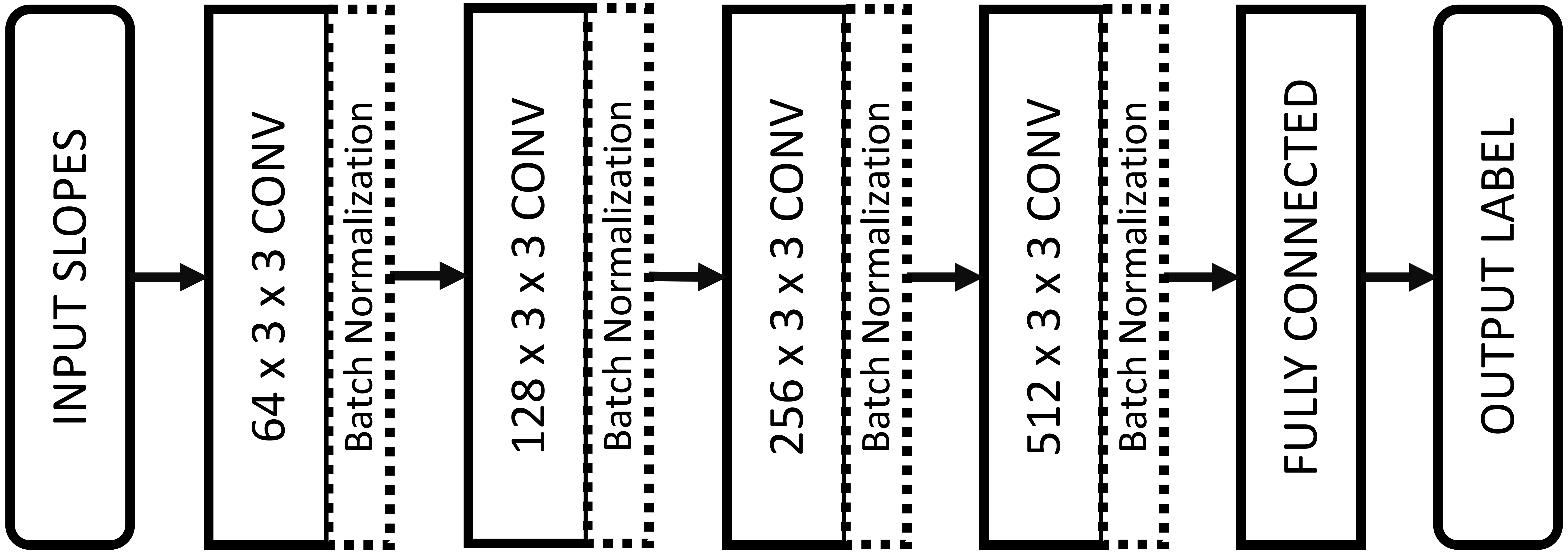
Are these slopes from the **training data set**?

Or are they **output from our neural network**?

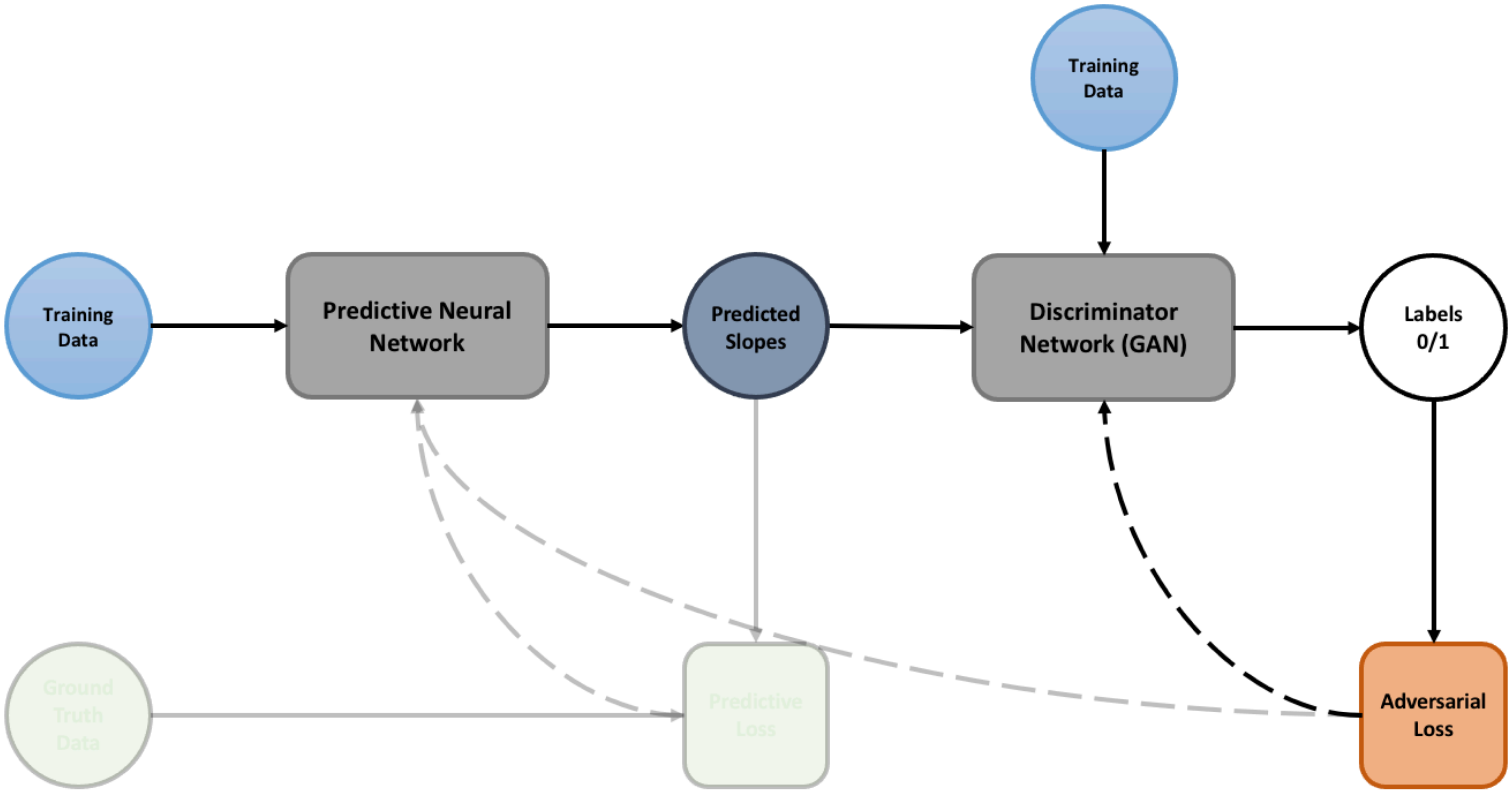
Closed Loop Divergence



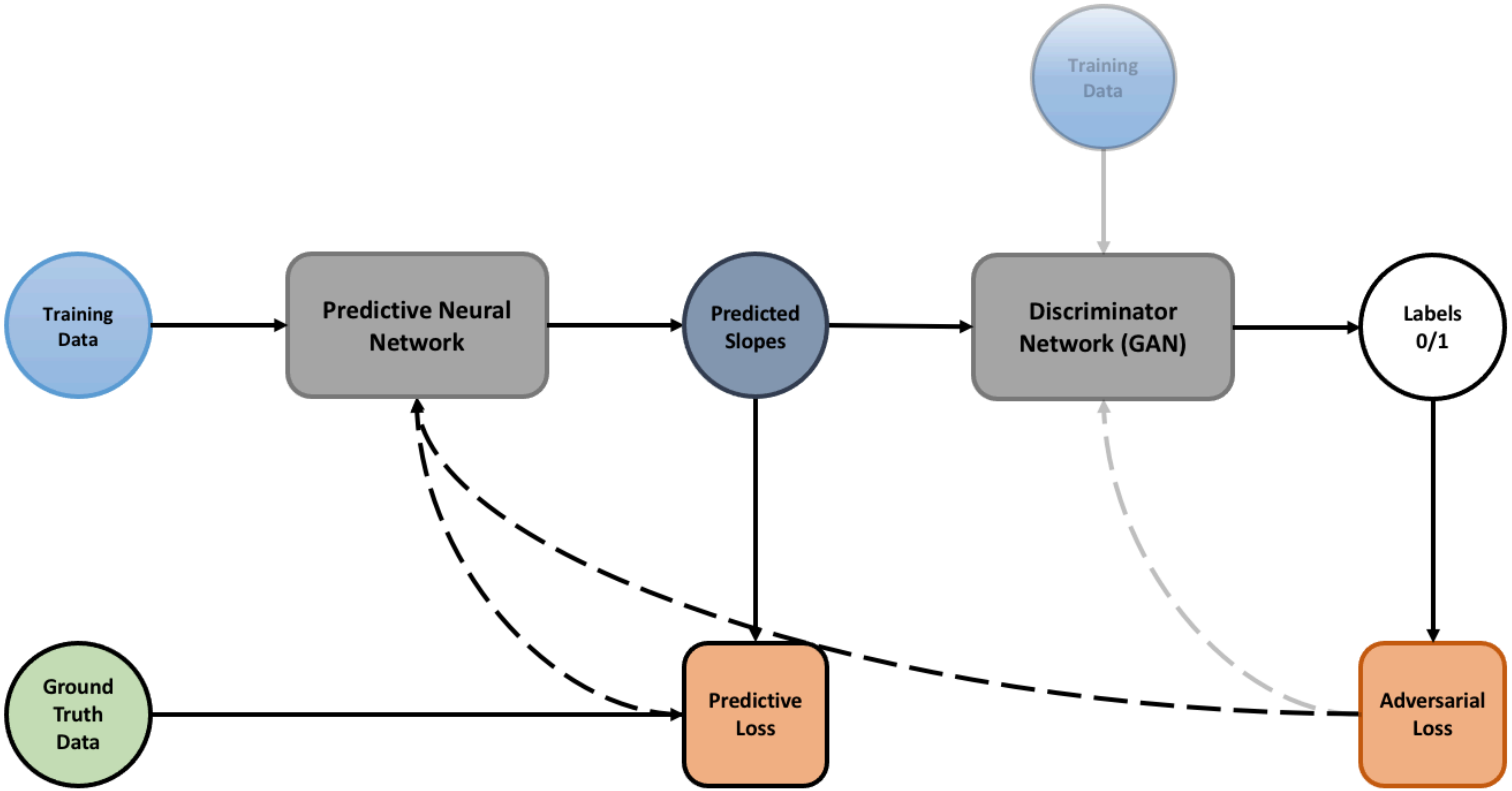
Our Models: Discriminator Network



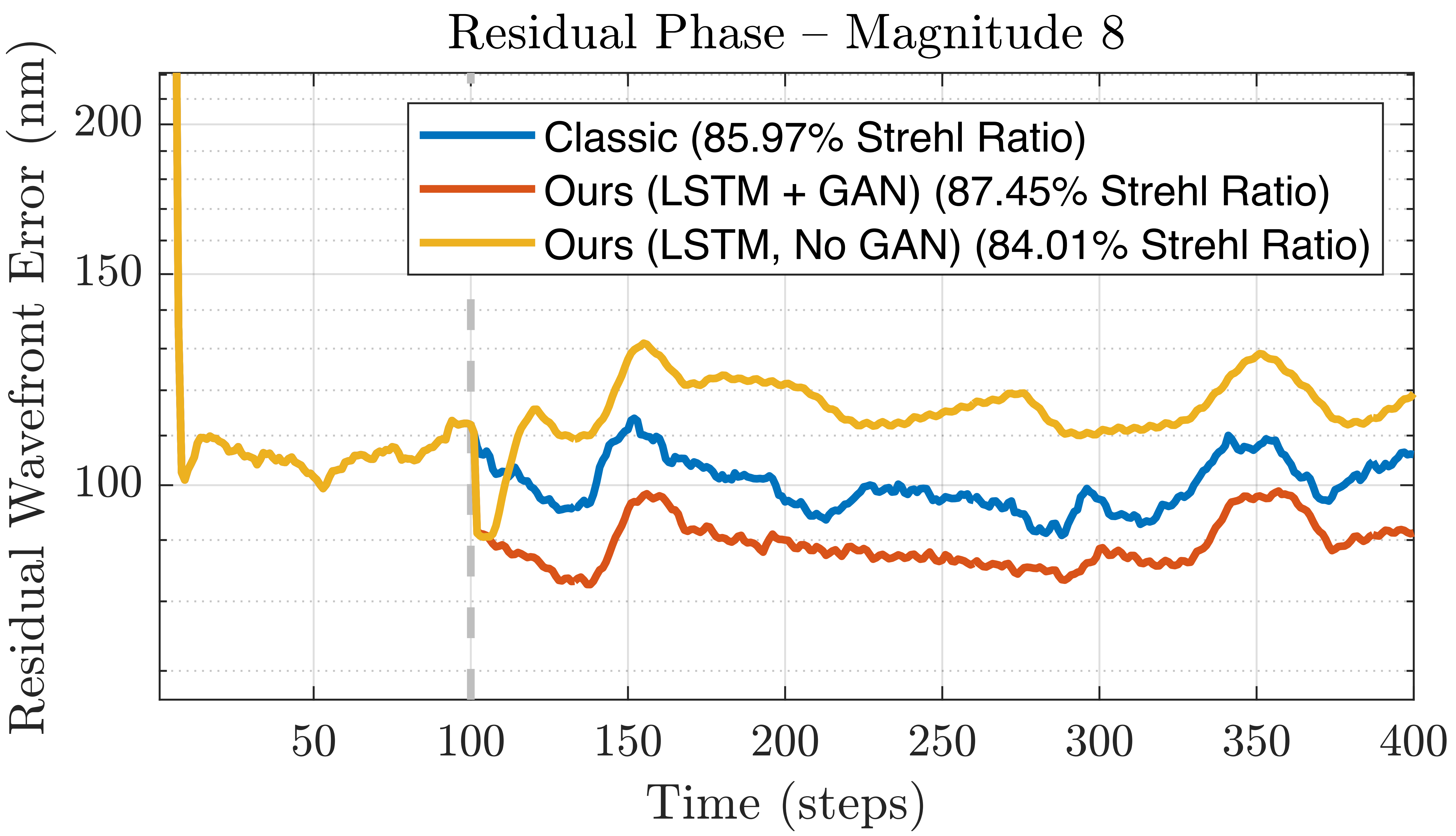
Full Training Map: Discriminator Update Step



Full Training Map: Predictive Update Step

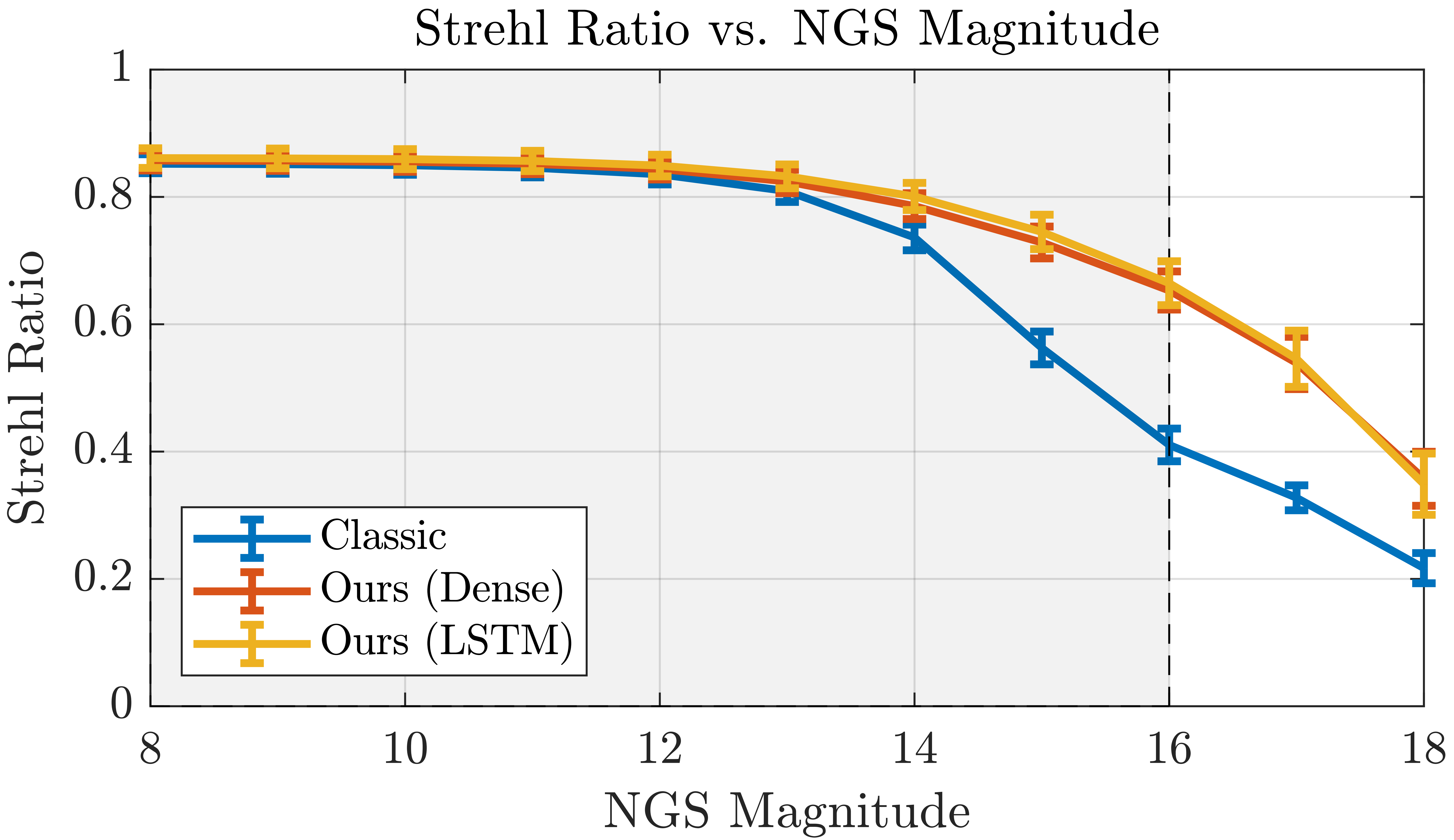


Closing The Loop With Adversarial Prior

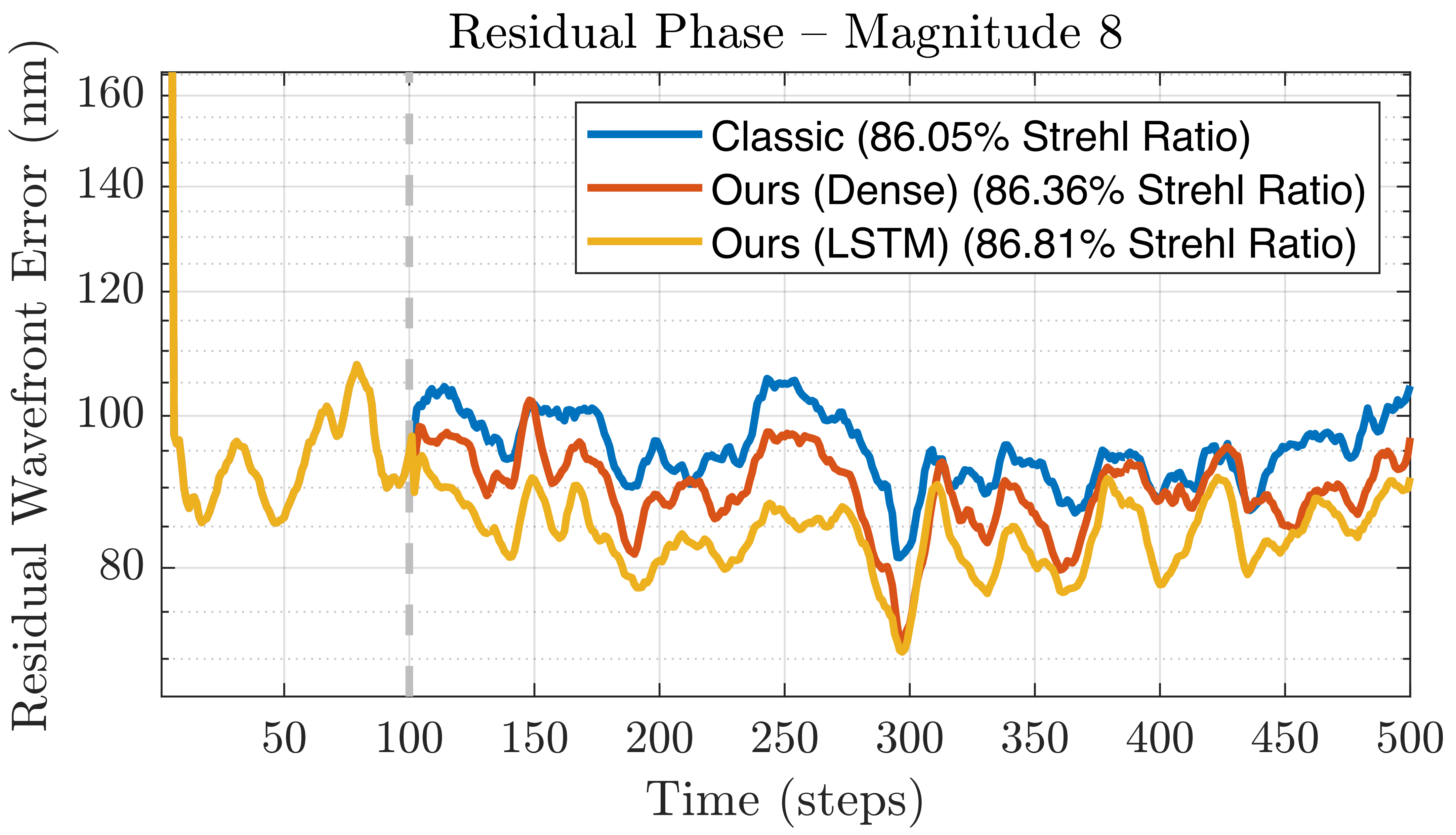


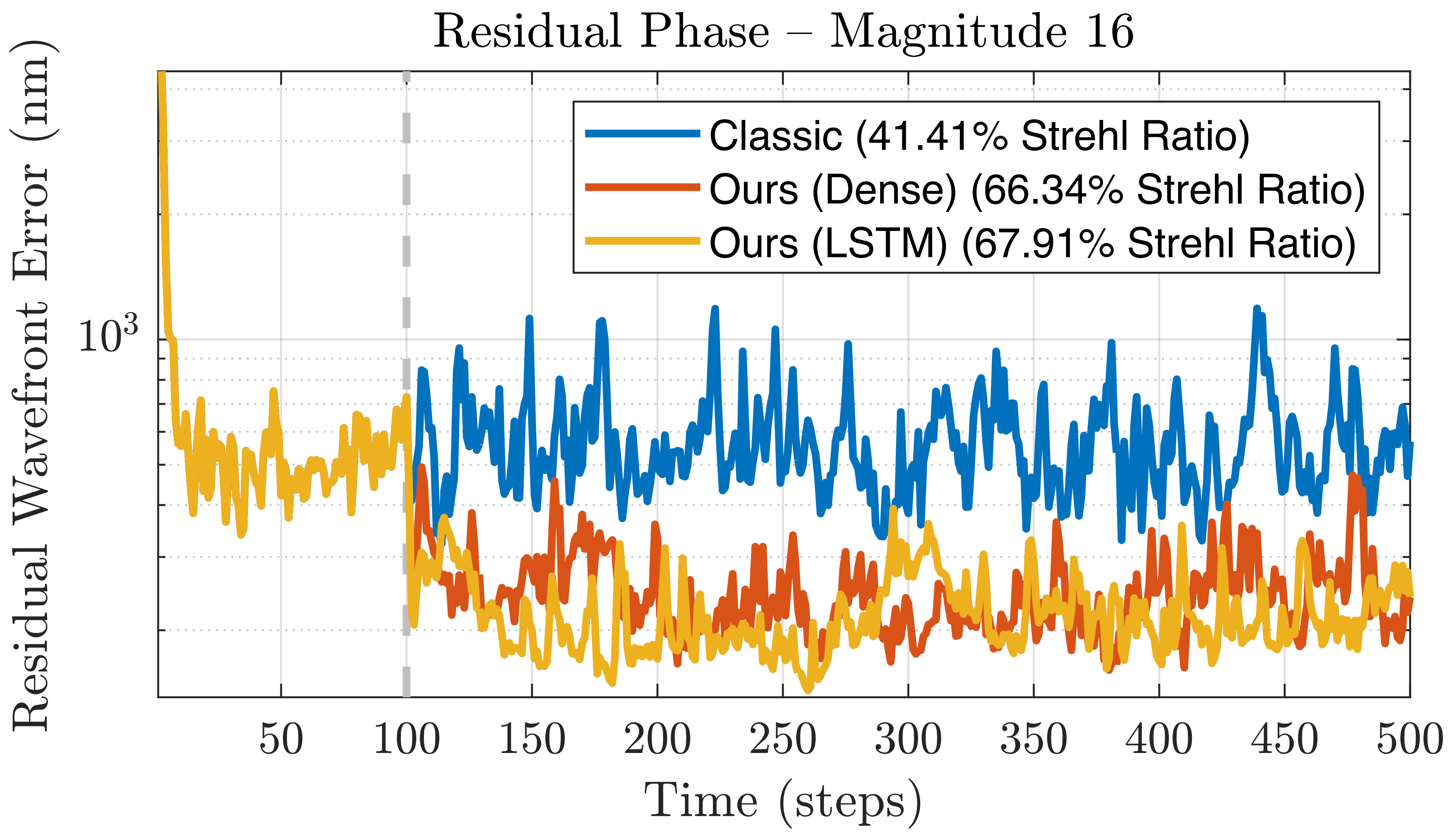
Results

Final Results: Strehl Ratios

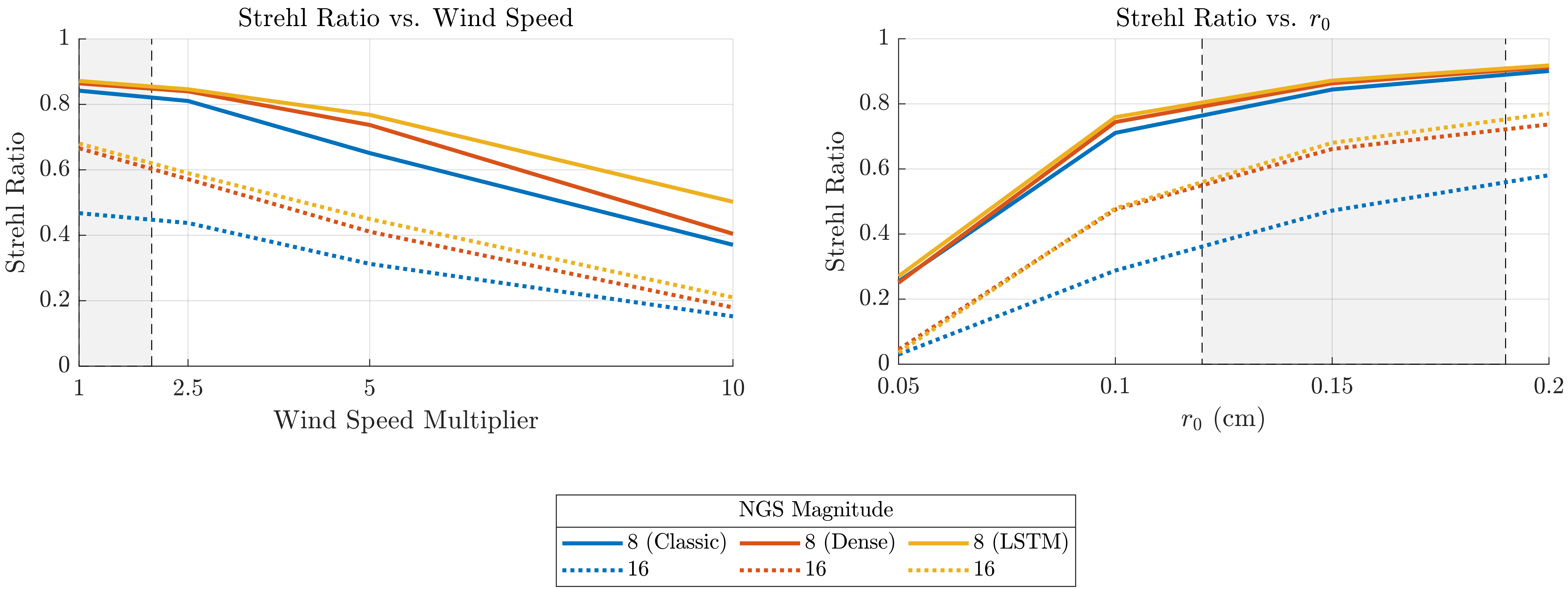


Final Results: Residual Wavefronts



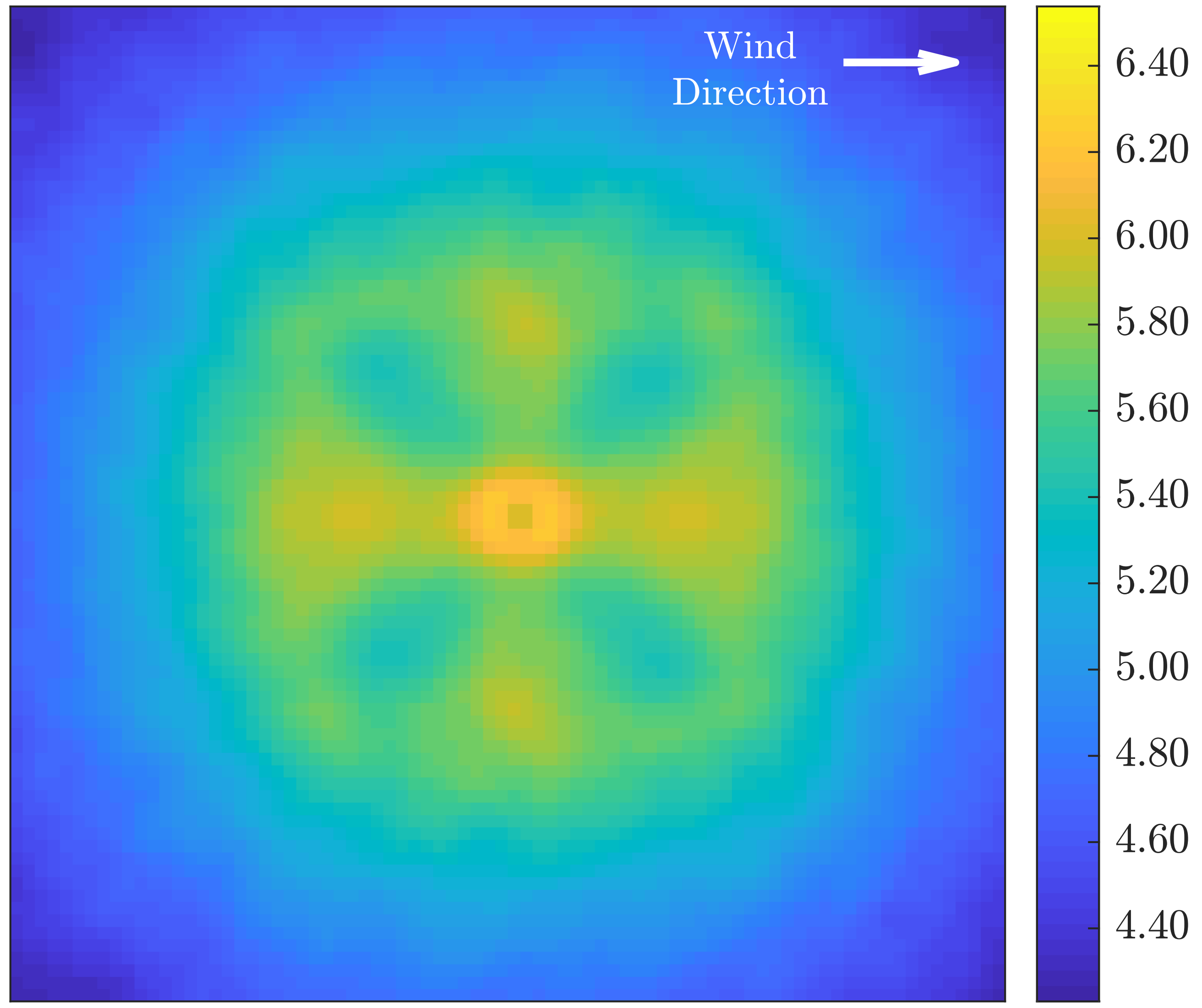


Model Generalization

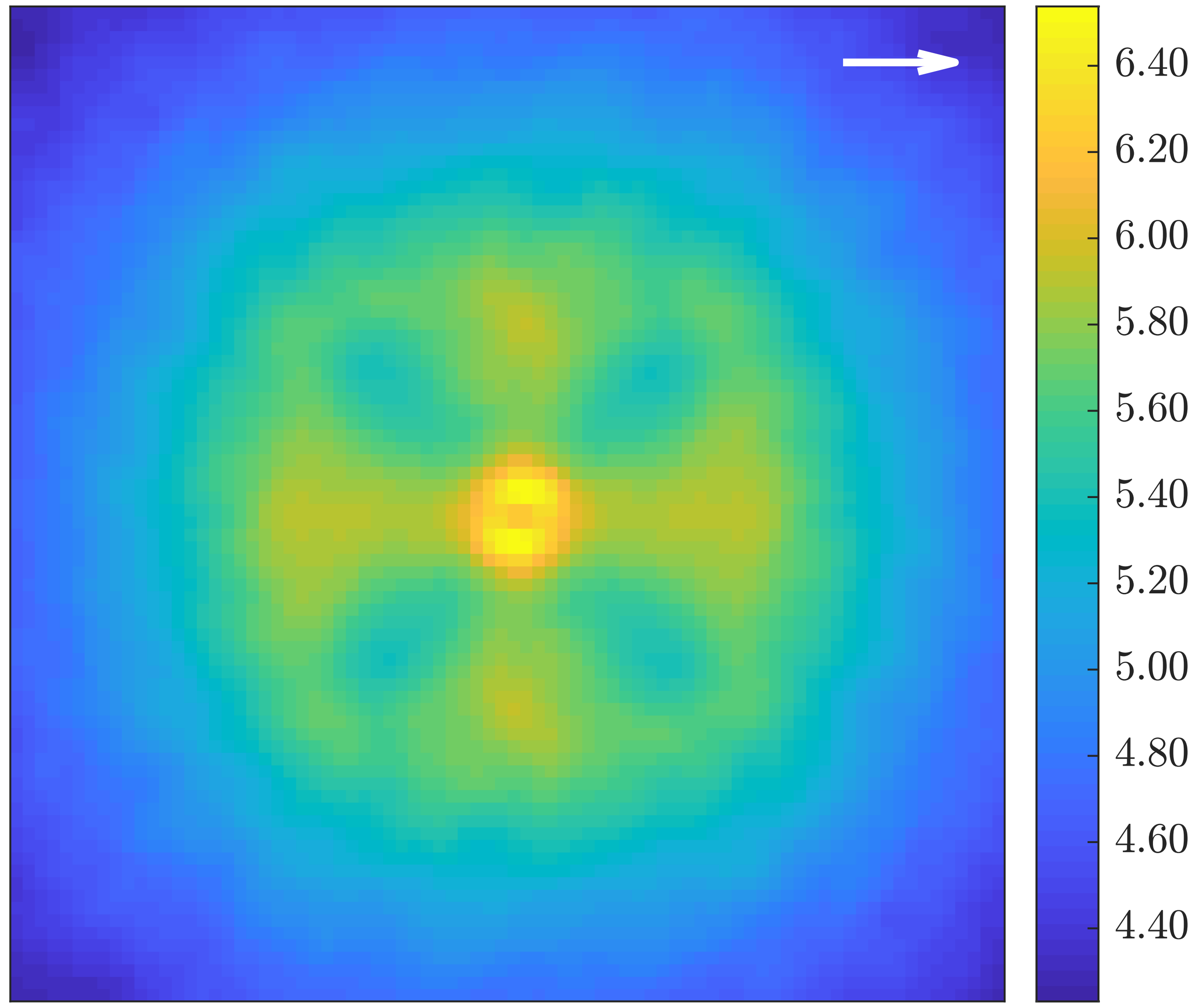


Residual Wavefront Power Spectral Density:

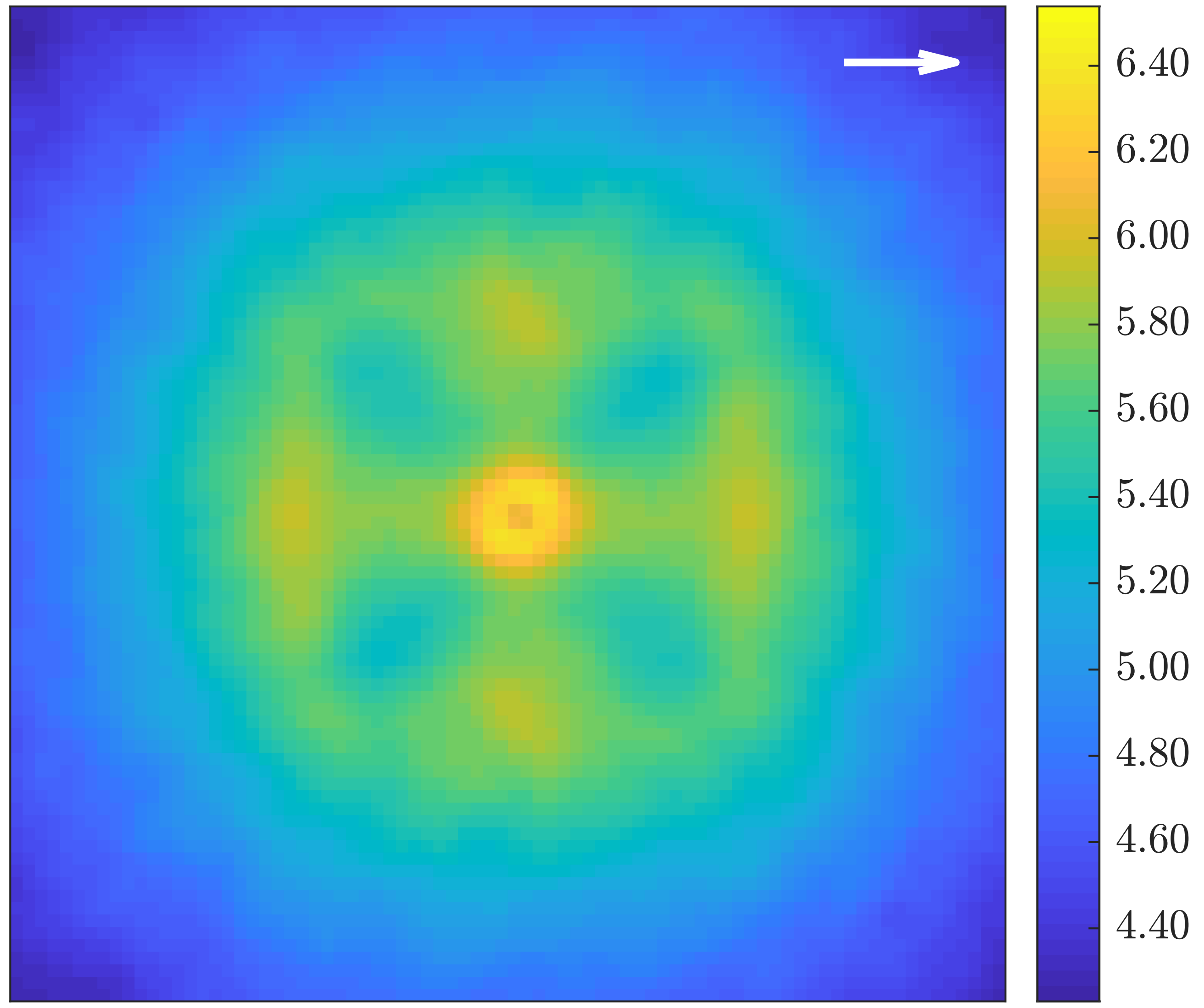
Magnitude 8



Classic Integrator
85.7% Strehl



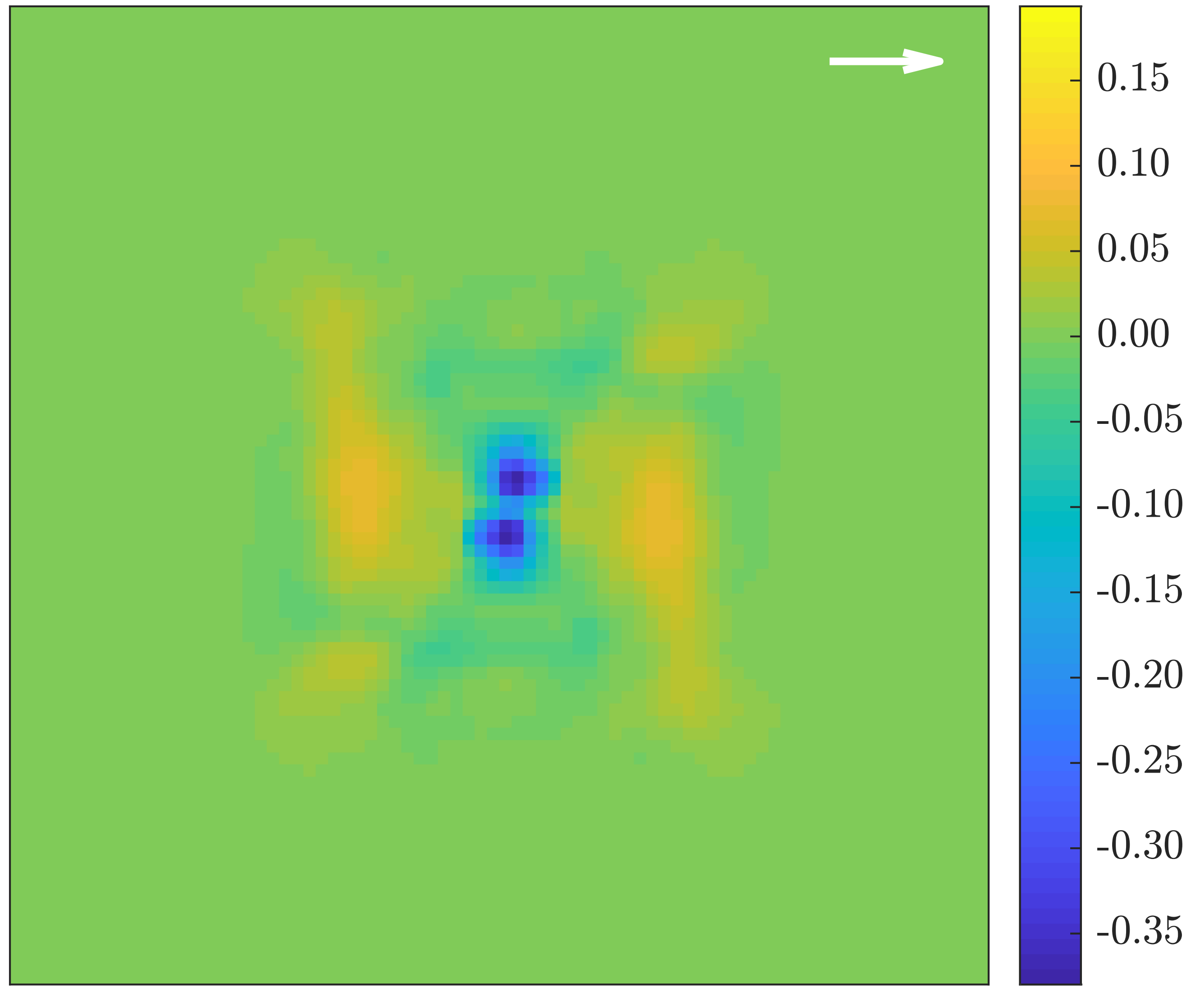
Dense Network
85.8% Strehl



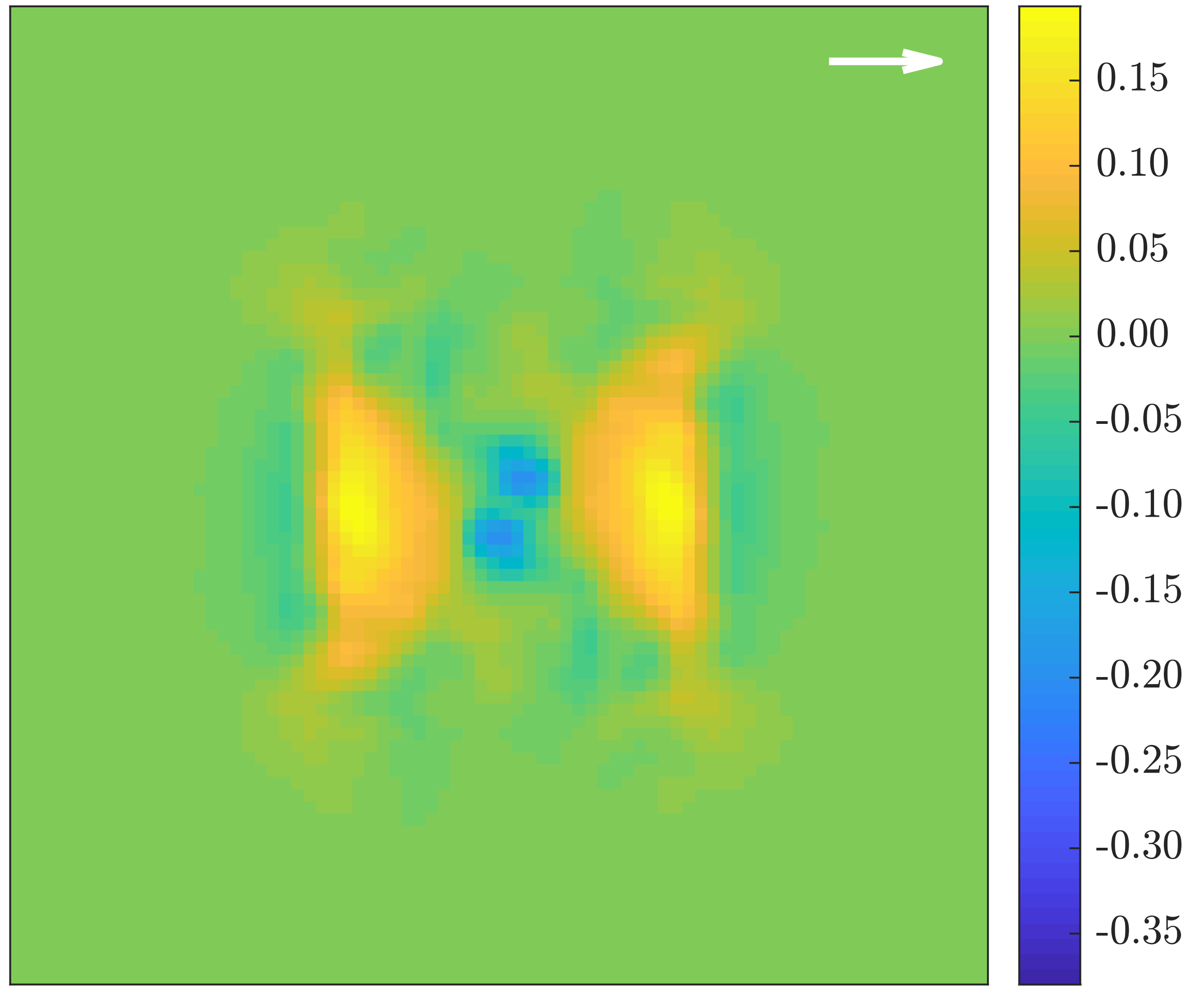
LSTM Network
86.1% Strehl

PSD Ratio Images:

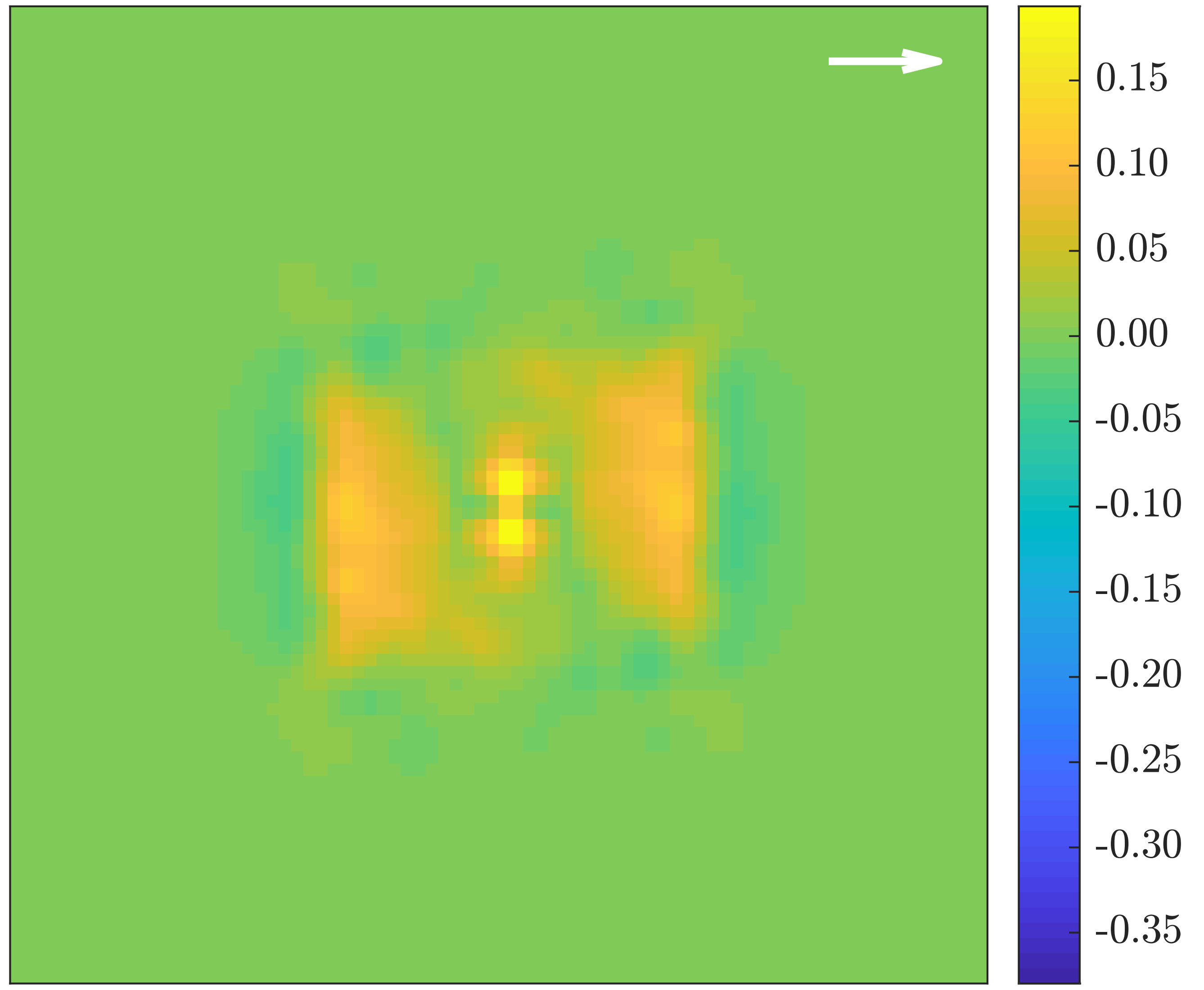
Magnitude 8



Classic / Dense
Ratio Image



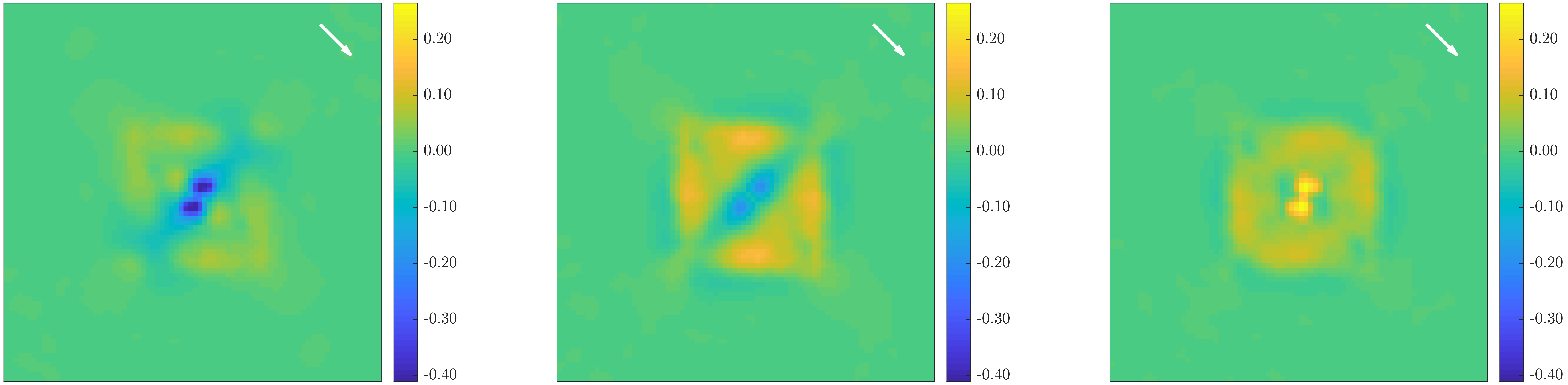
Classic / LSTM
Ratio Image



Dense / LSTM
Ratio Image

PSD Ratio Images:

Magnitude 8



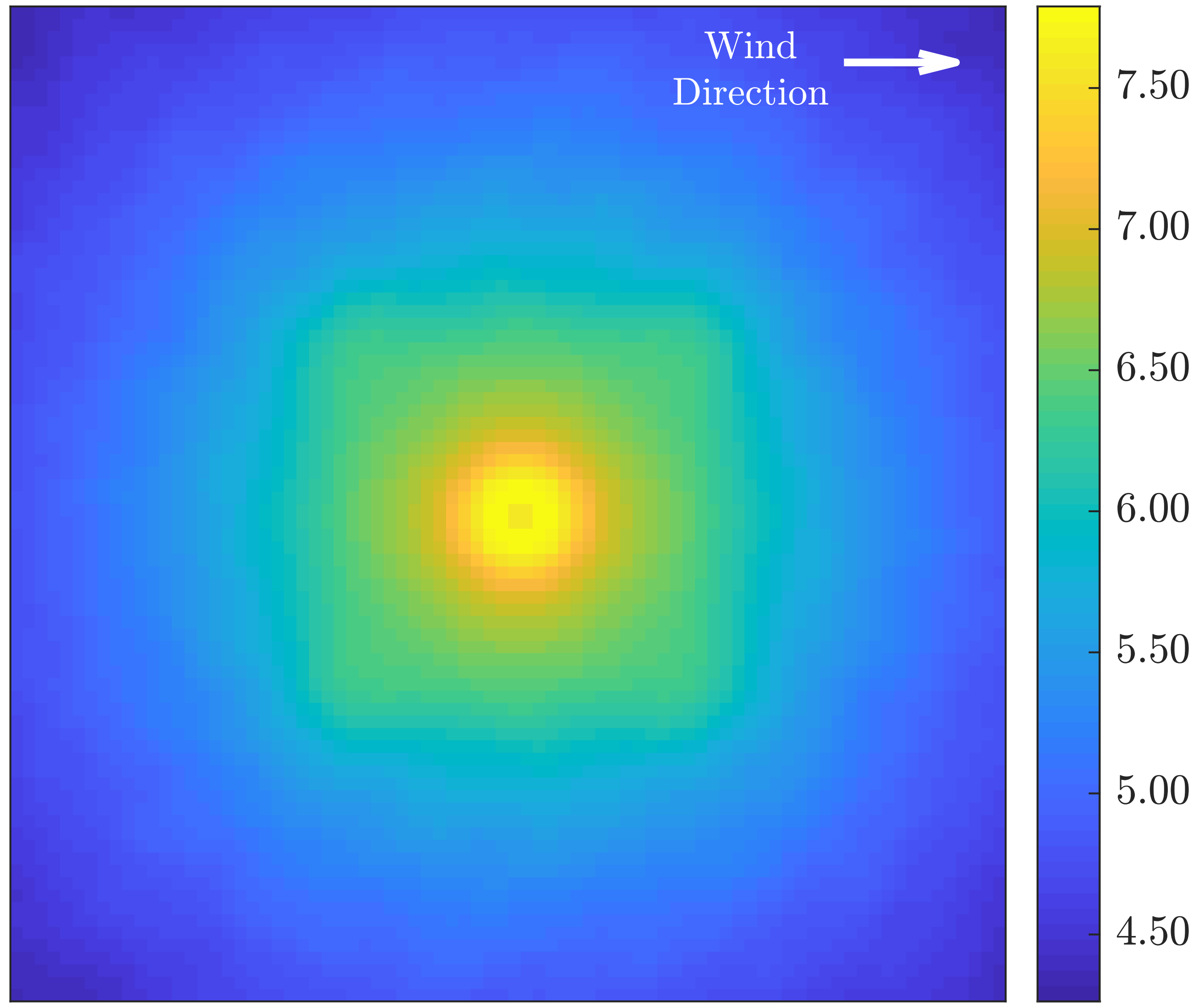
Classic / Dense
Ratio Image

Classic / LSTM
Ratio Image

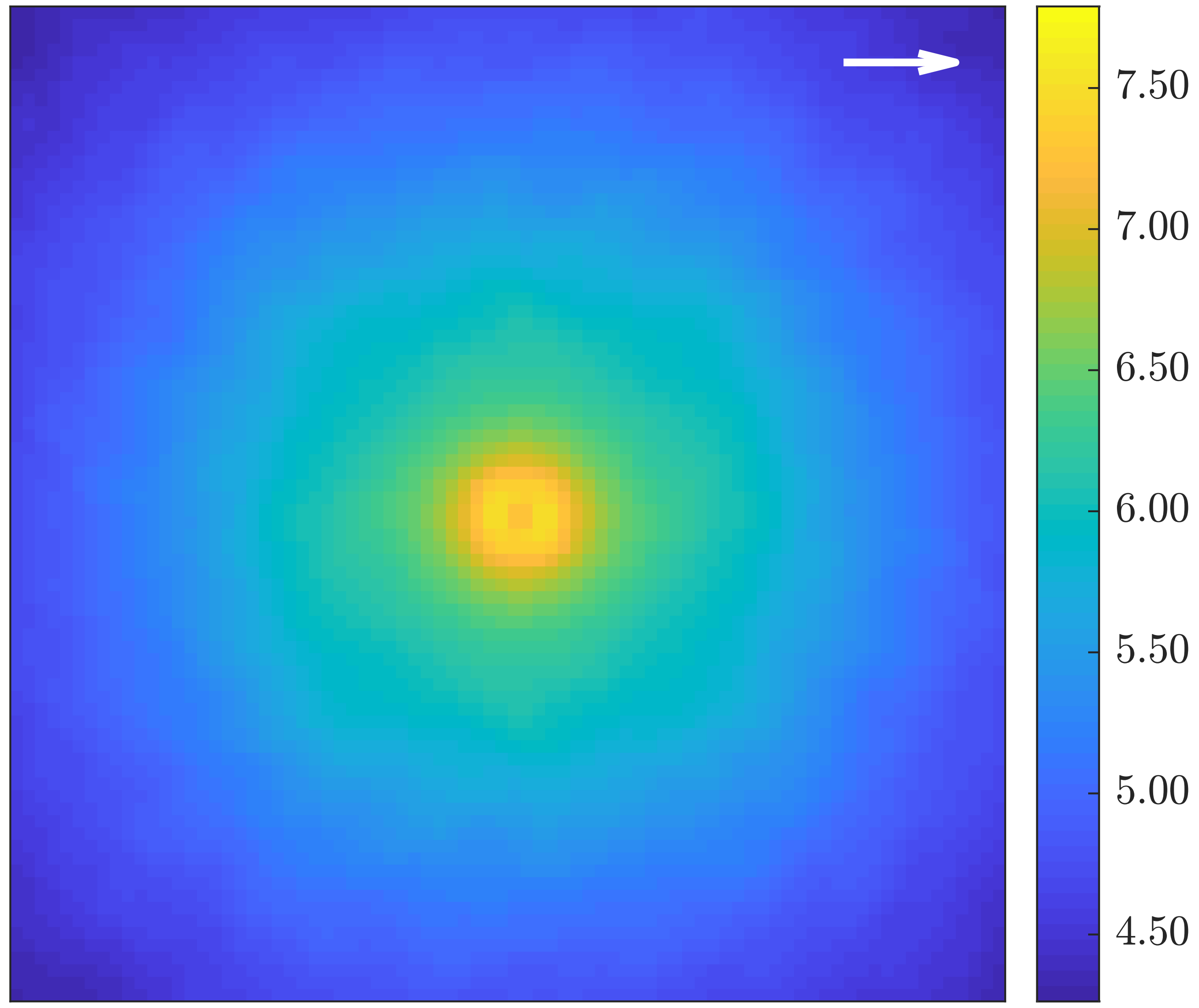
Dense / LSTM
Ratio Image

Residual Wavefront Power Spectrum Density:

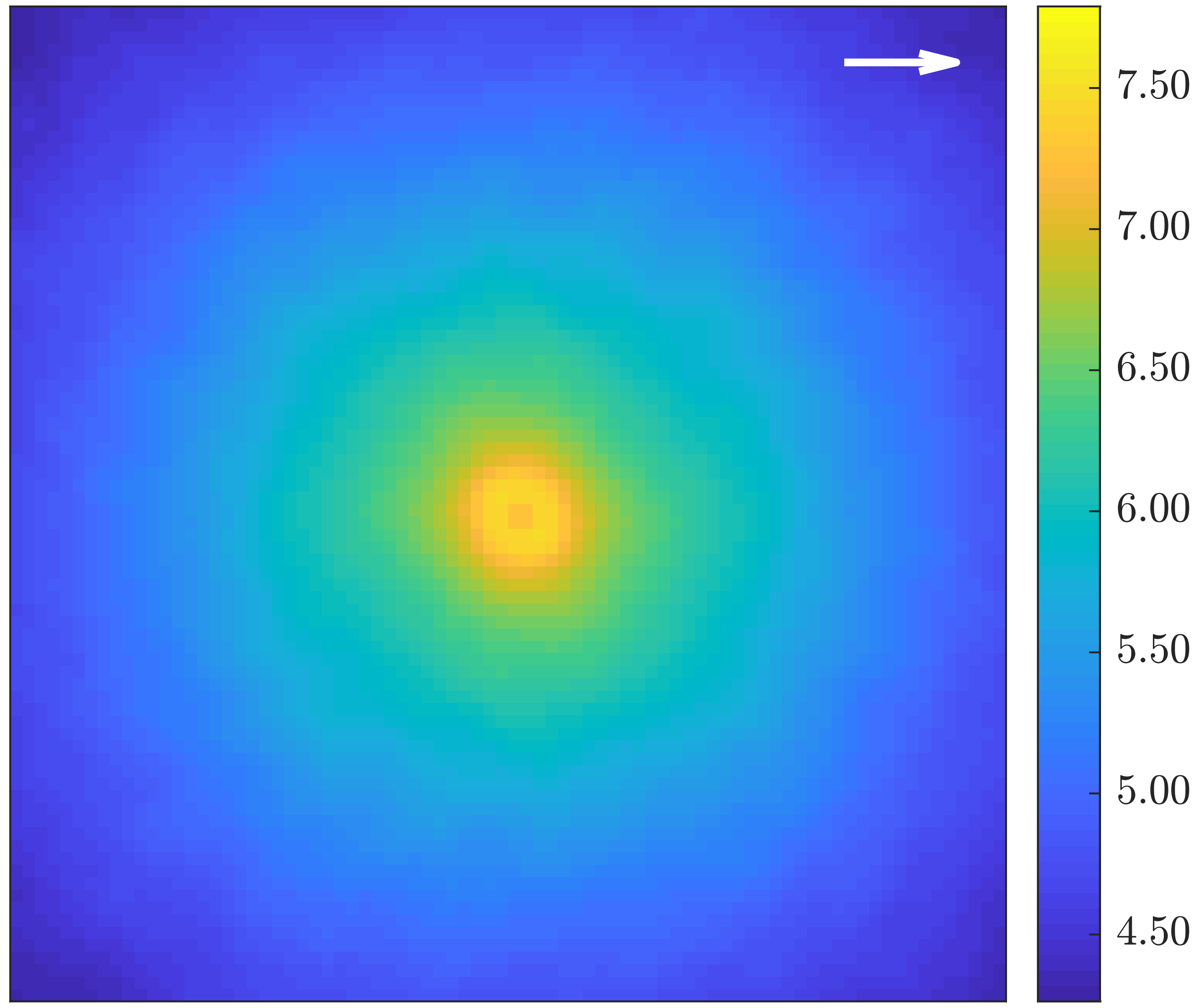
Magnitude 16



Classic Integrator
42.4% Strehl



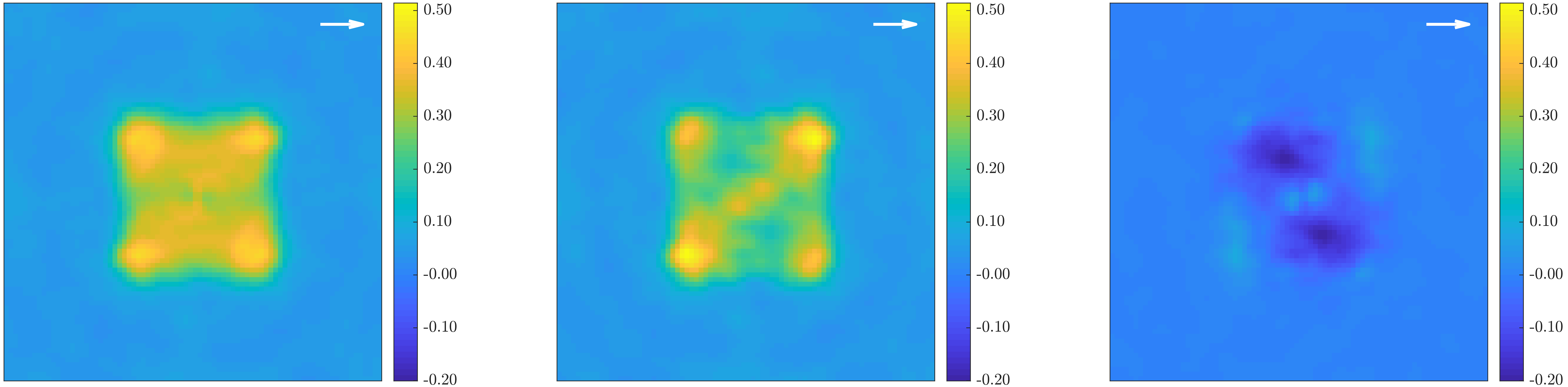
Dense Network
65.5% Strehl



LSTM Network
63.0% Strehl

PSD Ratio Images:

Magnitude 16



Classic / Dense Ratio Image | Classic / LSTM Ratio Image | Dense / LSTM Ratio Image

Conclusions

- Closed loop integrators can be trained robustly with a GAN prior
- Our CNN models outperform classical methods under all conditions
- LSTM models greatly improve servo-lag at low magnitudes
- Dense, feed-forward, networks perform better at removing noise at high magnitudes

Future Work & Directions

- **From Simulation to Hardware:**
  - Implement our models on an AO bench
  - Apply this knowledge to real systems (MMT, GIRMOS)
  - Optimization and real-time implementation
  - Explore GAN methods for creating realistic training data from experimental data
- **Improving our Models:**
  - Further investigate low-frequency PSD effects
  - Train and test for more specific use cases (i.e., high contrast scenario)
  - Integrate new CNN architecture techniques

Predictive AO Control with Convolutional Neural Networks

Robin Swanson, Masen Lamb, Carlos Correia, Suresh Sivanandam, Kyros Kutulakos
robin@cs.toronto.edu