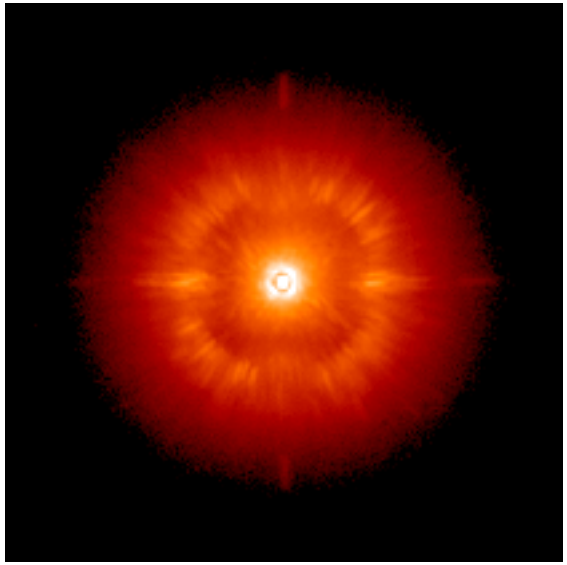
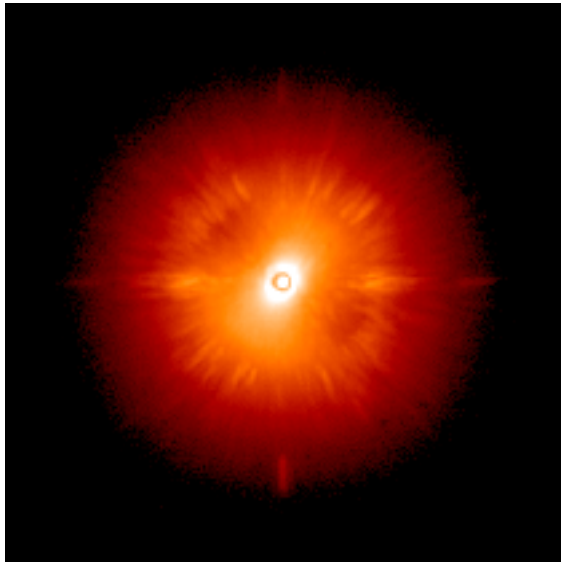
Machine Learning based Atmosphere Prediction for eXtreme Adaptive Optics

Markus Bonse, Markus Kasper, Patrick Schramowski, Kristian Kersting

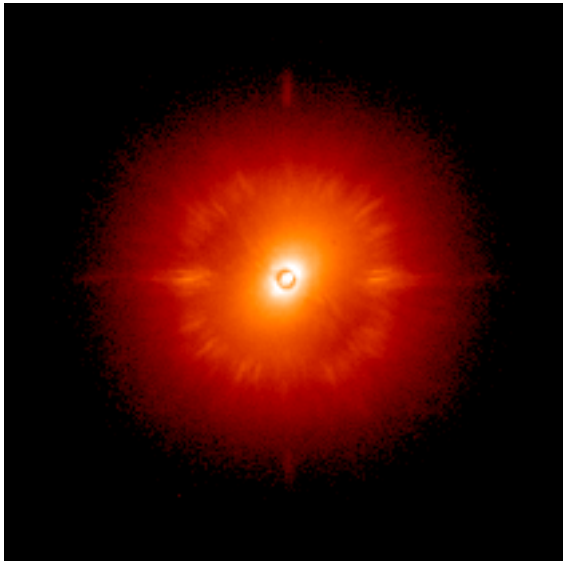
Temporal error on SPHERE



Small wind
V = 3 m/s

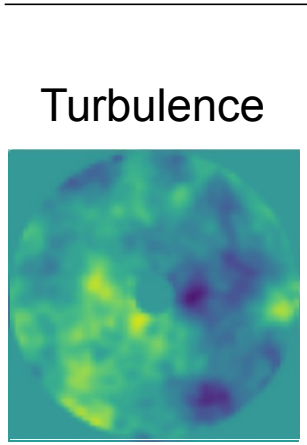


Medium Wind
V = 11 m/s



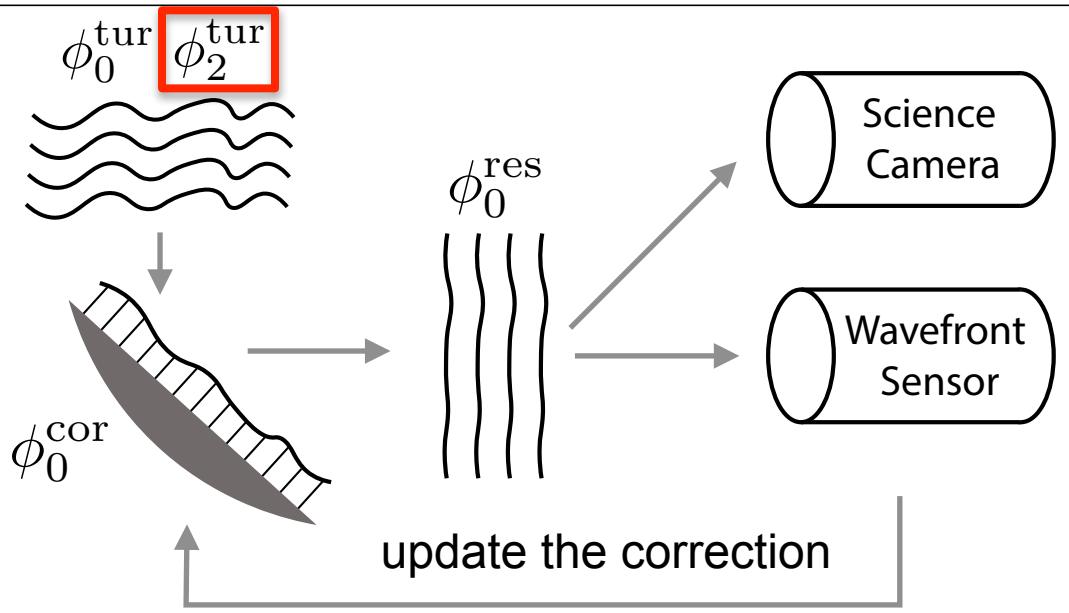
Strong wind
V = 22 m/s

Temporal Error



Turbulence

1 sec. -
1000 frames



Turbulence residual

1 sec. -
1000 frames

Problem: Measurements are lagging behind the turbulence evolution

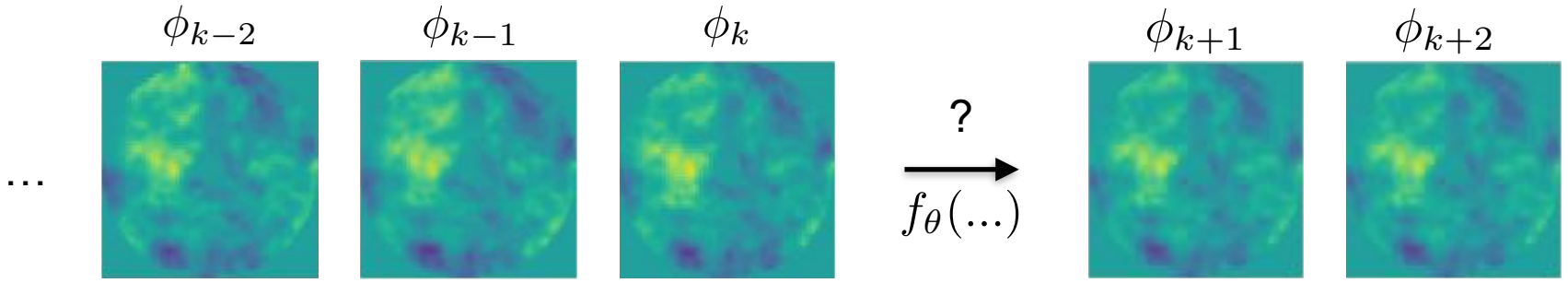
How to deal with the temporal error?

(1) Increase loop speed

(2) Prediction

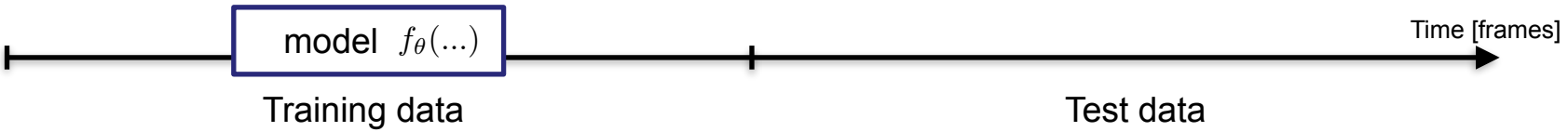
Open-loop prediction task

(2) Prediction



Goal: Find a model $f_\theta(...)$ which predicts the future $\hat{\phi}_{k+d}$

$$f_\theta\left(\phi_k, \phi_{k-1}, \phi_{k-2}, ...\right) = \hat{\phi}_{k+d}$$



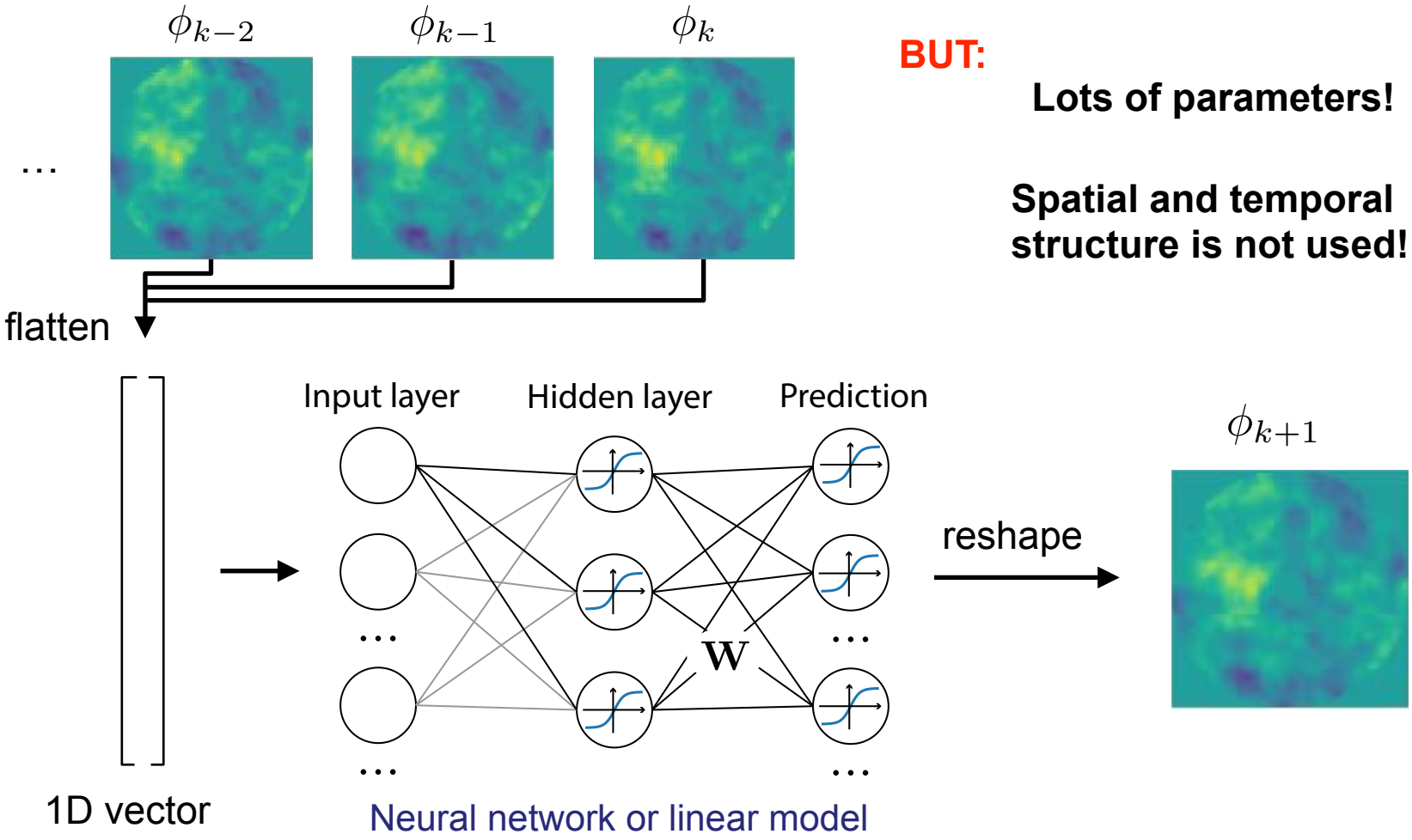
(1) Train the parameters θ of $f_\theta(...)$ on the **training data**.

(2) Use the trained $f_\theta(...)$ on the **test data**

Key Challenge: Generalisation - *good performance* on the **test data**

Classical Neural Networks and Linear Models

(2) Prediction



BUT:

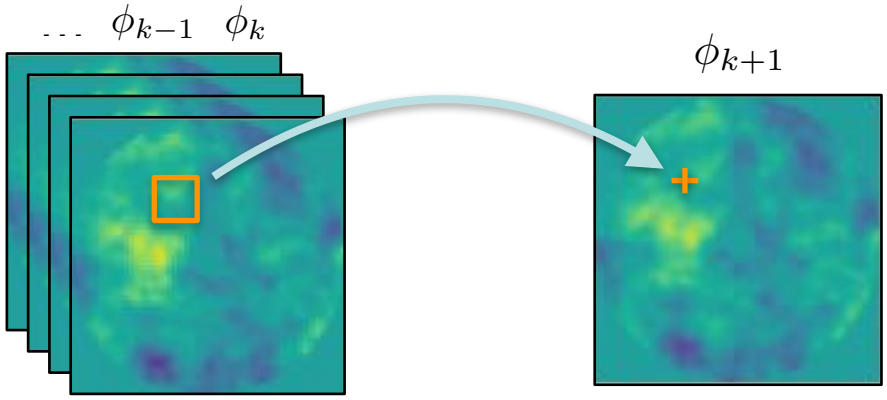
Lots of parameters!

Spatial and temporal structure is not used!

Encoding the nature of the turbulence

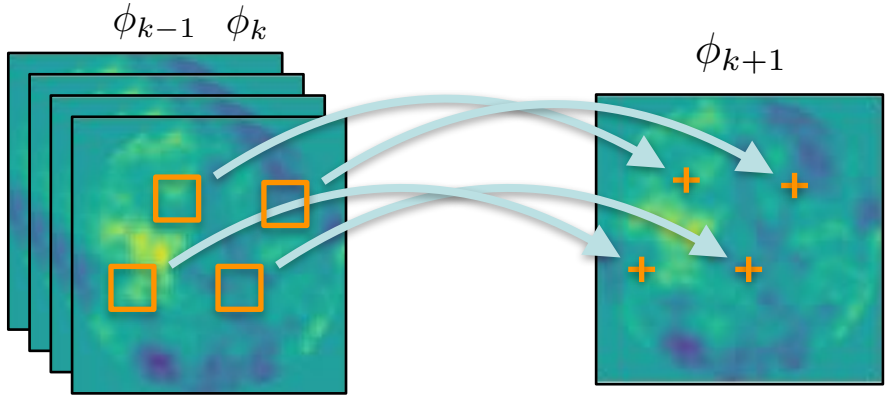
(2) Prediction

1.) Locality



Frozen-flow:
We do not expect the turbulence to jump

2.) Spatial invariance



Homogeneous:
The temporal evolution of the turbulence should be similar for all positions

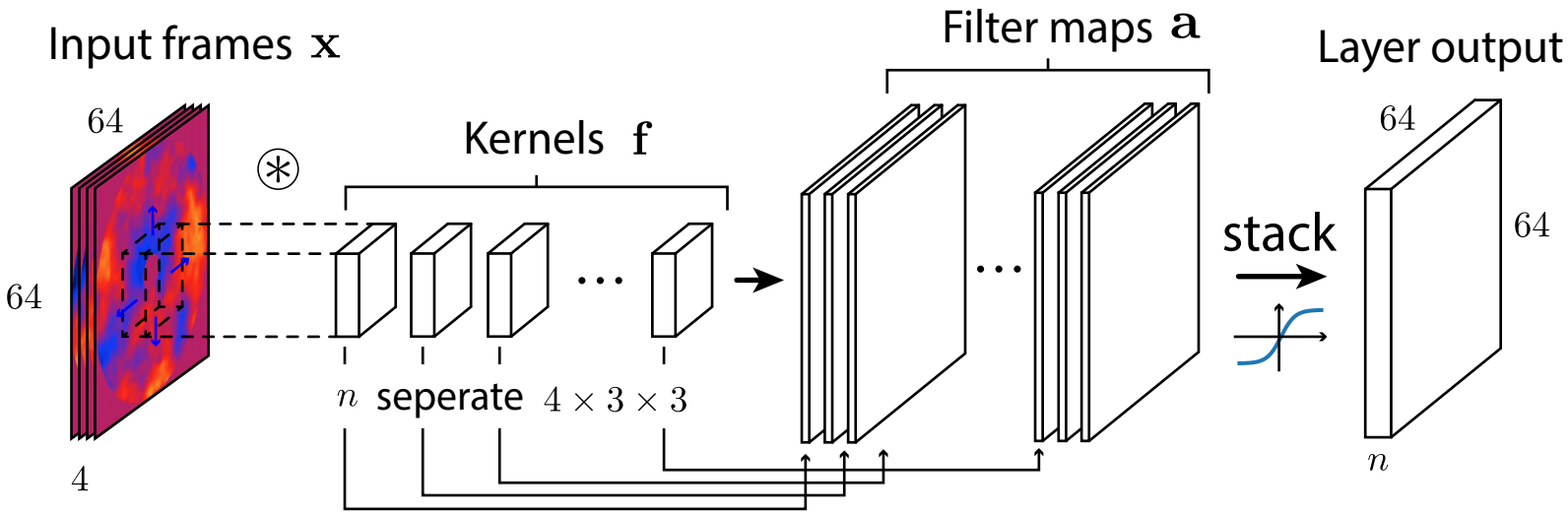
one local predictor for all positions

Convolutional Neural Networks

1.) Locality

2.) Spatial invariance

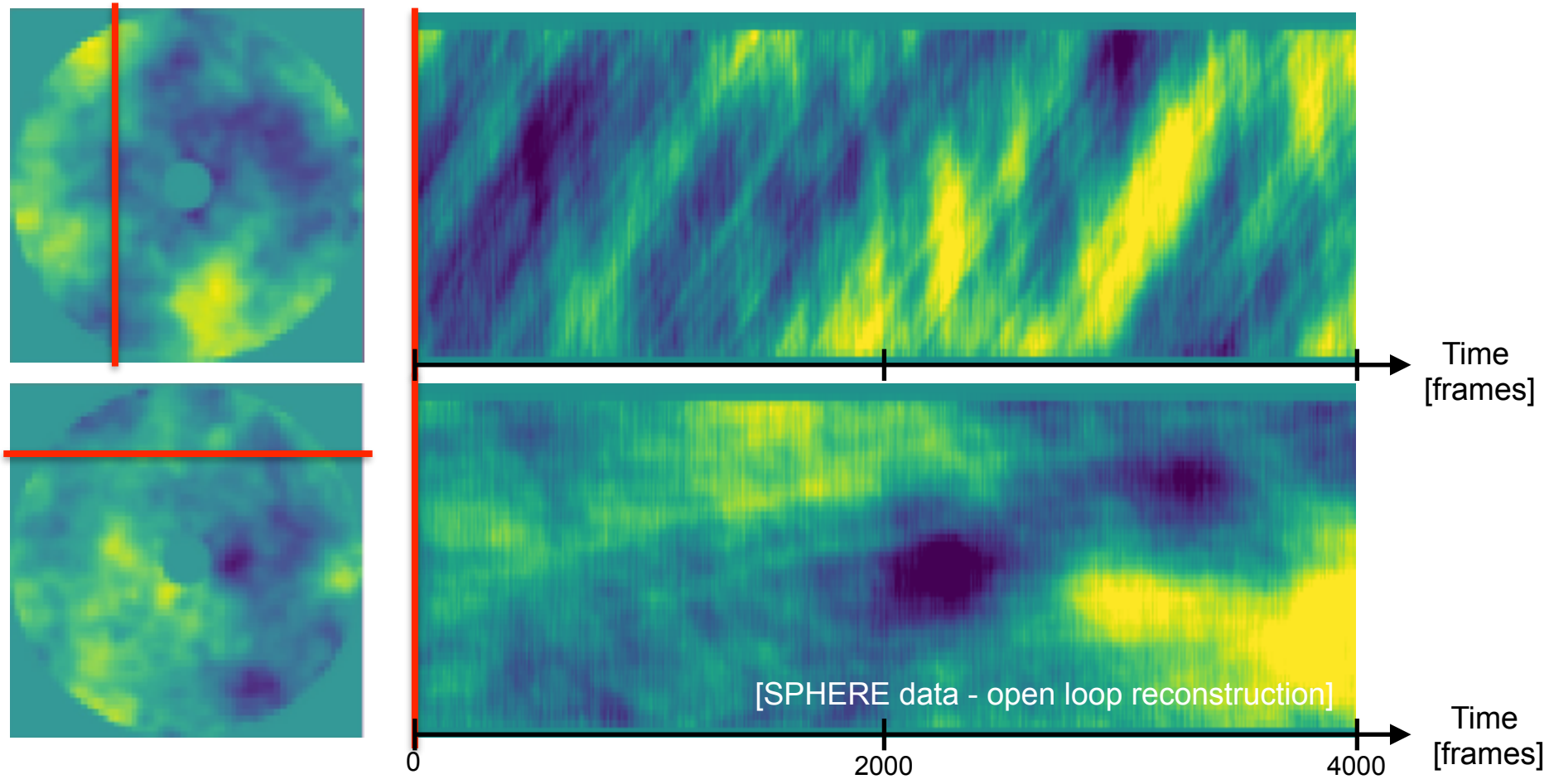
Convolutional Layer



Fully convolutional CNNs -
no dense layers!

How many past frames do we need?

Different conditions might require more or less temporal information…

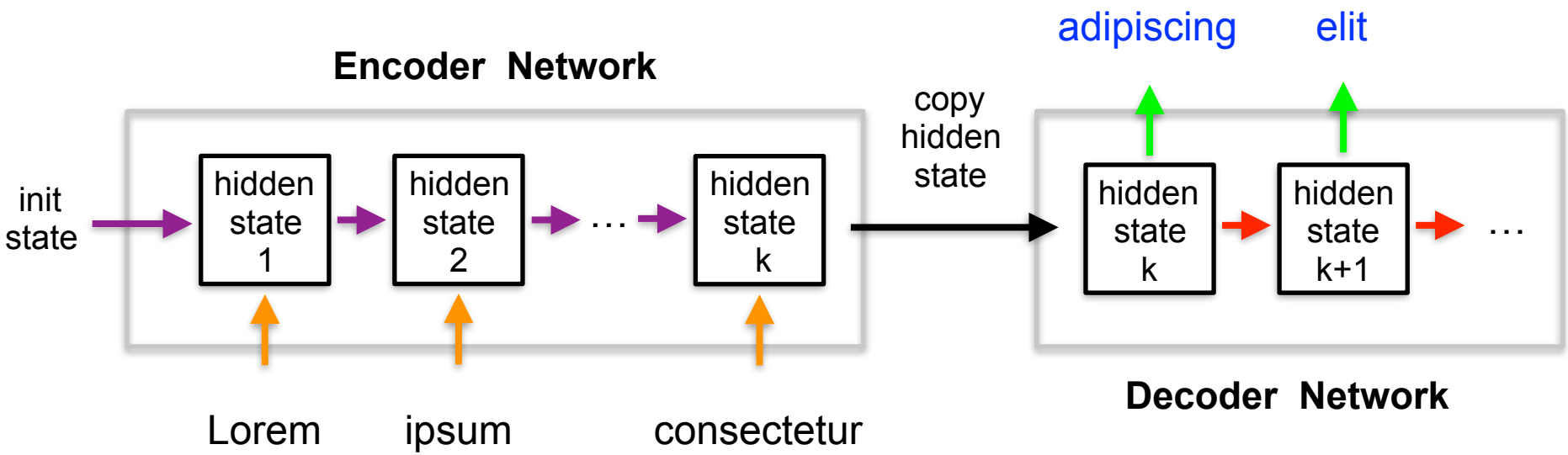


Convolutional Long Short Term Memory Networks

Combine **CNNs** with **Long Short Term Memory Networks**

Example with text:

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

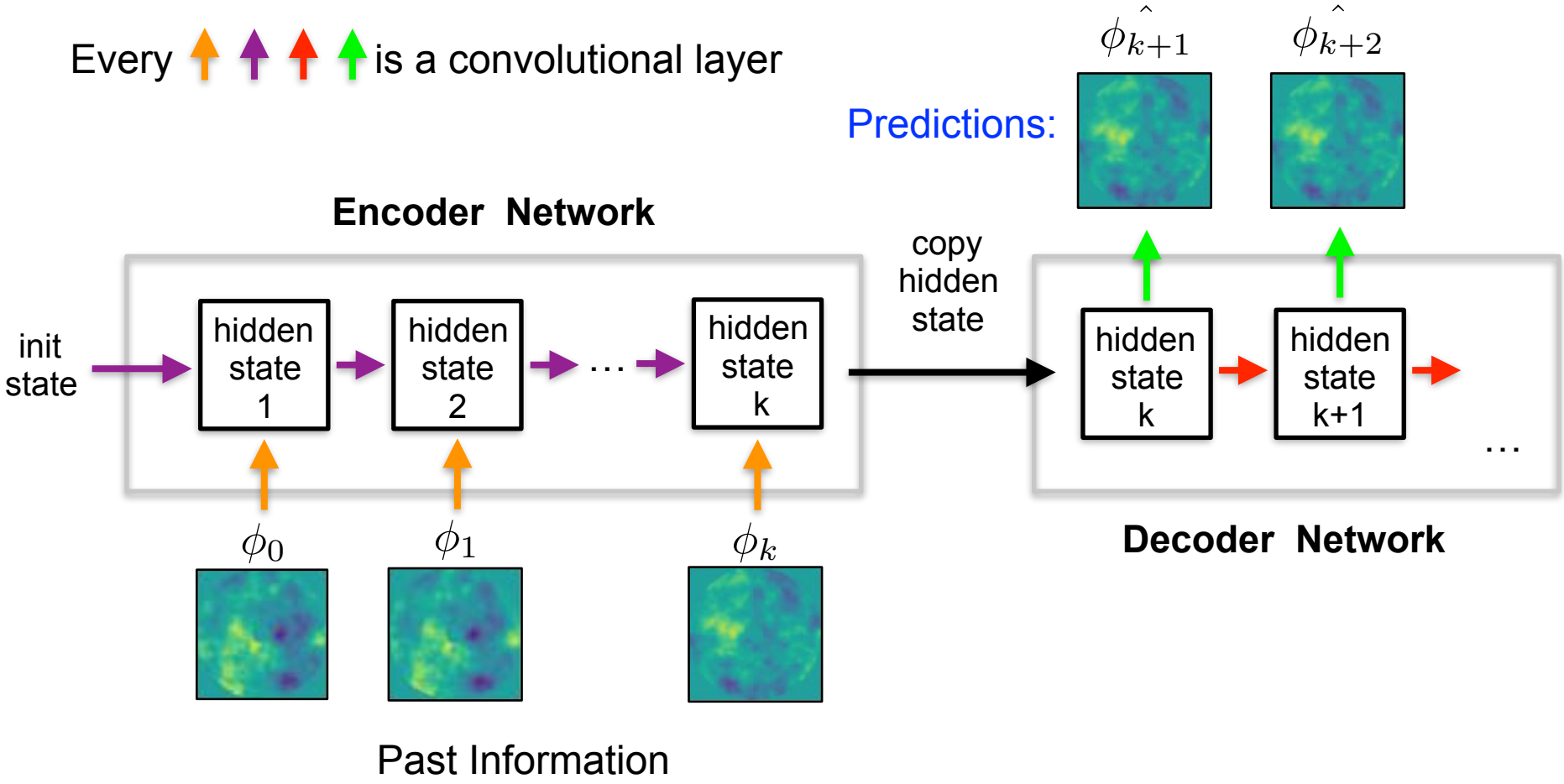


Learn how much past information is needed

[Shi Xingjian et al 2015]

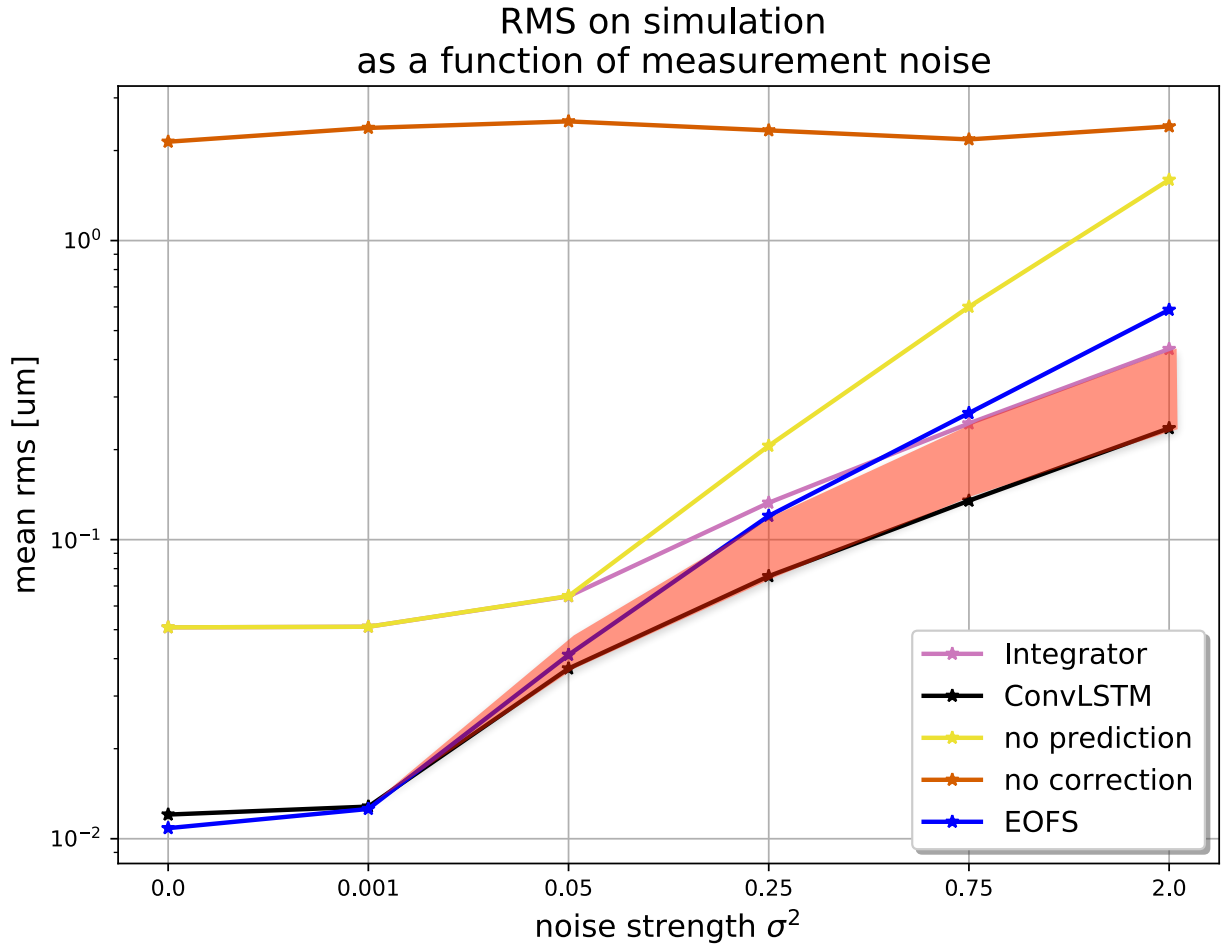
Convolutional Long Short Term Memory Networks

Combine **CNNs** with **Long Short Term Memory Networks**



[Shi Xingjian et al 2015]

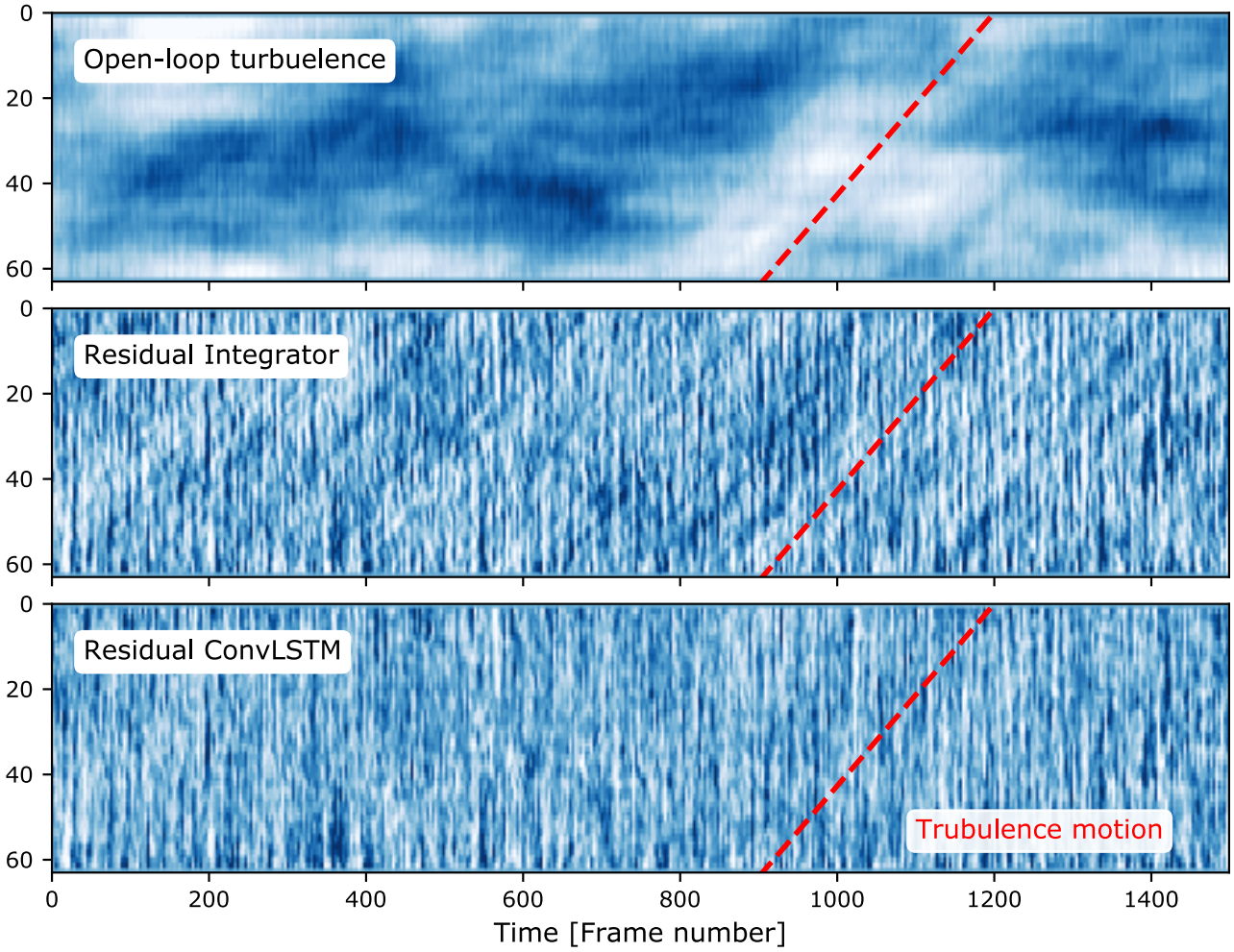
Noise Robustness on Simulations



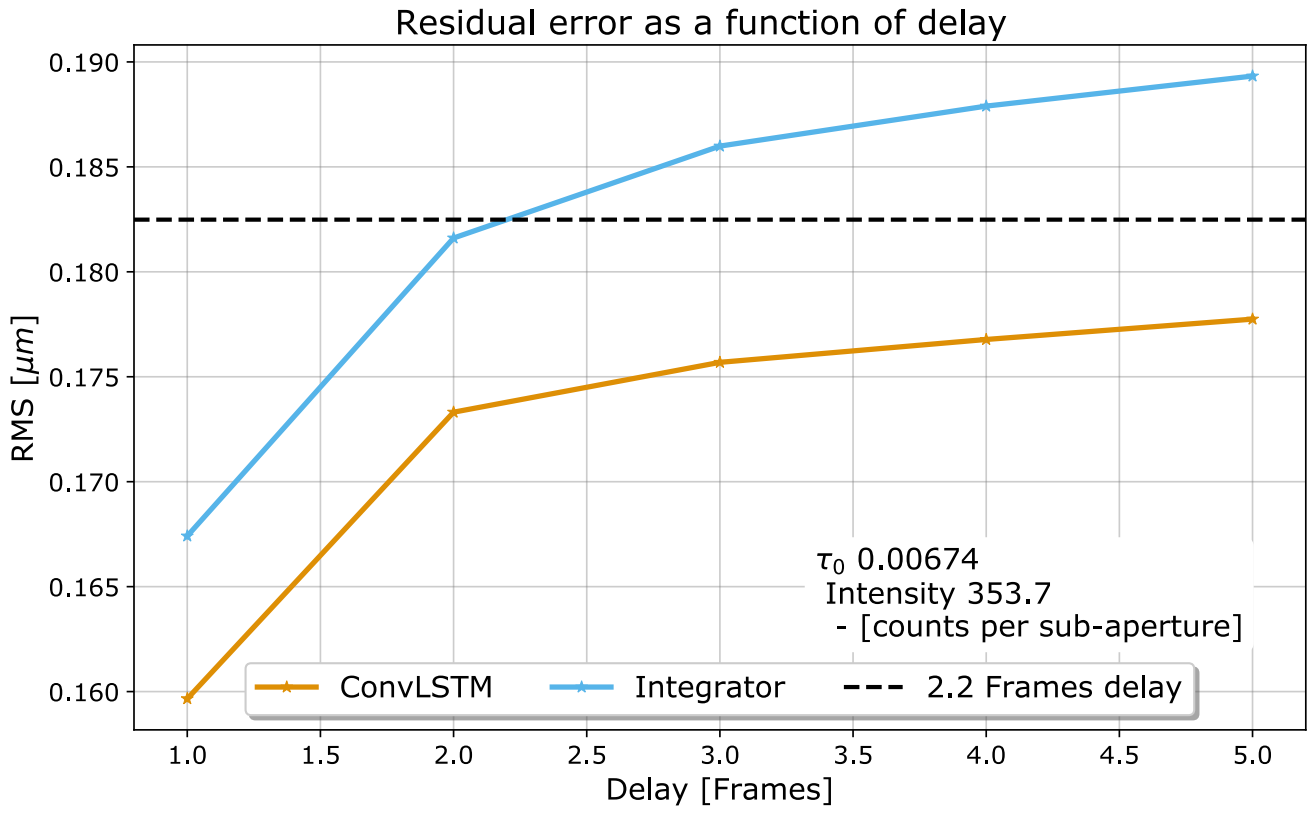
ConvLSTM improve robustness!

EOFS -
Linear Predictive Filter
[O. Guyon 2017]

Prediction on SPHERE data

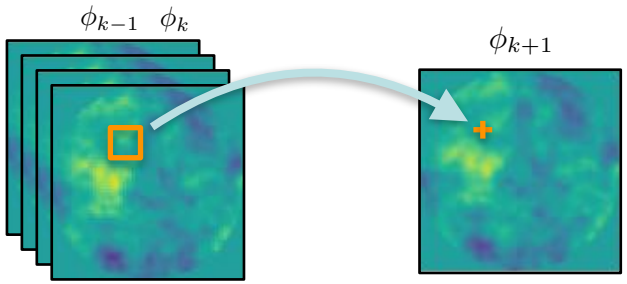


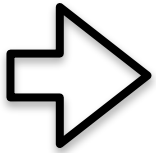
Prediction on SPHERE data

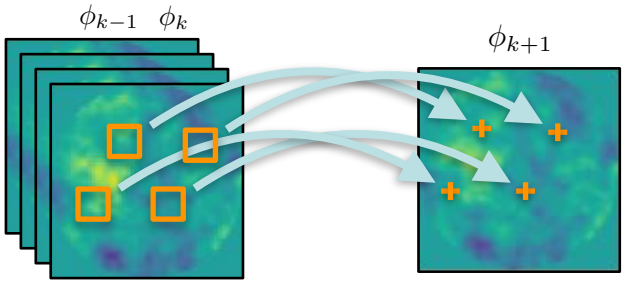


What you should remember…

Think about possible assumptions







What is special about my data?

Search for a *good* architecture

LSTM / RNN / GRU …?

GAN?

PixelCNN?

CNN?

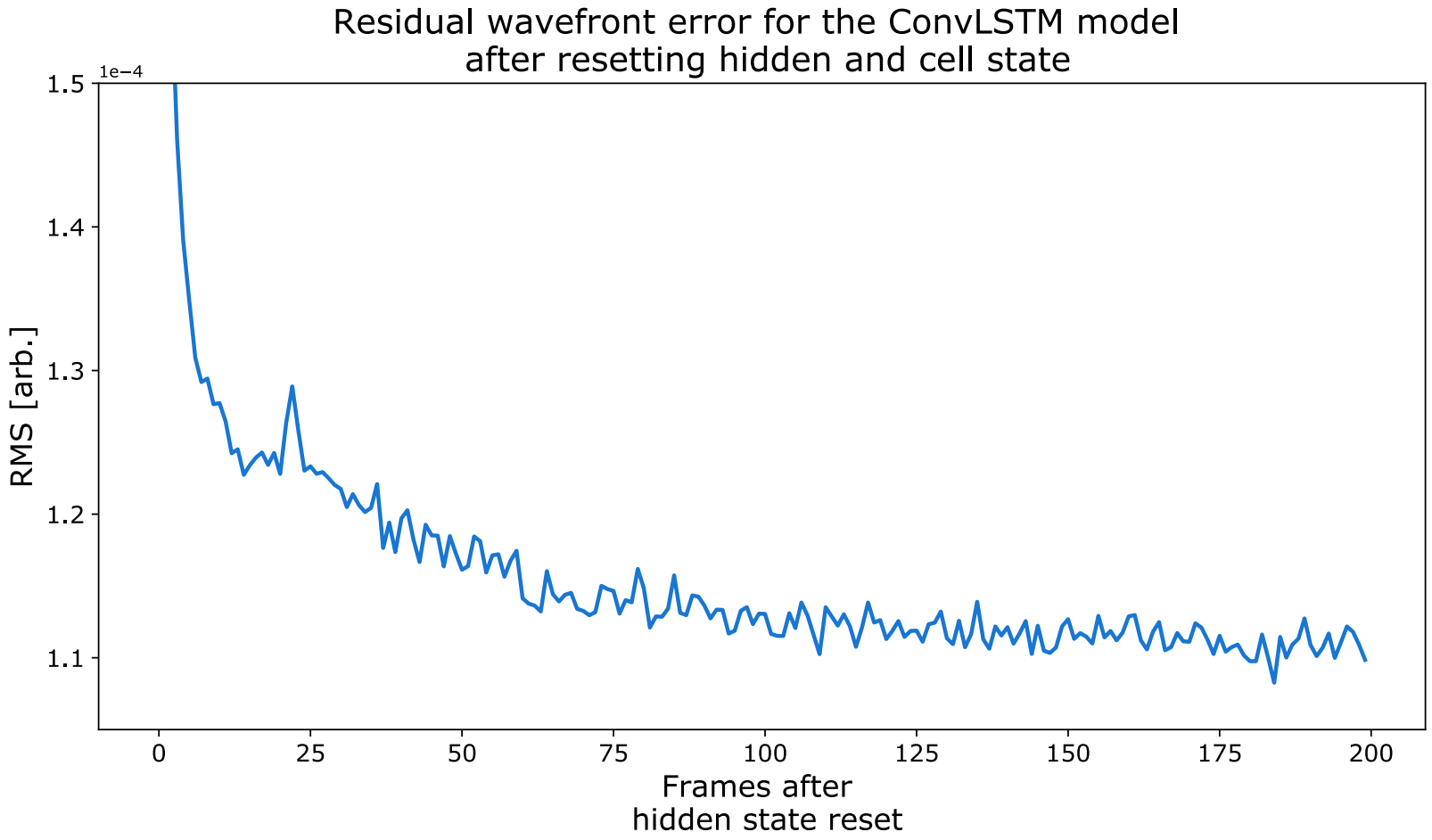
MLP?

Auto Encoder?

VAE?

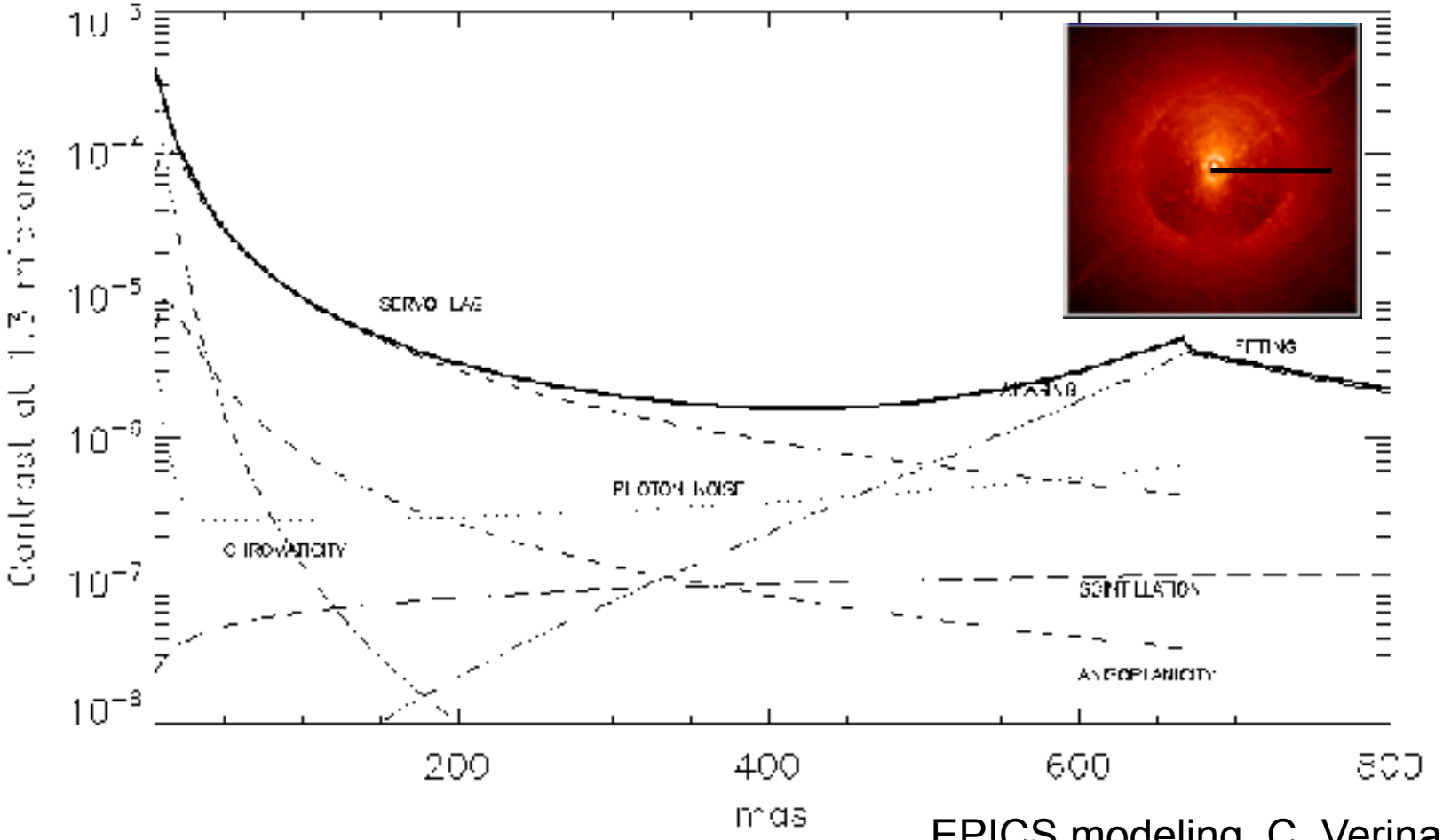
…?

One more thing - How important is time?



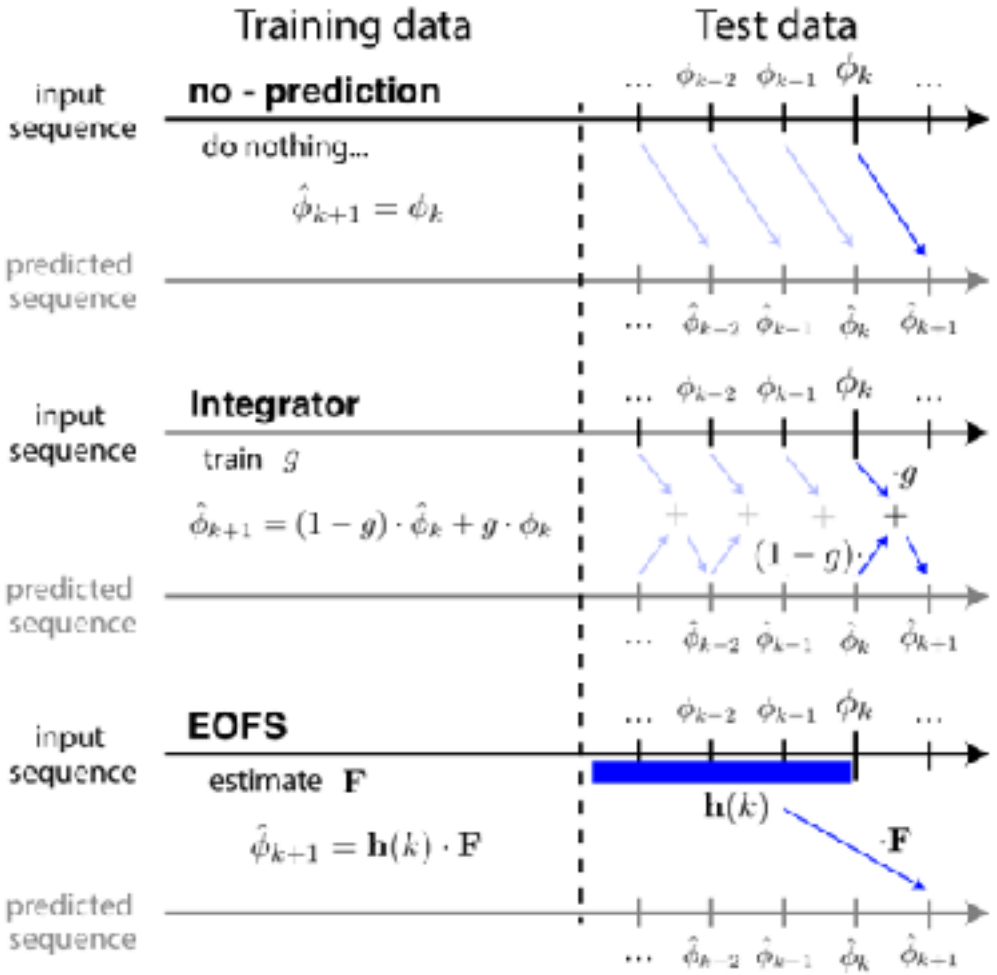
Appendix

eXtreme Adaptive Optics error budget

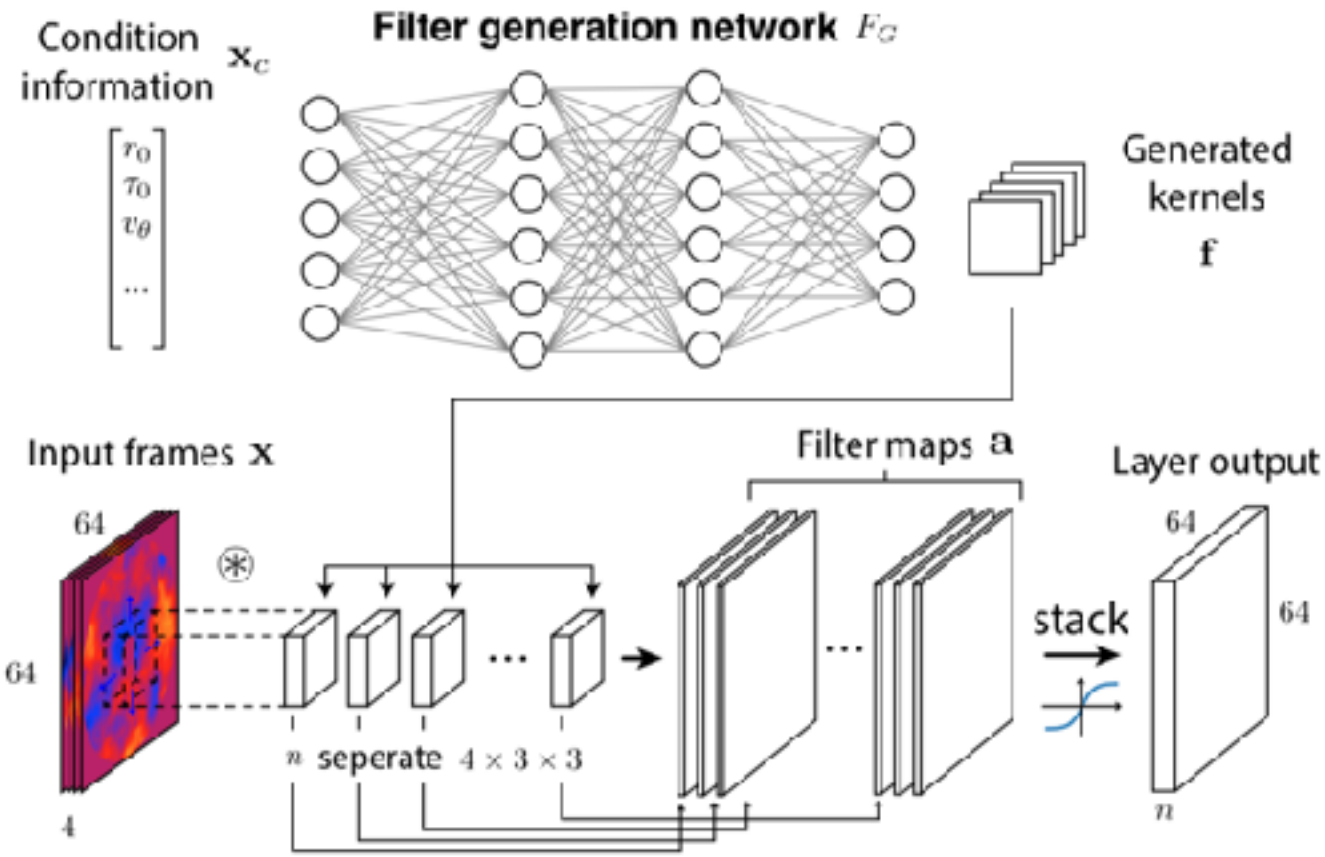


EPICS modeling, C. Verinaud

Baselines


Training data
Test data
input sequence
no - prediction
do nothing...
$\hat{\phi}_{k+1} = \phi_k$
predicted sequence
input sequence
Integrator
train g
$\hat{\phi}_{k+1} = (1-g)\cdot\hat{\phi}_k + g\cdot\phi_k$
$\cdot g$
$(1-g)\cdot$
predicted sequence
input sequence
EOFS
estimate $\mathbf{F}$
$\hat{\phi}_{k-1} = \mathbf{h}(k)\cdot\mathbf{F}$
$\mathbf{h}(k)$
$\cdot\mathbf{F}$
predicted sequence

Dynamic Filter Networks



ConvLSTM Layer