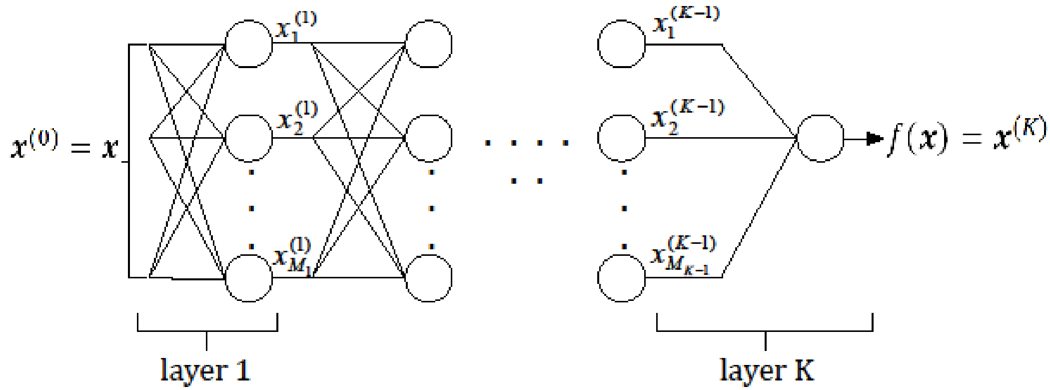
# Deep Learning Based Cooperative Scheduling with Distributed Non-linear Predictive Coding for Small Spacecraft Swarms

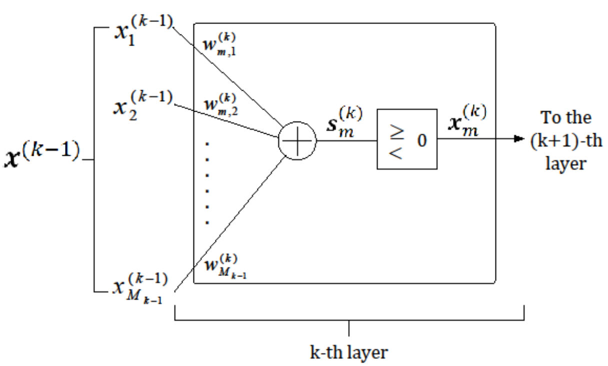
**Sudharman K. Jayaweera**
**Communications and Information Sciences Lab (CISL)**
Department of Electrical and Computer Engineering
University of New Mexico
USA.

**Brian McCollum and Mustafa Alkwaz**
Bluecom Systems & Consulting
Albuquerque, NM, USA.

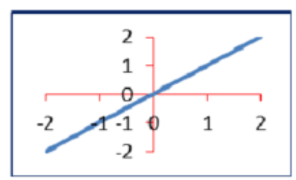
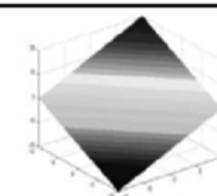
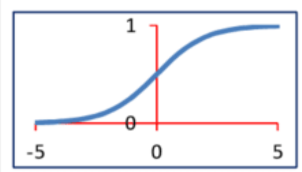
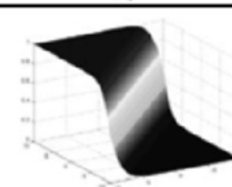
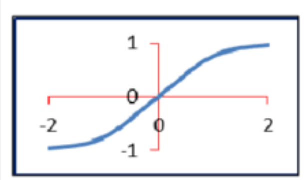
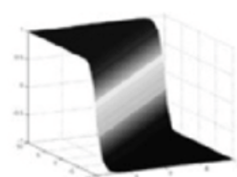
# Feed-Forward Neural Networks (FFNN)

- Most well-known Artificial Neural Network (ANN)
  - Several layers of fully-connected (FC) nonlinear elements called neurons
  - Output of each neuron is a nonlinear function of the **weighted** sum of its inputs
- Outputs of neurons in one layer become the inputs to the next layer



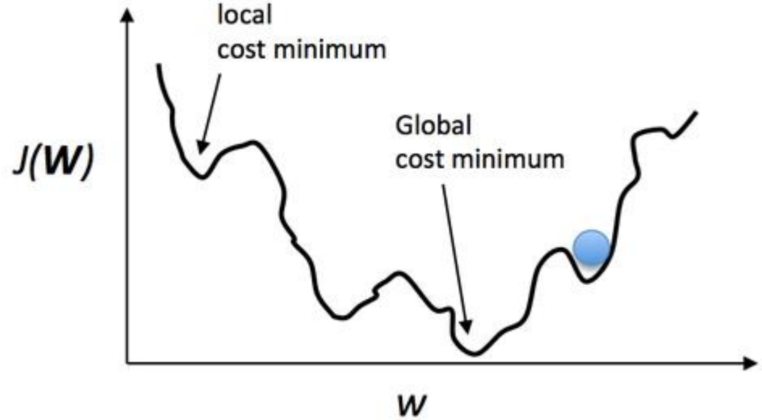



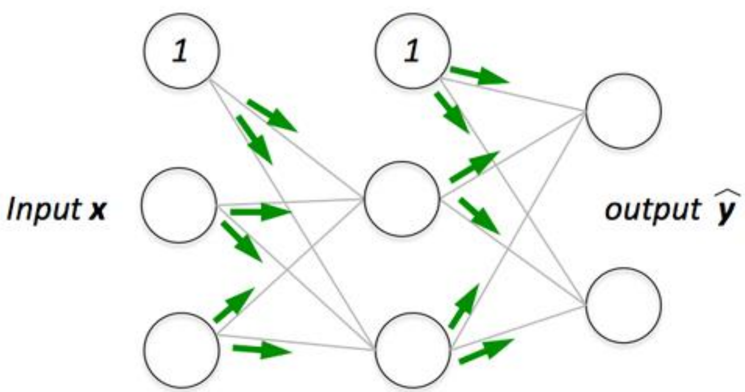

- Many possible activation functions for the nonlinearity of the neuron

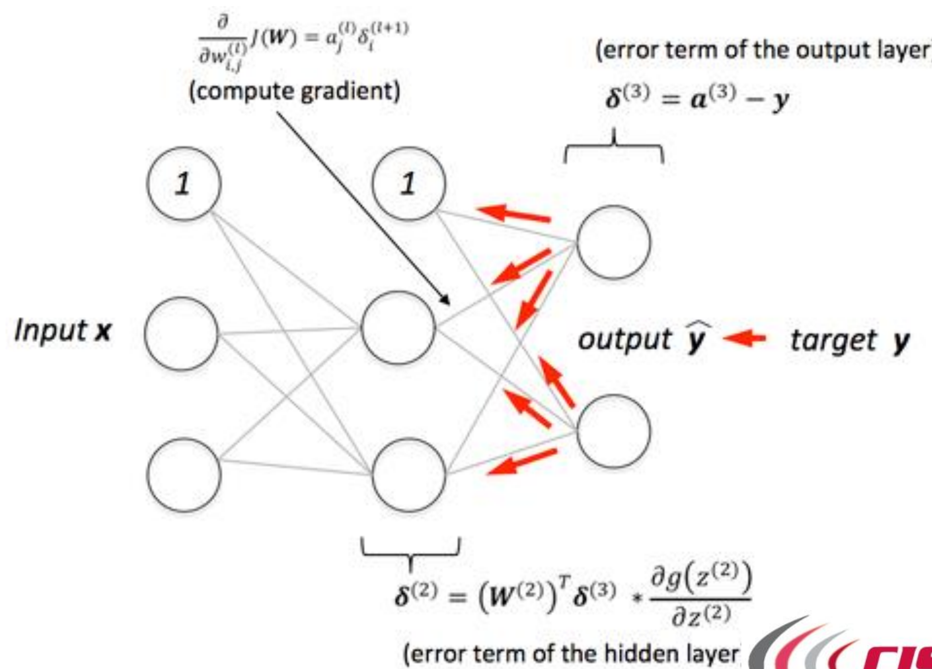
| Active Function Name | Formula | 2D Graphical Representation | 3D Graphical Representation | Description |
|---|---|---|---|---|
| Linear | $f(x) = x$, for all $x$ |  |  | The activation of the neuron is passed on directly as the output |
| Logistic (or sigmoid) | $f(x) = \frac{1}{1+e^{-x}}$ |  |  | A S-shaped curve, very popular because it is Monotonous and has a simple derivative, Range of logistic or sigmoid function is from 0 to 1 |
| Hyperbolic Tangent | $f(x) = tanh(x)$ <br> $f(x) = \frac{1+e^{-2x}}{1+e^{2x}}$ |  |  | A sigmoid curve similar to the logistic function. Often performs better than the logistic function because of its symmetry. Ideal for multilayer Perceptrons, particularly the hidden layers. Output value is between -1 and +1 |

# Training Neural Networks: Back Propagation Algorithm

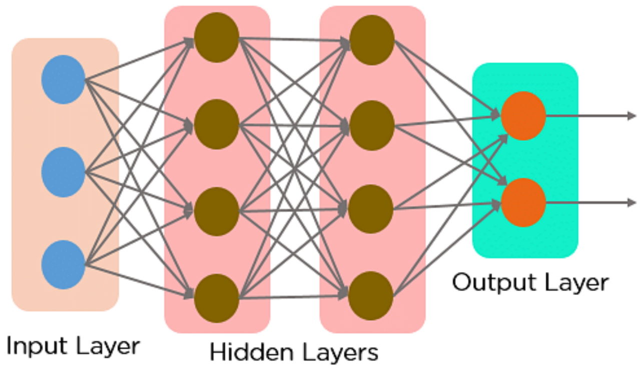
- FFNNs can be trained by using the Back Propagation (BP) Algorithm
  - Stochastic Gradient Descent (SGD) to minimize the loss (error) between the network's output and the true output
- Different loss functions and variations of SGD are possible

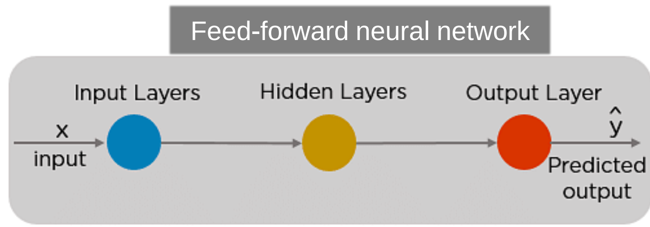


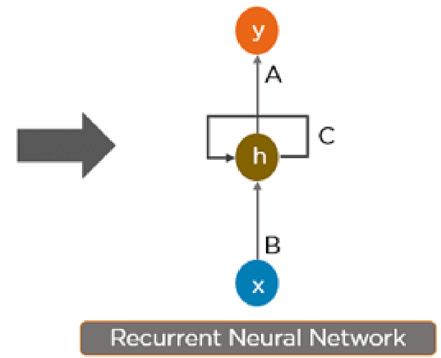



# Feed-Forward Vs. Recurrent Neural Networks (RNN)

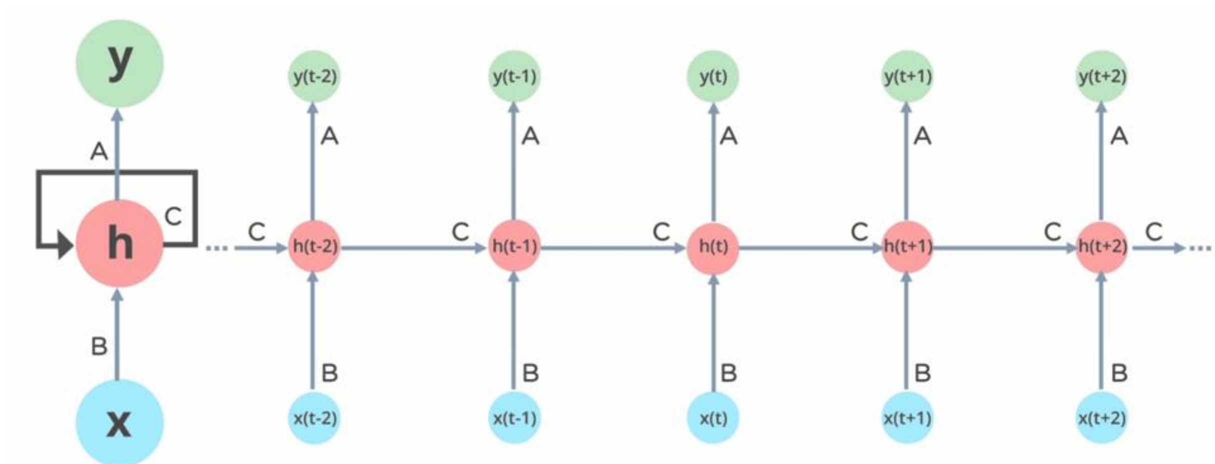



- In FFNNs information only flows from input to the output direction
  - No cycles or loops, No memory
  - Not ideal for handling sequential or time-series data

- Recurrent Neural Networks (RNN)
  - Outputs of layers/neurons are fedback as inputs
  - Each neuron has an internal hidden state (h) that is used to feedback information

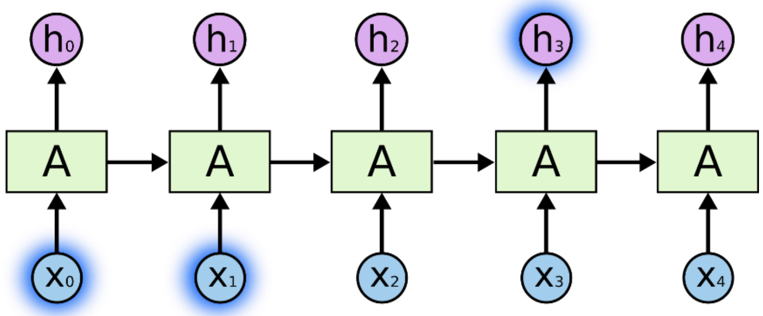


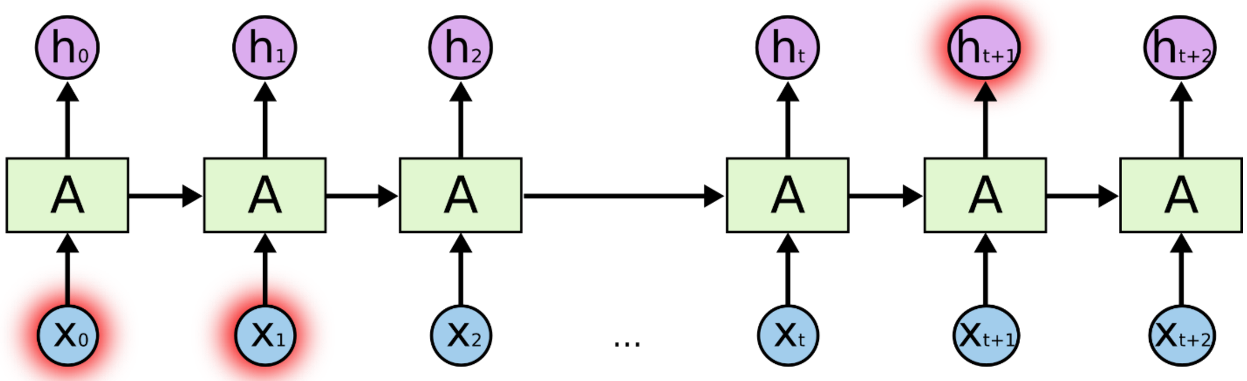

Ideal for handling sequential data (with correlations):

e.g. NLP (text mining, sentiment analysis), machine translation, time-series prediction

# Learning Long-term Dependencies?

- In theory, RNNs can be trained just the same way as FFNNs by modifying the BP algorithm to what is called **Back Propagation Through Time** (BPTT)
  - In practice, gradients can quickly vanish or explode rendering it ineffective

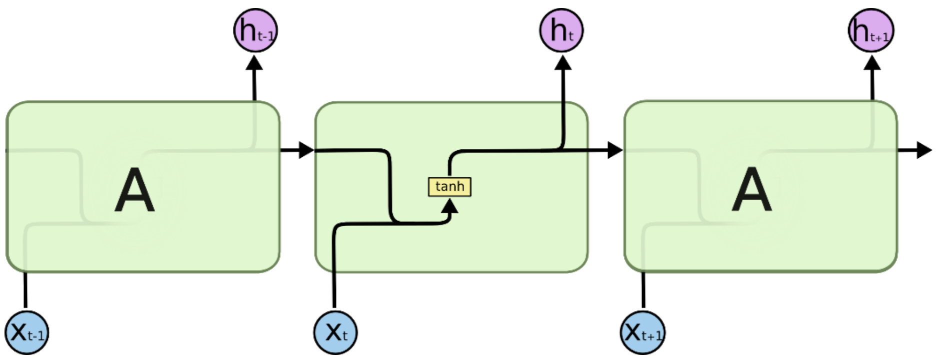



- Standard RNNs are not very effective in learning long-term dependencies

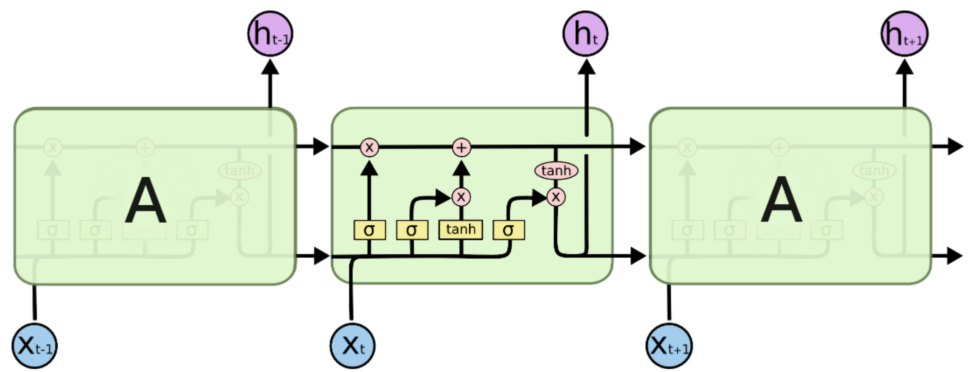
# Long Short-term Memory (LSTM)

- Long short-term Memory, LSTM, is a more elaborate type of RNN that has shown to be capable of learning long-term dependencies

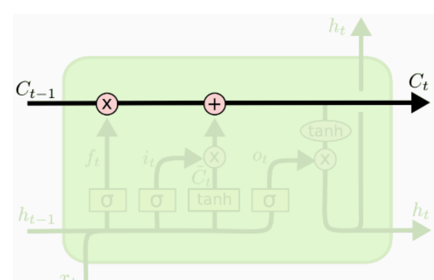



- Standard RNN only has a single layer that performs hidden state and input interactions

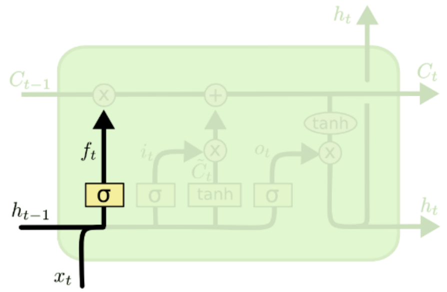
$$\mathbf{h}_t = \tanh\left(W_h \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$



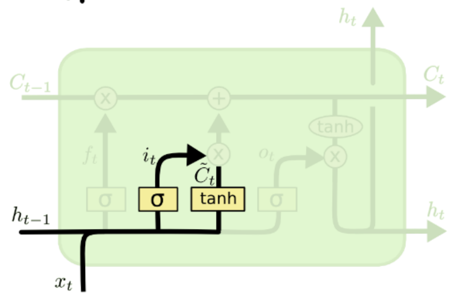

- In LSTM, there are four interacting layers
  - Forget gate, input gate, cell gate, output gate
- In addition to the hidden state, there is a another state that carries information from one time instant to another
  - Cell state



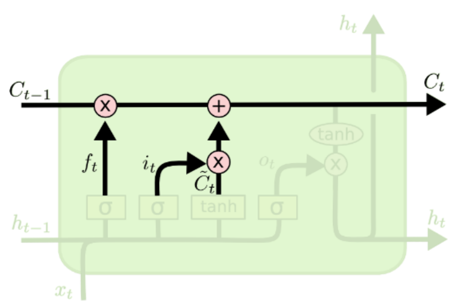
# What Does LSTM Do?



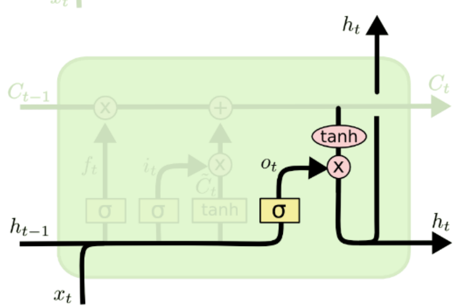

$$\mathbf{f}_t = \sigma\left(W_f \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$




$$\mathbf{i}_t = \sigma\left(W_i \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$
$$\tilde{\mathbf{C}}_t = \tanh\left(W_C \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$



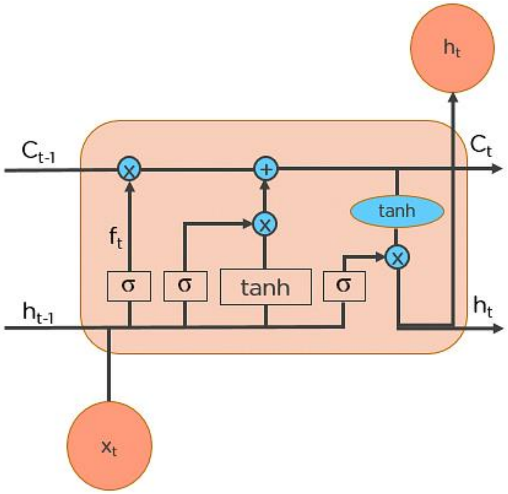

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t$$



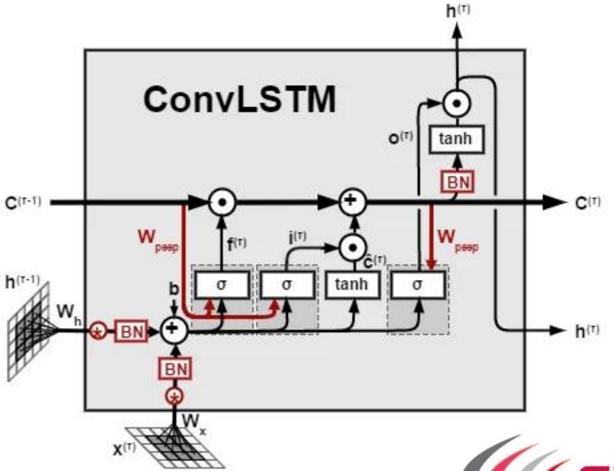

$$\mathbf{o}_t = \sigma\left(W_o \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh \mathbf{C}_t$$

# ConvLSTM: A Neural Model for Learning Spatio-temporal Correlations

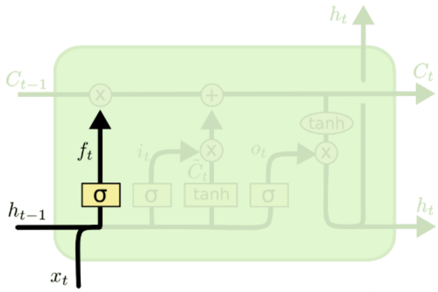
- Long Short-term Memory (LSTM) is good at learning long-term correlations in temporal data
  - They cannot learn spatial correlations!




- Convolutional LSTM (ConvLSTM):
  - Combines the convolution of CNNs with sequential processing of LSTMs
  - Replace matrix products with weights by convolution with a filter kernel
- Ideal for learning spatiotemporal correlations in image sequences
  - Keras: convlstm2d

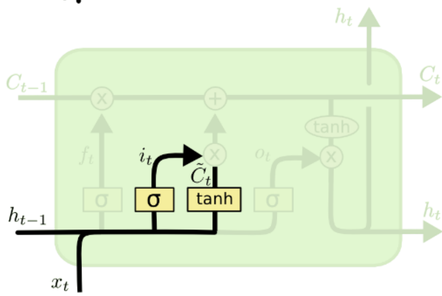


# What Does ConvLSTM Do?



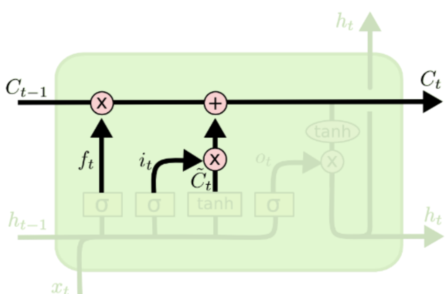
$$f_t = \sigma\left(W_f \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$

$$\mathbf{f}_t = \sigma\left(W_f \circledast [\mathbf{h}_{t-1}, \mathbf{x}_t]\right)$$



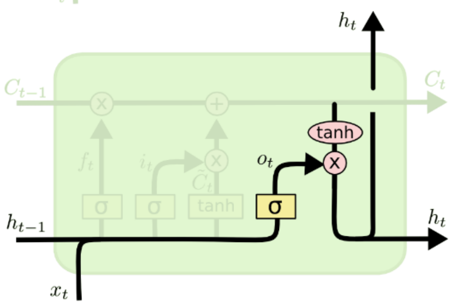
$$\begin{aligned} \mathbf{i}_t &= \sigma\left(W_i \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \\ \tilde{\mathbf{C}}_t &= \tanh\left(W_C \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \end{aligned}$$

$$\begin{aligned} \mathbf{i}_t &= \sigma\left(W_i \circledast [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \\ \tilde{\mathbf{C}}_t &= \tanh\left(W_C \circledast [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \end{aligned}$$



$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t$$

$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t$$

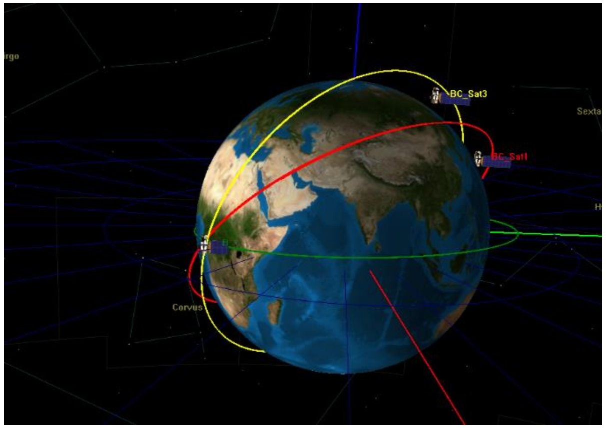


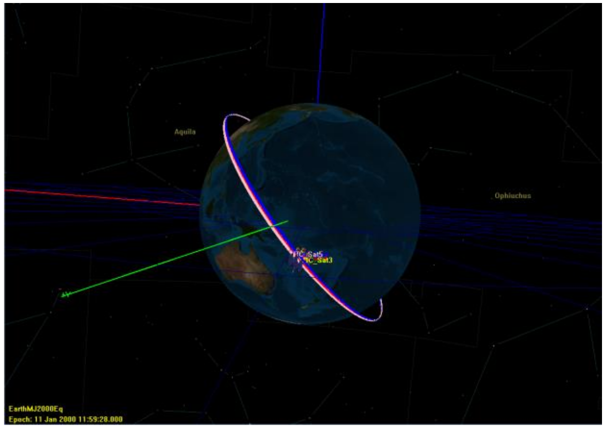
$$\begin{aligned} \mathbf{o}_t &= \sigma\left(W_o \times [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh \mathbf{C}_t \end{aligned}$$

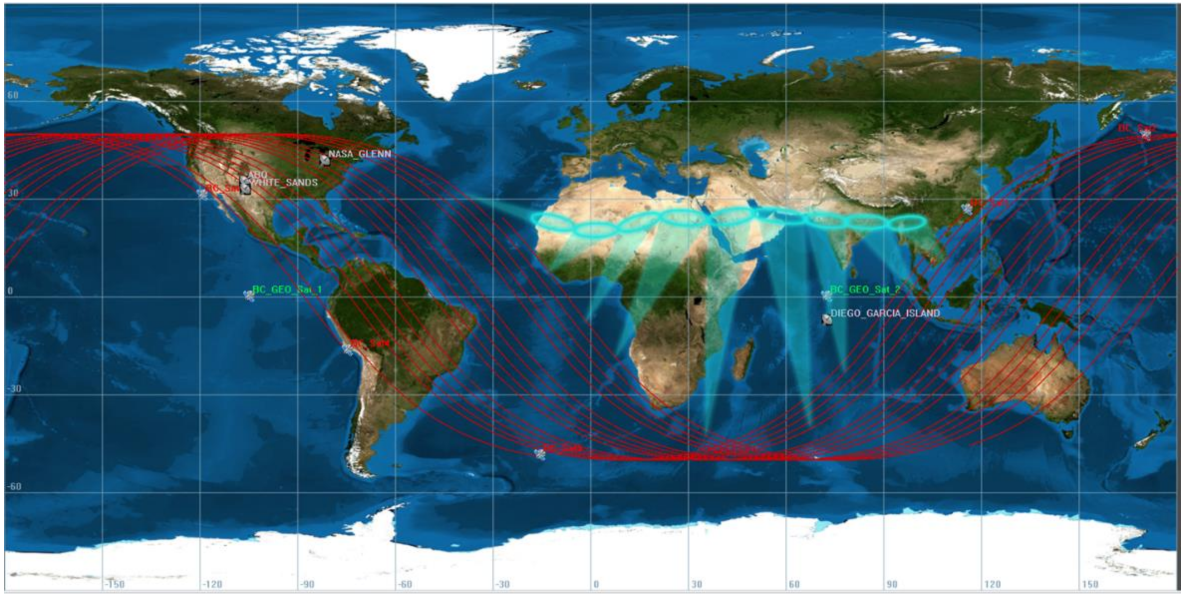
$$\begin{aligned} \mathbf{o}_t &= \sigma\left(W_o \circledast [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh \mathbf{C}_t \end{aligned}$$

# Remote Sensing with LEO Satellites

- A swarm of LEO satellites
- Pearl-of-string or cluster formations

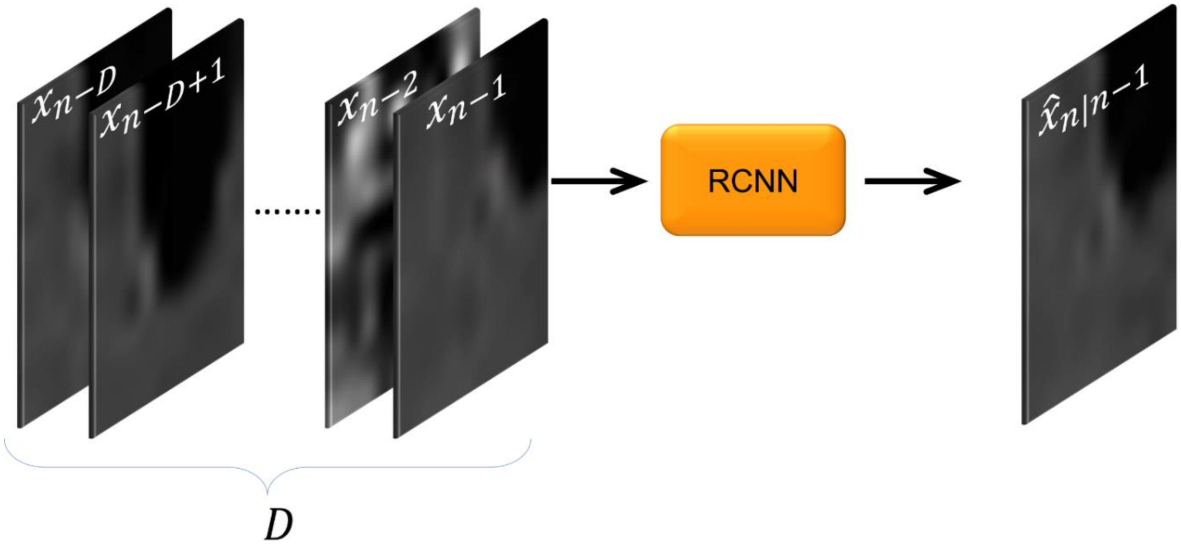





- Each satellite generates an image of the earth's surface inside its footprint on the earth
  - Periodic or even-driven

# DL based Nonlinear Predictive Coding (NLPC) for Remote Sensing




$$e_{pN+n}^{(j)} = x_{pN+n}^{(j)} - \hat{x}_{pN+n|pN+n-1}^{(j)}$$

$$\text{output bpp} = \begin{cases} q & \text{if } |e_n| \le \sigma \\ Q+q & \text{otherwise} \end{cases}$$

- Original image size: dxd pixels (e.g. 10x10)
- Original **bits per pixel (bpp)**: Q (e.g. Q=8)
- Pixel resolution: $1/2^Q$ (e.g. 1/256)
- RCNN training period: $N_t$
- Estimated std of pixel prediction errors of j-th satellite

$$\sigma^{(j)} = \sqrt{\frac{1}{N_t}\sum_{n=1}^{N_t}\frac{\left|\mathbf{X}_n^{(j)} - \hat{\mathbf{X}}_n^{(j)}\right|^2}{d^2}} = \sqrt{\frac{1}{N_t}\sum_{n=1}^{N_t}\frac{\left|e_n^{(j)}\right|^2}{d^2}}$$

- **Assumption** - Prediction error range: [-σ, σ]
- Quantization levels needed to keep the same original resolution:

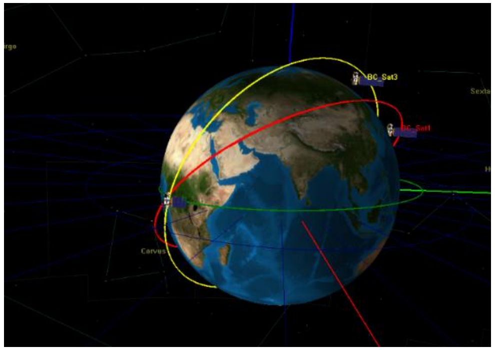
$$L = \frac{2\sigma}{1/2^Q} = 2^{Q+1}\sigma$$

- Minimum number of bits per pixel needed to encode the prediction error at the same original resolution

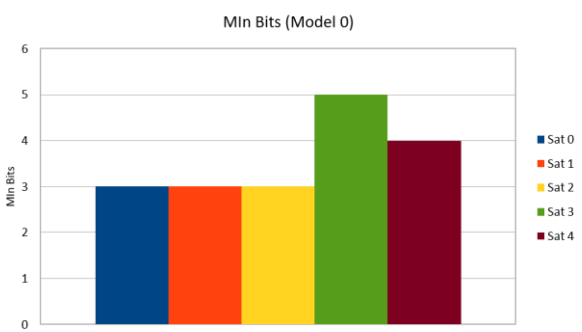
$$\begin{aligned} q &= 1 + \lceil \log_2(L) \rceil \\ &= 1 + \lceil \log_2\left(2^{Q+1}\sigma\right) \rceil = 1 + \lceil Q + 1 + \log_2 \rceil \end{aligned}$$

# RCNN NLPC Coding of Earth Images

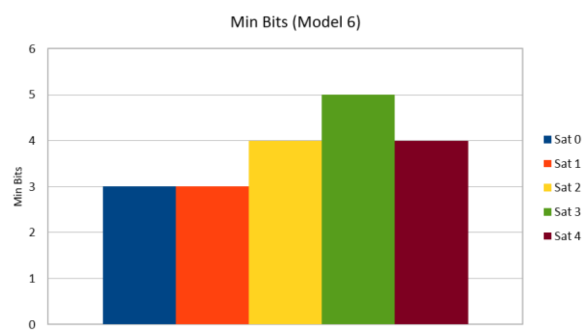
## Swarm of 5 LEO cubesats



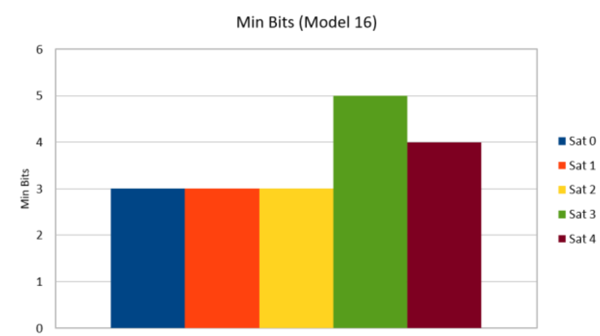
- Final layer of all models: Conv3D, Filters = 1, Kernel = (3,3,3), Activation = sigmoid
- Training over $N_t$ = 3000 time instants (observation points)
- Original images: 10x10 pixels (d=10) with Q=8 bpp




Model 0
- Number of Layers: 7
- Layers 1 – 6: Conv2Dlstm, Filters = 49, Kernel = (3x3), Activation = tanh
- Number of trainable parameters: 955648
- Runtime: 31414 seconds



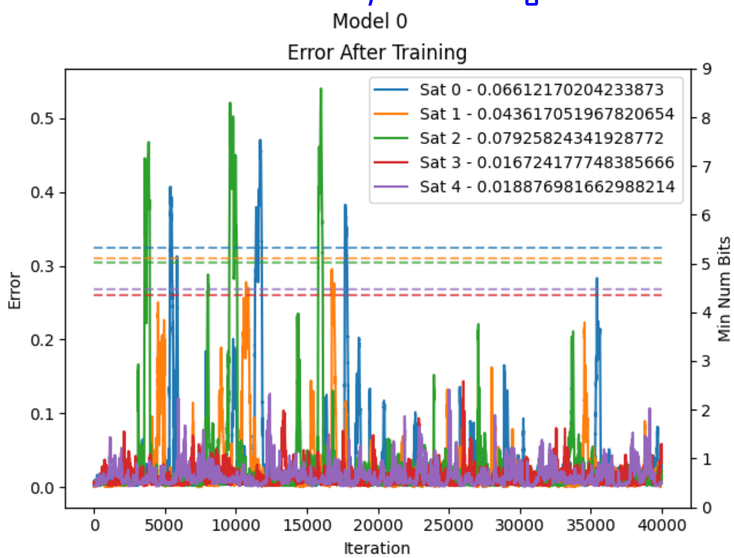

Model 6
- Number of Layers: 4
- Layers 1 – 3: Conv2Dlstm, Activation = relu
- Layer 1 -3 Filters: 128, 64, 32
- Layers 1- 3 Kernels: (5x5), (3x3), (1x1)
- Number of trainable parameters: 2108065
- Runtime: 21034 seconds

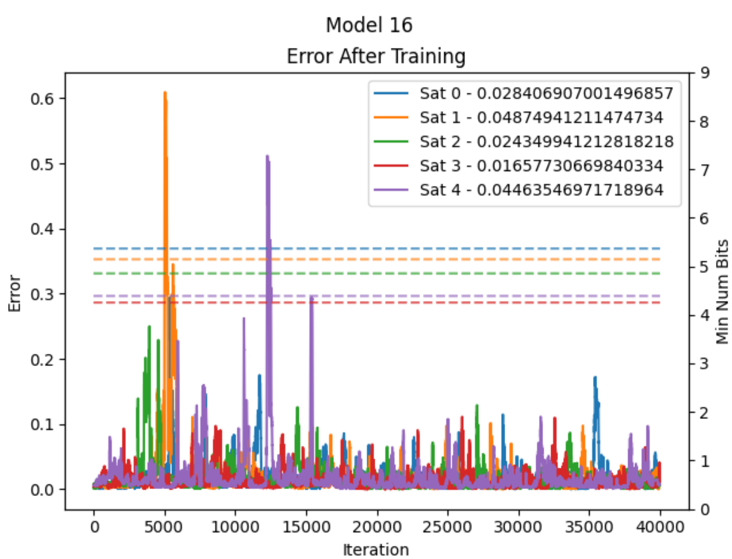



Model 16
- Number of Layers: 4
- Layers 1 – 3: Conv2Dlstm, Activation = relu
- Layer 1 -3 Filters: 49, 39, 29
- Layers 1- 3 Kernels: (5x5), (3x3), (1x1)
- Number of trainable parameters: 377926
- Runtime: 18420 seconds

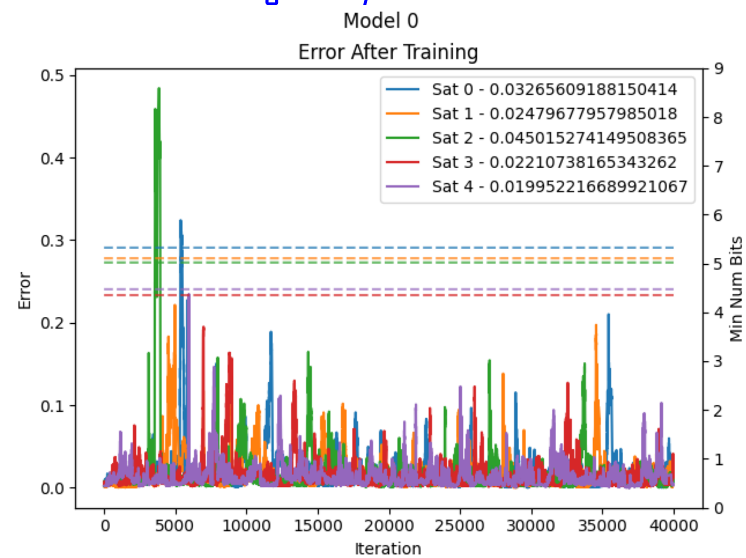
# Performance of RCNN NLPC Coding of Earth Images

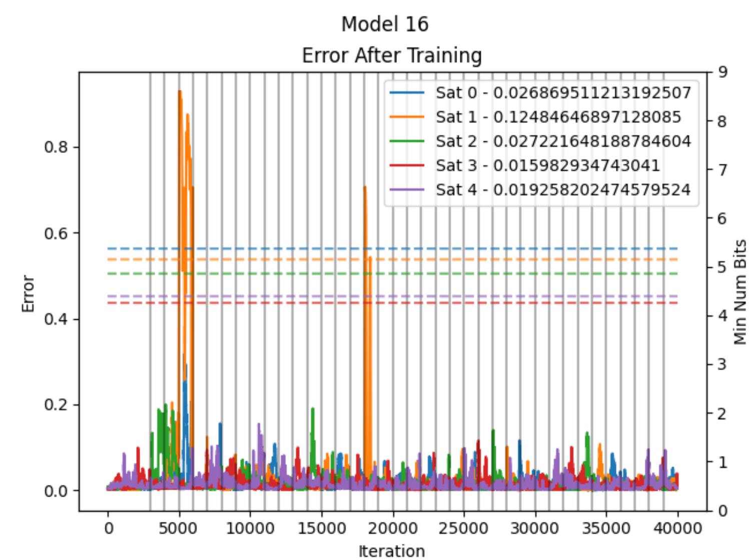
## Without any retraining



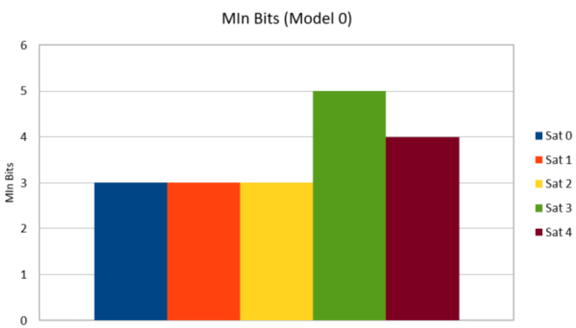

## With Retraining every 1000 time instants

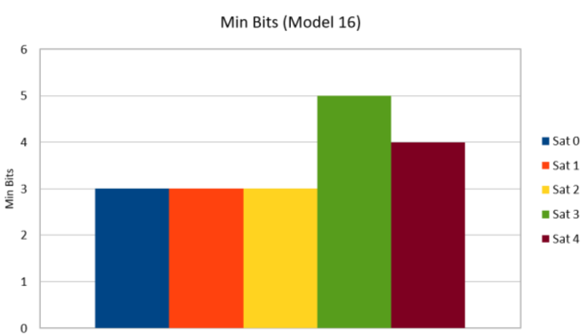


# Performance of RCNN NLPC Coding of Earth Images

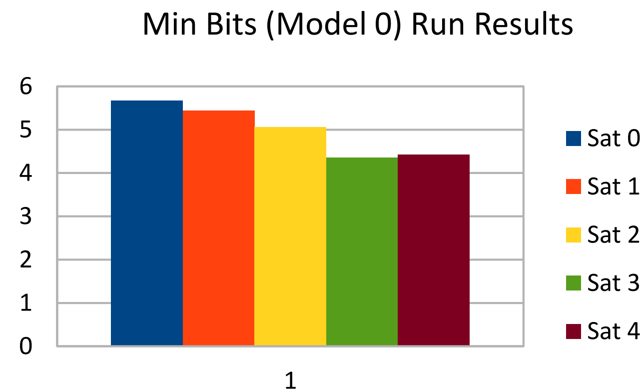
## Estimated Minimum Bits (q)



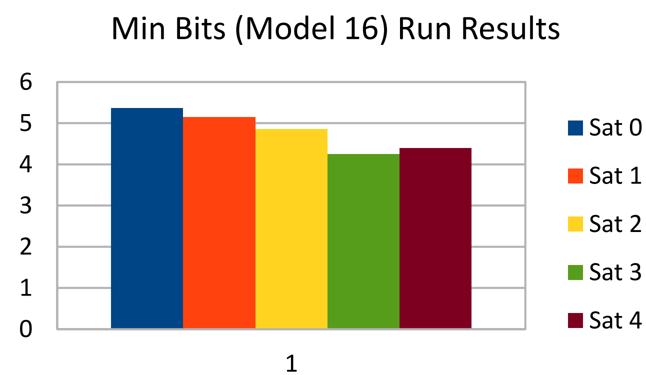




## True Bits used during runtime

| | |
|---|---|
| **Sat #1** | 5.67 |
| Sat #2 | 5.44 |
| Sat #3 | 5.06 |
| Sat #4 | 4.36 |
| Sat #5 | 4.43 |
| Average | 4.99 |

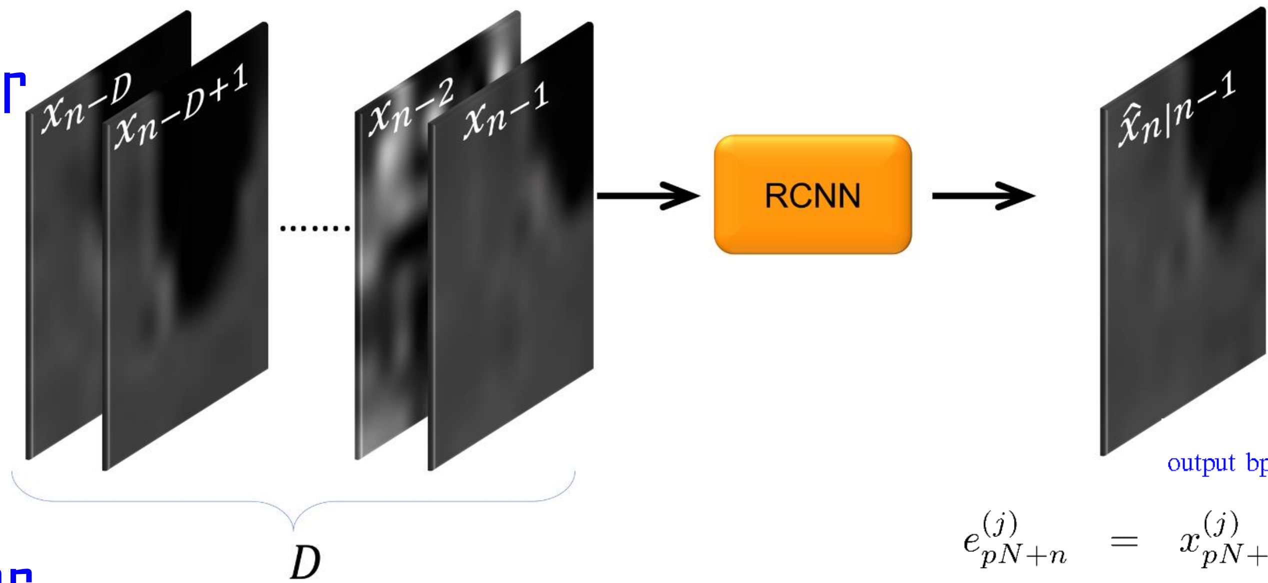



| | |
|---|---|
| **Sat #1** | 5.37 |
| Sat #2 | 5.15 |
| Sat #3 | 4.85 |
| Sat #4 | 4.25 |
| Sat #5 | 4.39 |
| Average | 4.80 |



# DL based Nonlinear Predictive Coding (NLPC) for Remote Sensing

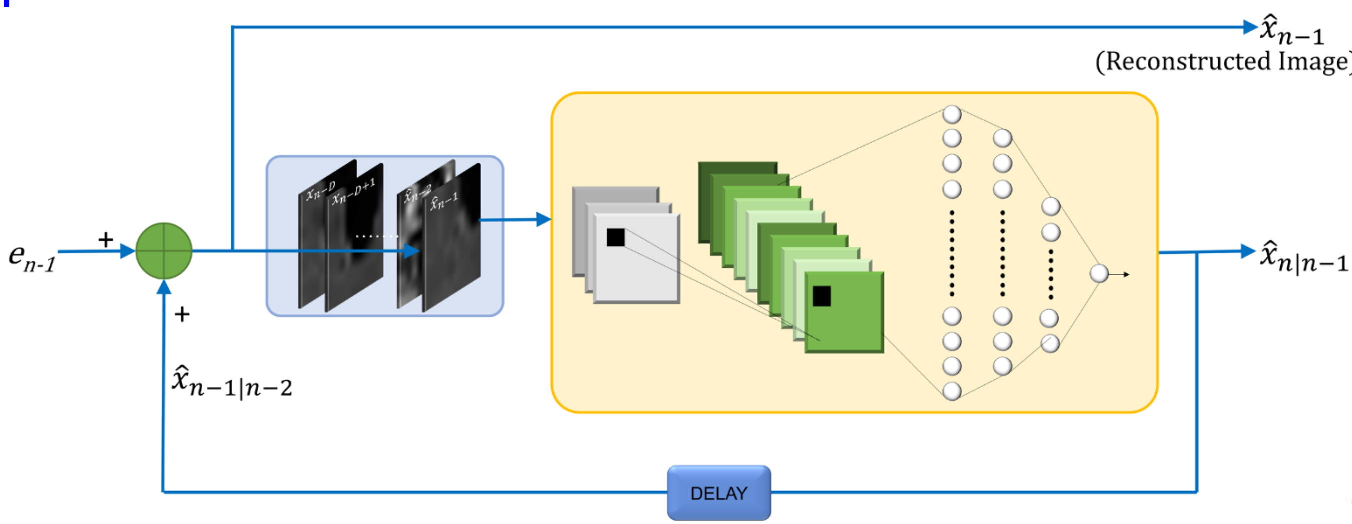
- Encoder




$$\text{output bpp} = \begin{cases} q & \text{if } |e_n| \le \sigma \\ Q+q & \text{otherwise} \end{cases}$$

$$e^{(j)}_{pN+n} = x^{(j)}_{pN+n} - \hat{x}^{(j)}_{pN+n|pN+n-1}$$

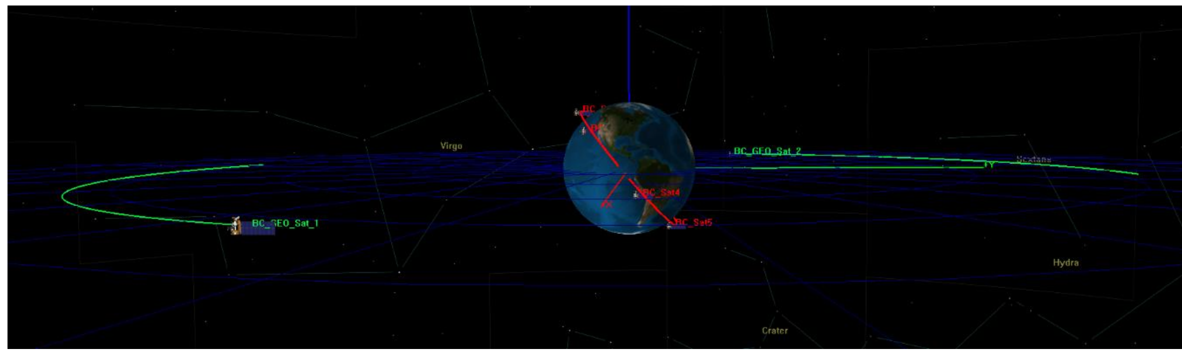
- Decoder

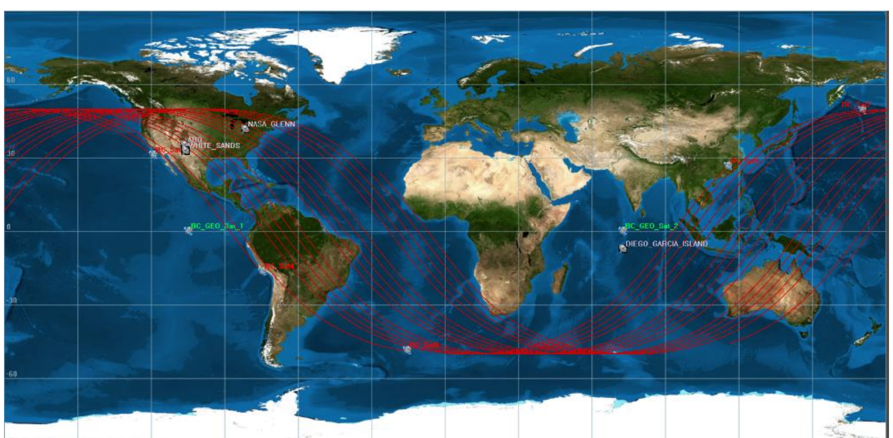


# DL based Distributed NLPC Compression in Cubesat Swarms

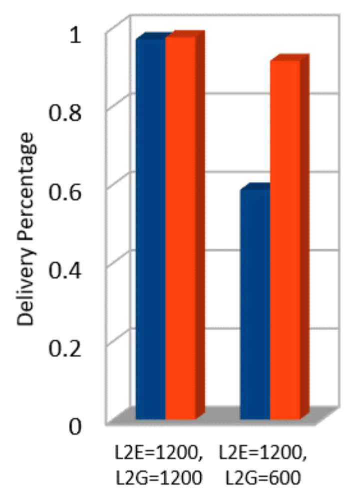
## Swarm of 5 LEO cubesats, 2 GEO relays and 2 Earth stations

- Fully distributed implementation for real-time encoding and decoding
- Regular updating of DNN's in space while maintaining synchronization with ground decoders
- 10x10 pixel grey images of earth footprint
- Bursty or periodic data
- Cluster or string-of-pearls formations

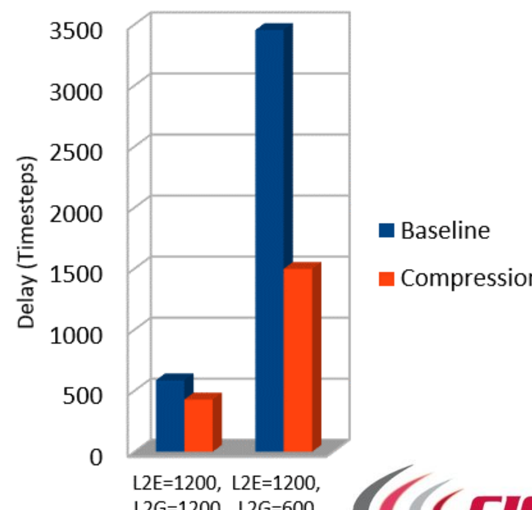


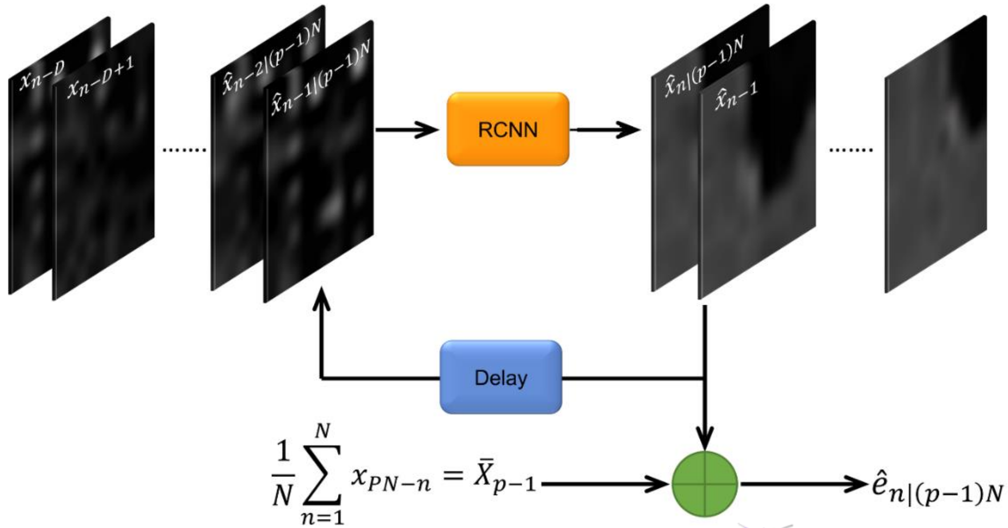

Percentage of Delivered Images



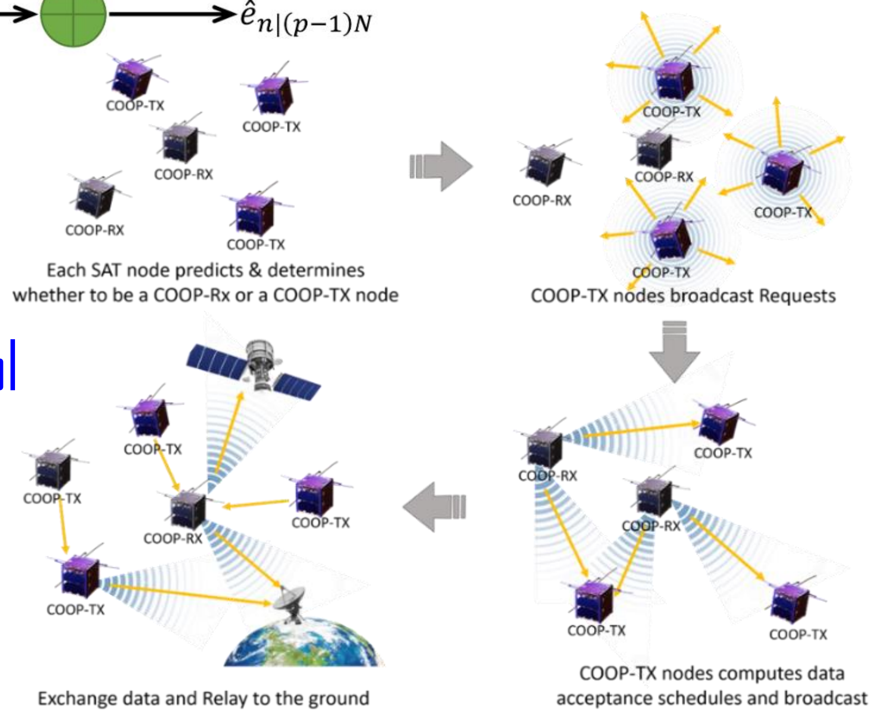

Average image Delivery Delay



# DL aided Cognitive Cooperative Scheduling for Cubesat Networks

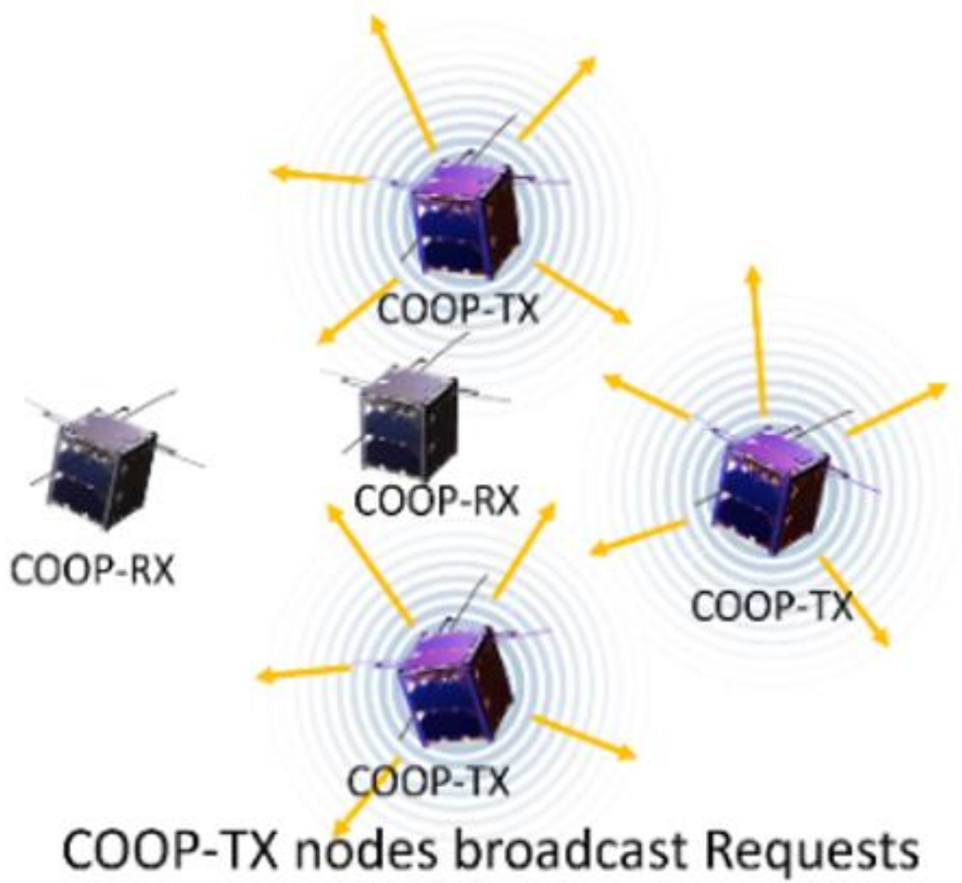



- Payload Prediction with Trained RCNN
- Predict and broadcast relay capacities
- Cooperative relay requests
- Multi-objective relay prioritization protocol for relay scheduling
- L2L buffer exchanges
- L2Geo/L2Earth data transmissions



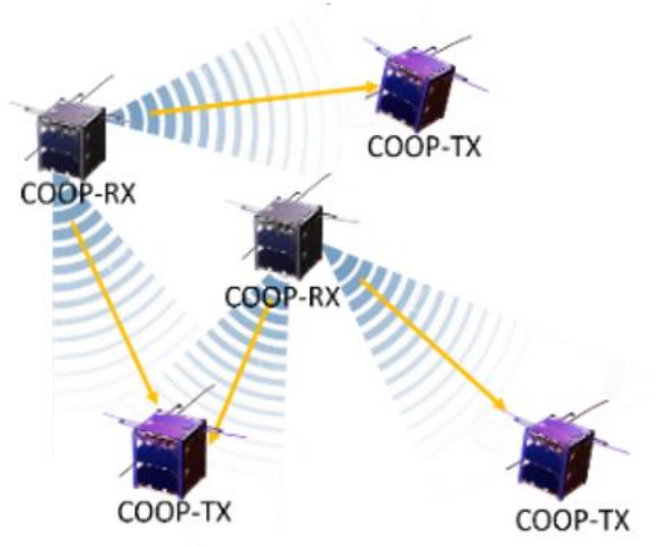

CISL

# Cooperative Relay Requests




$$\left(K_p^{(j)}, \bar{D}_p^{(j)}\right)$$

# Cooperative Relay Scheduling




COOP-TX nodes computes data

$$f_{p,f}^{(j)} = \frac{K_p^{(j)}}{\sum_{j' \in S_{T,p}^{(i)}} K_p^{(j')}}$$

$$F_p^{(j)} = \lambda_1 f_{p,f}^{(j)} + \lambda_2 f_{p,l}^{(j)}$$

Step 1: Order the allocation fractions $F_j$ in non-increasing order $F_{j_1} \geq F_{j_2} \geq \cdots, \geq F_{j_M}$ where $j_k \in S_T^{(i)}$, for $k = 1, \cdots, |S_T^{(i)}|$, and set $\hat{K}_p^{(i)} = K_p^{(i)}$.

Step 2: For $k = 1, 2, \cdots, |S_T^{(i)}|$:

$$K_{j_k,i} = \min\left\{F_{j_k}\hat{K}_p^{(i)}, K_p^{(j_k)}, C_{j_k,i}\right\}$$

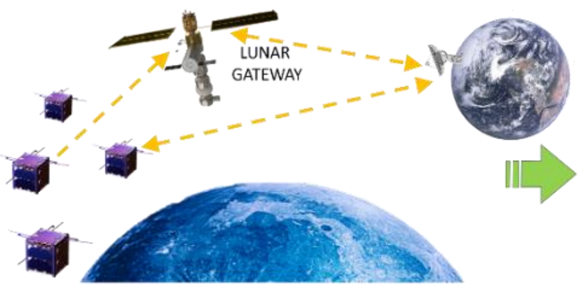
$$\hat{K}_p^{(i)} \leftarrow \hat{K}_p^{(i)} - K_{j_k,i}$$

and, for $m = k+1, \cdots, |S_T^{(i)}|$,

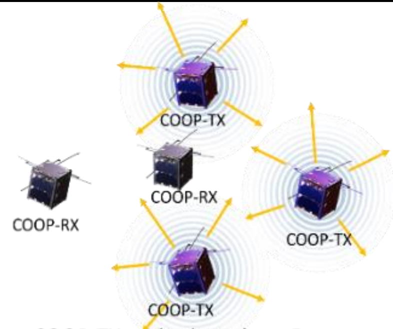
$$F_{j_m} \leftarrow \frac{F_{j_m}}{\sum_{m'=k+1}^{M} F_{j_{m'}}} \text{ for } m = k+1, \cdots, |S_T^{(i)}| \quad (7)$$

# DL based Protocol for Small spacecraft Swarms:

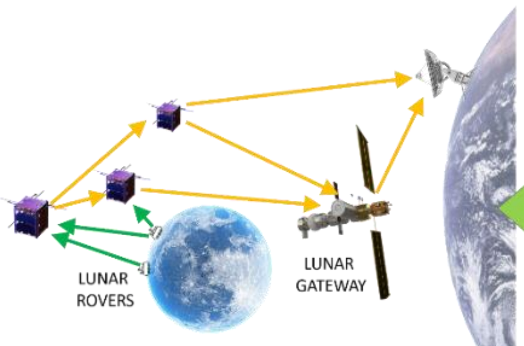
## Cognitive Cooperative Scheduling with Distributed NLPC Compression



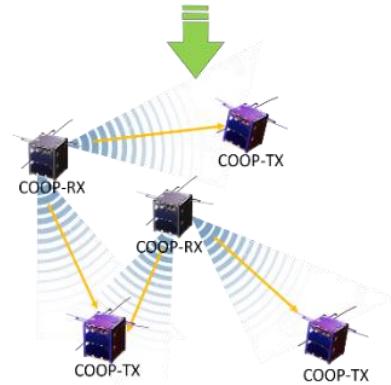

Each SmallSat predicts & determines whether to be a COOP-RX or a COOP-TX node




COOP-TX nodes broadcast Requests

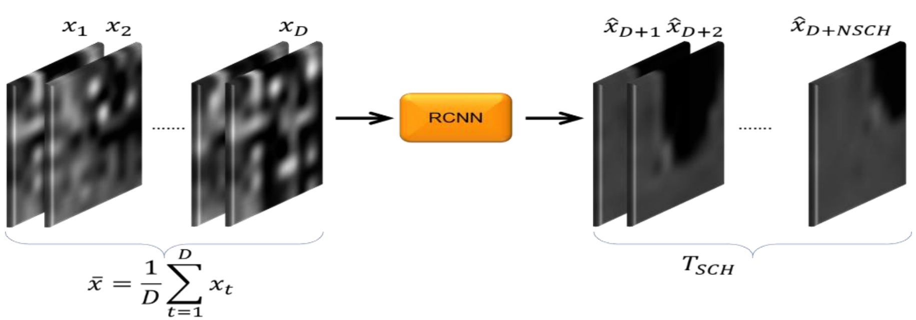


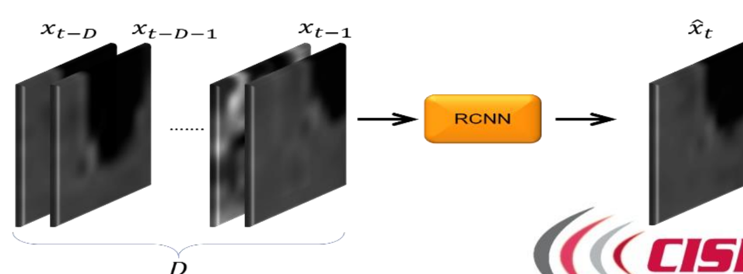

Execution :
SmallSats observe, preprocess and exchange data and relay to the earth either directly or via Lunar Gateway




COOP-TX nodes computes data acceptance schedules and broadcasts

- Deep learning based nonlinear predictive coding distributed data compression
- Distributed compression of data at sensor and cluster levels
- Cluster level cooperative relay scheduling using deep learning predictions
- Multi-objective optimization for data prioritization



# A Measure of Fairness

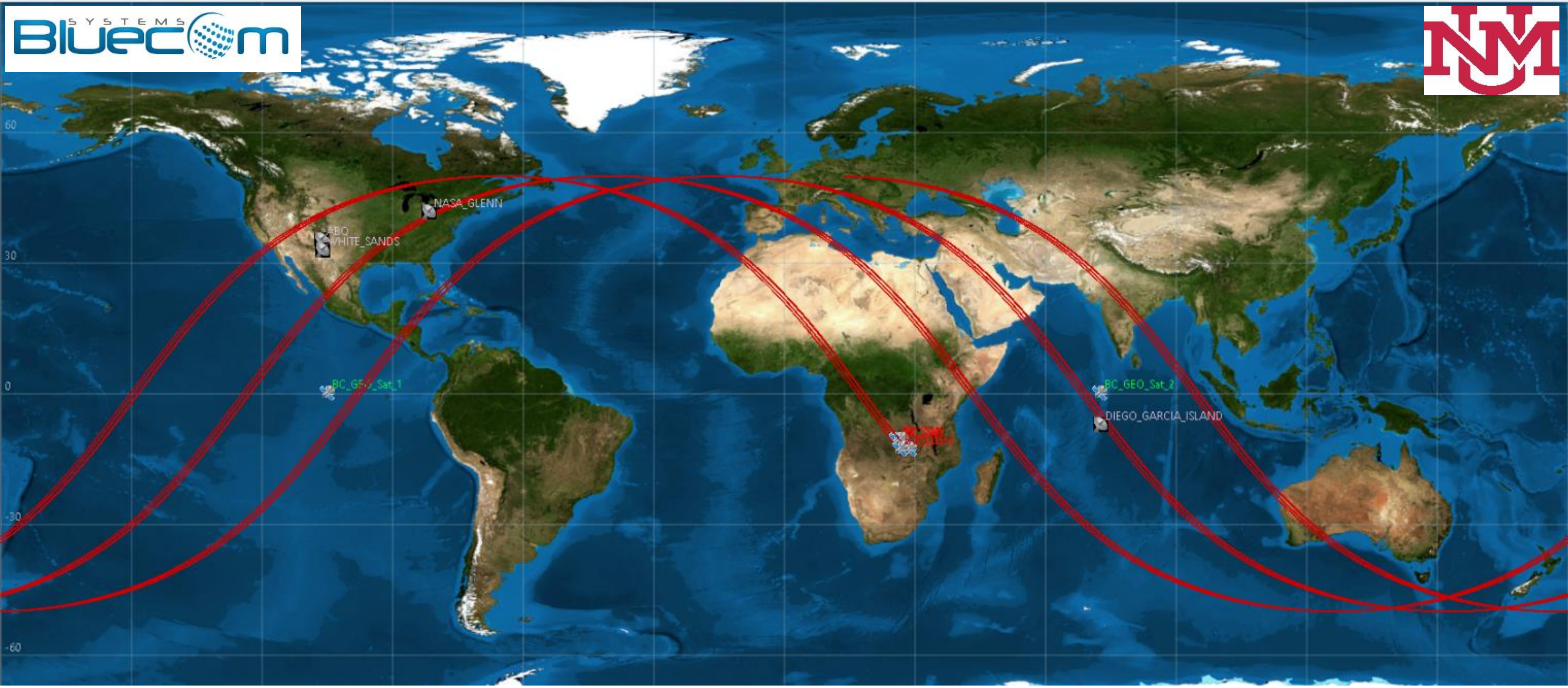
$$\begin{aligned}
\text{number of images generated by sat } i &= X_i \\
\text{number of delivered images from sat } i &= Y_i \\
\text{fraction of sat } i \text{ images delivered} &= \frac{Y_i}{X_i} \triangleq Z_i
\end{aligned}$$

## Jayne's Fairness Metric

$$\begin{aligned}
\text{Jain's Fairness Index} &= \frac{\left(\bar{\mathbf{Y}}\right)^2}{\bar{\mathbf{Y}^2}} = \frac{\left(\bar{\mathbf{Y}}\right)^2}{\mathrm{Var}(Y) + \left(\bar{\mathbf{Y}}\right)^2} \\
&= \frac{\left(\frac{1}{N}\sum_{i=1}^{N} y_i\right)^2}{\frac{1}{N}\sum_{i=1}^{N} y_i^2}
\end{aligned}$$

## Modified Jayne's Fairness Metric

$$\begin{aligned}
\text{Modified Jain's Fairness Index} &= \frac{\left(\bar{\mathbf{Z}}\right)^2}{\bar{\mathbf{Z}^2}} \\
&= \frac{\left(\frac{1}{N}\sum_{i=1}^{N} z_i\right)^2}{\frac{1}{N}\sum_{i=1}^{N} z_i^2} = \frac{\left(\frac{1}{N}\sum_{i=1}^{N} \frac{y_i}{x_i}\right)^2}{\frac{1}{N}\sum_{i=1}^{N} \left(\frac{y_i}{x_i}\right)^2}
\end{aligned}$$




## DL aided Cognitive Cooperative Scheduling in Cubesat Swarms

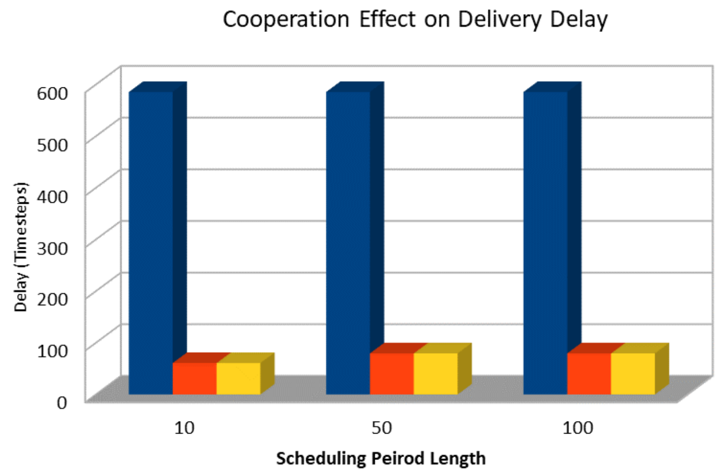
- Swarm of 5 LEO cubesats, 2 GEO relays and 2 Earth stations

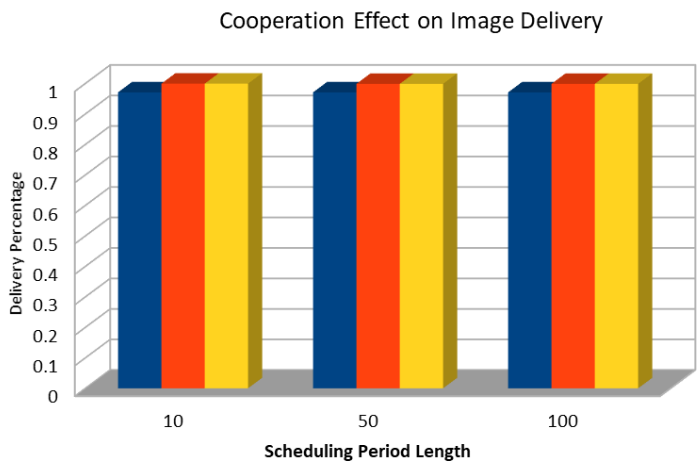
# DL aided Cognitive Cooperative Scheduling in Cubesat Swarms

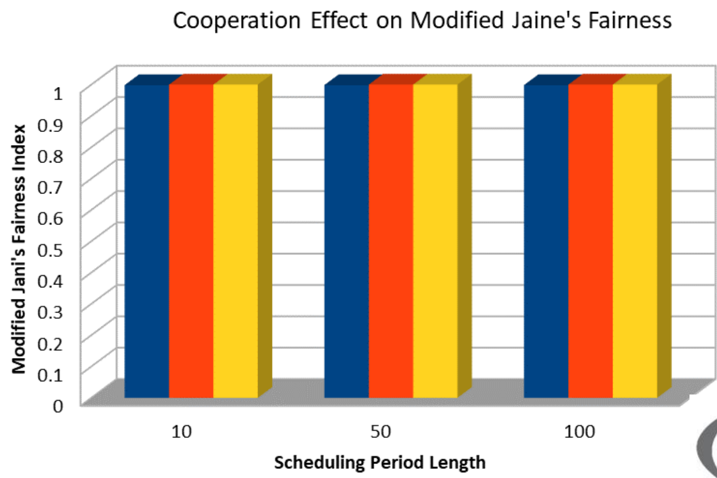
## Uniform Data and Link Capacities

- L2E Capacities: [1200, 1200, 1200, 1200, 1200]
- L2G Capacities: [1200, 1200, 1200, 1200, 1200]

BASELINE
DELAY
FAIRNESS



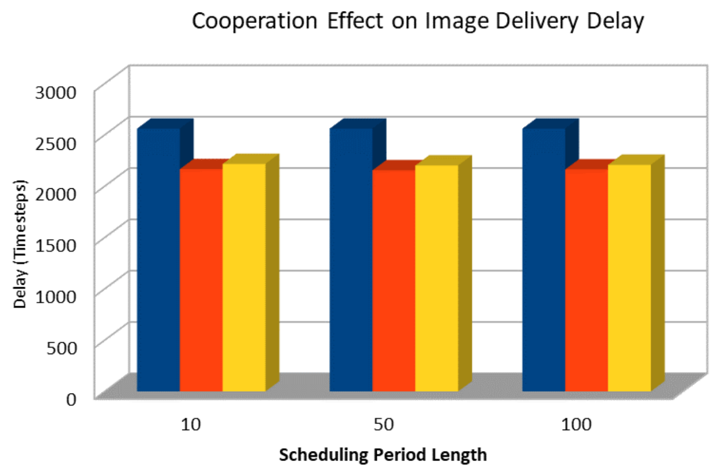



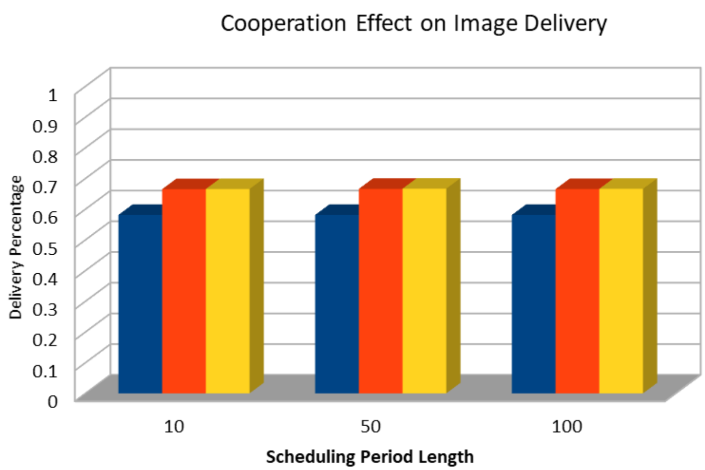



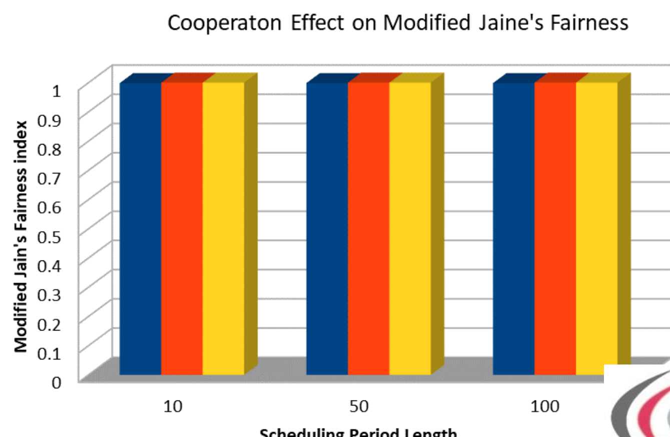
# DL aided Cognitive Cooperative Scheduling in Cubesat Swarms

## Poisson Data with Unequal Link Capacities

- Poisson parameters [2, 1, 1, 3, 3]
- L2E Capacities: [2400, 2400, 2400, 2400, 2400]
- L2G Capacities: [2400, 2400, 0, 2400, 0]

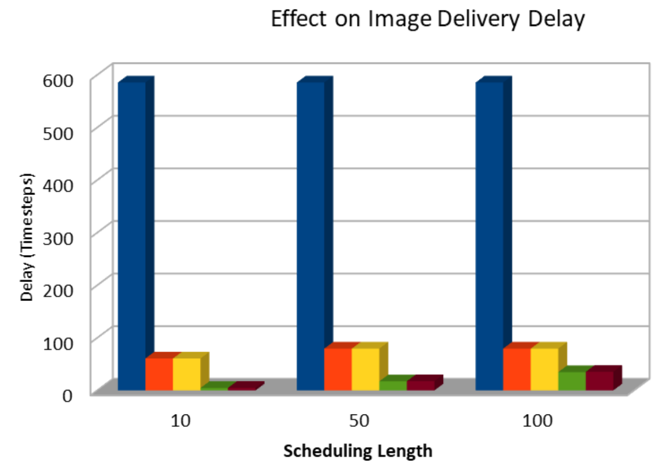
# DL aided Cognitive Cooperative Scheduling with Distributed NLPC Compression in Cubesat Swarms

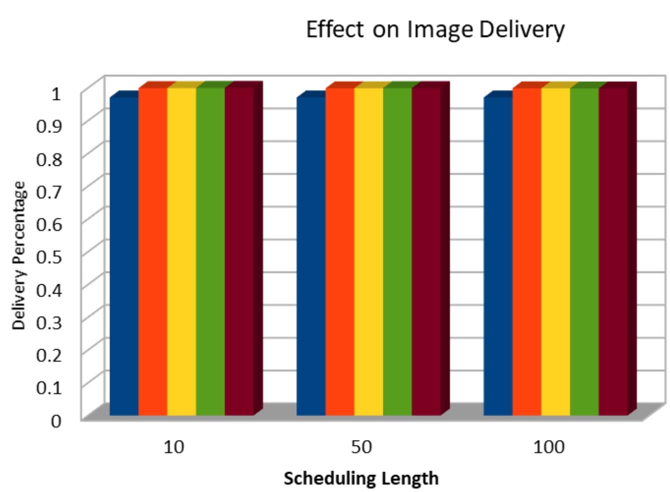
## Uniform Data and Link Capacities

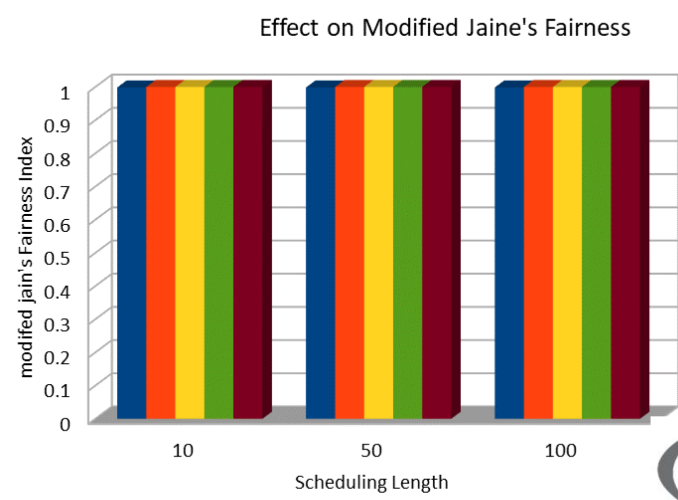
- L2E Capacities: [1200, 1200, 1200, 1200, 1200]
- L2G Capacities: [1200, 1200, 1200, 1200, 1200]

Swarm of
5 LEOs, 2 GEO relays &
2 Earth stations

BASELINE
DELAY
FAIRNESS
COMP + DELAY
COMP + FAIRNESS



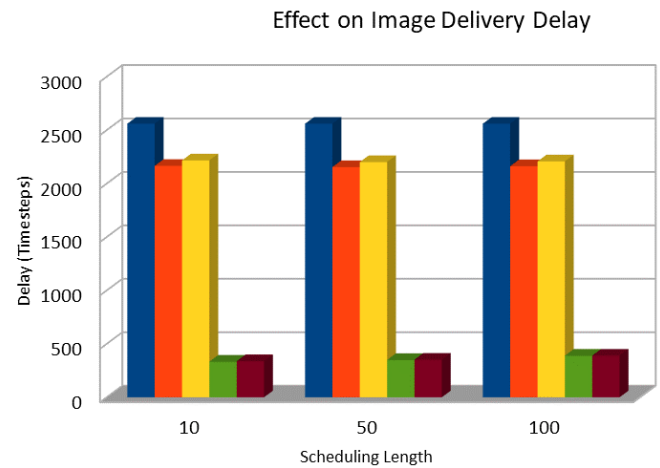
# DL aided Cognitive Cooperative Scheduling with Distributed NLPC Compression in Cubesat Swarms

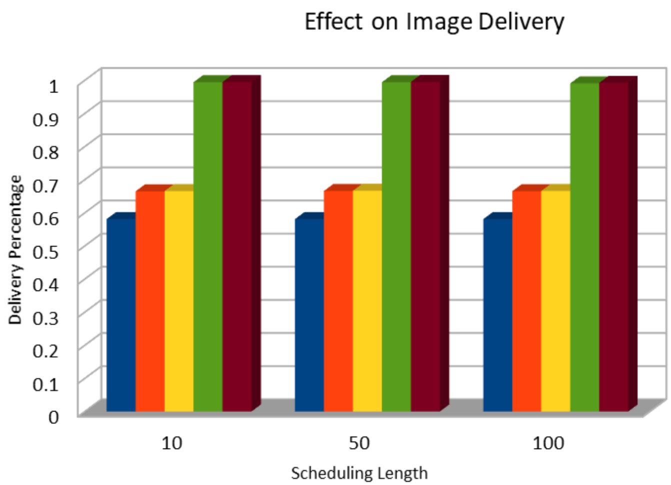
## Poisson Data with Unequal Link Capacities

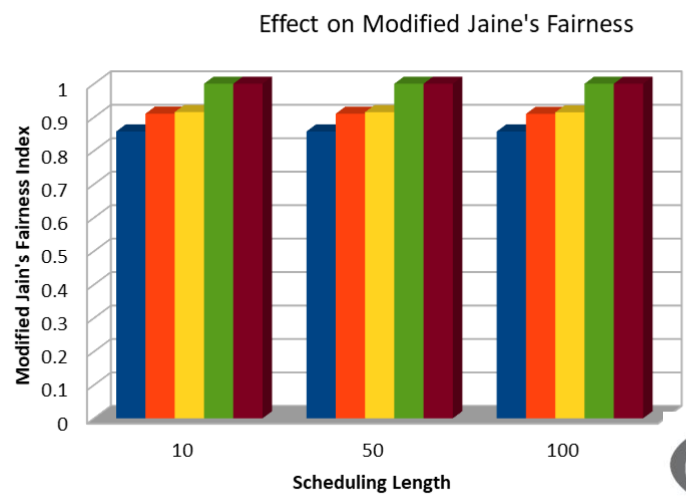
- Poisson parameters [2, 1, 1, 3, 3]
- L2E Capacities: [2400, 2400, 2400, 2400, 2400]
- L2G Capacities: [2400, 2400, 0, 2400, 0]

Swarm of
5 LEOs, 2 GEO relays &
2 Earth stations

BASELINE
DELAY
FAIRNESS
COMP + DELAY
COMP + FAIRNESS



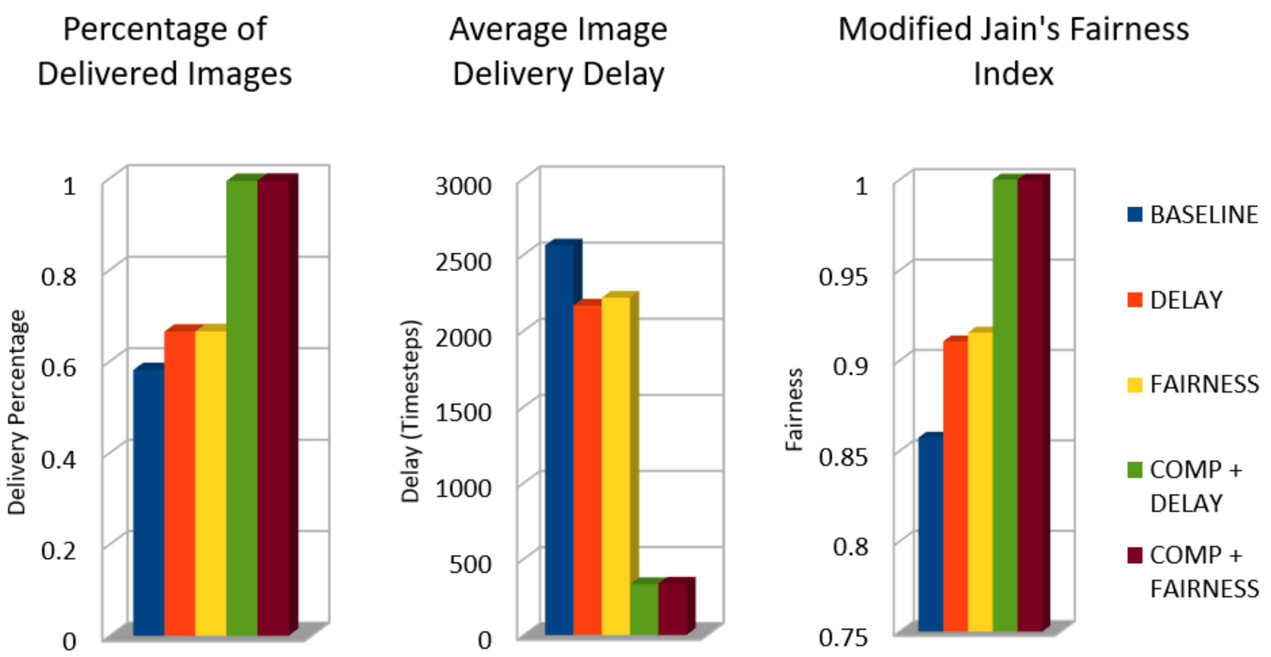



# Summary Performance

**Poisson Data with Unequal Link Capacities**

- Poisson parameters [2, 1, 1, 3, 3]
- L2E Capacities: [2400, 2400, 2400, 2400, 2400]
- L2G Capacities: [2400, 2400, 0, 2400, 0]

Swarm of
5 LEOs, 2 GEO relays &
2 Earth stations



# Thank you.