Science use case 1: Deep Learning in astrophysics

Elise Jennings

ALCF Data Science group

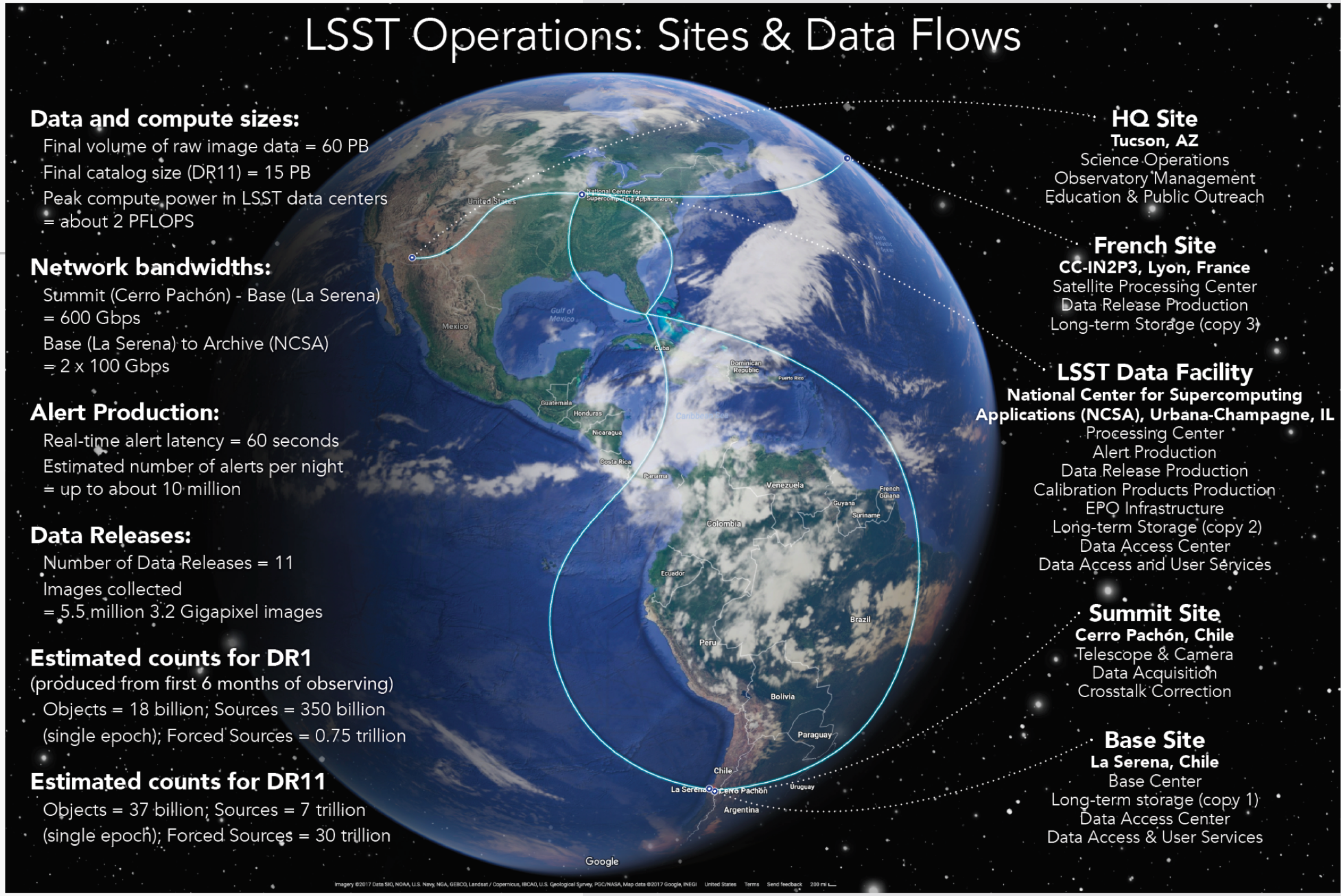
ejennings@anl.gov

Argonne NATIONAL LABORATORY

Deep Learning in Cosmology/Astronomy

Big data challenges:

- Volume of data (Gigabytes,Terabytes, Petabytes, Exabytes)
- Complexity of data
- Rate of data (massive data streams, real time knowledge extraction)
- Big scientific data visualization
- Hardware and software co-design

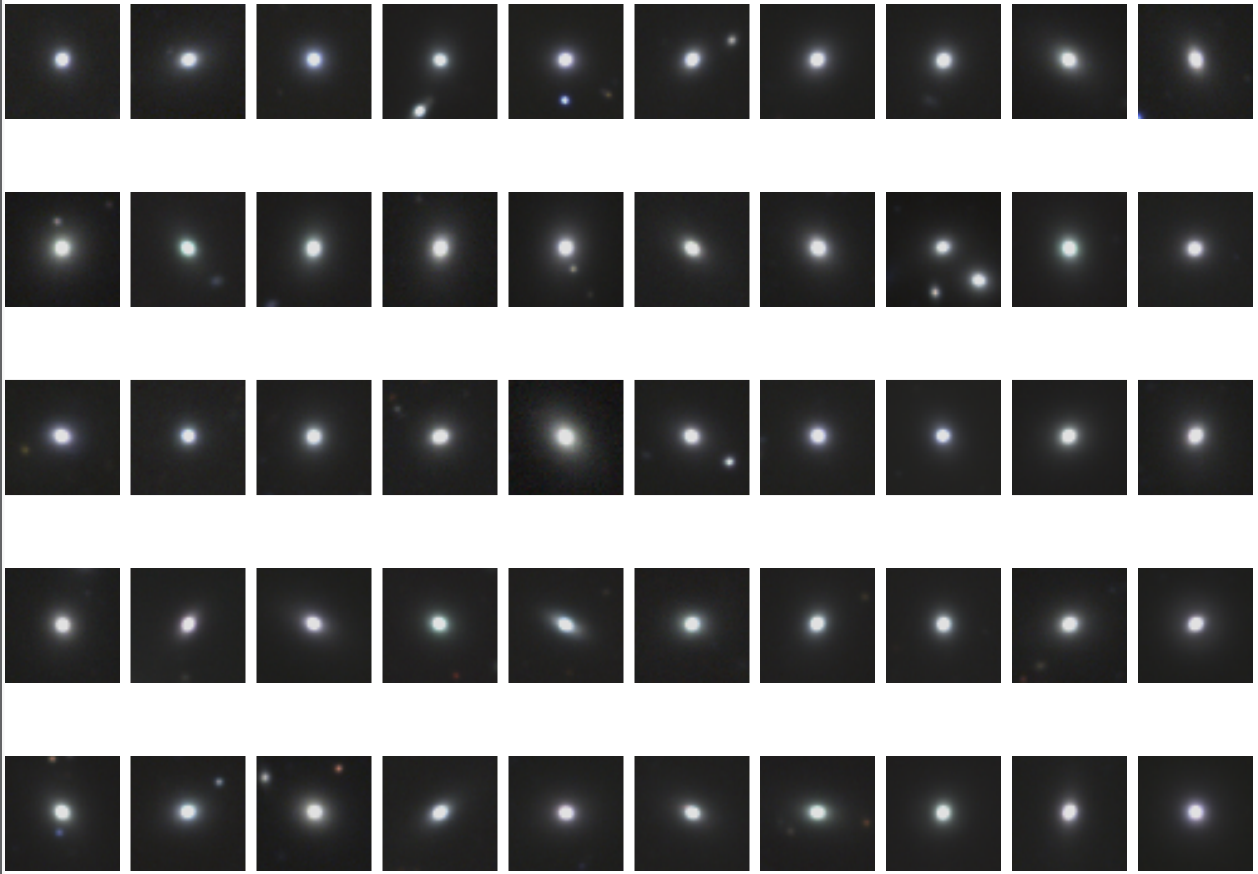


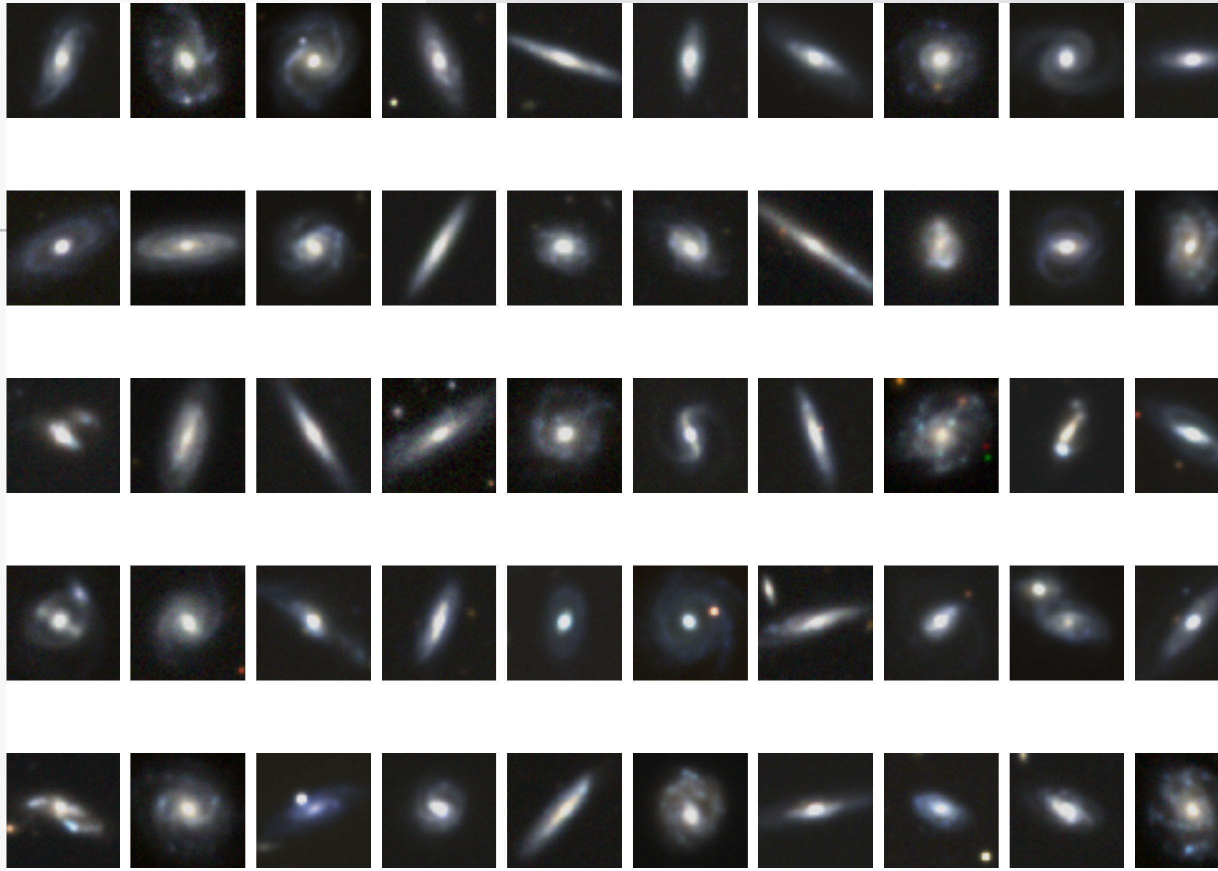
Deep Learning in Cosmology/Astronomy

Applications:

- Mock catalogue creation
- Augment N-body simulations
- Detect gravitational lenses
- Detect transients in real time
- Detect gravitational waves in real time
- Classify galaxy images
- Parameter estimation from time series data

Deep Transfer Learning to classify galaxy images





Deep Transfer Learning to classify galaxy images

Network:

Xception + a few custom defined fully connected layers

Weights:

pretrained weights with the ImageNet dataset

Data:

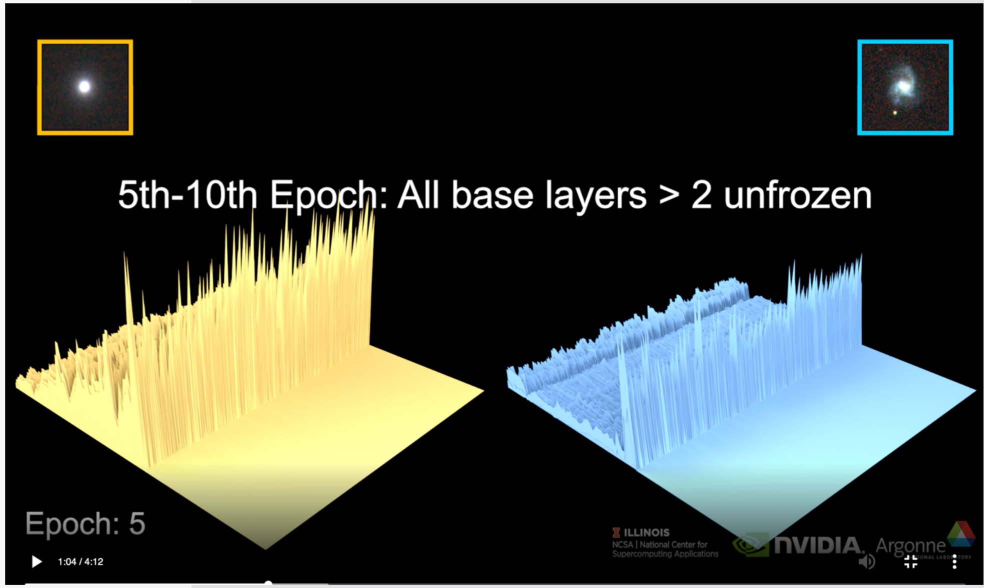
resized all the galaxy images 299 × 299 pixels

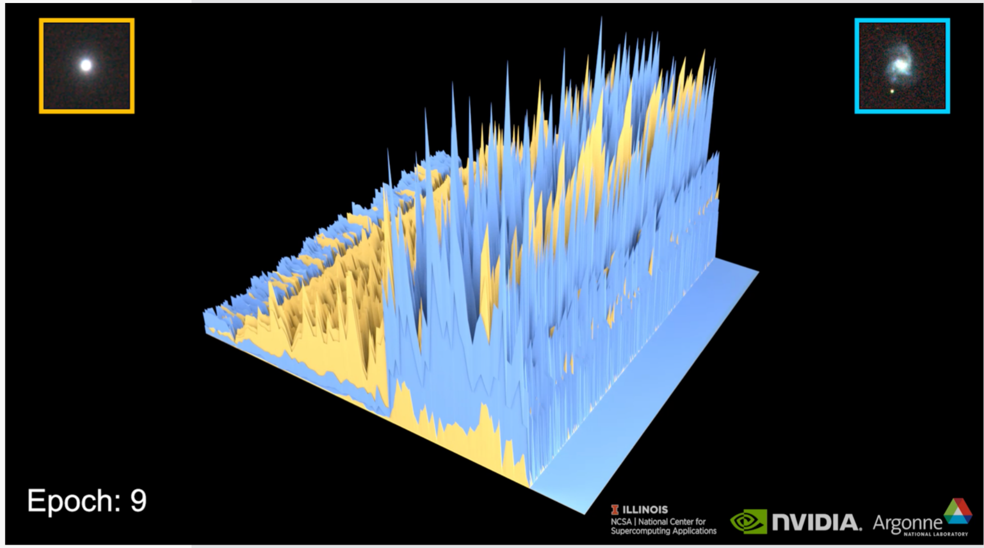
Method:

Progressively unfreeze earlier layers of the whole network

Fine tune their weights for a few epochs of training

Retain earlier layers of a trained network: versatile filters for features like lines and edges





Argonne NATIONAL LABORATORY

Using the network as a feature extractor

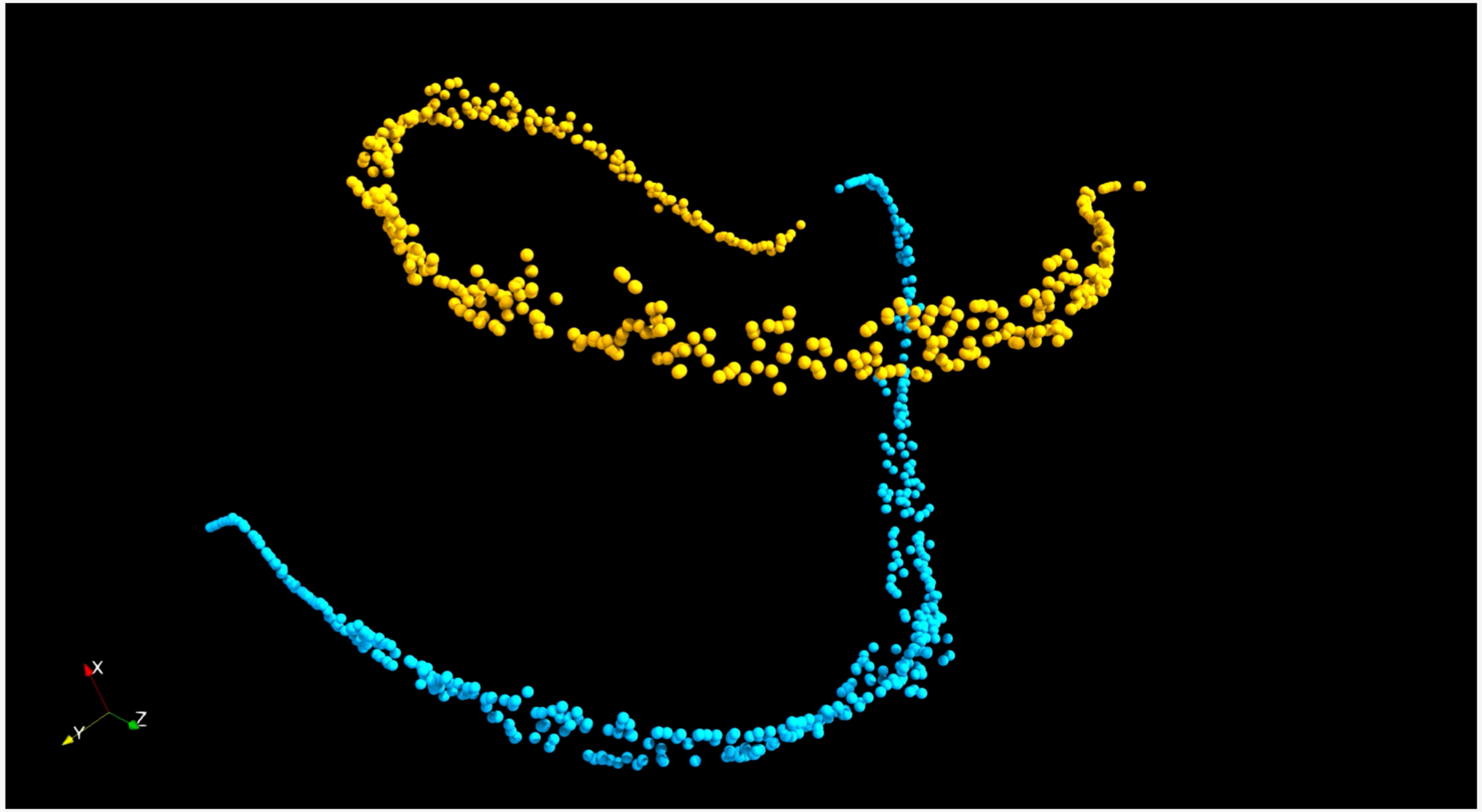
Network output:

output activation values from second-to-last layer

3D representation

t-Distributed Stochastic Neighbor Embedding (t-SNE)

Addresses common problem of large unlabeled datasets



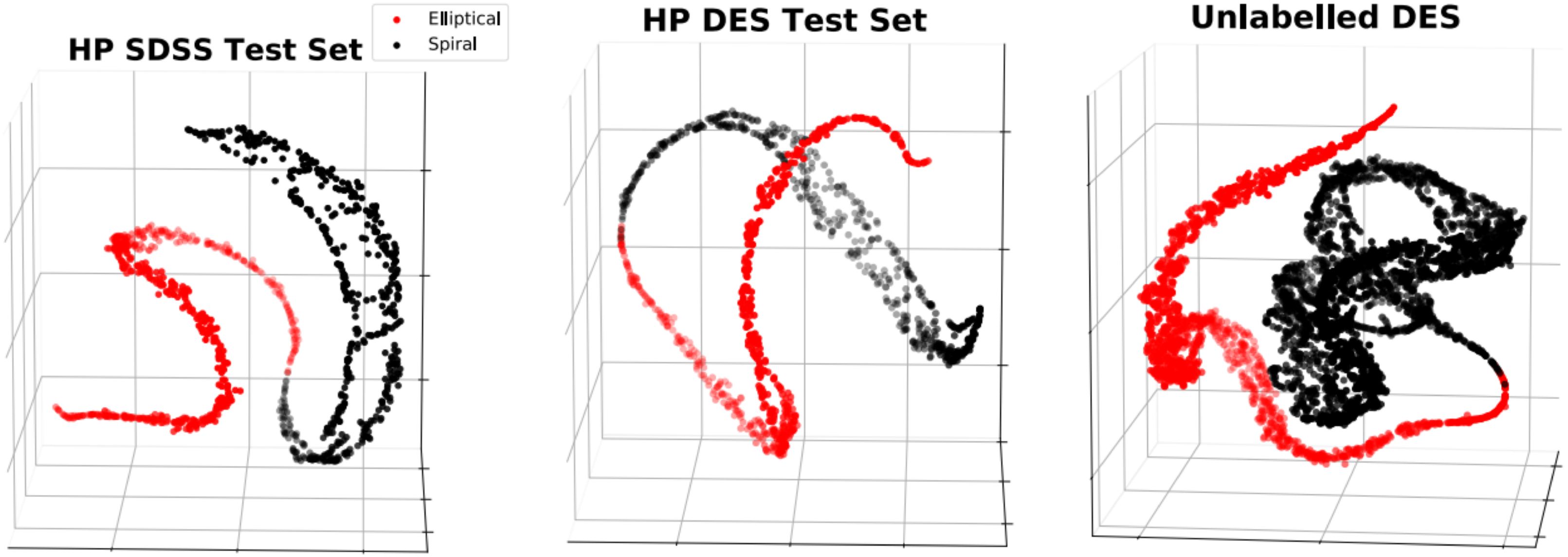
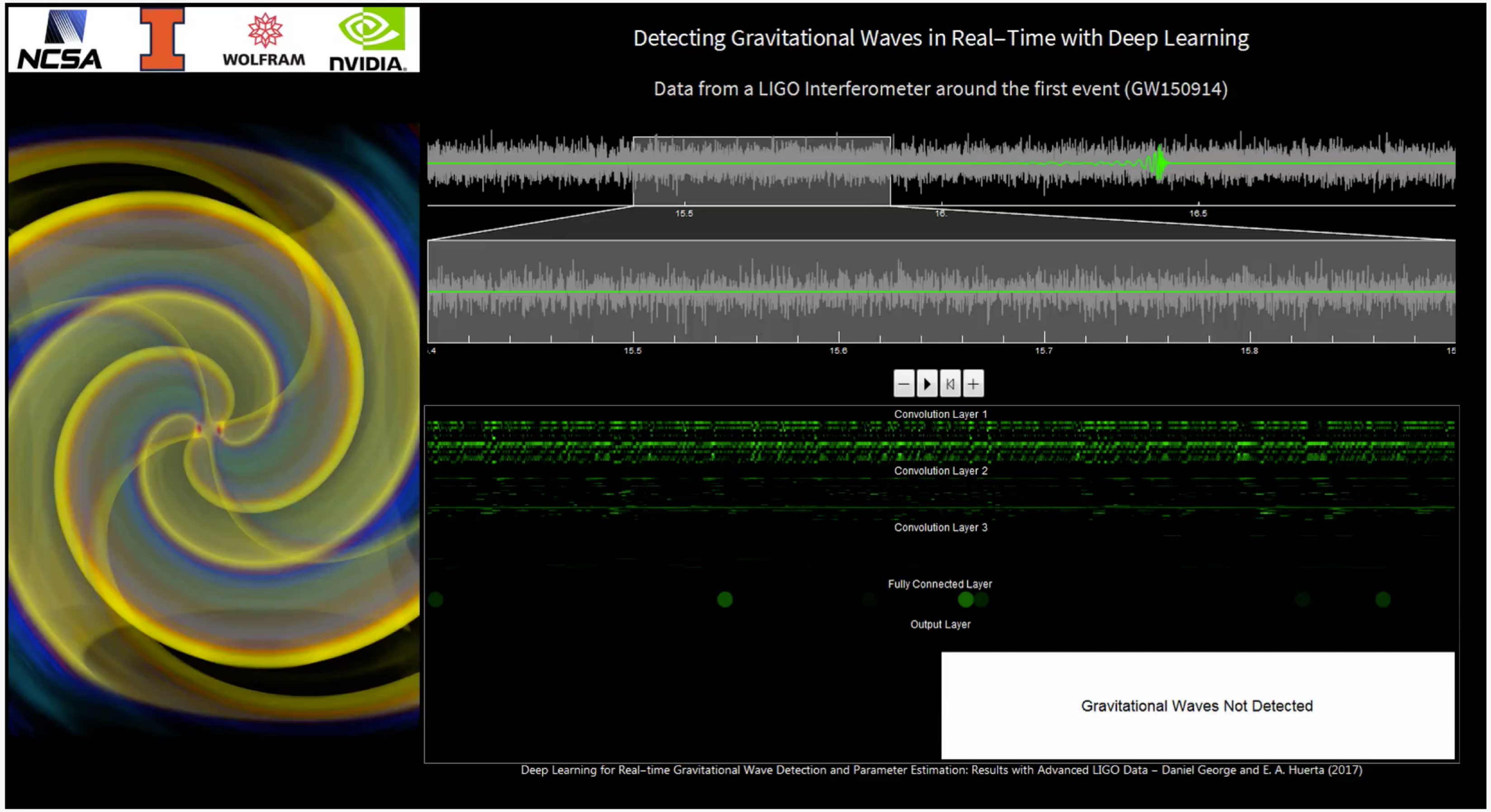


FIG. 4: t-SNE visualization of the clustering of HP SDSS and DES test sets, and unlabelled DES test.

ADSP mmaADSP
Khan et al 2019

Deep Learning driven science

Multimessenger Astrophysics through the NCSA-Argonne Collaboration
PIs: Huerta, Zhao, Haas, Saxton (NCSA)



Novel data-parallel deep learning fusing HPC and AI for MultiMessenger Astrophysics (MMA).

Huge potential for scientific discovery

- Convergence of all-sky GW observations (LIGO) with deep, high-cadence electromagnetic observations (LSST)
- Novel visualization of Neural Networks

Deep Learning for Multi-Messenger Astrophysics. A Gateway for Discovery in the Big Data Era, Huerta et al., Nature Review Physics

Deep Learning at Scale for Gravitational Wave Parameter Estimation of Binary Black Hole Mergers

- DL at scale for parameter estimation of Binary Black Hole (BH) mergers
  (spins are aligned or anti-aligned, evolve on quasi-circular orbits)

- Densely sample 4-D signal manifold ~300,000 simulated waveforms

- Enhance dataset to 10^7 samples
  time invariance in the data stream detectors
  scale invariance in range of SNR
  Add non-Gaussian and non-stationary noise.

- Distinct NN models to estimate
  Individual BH masses
  BH remnants - final spin, GW quasi-normal frequencies
  Curriculum learning with decreasing SNR

- Inference carried out for each binary BH merger observed so far from LIGO and Virgo detectors
  Parameters reconstructed within 2 milliseconds
  Consistent with Bayesian analyses (days to weeks)

Data

Number of features:

8192Hz (nsteps) downsampled compared to real data frequency

Number of samples:

300,000 simulated unique waveforms (1D time series data), augmented to 10^7 waveforms ~28GB. Collaboration with Princeton 50M waveforms

NN for spin of BHs – 300M waveforms

Encoding the data:

HDF5, TFRecords

Feature selection

– in network Root and Leaf model

$$m_1 \in [9M_\odot, 65M_\odot]$$

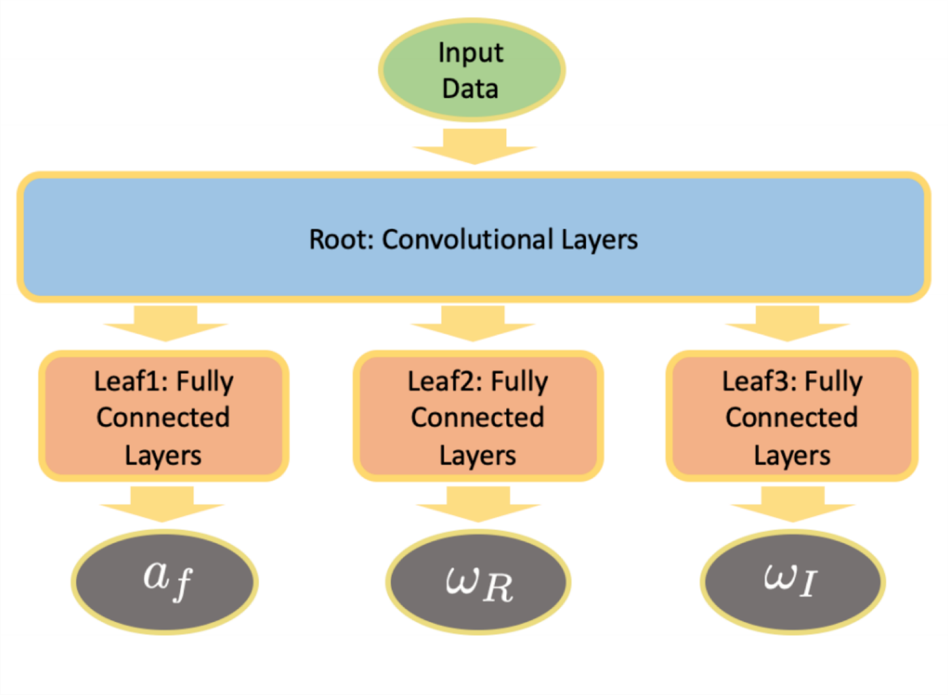
$$m_2 \in [5.2M_\odot, 42M_\odot]$$

$$a_{\{1,2\}} \in [-0.8, 0.8]$$

Models

Model 1:

- BH remnants - final spin, GW quasi-normal frequencies
- HSD-CNN Hierarchically self decomposing CNN (SaiRam et al 2018), subnetworks for specific set of classes
- EraseReLU (Dong et al 2017): erasing ReLUs of certain layers to enhance propagation of useful information



(kernel size, # of output channels, stride, dilation rate, max pooling kernel, max pooling stride)

Root Layer: Convolutional	$(16, 64, 1, 1, 4, 4)$ $(16, 128, 1, 2, 4, 4)$ $(16, 256, 1, 2, 4, 4)$ $(32, 256, 1, 2, 4, 4)$ $(4, 128, 1, 2, 0, 0)$ $(4, 128, 1, 2, 0, 0)$ $(2, 64, 1, 1, 0, 0)$	ReLU

Leaf Layer :

FC $(1024, 0.0)$ ReLU

FC $(1024, 0.0)$ ReLU

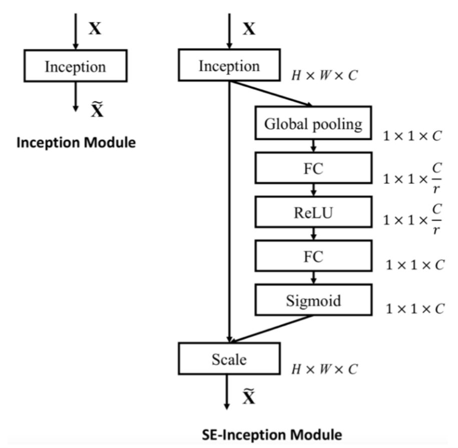
FC $(1024, 0.0)$ Identity

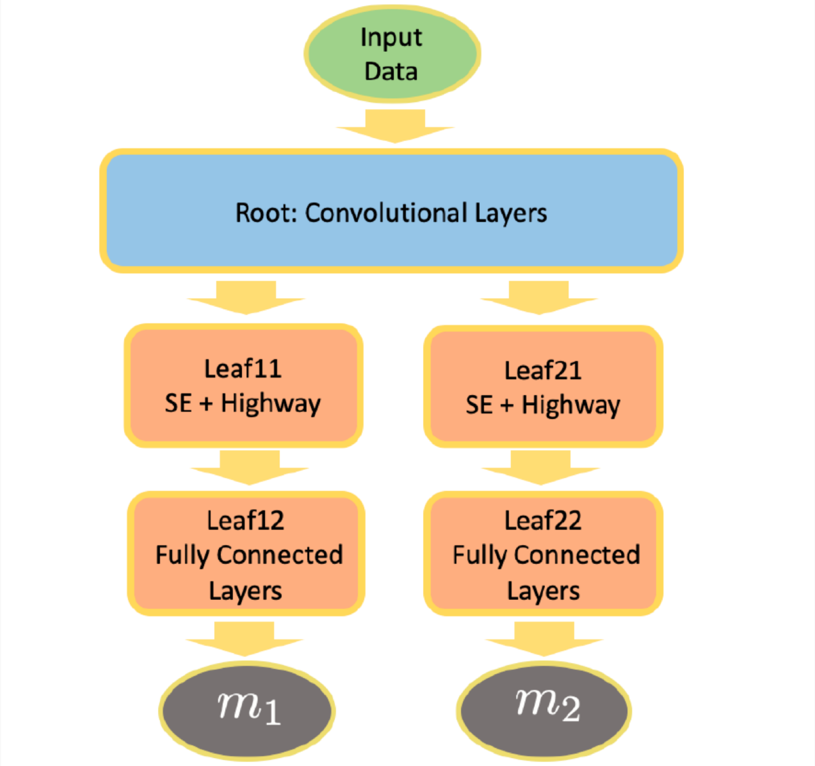
FC $(1, 0.0)$ Tanh

Models

Model 2:

- BH masses
- Squeeze-Excitation structure(Hu et al 2018) models interdependencies in channels
- ’Highway’ block (Srivastava et al 2015) learns the residual components, short cut for learning features





(kernel size, # of output channels, stride, dilation rate, max pooling kernel, max pooling stride)

Layer Component	Layer Configurations	Activation Functions
Root Layer: Convolutional	$(16, 64, 1, 2, 4, 4)$ $(16, 128, 1, 2, 4, 4)$ $(16, 128, 1, 2, 4, 4)$	ReLU
Leaf Layer: SE	$(128, 3)$ $(128, 3)$	ReLU
Leaf Layer: Highway	$(4, 128, 2, 30)$	ReLU

Leaf Layer :

FC $(1024, 0.1)$ ReLU

FC $(1024, 0.1)$ ReLU

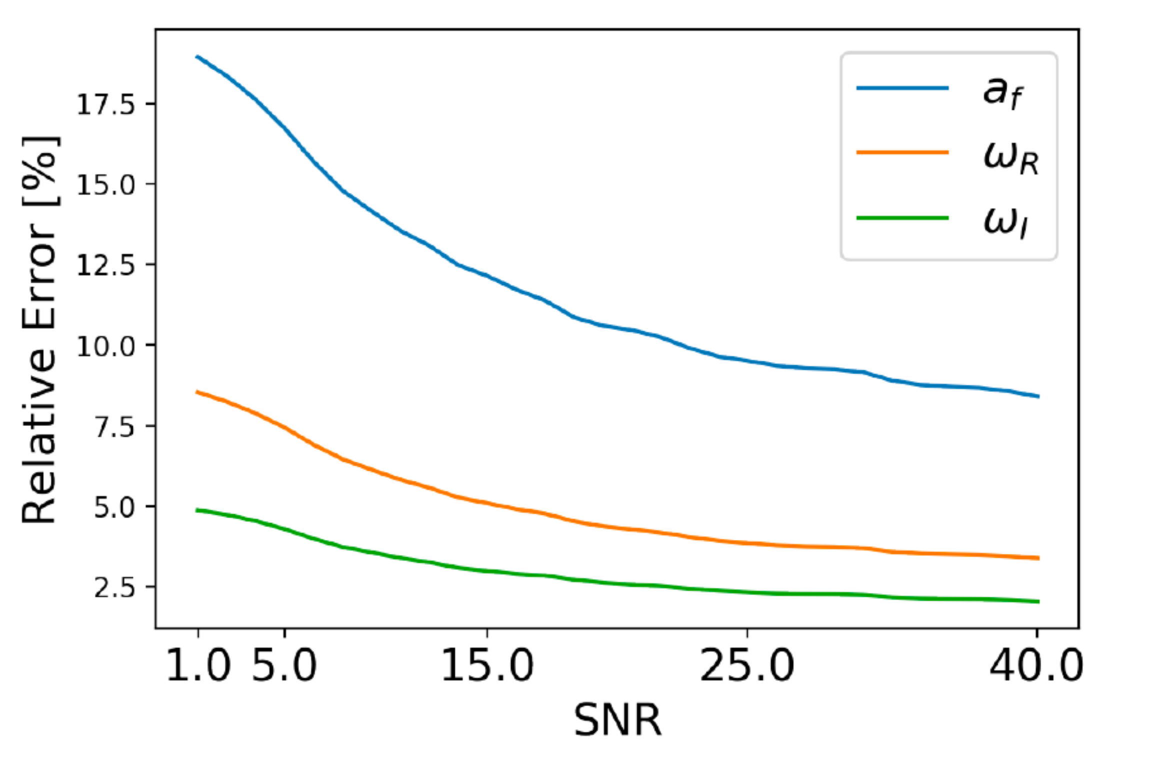
FC $(1024, 0.0)$ Identity

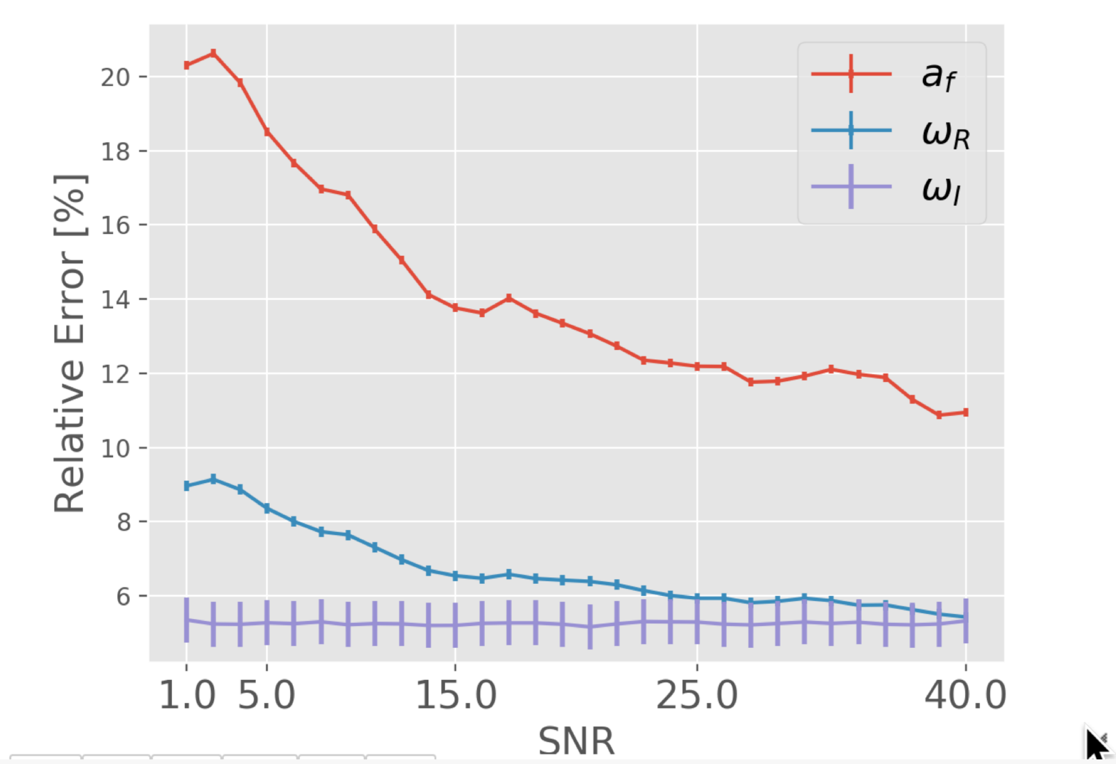
FC $(1, 0.0)$ Tanh

Bayesian Neural Network implementation

- L2loss re-defined to be ELBO loss
- Posterior distribution = Normal with network outputs as mean, std could be implemented also

```
labels_distribution = tfd.Normal(loc=outputs,scale=fixed_var*tf.ones(1))

sample_distribution = labels_distribution.sample()

# Compute the -ELBO as the loss, averaged over the batch size.
neg_log_likelihood = -
tf.reduce_mean(input_tensor=labels_distribution.log_prob(output_vector))

KL = sum(model.losses)/ tf.cast(self.N,dtype=tf.float32)

elbo_Loss = neg_log_likelihood + alpha_KL * KL
```


17.5
15.0
12.5
10.0
7.5
5.0
2.5
Relative Error [%]
1.0 5.0 15.0 25.0 40.0
SNR
a_f
ω_R
ω_I


20
18
16
14
12
10
8
6
Relative Error [%]
1.0 5.0 15.0 25.0 40.0
SNR
a_f
ω_R
ω_I

Thank you !