# AI FOR SCIENCE

## NUMERICAL WEATHER PREDICTION - OVERVIEW

David Hall Senior Solutions Architect, NVIDIA GTC March 2019 dhall@nvidia.com

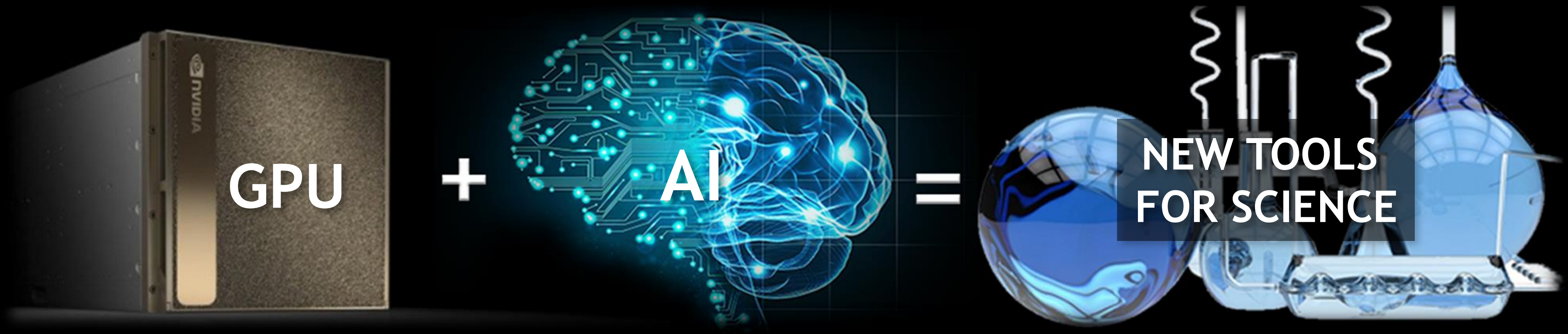
# OVERVIEW

NVIDIA GPUs are powering modern supercomputers

Using them effectively is increasingly important

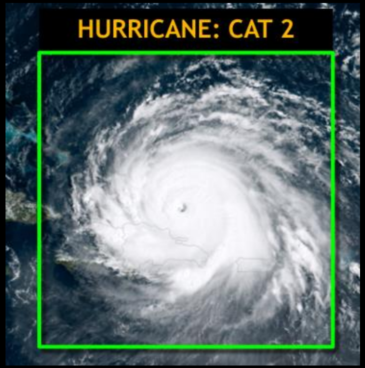
Modern AI is a perfect fit for GPUs

AI + GPUs provides a powerful new set of tools for science

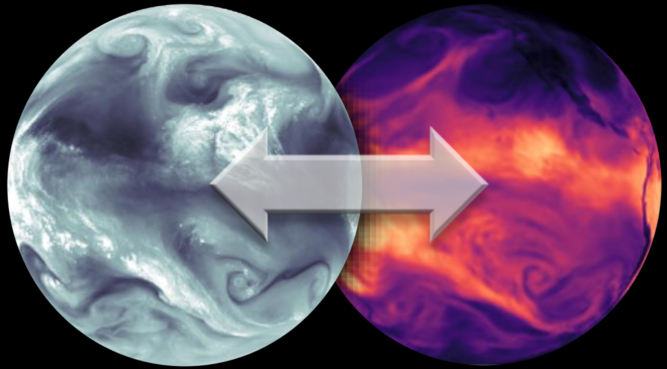


# OVERVIEW

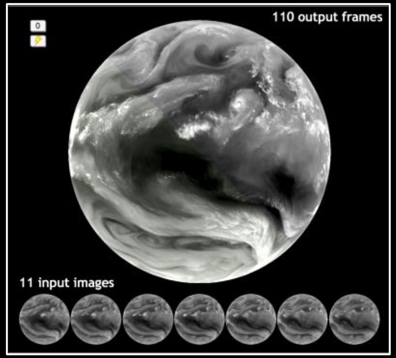
## DETECTION
Tropical Storm Detection



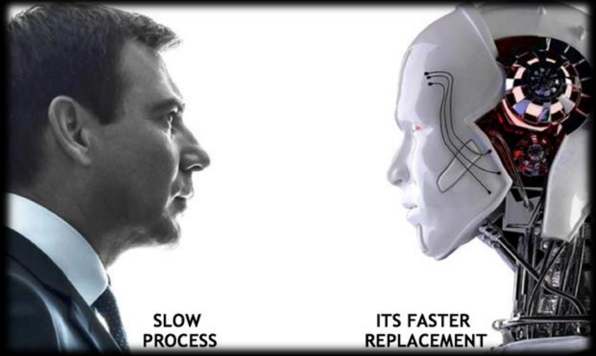

## TRANSLATION
Inverse Modeling for Data Assimilation



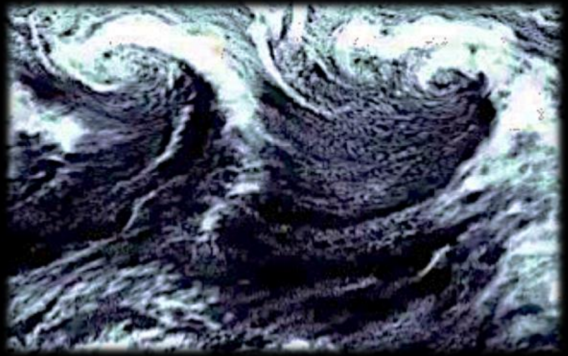
## ENHANCEMENT
Slow Motion Satellite Loop




## EMULATION
Model Acceleration Without Porting




## PARAMETRIZATION
More Accurate Physics from Data

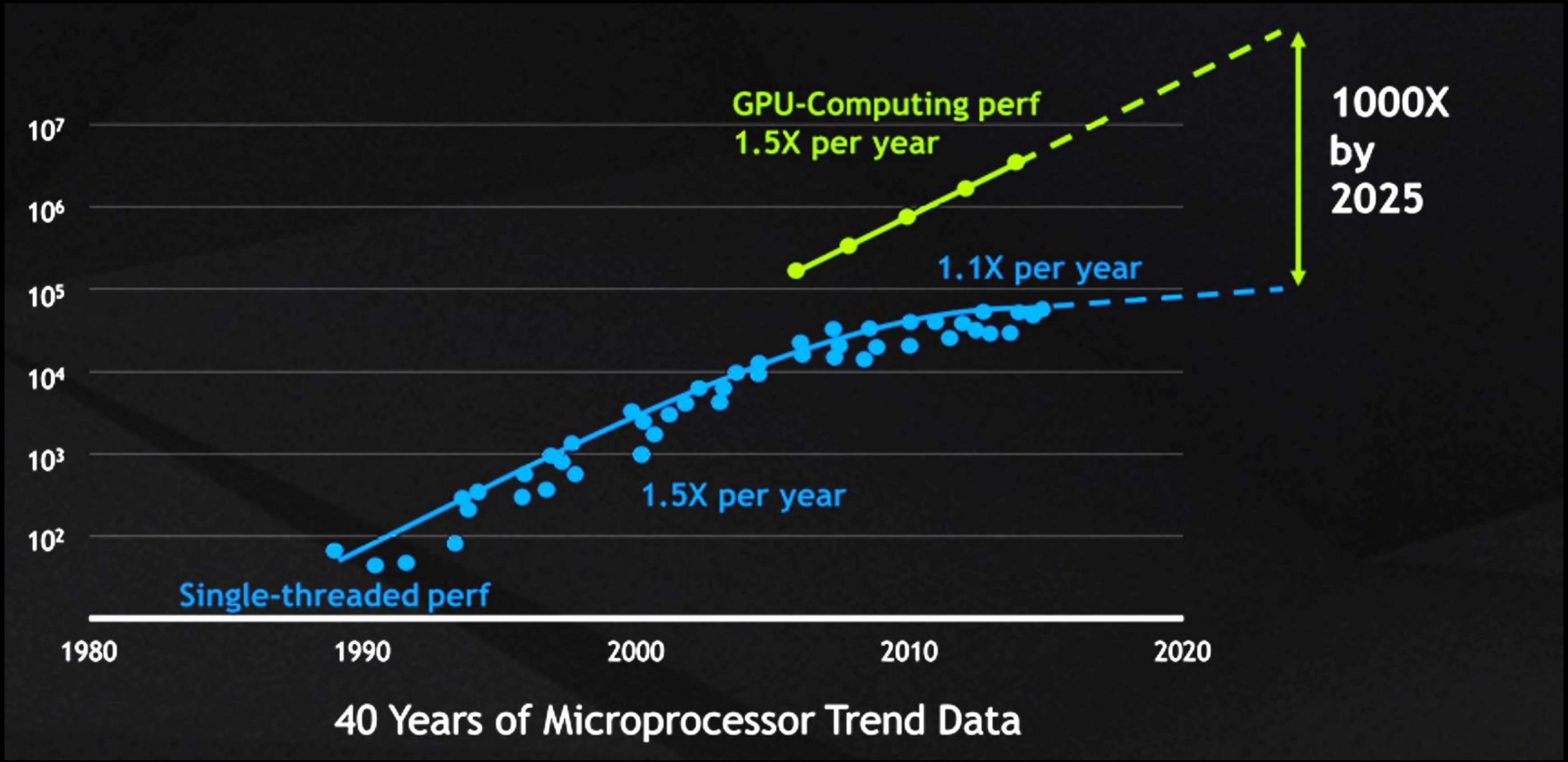


# ARTIFICIAL INTELLIGENCE ON GPUS

CPU performance growth has stalled and NVIDIA GPUs are powering current and next generation supercomputers. It is important for researchers and practitioners to learn to use these resources effectively. Artificial intelligence is a natural solution. It makes effective use of GPUs and has the potential to improving all aspects of scientific computing.

# GPUS ARE DRIVING PERFORMANCE GROWTH

## The performance gap between CPUs and GPUs is growing rapidly




- Dennard scaling has come to an end
- CPU growth has slowed to 10% per year
- GPU performance is growing at 150% per year
- 1000x performance gap projected by 2025

# MODERN SUPERCOMPUTERS ARE GPU MACHINES

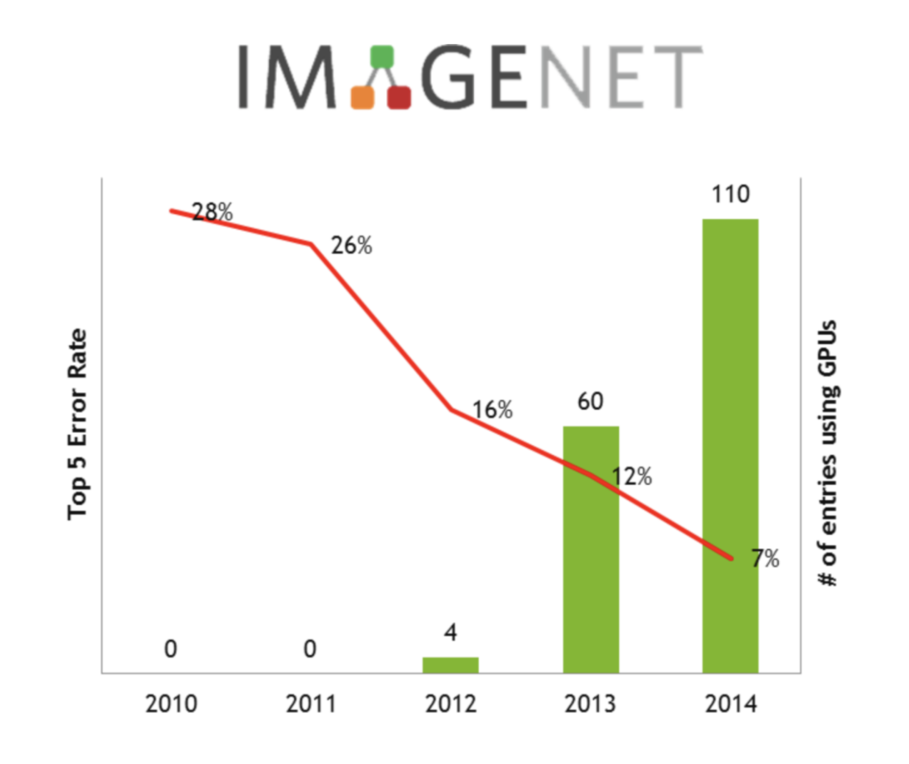
## Most high end supercomputers are loaded with NVIDIA Volta GPUs


| Rank | System | Cores | Rmax (TFlop/s) | Rpeak (TFlop/s) | Power (kW) |
|---|---|---|---|---|---|
| 1 | **Summit** - IBM Power System AC922, IBM POWER9 22C 3.07GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM<br>DOE/SC/Oak Ridge National Laboratory<br>United States | 2,397,824 | 143,500.0 | 200,794.9 | 9,783 |
| 2 | **Sierra** - IBM Power System S922LC, IBM POWER9 22C 3.1GHz, NVIDIA Volta GV100, Dual-rail Mellanox EDR Infiniband , IBM / NVIDIA / Mellanox<br>DOE/NNSA/LLNL<br>United States | 1,572,480 | 94,640.0 | 125,712.0 | 7,438 |
| 3 | **Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC<br>National Supercomputing Center in Wuxi<br>China | 10,649,600 | 93,014.6 | 125,435.9 | 15,371 |
| 4 | **Tianhe-2A** - TH-IVB-FEP Cluster, Intel Xeon E5-2692v2 12C 2.2GHz, TH Express-2, Matrix-2000 , NUDT<br>National Super Computer Center in Guangzhou<br>China | 4,981,760 | 61,444.5 | 100,678.7 | 18,482 |
| 5 | **Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc.<br>Swiss National Supercomputing Centre (CSCS)<br>Switzerland | 387,872 | 21,230.0 | 27,154.3 | 2,384 |

- Supercomputing centers recognize the advantage of GPUS
- Most high end supercomputers are now GPU machines
- This trend is likely to continue
- Important to learn to use GPUs effectively

# AI IS PERFECTLY SUITED FOR GPUS

## ImageNet 2012: A Revolution in Computer Vision




### ImageNet Classification with Deep Convolutional Neural Networks

**Alex Krizhevsky**
University of Toronto
kriz@cs.utoronto.ca

**Ilya Sutskever**
University of Toronto
ilya@cs.utoronto.ca

**Geoffrey E. Hinton**
University of Toronto
hinton@cs.utoronto.ca

**Abstract**

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

- Luckily, AI is a perfect fit for GPUs
- Alex Krizhevsky demonstrated this in 2012 @ Imagenet

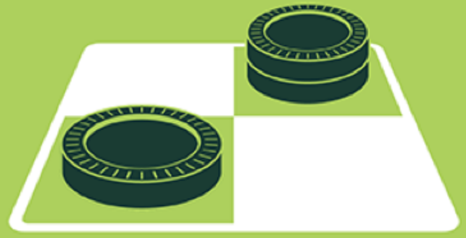
- His simple DNN defeated the best expert coded solutions
- Deep learning has been growing like wildfire since

# THREE ROADS TO AI

## There are three main flavors of AI, and each can be GPU accelerated

ARTIFICIAL INTELLIGENCE

**EXPERT SYSTEMS**
EXECUTE HAND-WRITTEN ALGORITHMS AT HIGH SPEED



Accelerate with

**GPU accelerated Libraries**

**OpenACC Directives**

**CUDA Kernels**

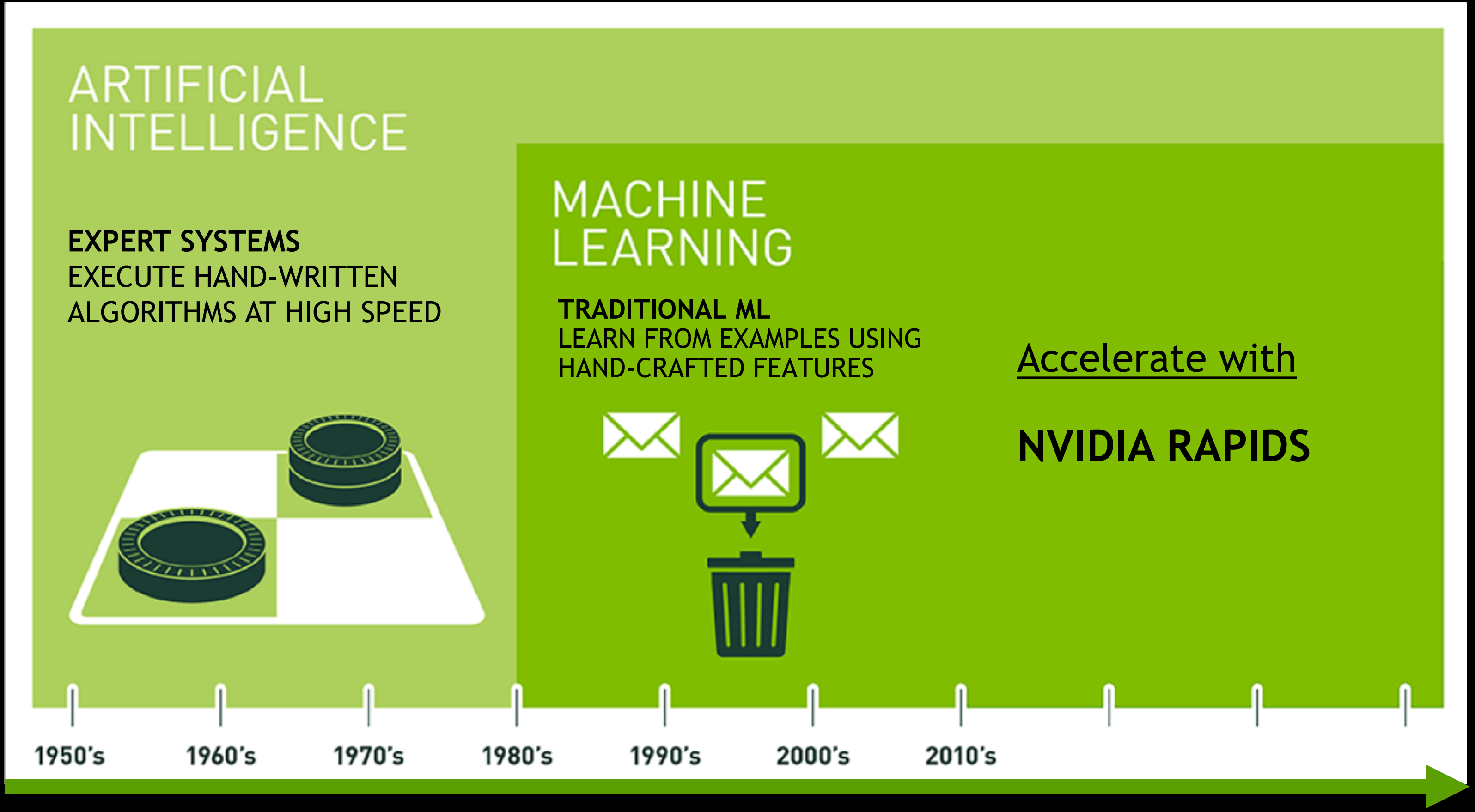
1950's 1960's 1970's 1980's 1990's 2000's 2010's

**INCREASING COMPLEXITY AND AUTONOMY OVER TIME**

- There are 3 main types of AI
- Expert systems accelerated through libraries, OpenACC, CUDA
- ML is accelerated with NVIDIA's RAPIDS
- DL is accelerated via cuDNN in most DL frameworks

# THREE ROADS TO AI

## There are three main flavors of AI, and each can be GPU accelerated

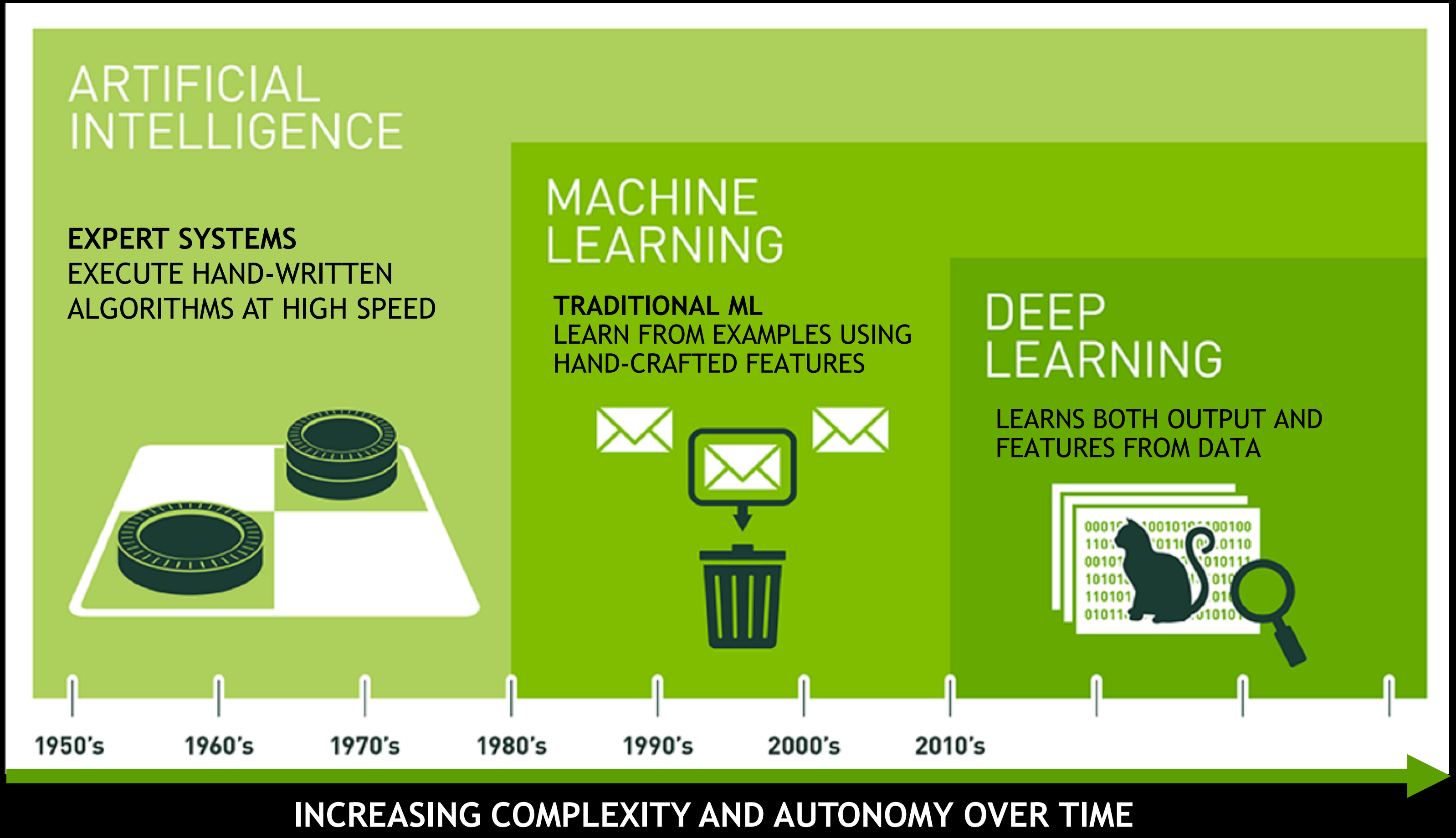



INCREASING COMPLEXITY AND AUTONOMY OVER TIME

- There are 3 main types of AI
- Expert systems accelerated through libraries, OpenACC, CUDA
- ML is accelerated with NVIDIA's RAPIDS
- DL is accelerated via cuDNN in most DL frameworks

# THREE ROADS TO AI

There are three main flavors of AI, and each can be GPU accelerated




- There are 3 main types of AI
- Expert systems accelerated through libraries, OpenACC, CUDA
- ML is accelerated with NVIDIA's RAPIDS
- DL is accelerated via cuDNN in most DL frameworks

# EXPERT SYSTEM
## GARY KASPAROV VS DEEP BLUE 1997

Deep Blue: an expert system for playing chess

Experts hand-coded heuristics for pieces and positions

High speed search enabled super-human performance

Defeated world chess champion in 1997

# DEEP LEARNING
## LEE SEDOL
## VS ALPHA-GO 2016

Go is much too large to be beaten by brute force.

A game of human intuition

Unbeatable by machines…

AlphaGo: Deep reinforcement learning and self competition

Defeated top world Go champions in 2016-2017

Also world champion in Chess and Shogi

# NWP IS AN EXPERT SYSTEM

## Expert knowledge encoded as software, executed at high speed.

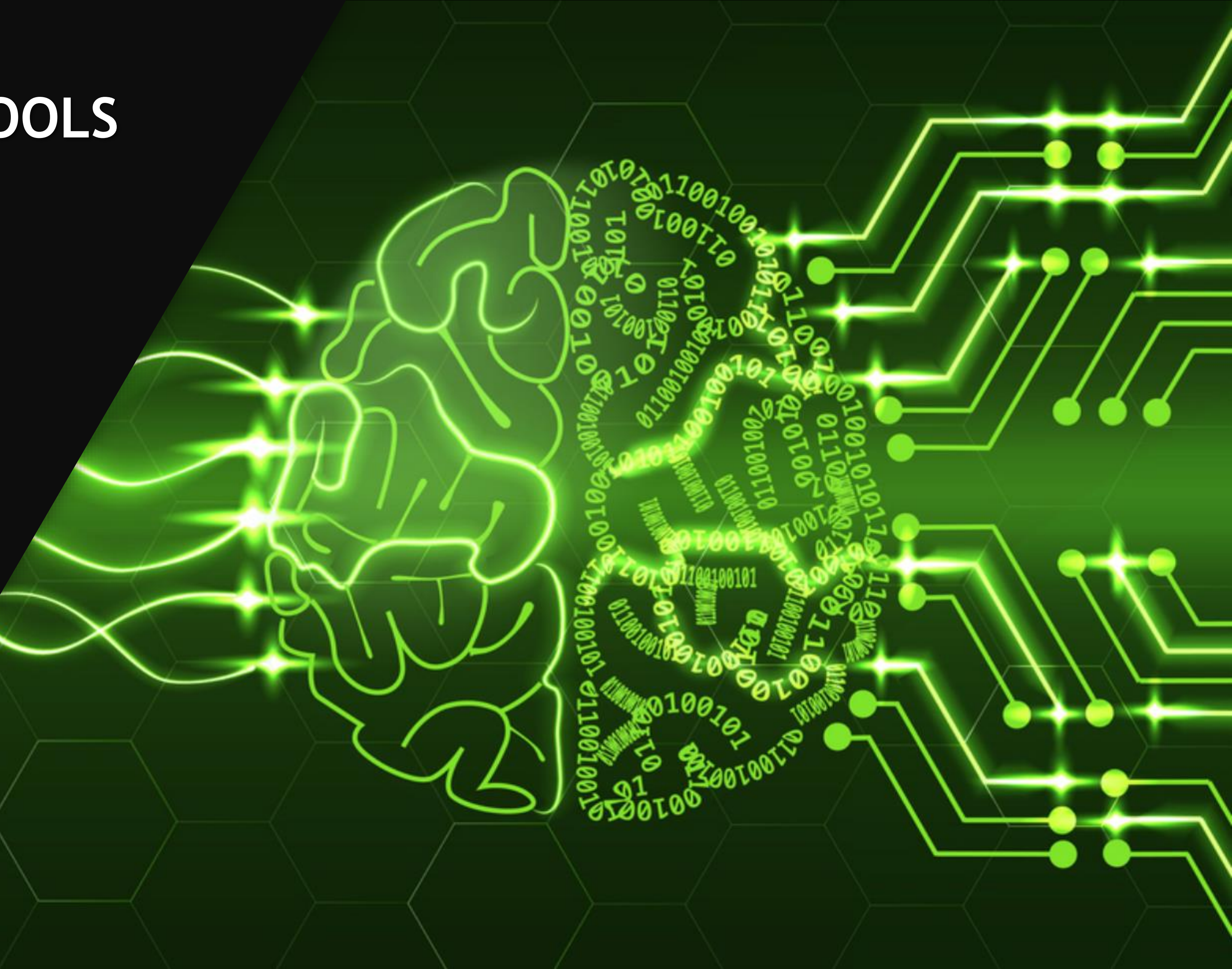

COLLECTION THINNING ASSIMILATION DYNAMICS PARAMETRIZATION FORECASTING

- Encodes knowledge of experts as algorithms
- So familiar, most people don't think of it as AI

- Deep learning provides a new set of tools
- All stages of NWP may be augmented by deep learning

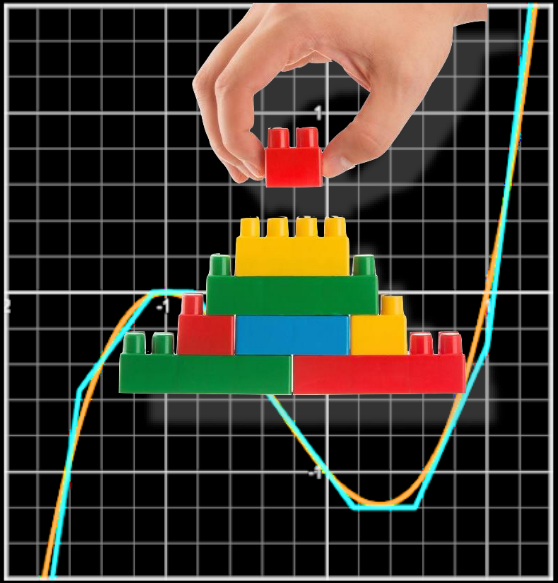
# DEEP LEARNING: A NEW SET OF TOOLS FOR SCIENCE

Deep learning provides a new approach for building complex software components, by constructing functions automatically from a large set of examples. This approach complements traditional algorithm development, providing a means of devising algorithms too complex, subtle, or unintuitive to code by hand.

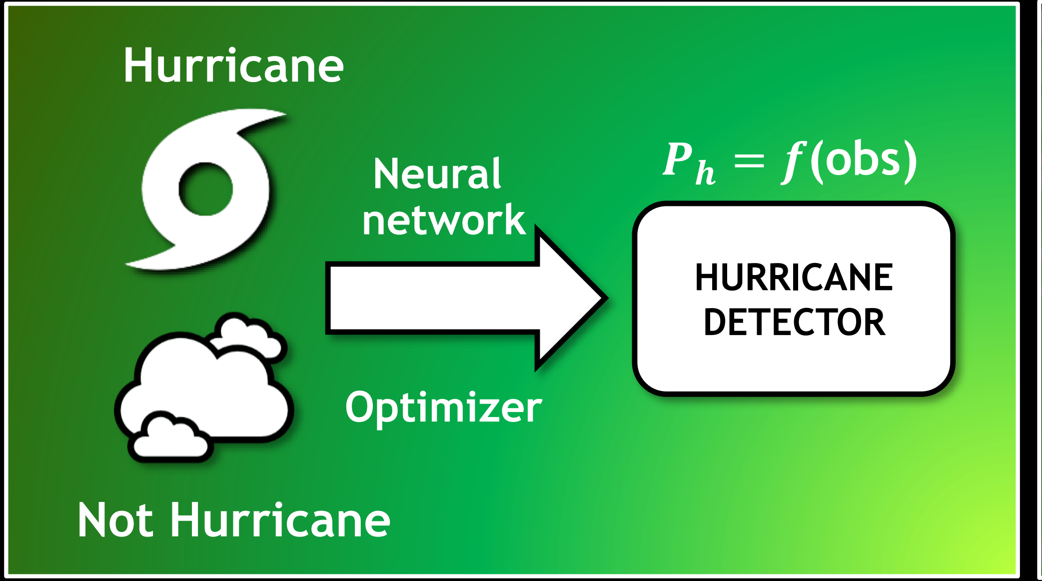


# SOFTWARE BY EXAMPLE

## Supervised deep Learning builds functions from input/output pairs



Functions are the building blocks of software. DL can approximate any function.



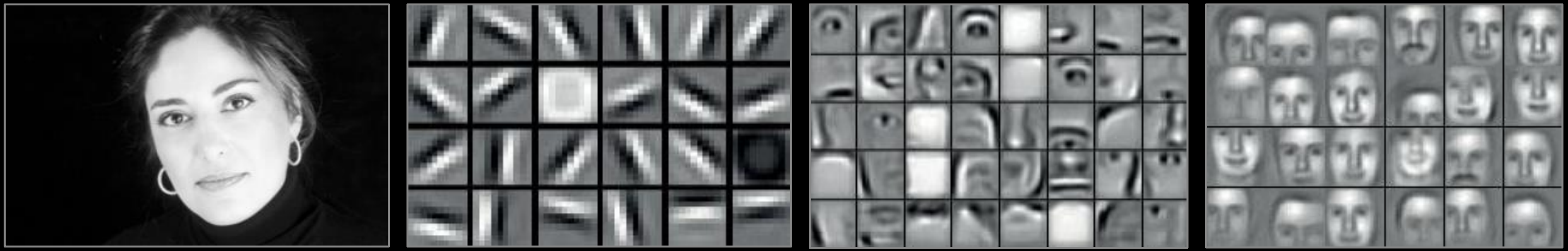

Some functions are too challenging to code by hand. DL builds complex functions from a set of examples.

Mix freely with conventional software and algorithms

# DL LEARNS FEATURES FROM DATA

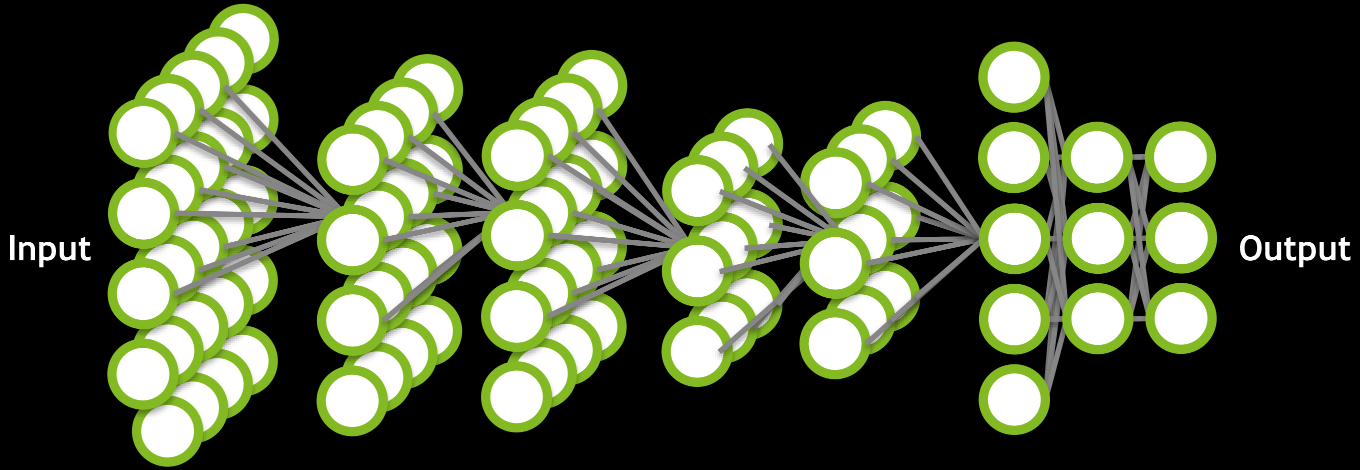
## Deep learning automatically finds feature hierarchies

Input data (pixels values) | low-level features | mid-level features | high-level features | output



$P_{face}$

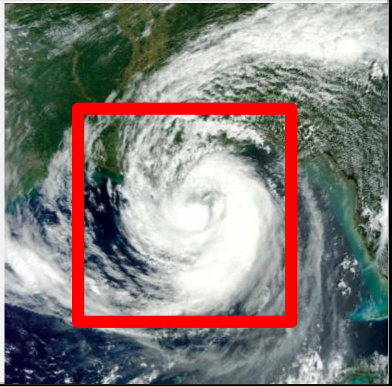
- Example: face detection
- Learns lines, noses, faces
- Returns $P_{face} = F(\text{pixels})$
- Greater depth → greater abstraction
- 1000s of subtly different feature detectors
- Different data produces a different algorithm



## DETECTION

- **Tropical storms**

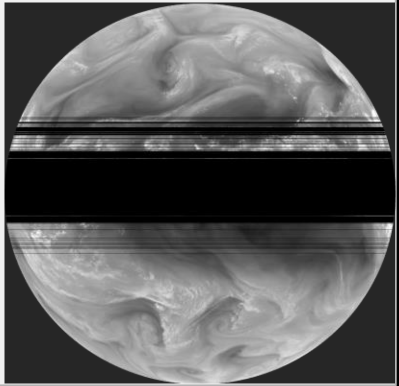
Extra-tropical cyclones
Atmospheric rivers
Cyclogenesis events
Convection initiation
Change detection



## ENHANCEMENT

Frame repair
Sequence repair
- **Slow motion**

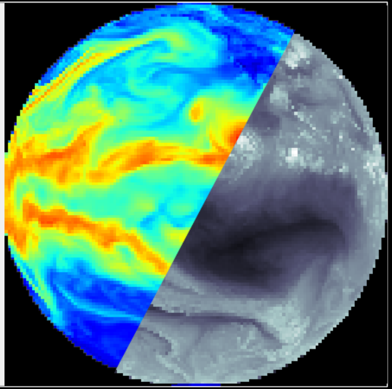
Super-resolution
Cloud removal
Data augmentation



## TRANSLATION

- **Data Assimilation**

Forecast verification
Model inter-comparison
Common data formatting
Colorization
Digital Elevation from Imagery



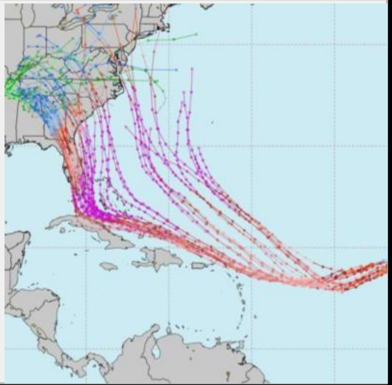
## EMULATION

Physics Acceleration
Turbulence
- **Radiation**

Convection
Microphysics
Dynamics Acceleration

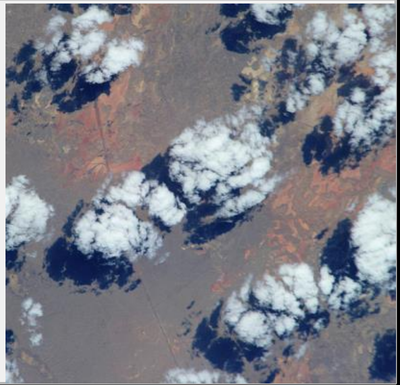
## PREDICTION

Uncertainty prediction
Storm track
Storm intensity
Fluid motion
Now casting
Satellite frame prediction



## PARAMETRIZATION

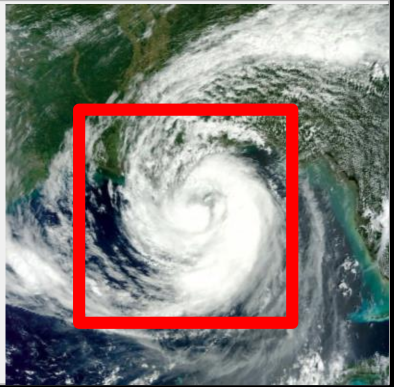
New parametrizations
From higher resolution model
- **From observational data**



## DETECTION

- **Tropical storms**

Extra-tropical cyclones
Atmospheric rivers
Cyclogenesis events
Convection initiation
Change detection

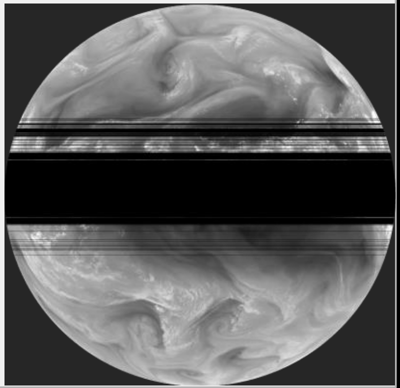


## ENHANCEMENT

Frame repair
Sequence repair

- **Slow motion**

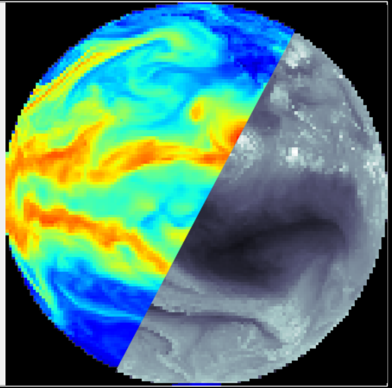
Super-resolution
Cloud removal
Data augmentation



## TRANSLATION

- **Data Assimilation**

Forecast verification
Model inter-comparison
Common data formatting
Colorization
Digital Elevation from Imagery



## EMULATION

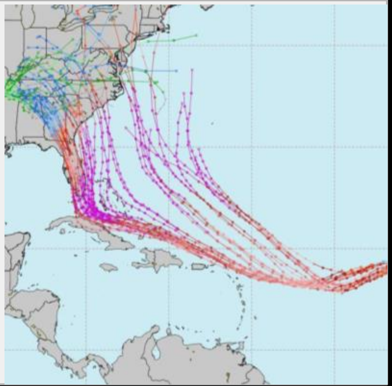
Physics Acceleration
Turbulence

- **Radiation**

Convection
Microphysics
Dynamics Acceleration

## PREDICTION

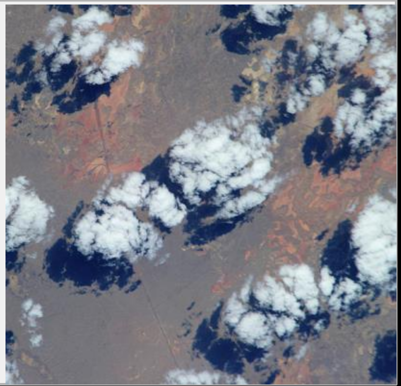
Uncertainty prediction
Storm track
Storm intensity
Fluid motion
Now casting
Satellite frame prediction



## PARAMETRIZATION

New parametrizations
From higher resolution model

- **From observational data**



# SELECTED DEEP LEARNING EXAMPLES

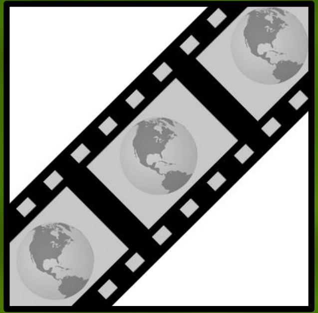
REGION OF INTEREST DETECTION

DATA THINNING

DATA-TO-DATA TRANSLATION

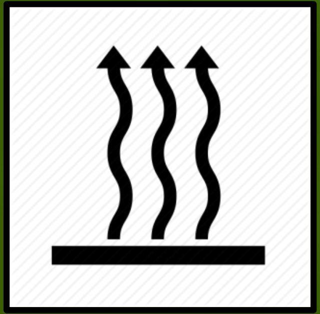
DATA ASSIMILATION

SLOW MOTION ENHANCEMENT



ERROR CORRECTION

CRTM EMULATION



ACCELERATION

SOIL MOISTURE PARAMETRIZATION

BETTER PHYSICS

# 1. STORM DETECTION: AI ASSISTED DATA ANALYSIS

The quantity of data produced by models, satellites and other sensors has become impractical to analyze manually. AI can help by detecting important features, tends, and anomalies. Applications include storm tracking, data thinning, advanced warning systems, search and rescue, route planning, and more.

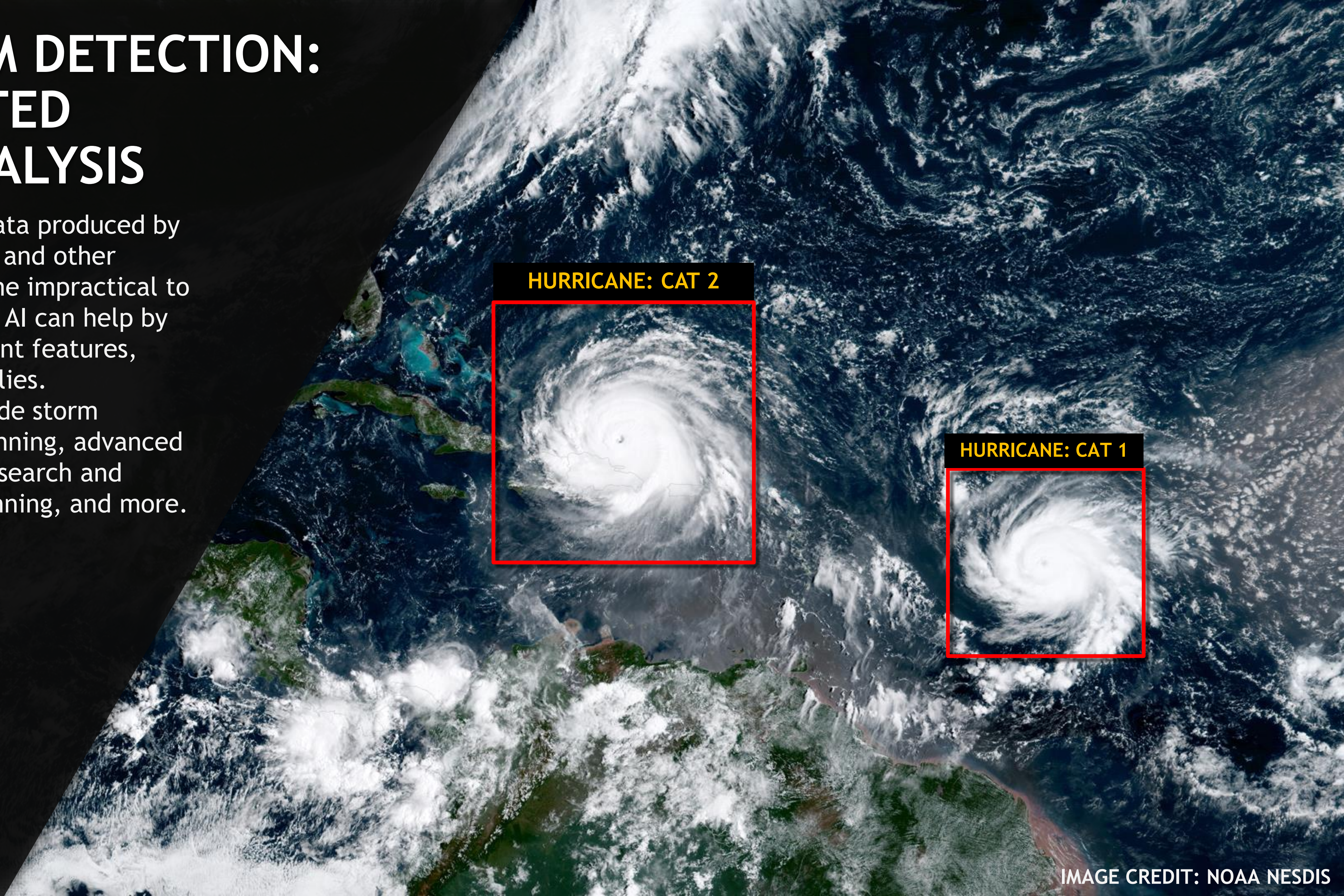



IMAGE CREDIT: NOAA NESDIS

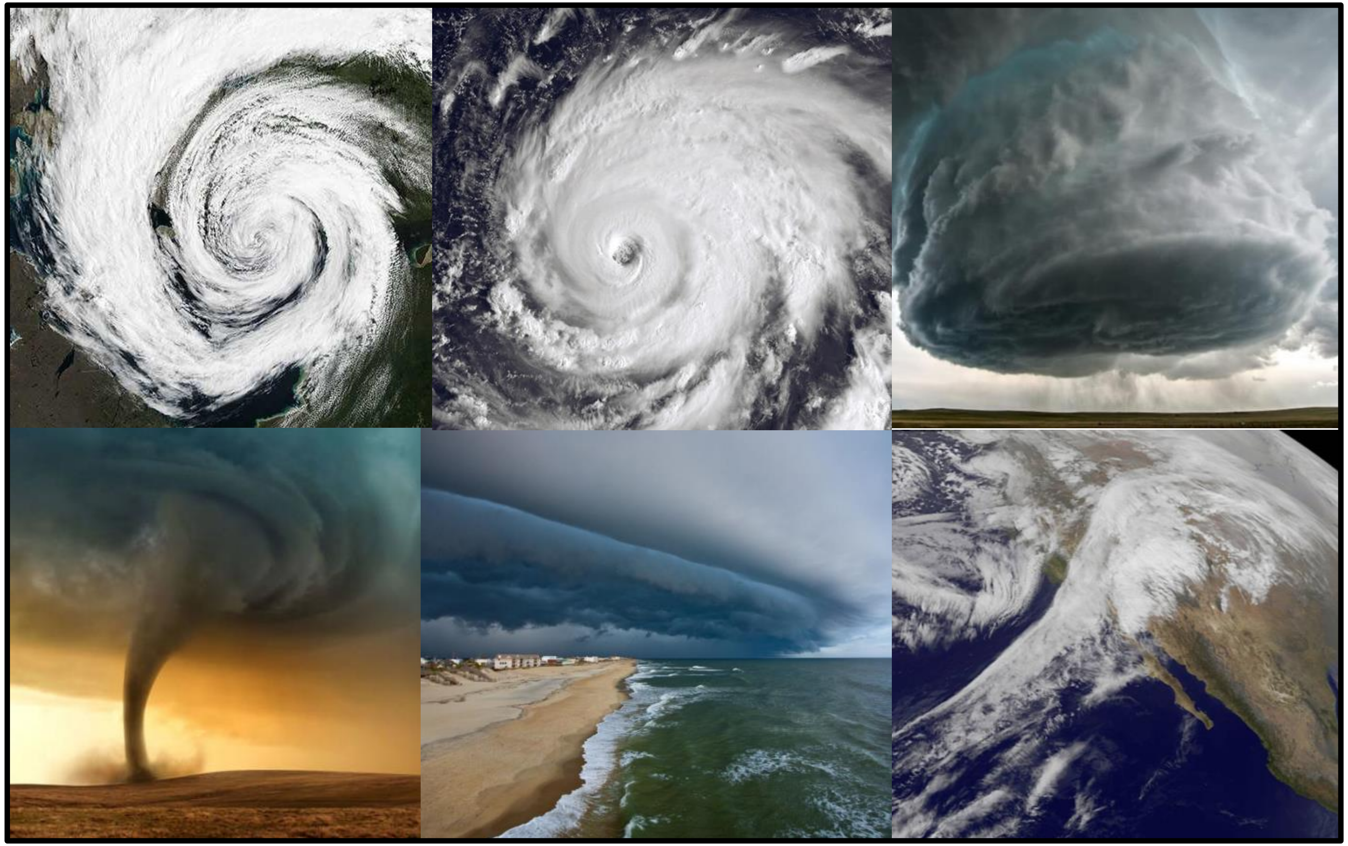
# STORM DETECTION

## Automatically locate and classify significant weather events

Some events have a large impact on the weather

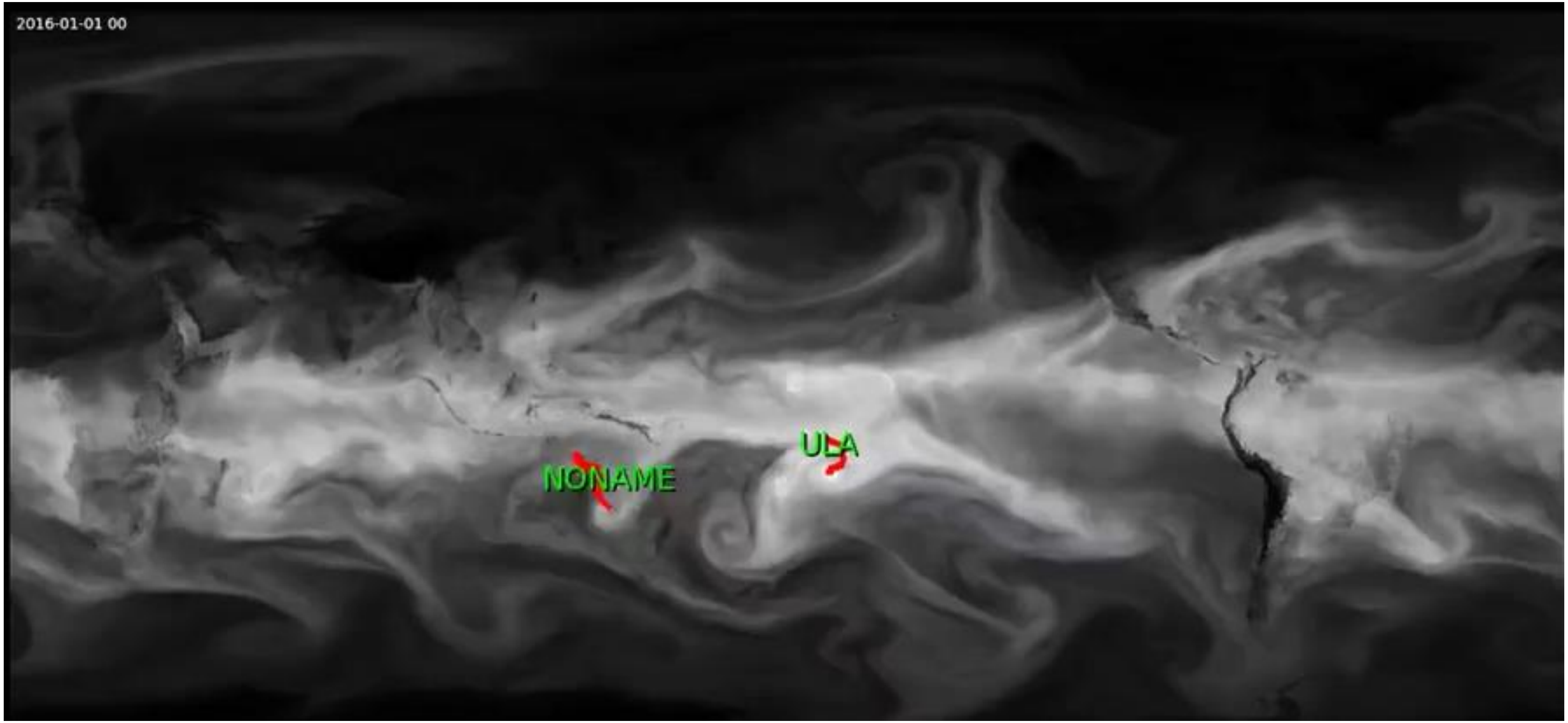
Detect such events automatically

- Tropical Cyclones
- Extra-tropical cyclones
- Atmospheric Rivers
- Storm Fronts
- Convection Initiation
- Cyclogenesis



# LOCATE KNOWN STORMS

## Use expert labeled IBTrACS database to locate storms in model data

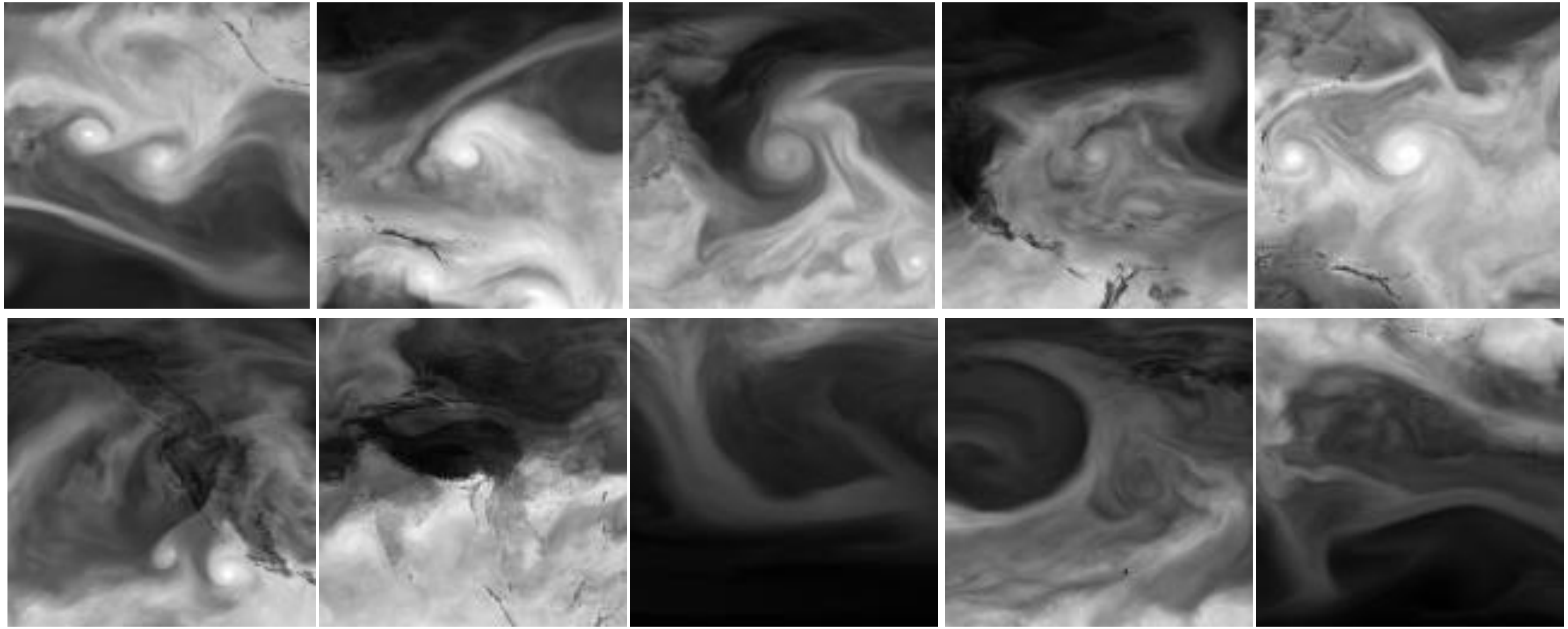


# EXTRACT TRAINING AND TEST EXAMPLES

## Extract storm/no-storm examples for supervised learning

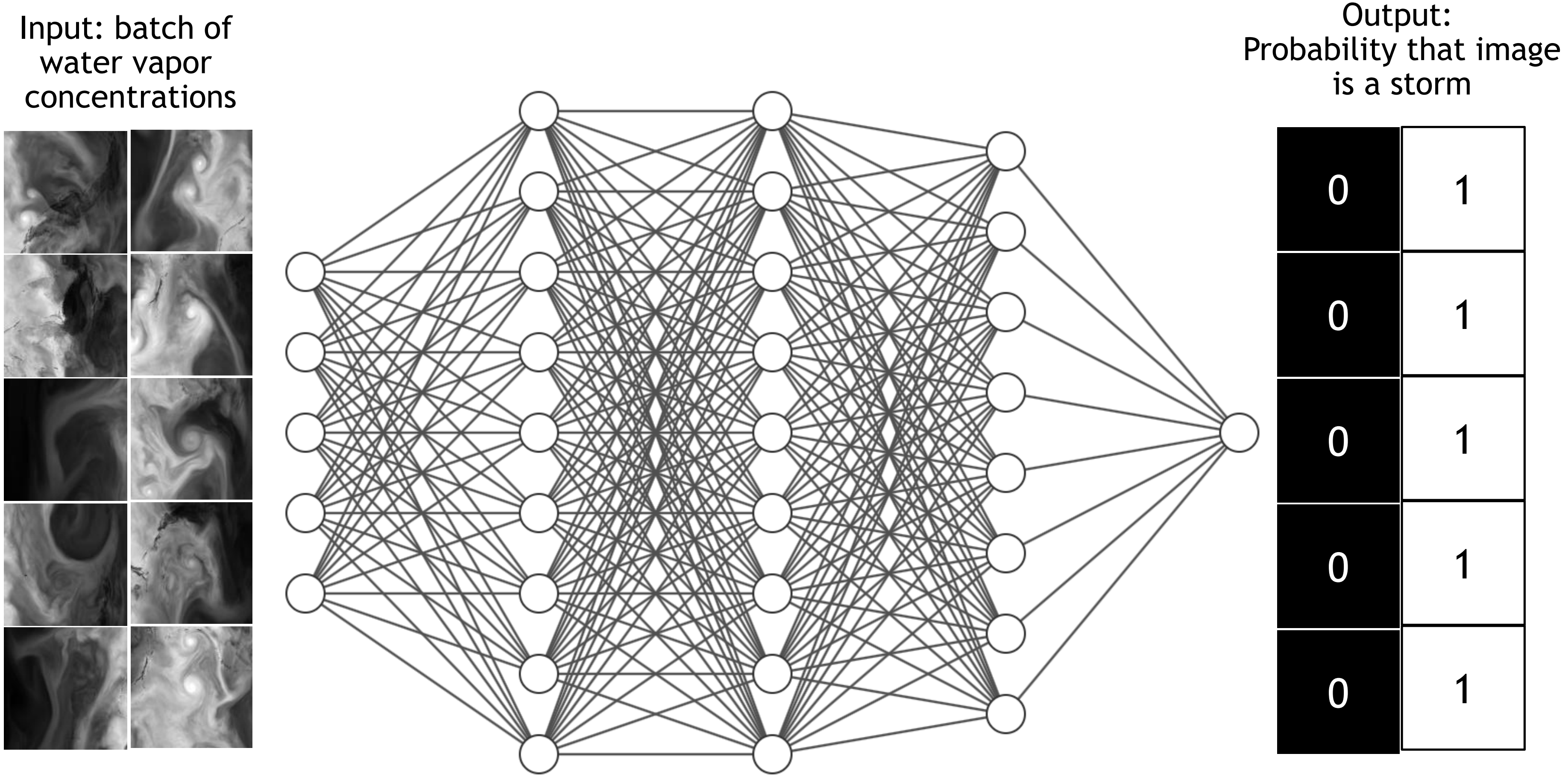
Positive Examples

Negative Examples

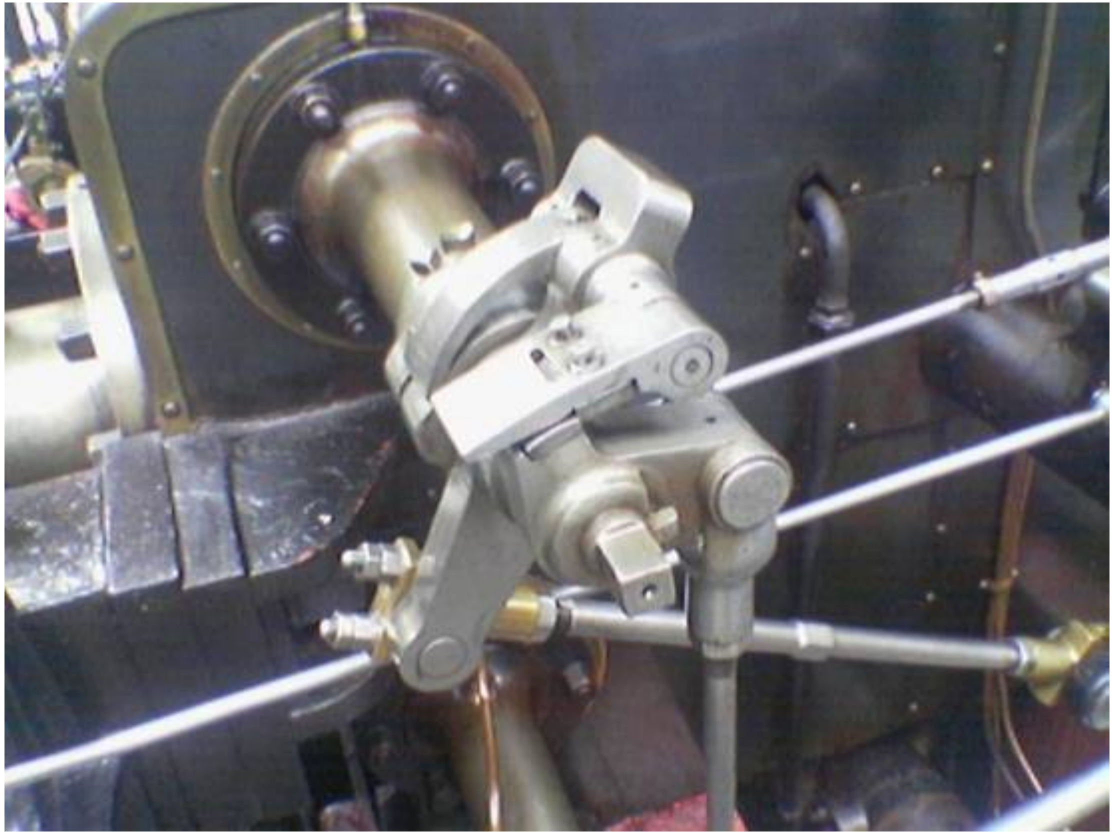


# TRAIN: SEARCH FOR FUNCTION THAT FITS THE DATA

## Training phase



# CONVOLUTION EXAMPLE: SOBEL FILTER



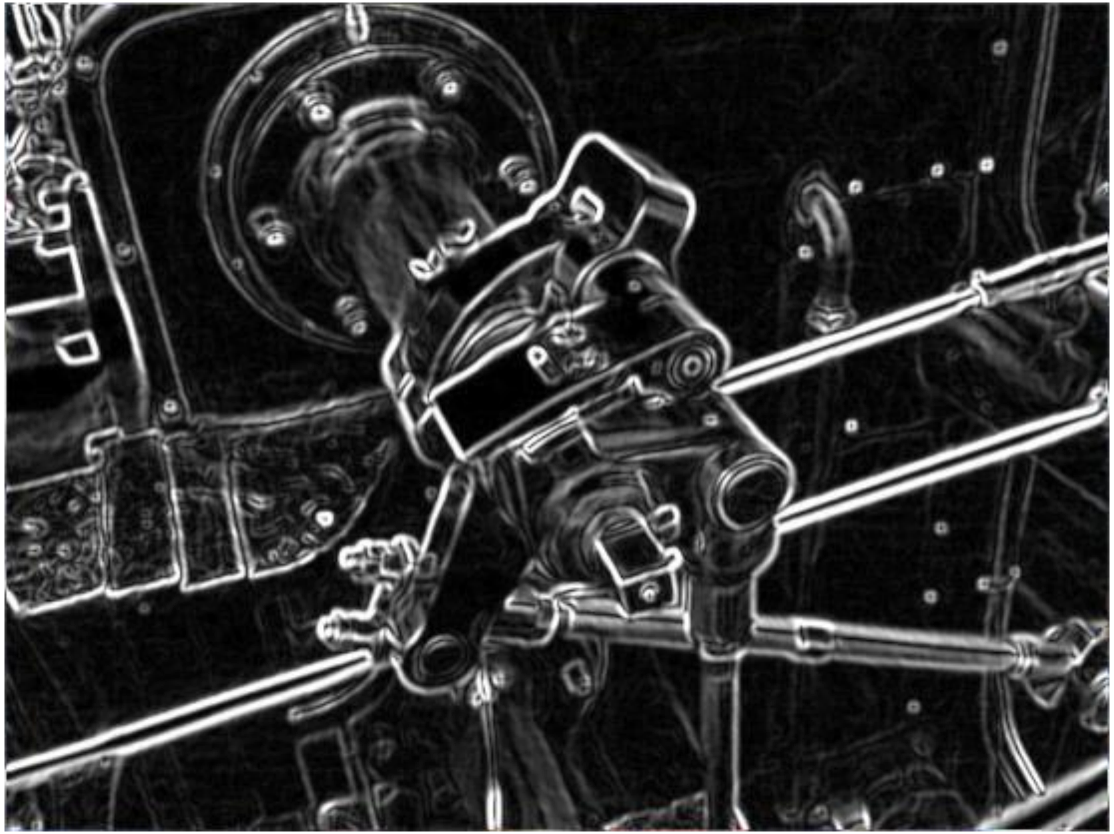
$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

Image source: https://en.wikipedia.org/wiki/Sobel_operator

# CONVOLUTION EXAMPLE: SOBEL FILTER



$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

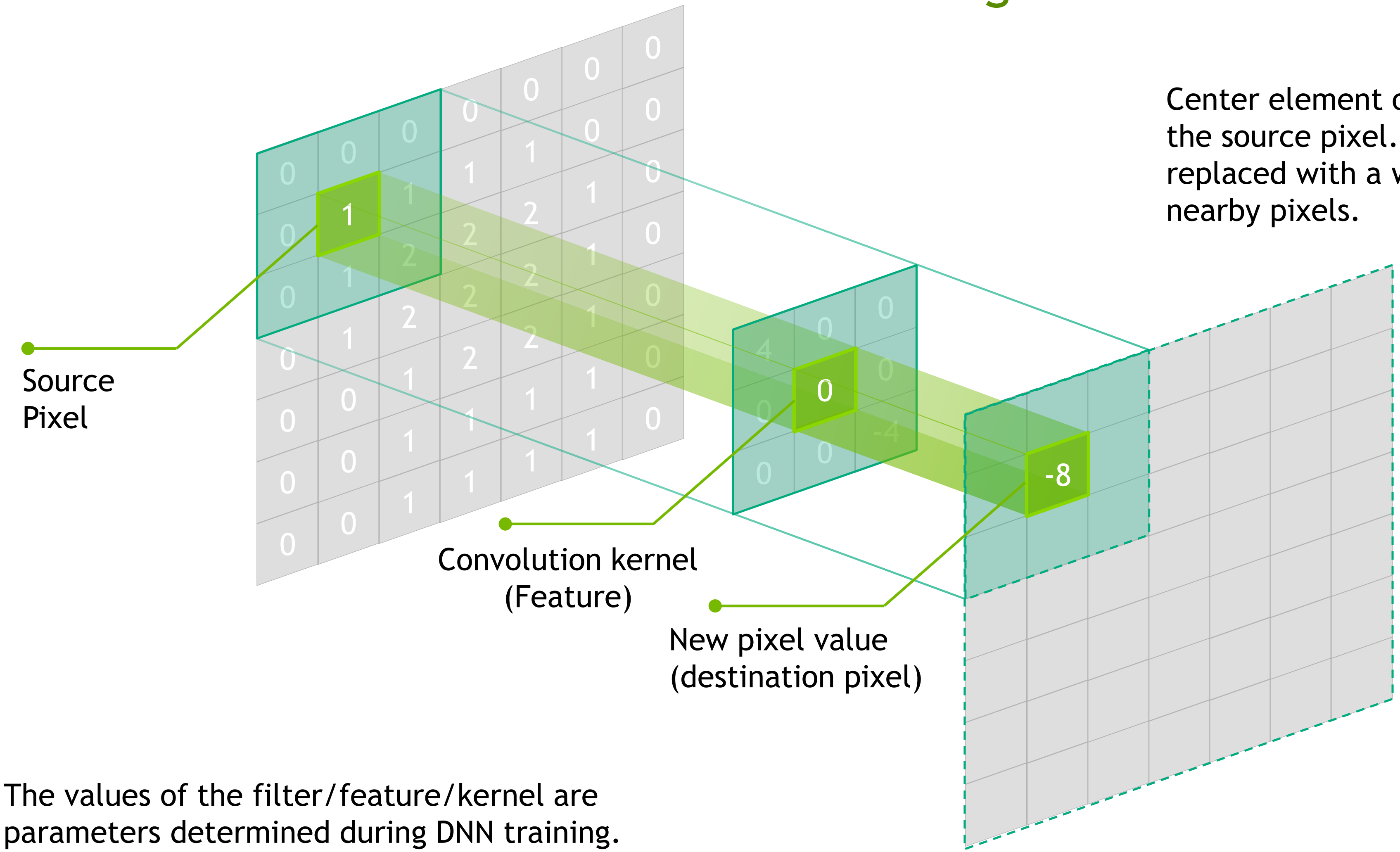
$$G = \sqrt{{G_x}^2 + {G_y}^2}$$

Image source: https://en.wikipedia.org/wiki/Sobel_operator

# CONVOLUTIONAL NEURAL NETWORK

## Network of convolutional filters assigned automatically from data

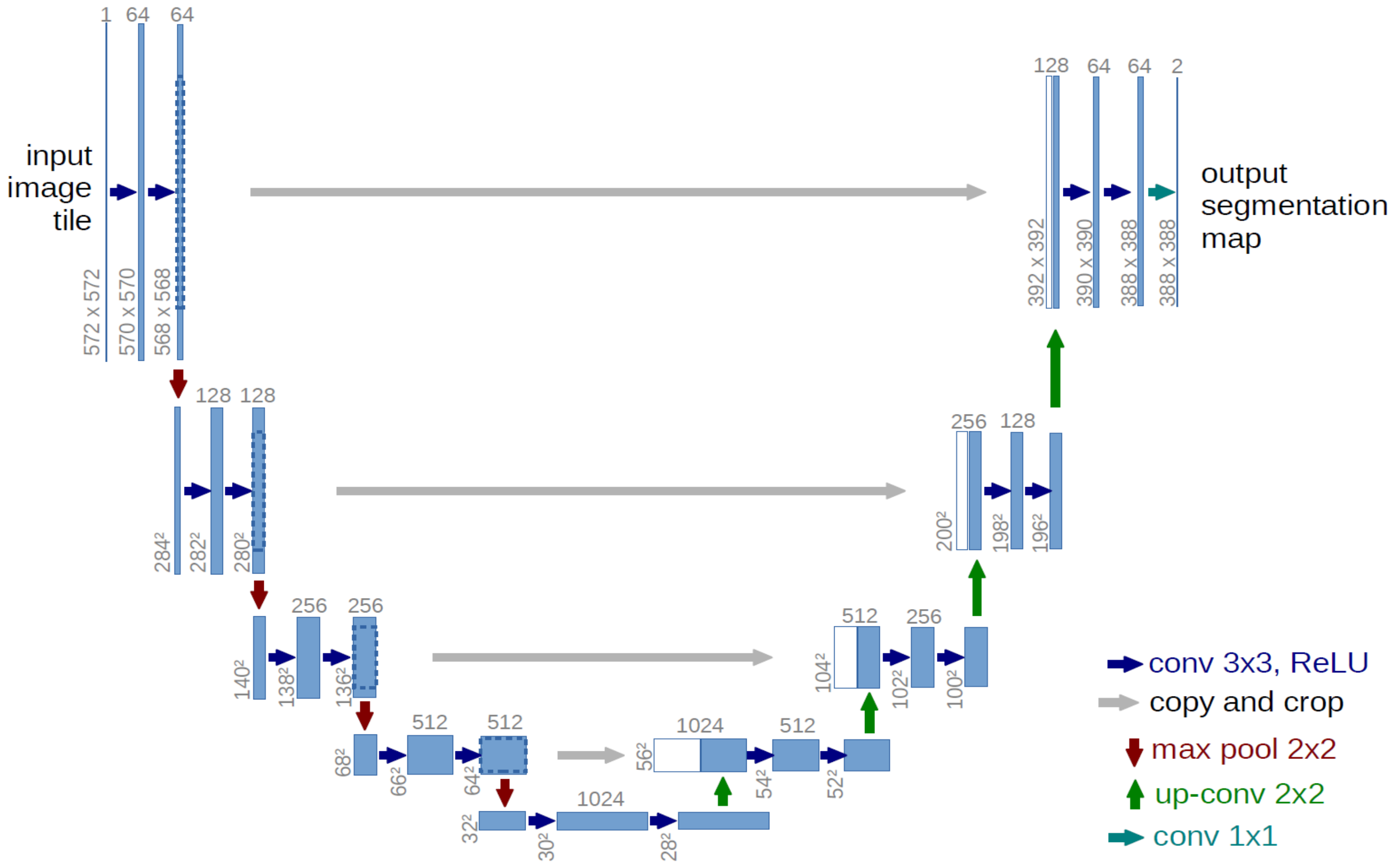
Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.




The values of the filter/feature/kernel are parameters determined during DNN training.

# U-NET: CONVOLUTIONAL NEURAL NETWORK

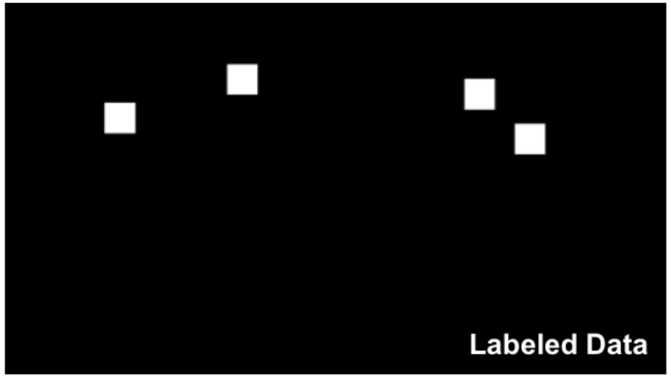
## Image segmentation at multiple scales



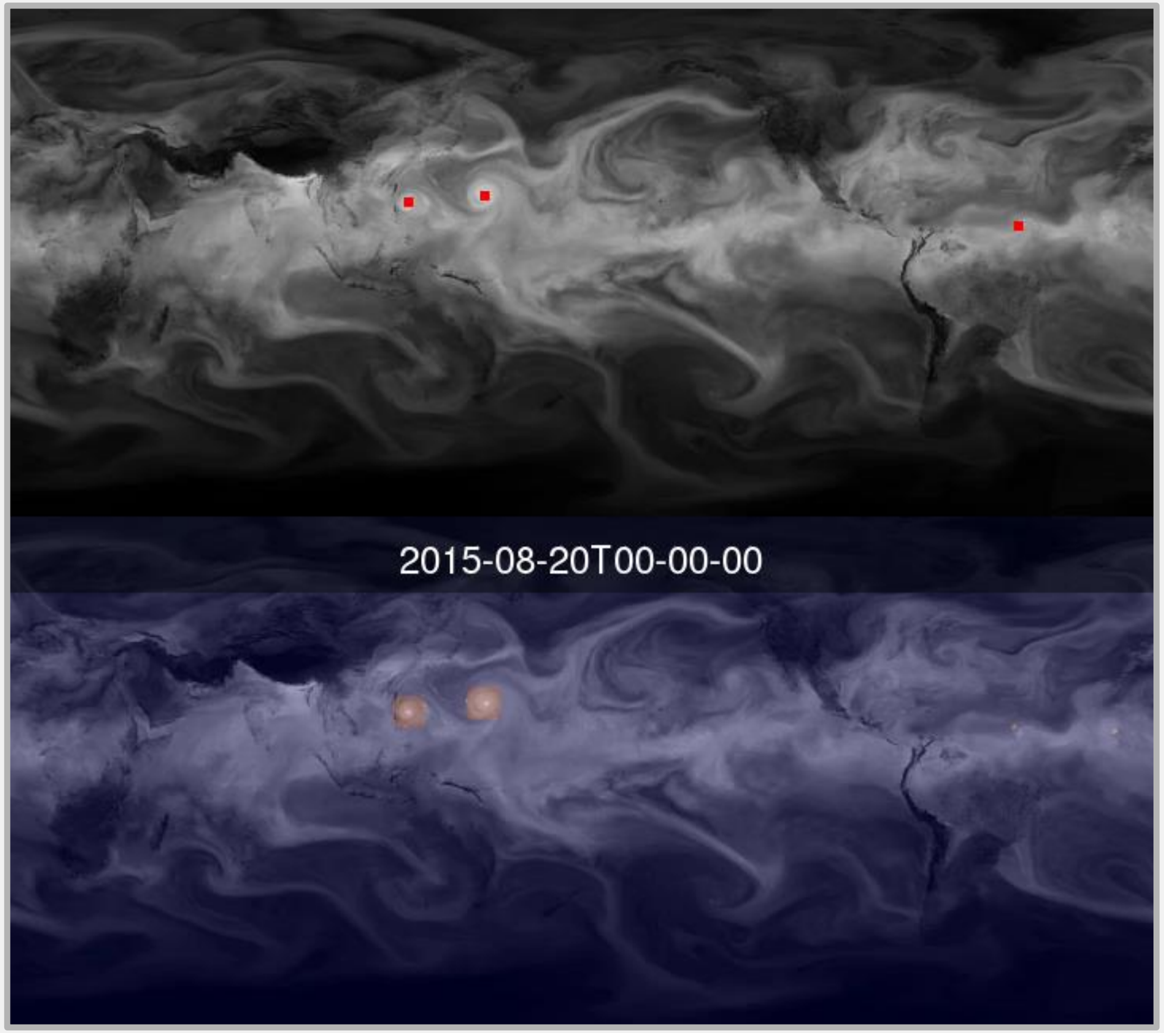
# TROPICAL STORMS GFS MODEL DATA

Jebb Stewart, Christina Bonfonti, Mark Govett NOAA, David Hall NVIDIA

| | |
|---|---|
| **INPUT** | GFS PWAT + IBTRACKS |
| **OUTPUT** | DETECTION CONFIDENCE |
| **TRAINING SET** | 2010-2015 |
| **TEST SET** | 2016 |
| **NETWORK** | U-NET |




**Ground Truth**



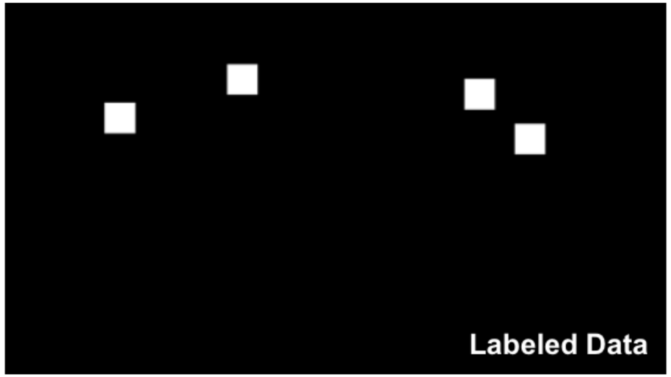

**Prediction**

Automatically detect future storms. No need to define precise heuristics. Storms defined implicitly by example.

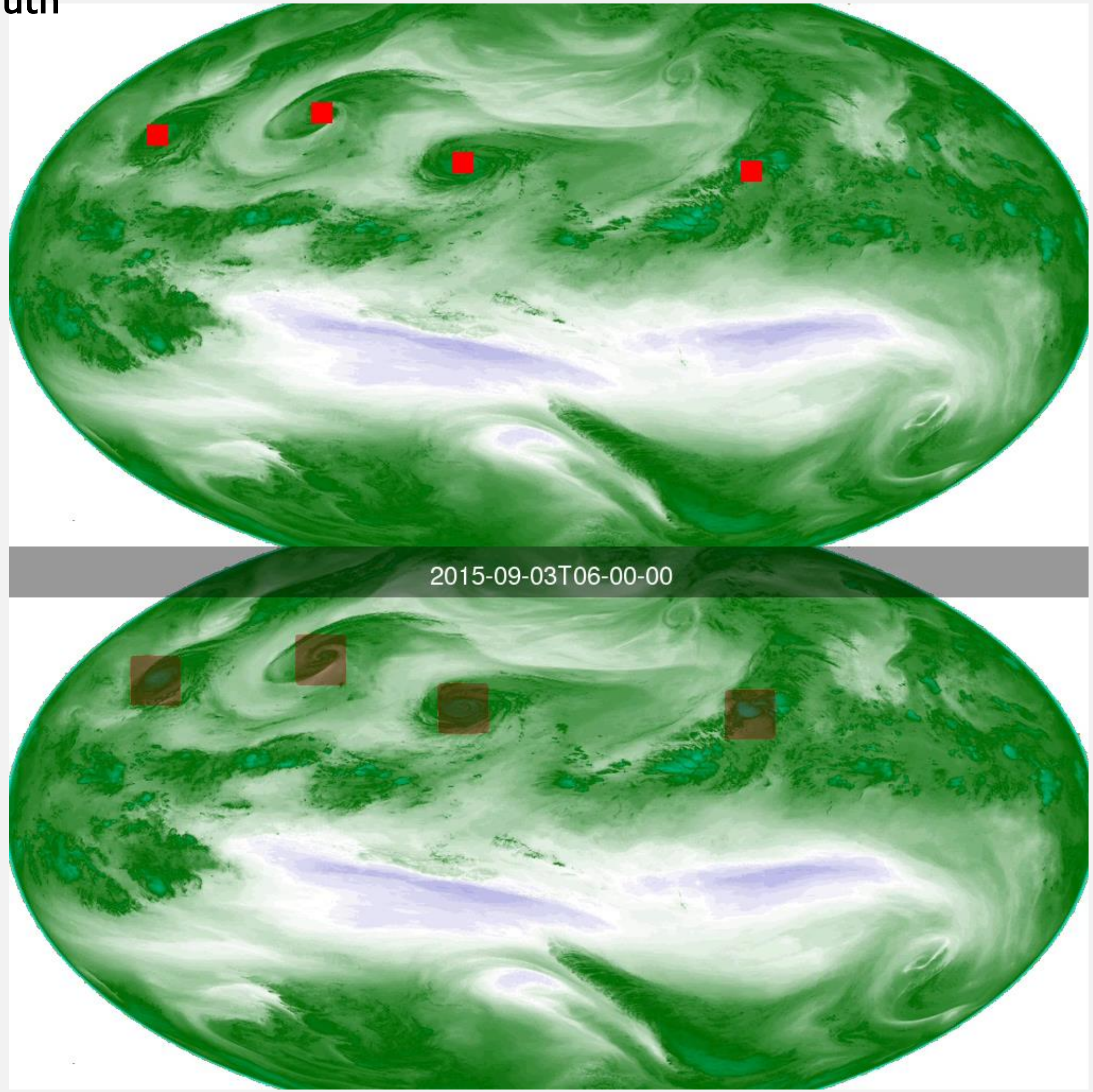
# TROPICAL STORMS GOES SATELLITE

Jebb Stewart, Christina Bonfonti, Mark Govett NOAA, David Hall NVIDIA

| | |
|---|---|
| **INPUT** | GOES UPPER TROPO WV |
| **OUTPUT** | DETECTION CONFIDENCE |
| **TRAINING SET** | 2010-2013 |
| **TEST SET** | 2015 |
| **NETWORK** | U-NET |




Ground Truth



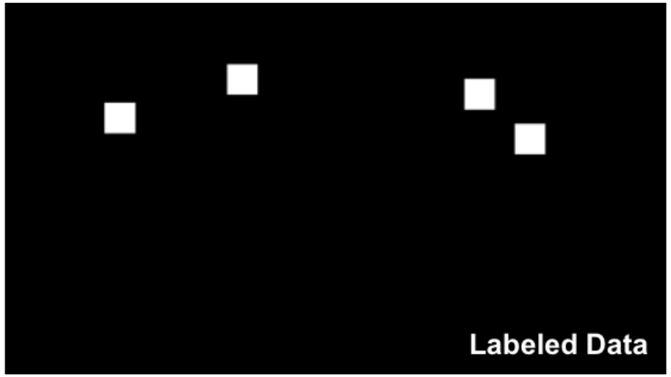

Prediction

- Uses only upper tropo water vapor
- Accurate near image center
- Has some trouble Earth’s limb

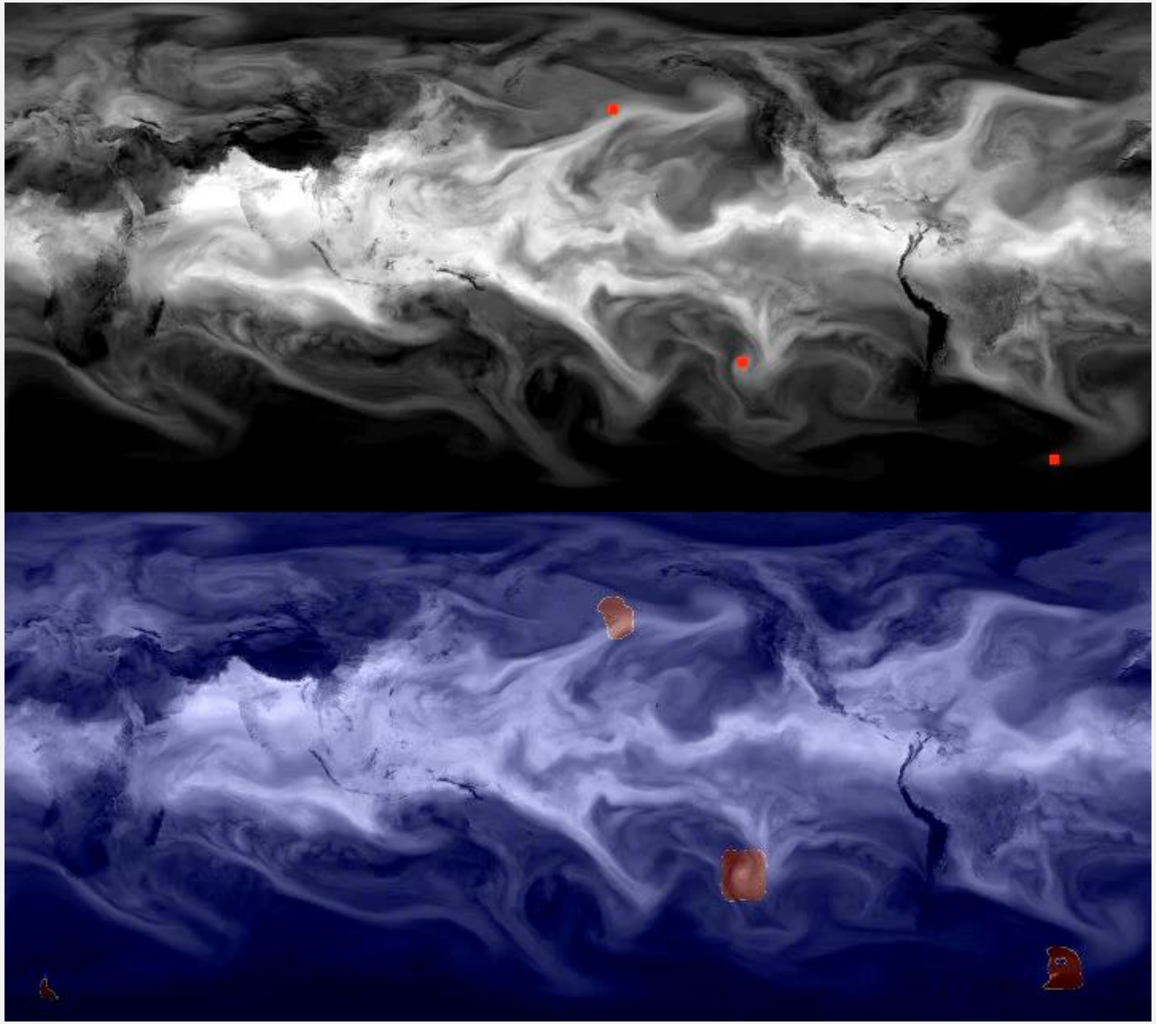
# EXTRATROPICAL CYCLONES GFS MODEL DATA

Christina Bonfonti , Jebb Stewart, Mark Govett NOAA, David Hall NVIDIA

| | |
|---|---|
| **INPUT** | GFS PWAT + HEURISTIC |
| **OUTPUT** | DETECTION CONFIDENCE |
| **TRAINING SET** | 2011-2014 |
| **TEST SET** | 2015 |
| **NETWORK** | U-NET |




Ground Truth



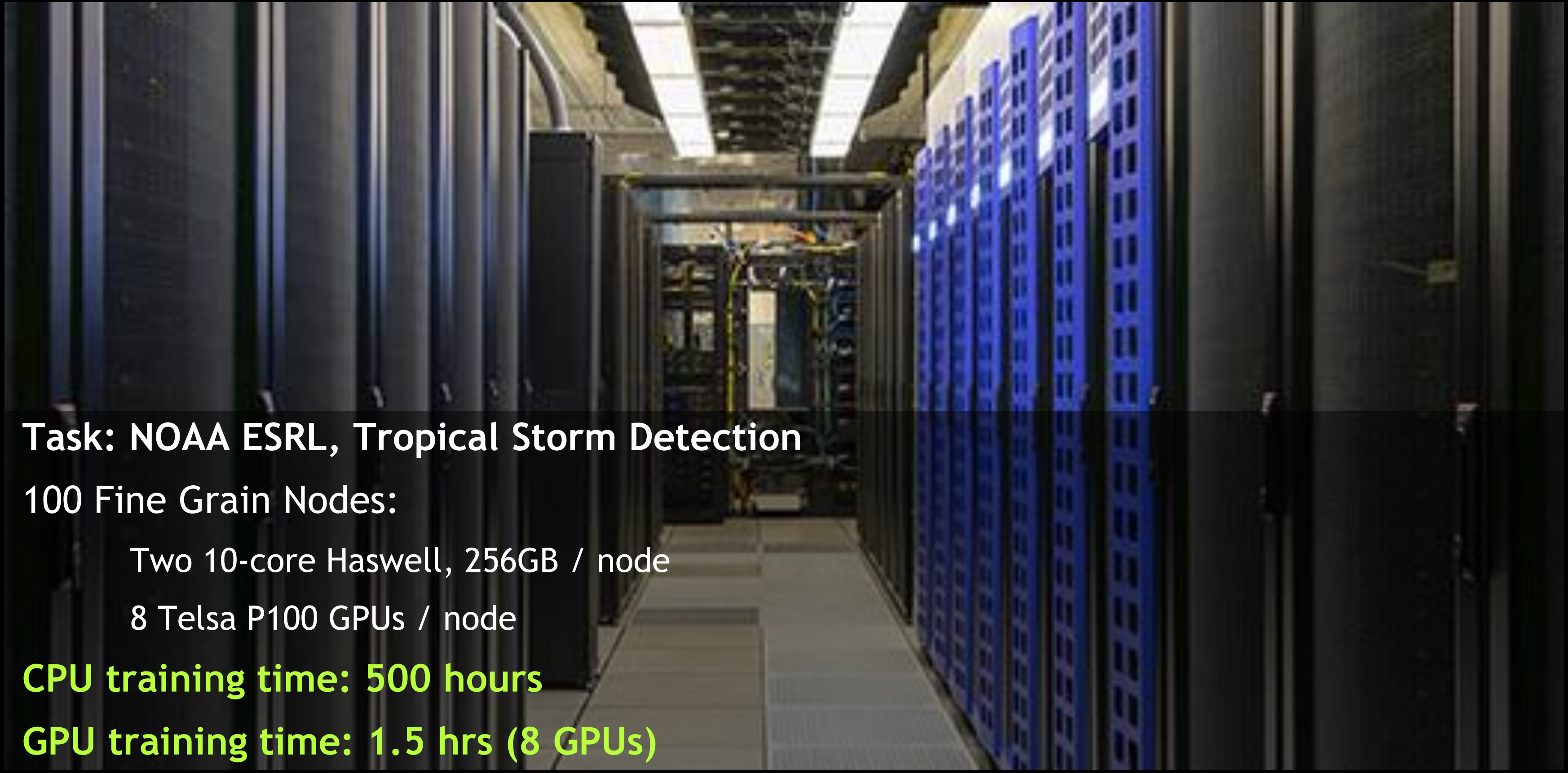
Prediction

- Data labelled using a heuristic (T,P,wind)
- Trained network needs only water-vapor
- Fast and simple detection

# GPU VS CPU TRAINING

## GPUs enabled a 300x speedup in training time

**NOAA's Theia Supercomputer**



**Task: NOAA ESRL, Tropical Storm Detection**

100 Fine Grain Nodes:

- Two 10-core Haswell, 256GB / node
- 8 Telsa P100 GPUs / node

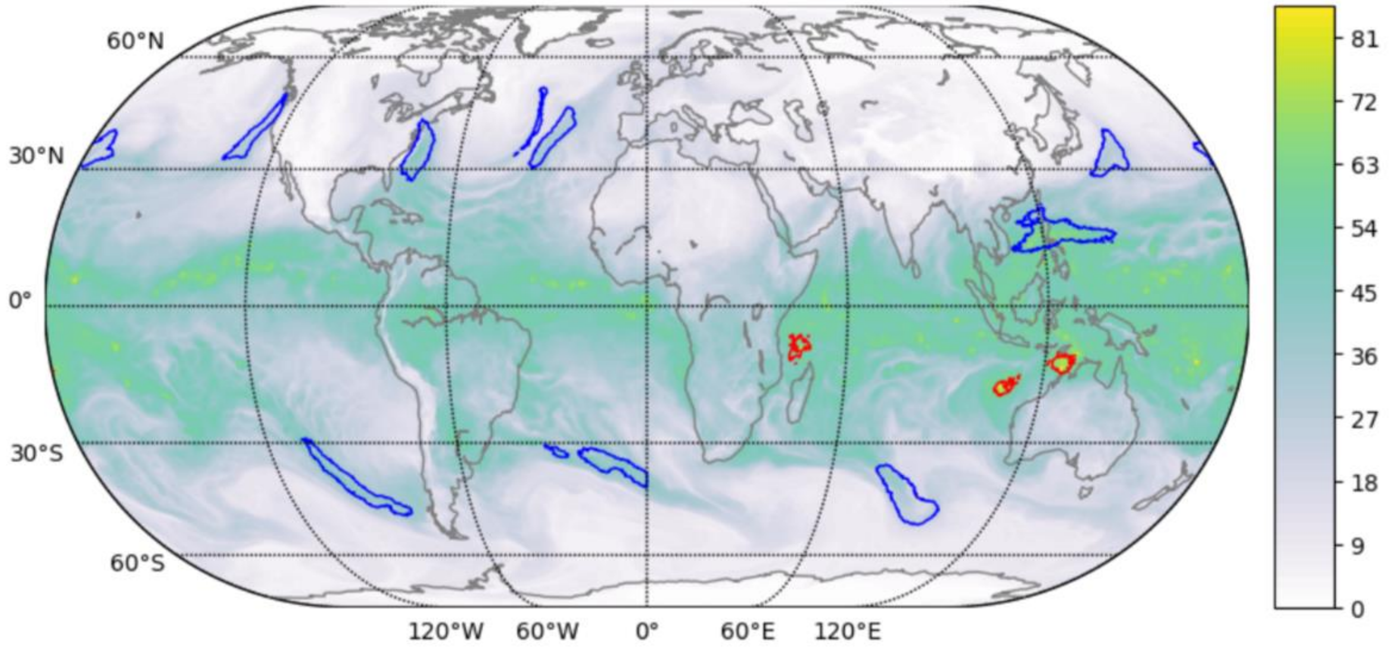
**CPU training time: 500 hours**

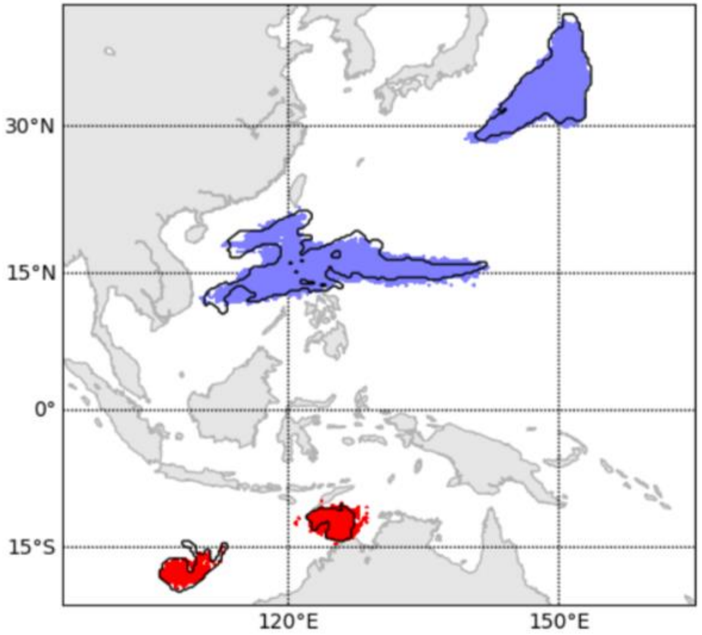
**GPU training time: 1.5 hrs (8 GPUs)**

# DETECTION AT SCALE: GORDON BELL PRIZE

## Exascale Deep Learning for Climate Analytics


Thorsten Kurth* tkurth@lbl.gov
Sean Treichler† sean@nvidia.com
Joshua Romero† joshr@nvidia.com
Mayur Mudigonda* mudigonda@berkeley.edu
Nathan Luehr† nluehr@nvidia.com
Everett Phillips† ephillips@nvidia.com
Ankur Mahesh* amahesh@lbl.gov
Michael Matheson‡ mathesonma@ornl.gov
Jack Deslippe* jrdeslippe@lbl.gov
Massimiliano Fatica† mfatica@nvidia.com
Prabhat* prabhat@lbl.gov
Michael Houston† mhouston@nvidia.com

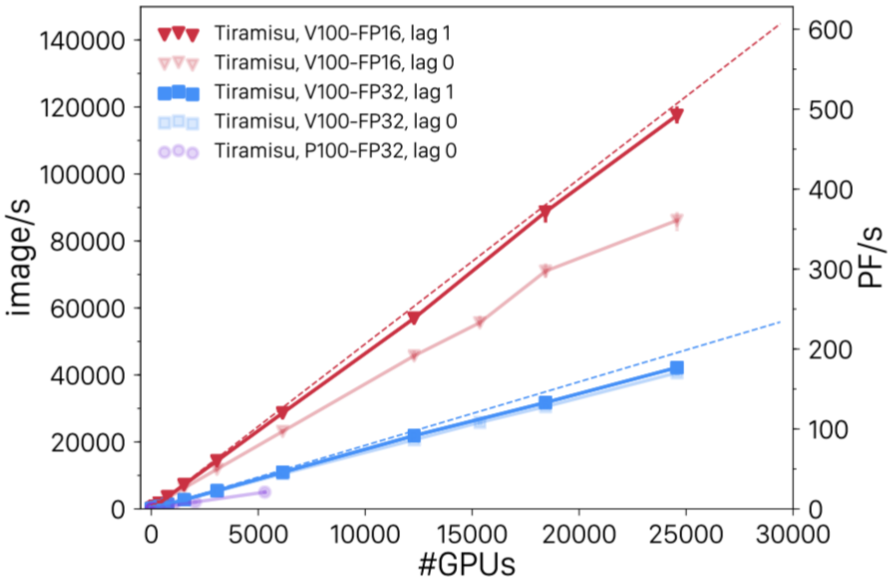




Segmentation of Tropical Storms and Atmospheric Rivers on Summit using convolutional neural networks.

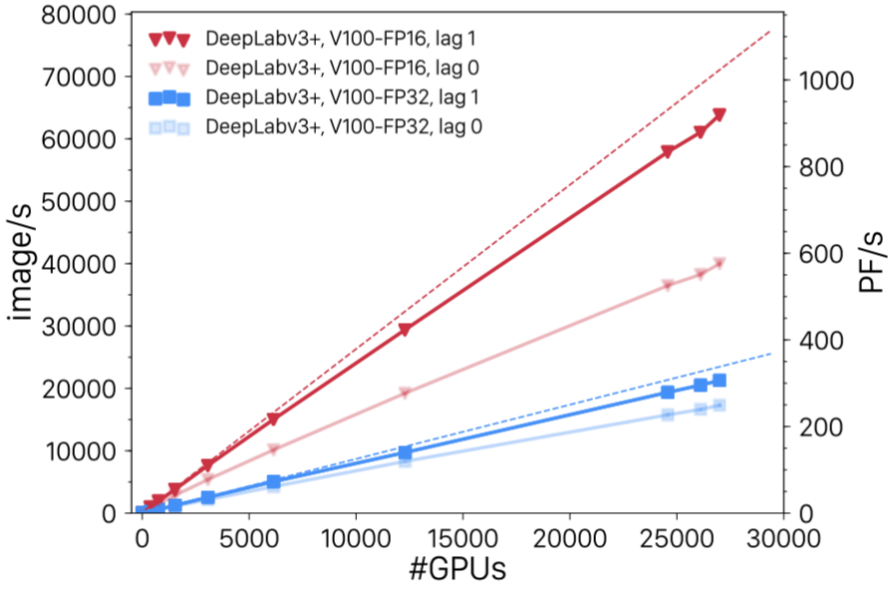
# DETECTION AT SCALE: GORDON BELL PRIZE

# Exascale Deep Learning for Climate Analytics


Thorsten Kurth* tkurth@lbl.gov
Sean Treichler† sean@nvidia.com
Joshua Romero† joshr@nvidia.com
Mayur Mudigonda* mudigonda@berkeley.edu
Nathan Luehr† nluehr@nvidia.com
Everett Phillips† ephillips@nvidia.com
Ankur Mahesh* amahesh@lbl.gov
Michael Matheson‡ mathesonma@ornl.gov
Jack Deslippe* jrdeslippe@lbl.gov
Massimiliano Fatica† mfatica@nvidia.com
Prabhat* prabhat@lbl.gov
Michael Houston† mhouston@nvidia.com


(a) Tiramisu

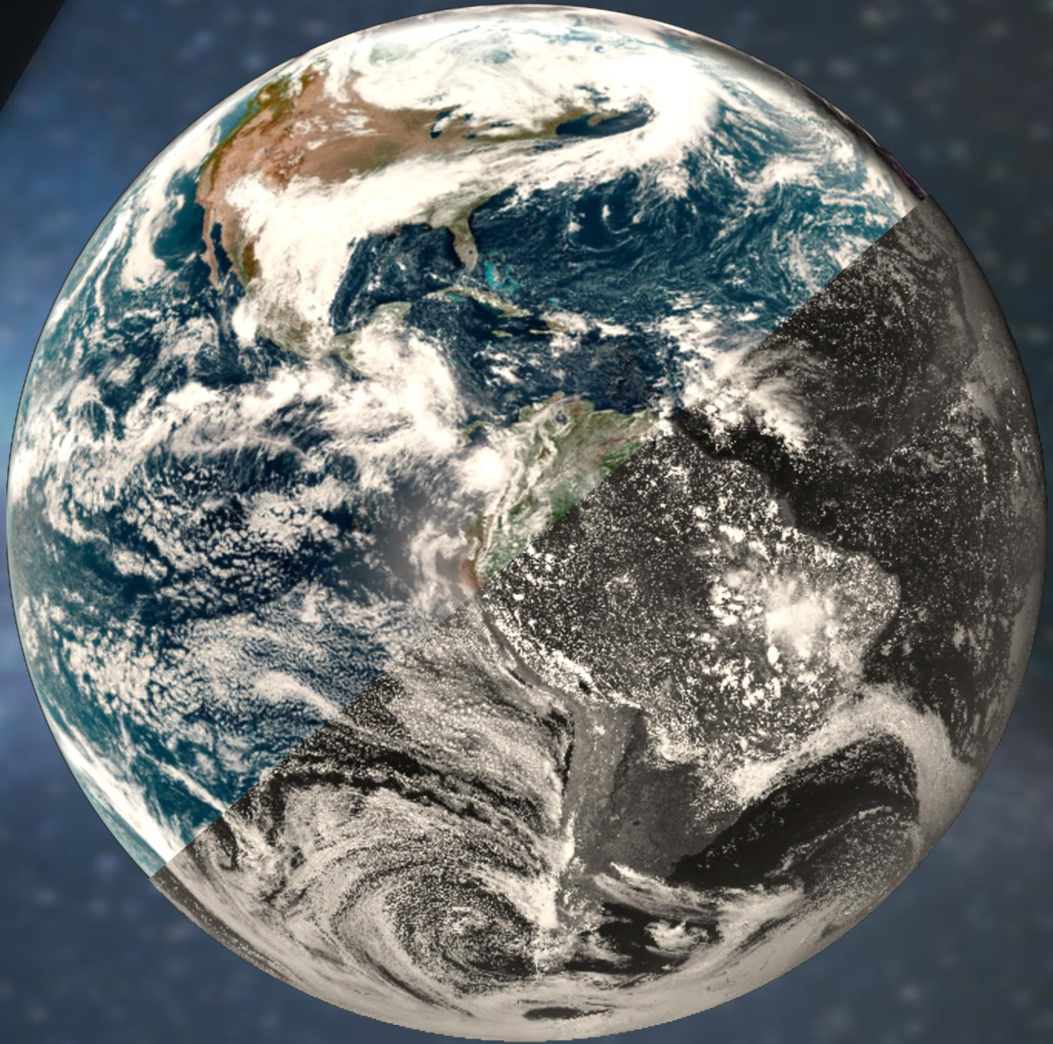



(b) DeepLabv3+

Nearly perfect weak scaling up to 25k GPUS. 1 Exa-flop of performance. 100 years of climate model data in hours
Demonstrates the power of this approach for large-scale data analysis

# 2. TRANSLATION: IMPROVED DATA ASSIMILATION VIA INVERSE MODELING

Deep learning can automatically construct maps between any two related coordinate systems. This can be used to convert satellite observations into model variables, with applications to data assimilation. It also has the potential to enable us to combine information from multiple models or satellites into a single dataset of greater accuracy and completeness.



GOES-16 CIRA GEO COLOR / GOES-15 RED BAND

# IMAGE TO IMAGE TRANSLATION

## Conditional GANS can translate one type of image into another

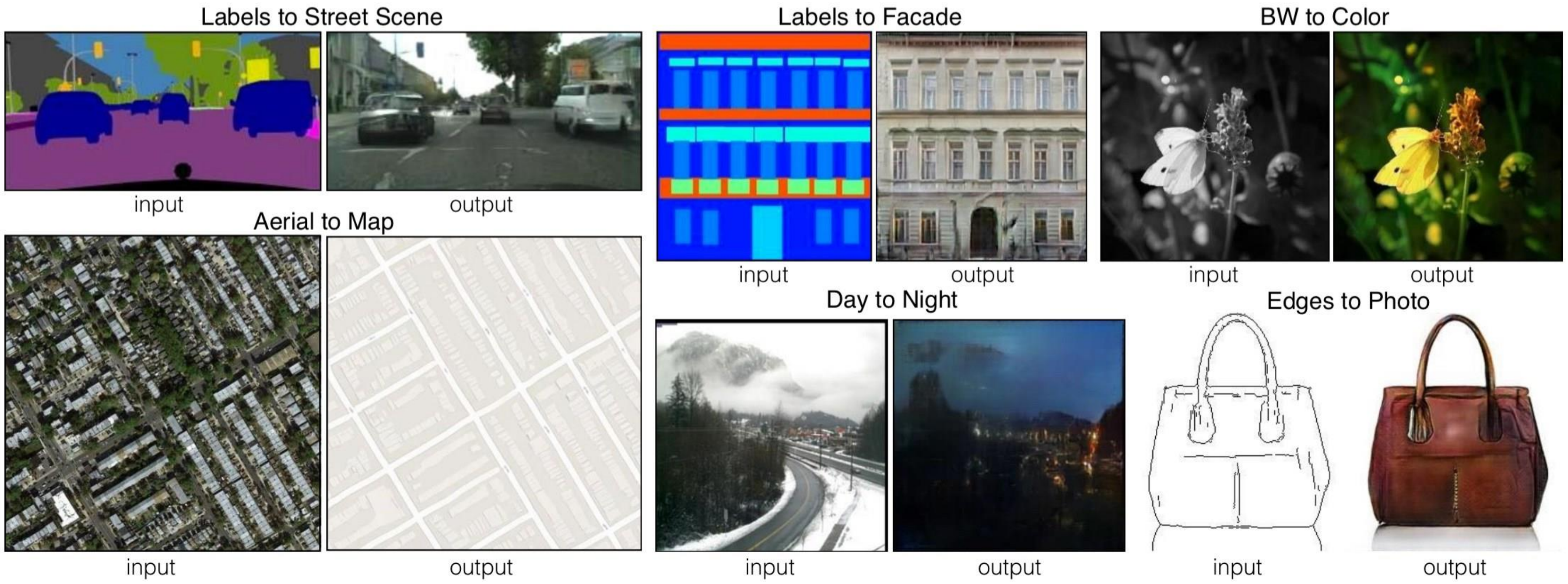
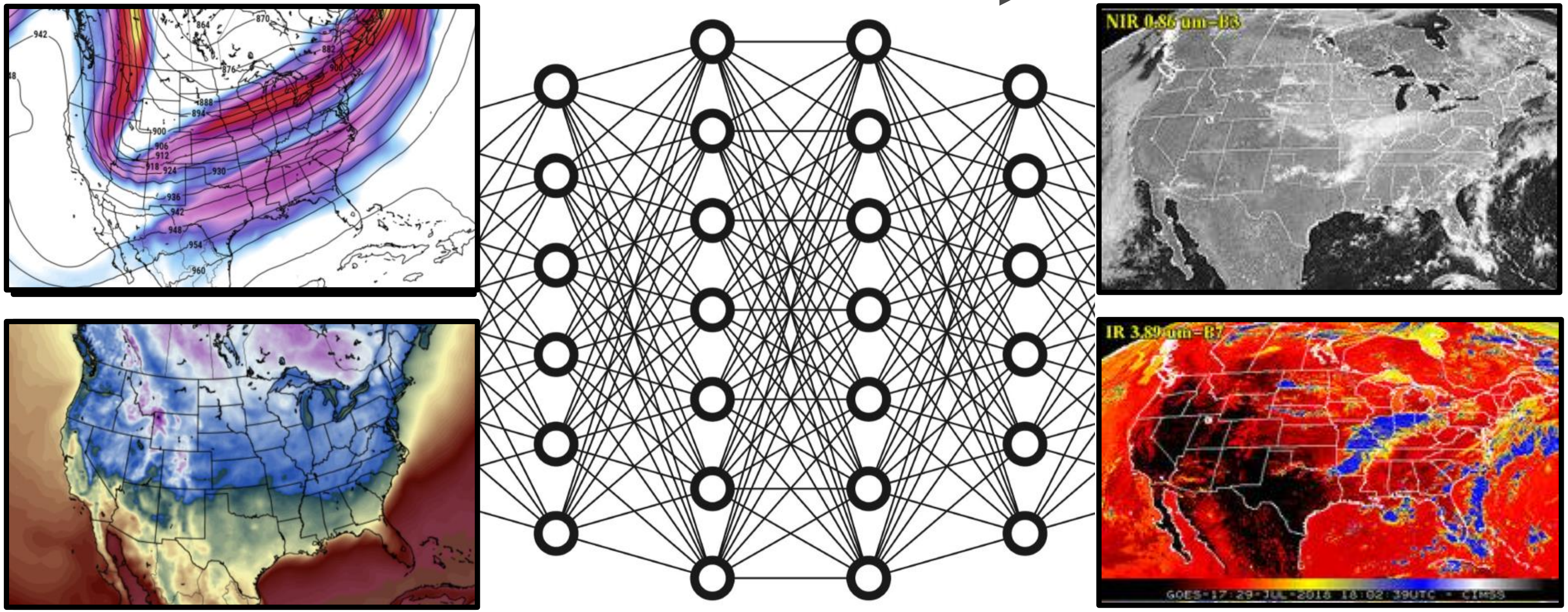



Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

# MAP: MODEL TO SATELLITE (FORWARD OPERATOR)

## Model analyses to satellite observations

SATELLITE RADIANCES    Convolutional Neural Network    MODEL VARIABLES

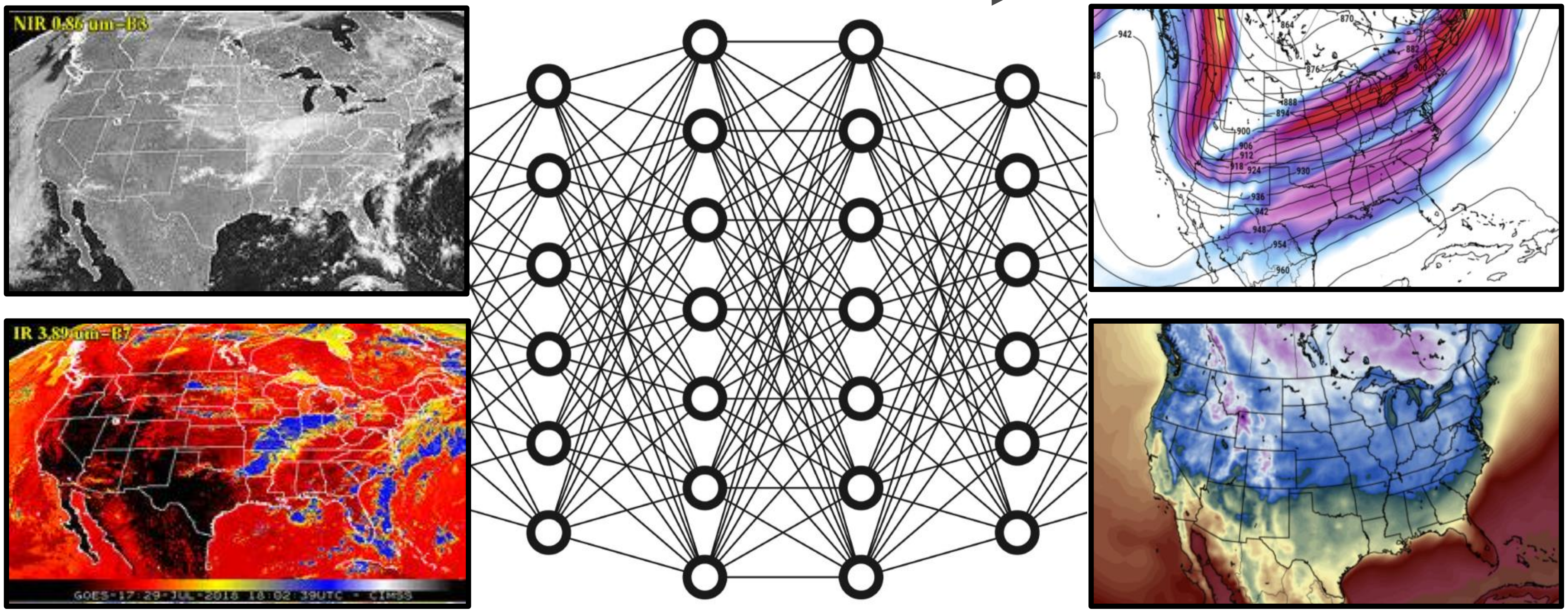



**Maps from 3d fields to 3d fields, rather than one column at a time**
**Can use spatial patterns to guide predictions**

# MAP: SATELLITE TO MODEL (INVERSE OPERATOR)

## Satellite observations to model analyses

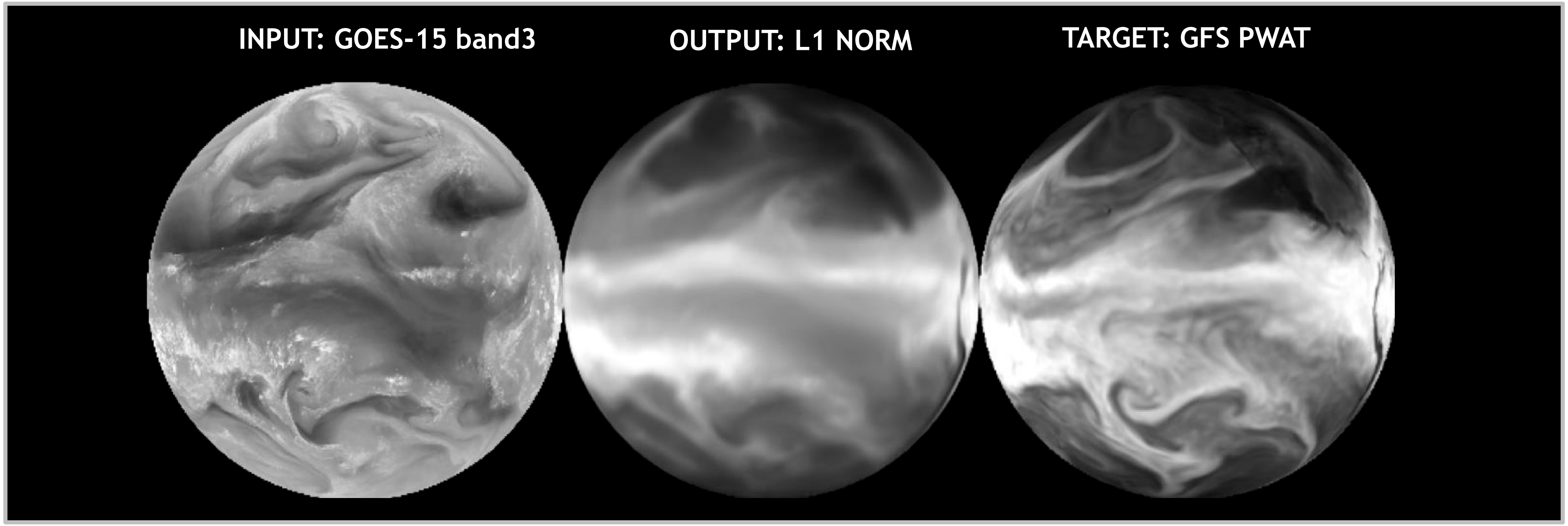
SATELLITE RADIANCES | Convolutional Neural Network | MODEL VARIABLES




**Hard to construct an inverse model by hand, but no more difficult for a neural network than the forward model.**

# RESULTS: REGRESSION

## One-to-many map results in 'regression to the mean'

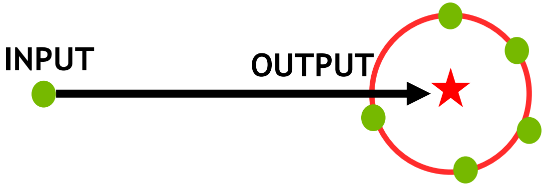



Example of incomplete information: upper-tropo WV to total column WV

L1 output is *average* of multiple plausible states

Not consistent with any single realizable state

Adding bands can more fully constrain the output



# RESULTS: CONDITIONAL GAN

## Physically plausible state from incomplete data

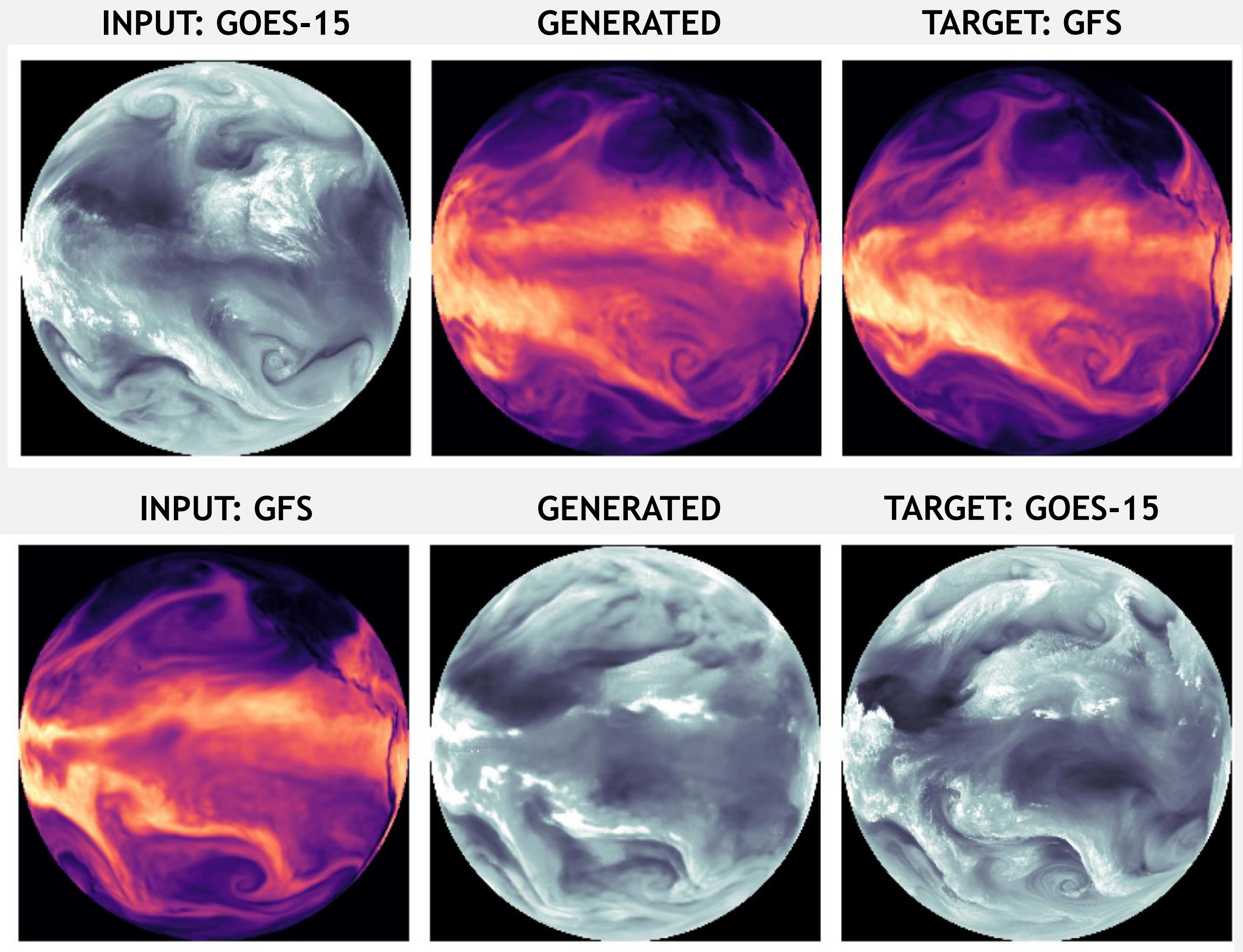
| | |
|---|---|
| OBSERVATION | GOES-15 band 3 |
| MODEL VAR | GFS Precipitable water |
| Training | 2014-2016 |
| Test | 2013 |

Adversarial model outputs a physically plausible state

Like an ensemble member from uncertain initial conditions

Both forward and inverse maps

For data assimilation and forecast verification



# APPLICATIONS TO DATA ASSIMILATION

## Accelerate forward and/or inverse models

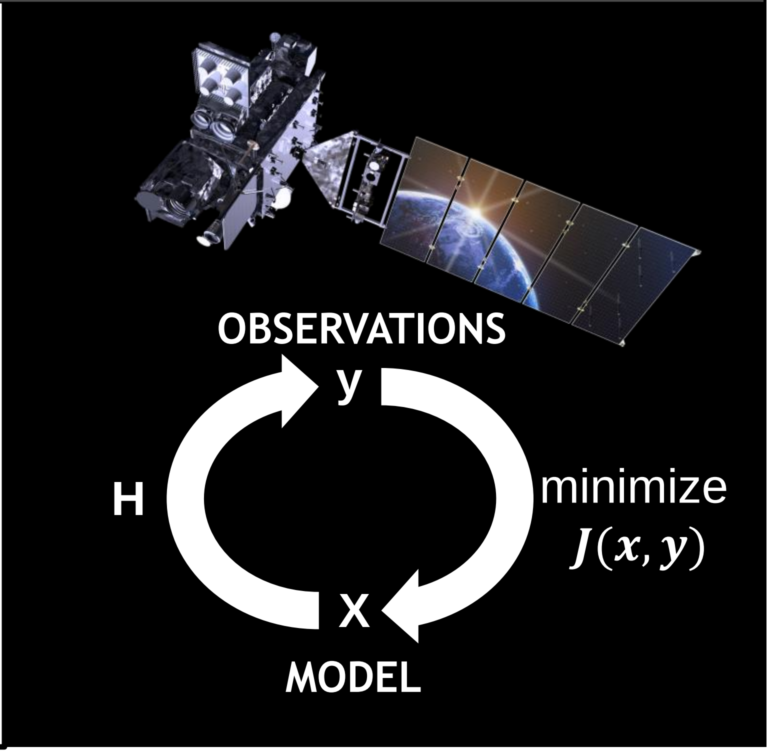
| Background state | Observations | Forward Operator | Background Error Covariances | Observation Error Covariances |
|---|---|---|---|---|
| $\boldsymbol{x}_b$ | $\boldsymbol{y}$ | $\boldsymbol{H}[\boldsymbol{x}]$ | $\boldsymbol{B}$ | $\boldsymbol{R}$ |

- 3DVAR: Iterate to minimize loss J(x). H is expensive!

$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_b)^T \boldsymbol{B}^{-1} (\boldsymbol{x} - \boldsymbol{x}_b) + (\boldsymbol{y} - \boldsymbol{H}[\boldsymbol{x}])^T \boldsymbol{R}^{-1} (\boldsymbol{y} - \boldsymbol{H}[\boldsymbol{x}])$$

1. Accelerate H by replacing it with DL forward map
2. Apply DL inverse map, then solve for x directly!

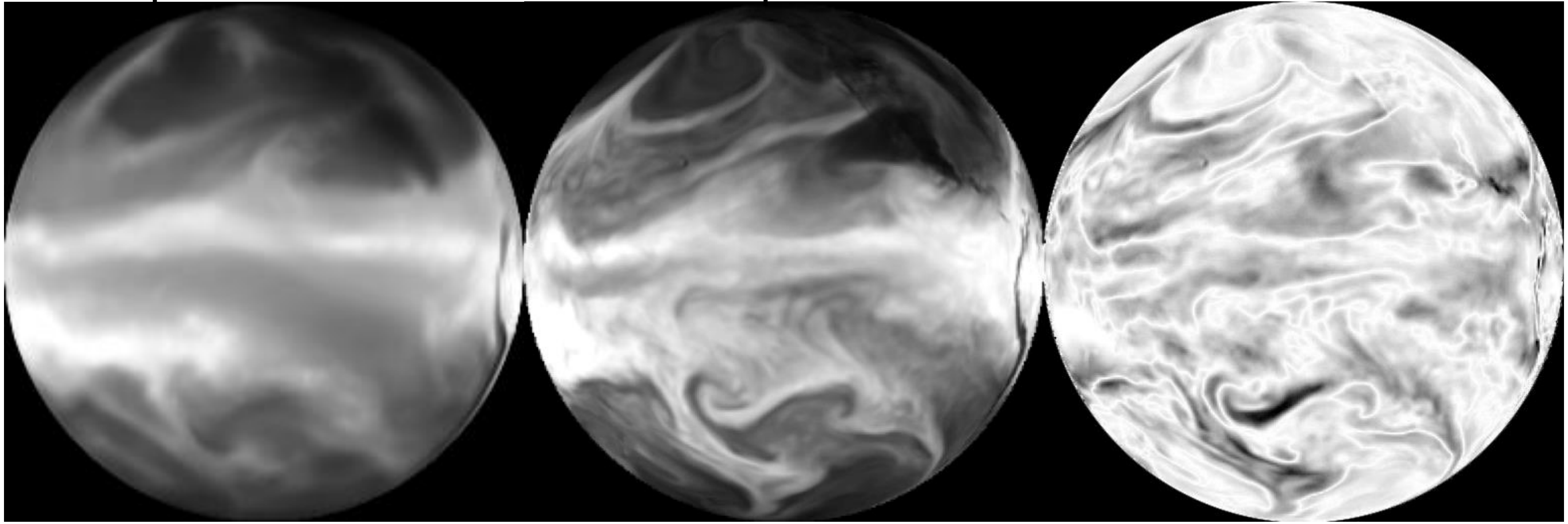
$$J(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}_b)^T \boldsymbol{B}^{-1} (\boldsymbol{x} - \boldsymbol{x}_b) + (\boldsymbol{x} - \boldsymbol{x}_o)^T \widetilde{\boldsymbol{R}}^{-1} (\boldsymbol{x} - \boldsymbol{x}_o)$$



# NEED FOR UNCERTAINTY QUANTIFICATION

## Some pixels are certain, others are are completely uncertain

**Mean of Possible Outputs** **One specific output** **Confiden ce**



Need pixel-level variances and covariances to combine with other data sources

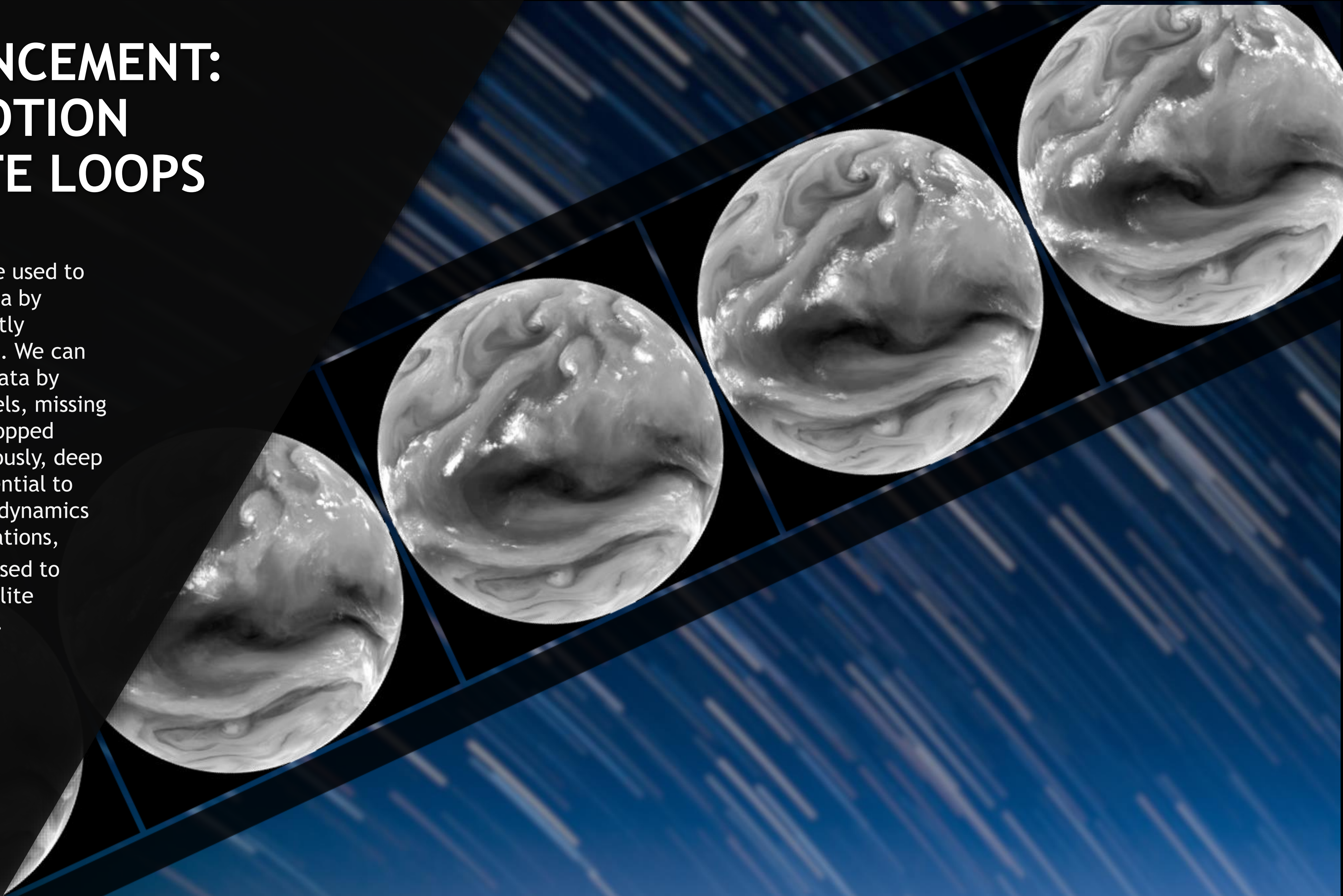
Use Bayesian neural networks to explicitly model uncertainties

Or use *“Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles”*

# 3. ENHANCEMENT: SLOW MOTION SATELLITE LOOPS

Deep learning may be used to enhance satellite data by learning to intelligently interpolate it in time. We can also repair damage data by imputing missing pixels, missing channels, or even dropped frames. More ambitiously, deep learning has the potential to learn the underlying dynamics directly from observations,

Which may then be used to estimate future satellite observations directly.

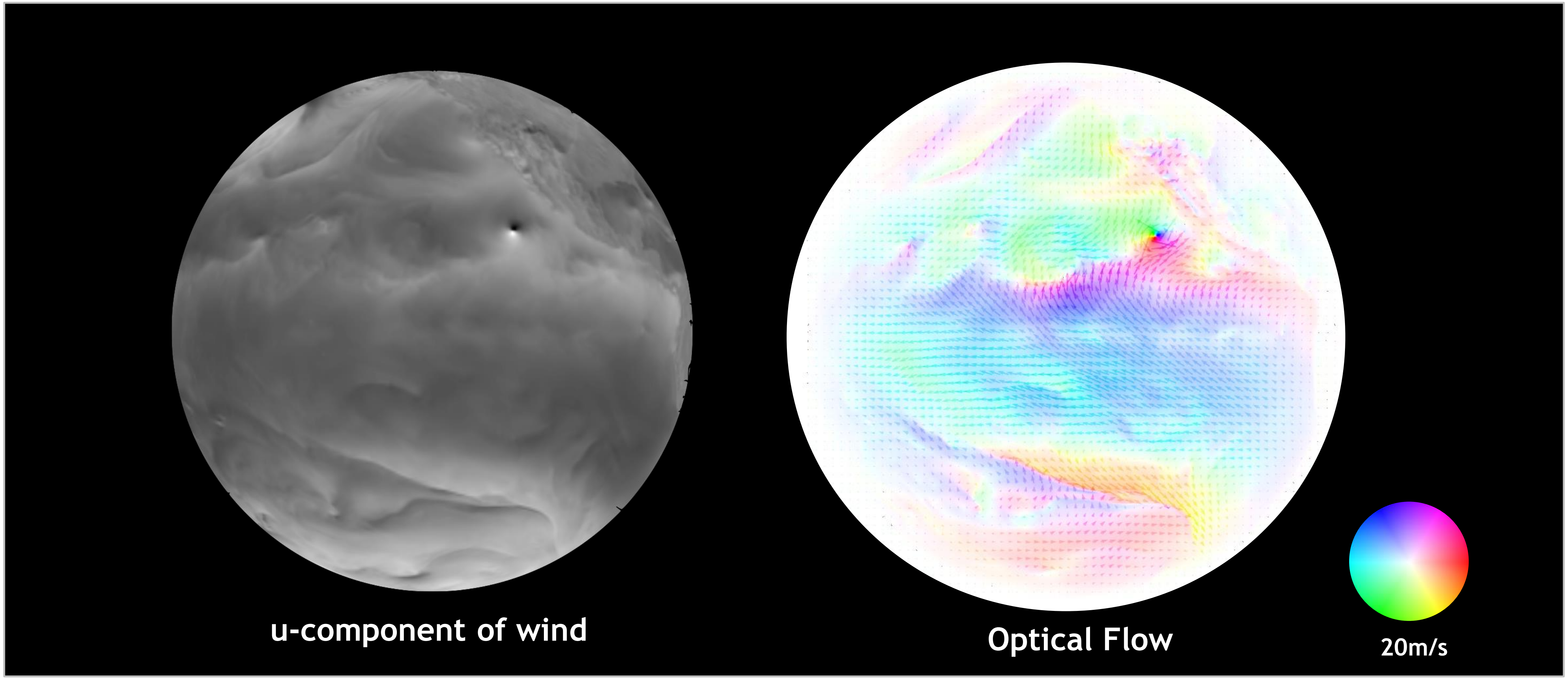


# NVIDIA SUPER SLOW-MOTION

## Deep learning for temporal interpolation

# OPTICAL FLOW FROM MODEL WINDS

## Estimate motion vectors from upper tropospheric model winds

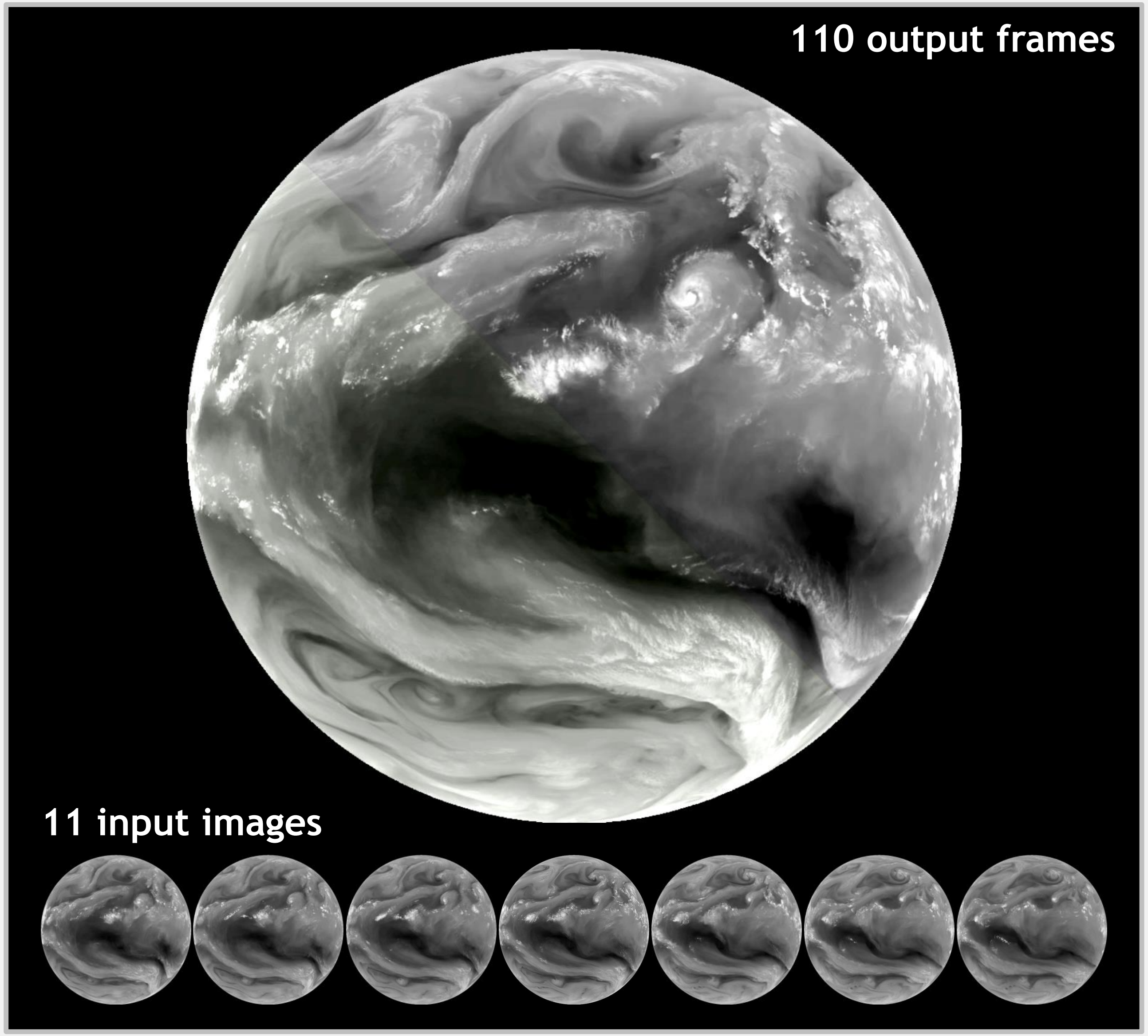


# SLOW MOTION SATELLITE LOOP

David Hall NVIDIA

| | |
|---|---|
| INPUT | GOES-15 band 3, GFS winds |
| OUTPUT | Interpolated GOES-15 |
| INPUT FREQ | 1 every 3 hours |
| OUTPUT FREQ | 1 every 18 minutes |

Applications:
- Visualization
- Data Augmentation
- Replace dropped frames
- Reduce storage requirements



# PARAMETER INFERENCE

## Fine tune winds from observations

Improve estimate of advective winds

Treat model winds as an initial guess

Advect observations forward from frame n

Compute a loss function using frame n+1

Back-propagate to obtain gradient

Optimize to fine tune wind speeds

# MODEL INFERENCE

## Learn both the winds and the ODE from observations

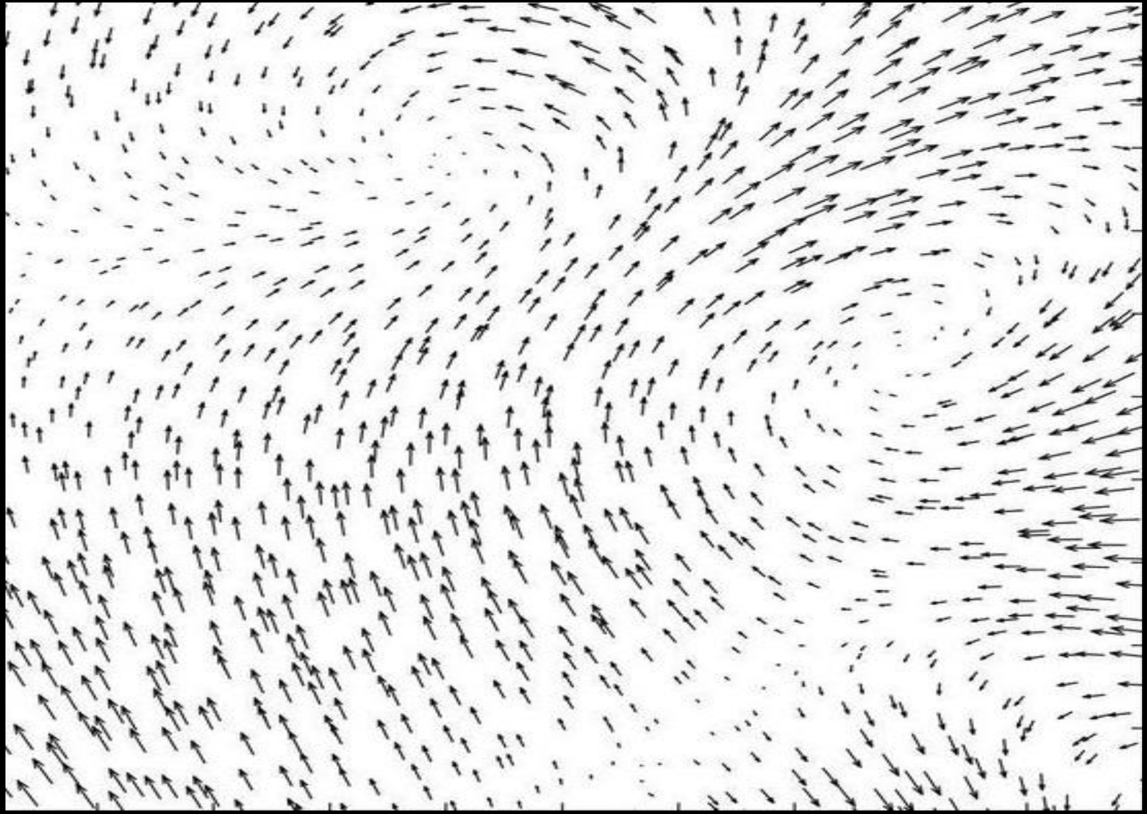
Use robust ODE solver for time integration

Represent derivatives via a neural net

Compute loss function following RK-NN paper

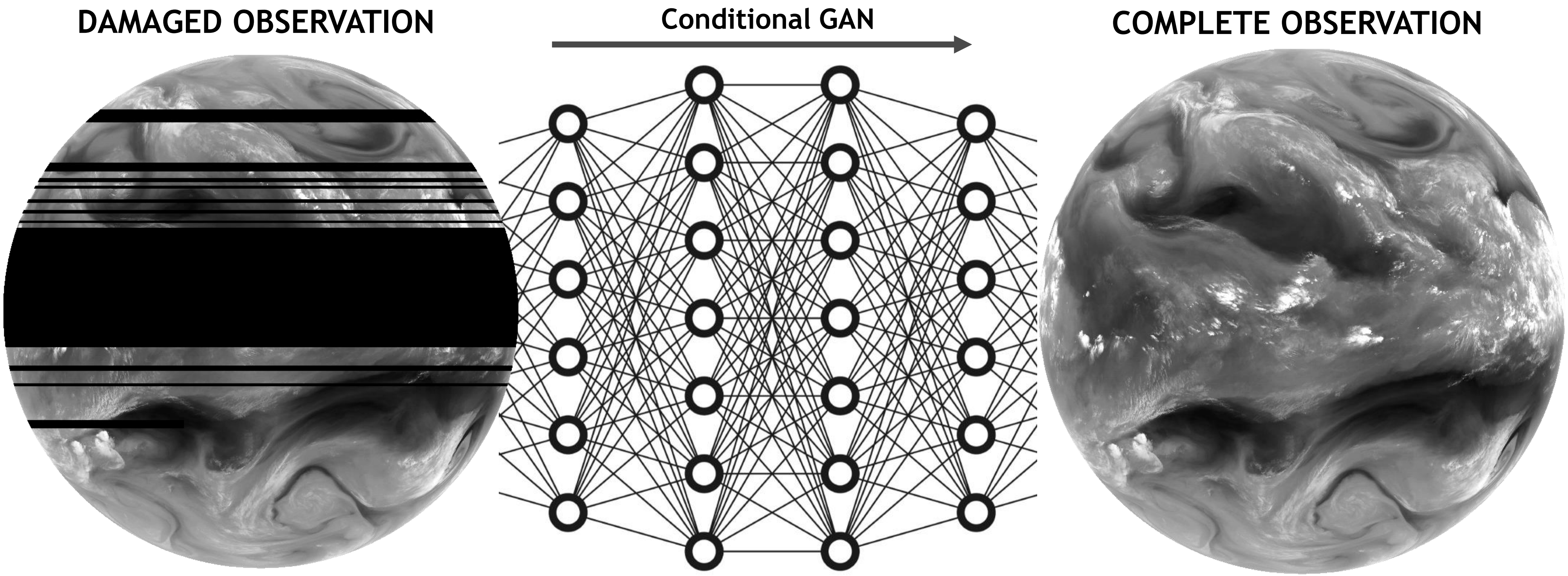
Obtain gradients via Adjoint Sensitivity

Automatically learn dynamics from data



# IMPUTE MISSING DATA

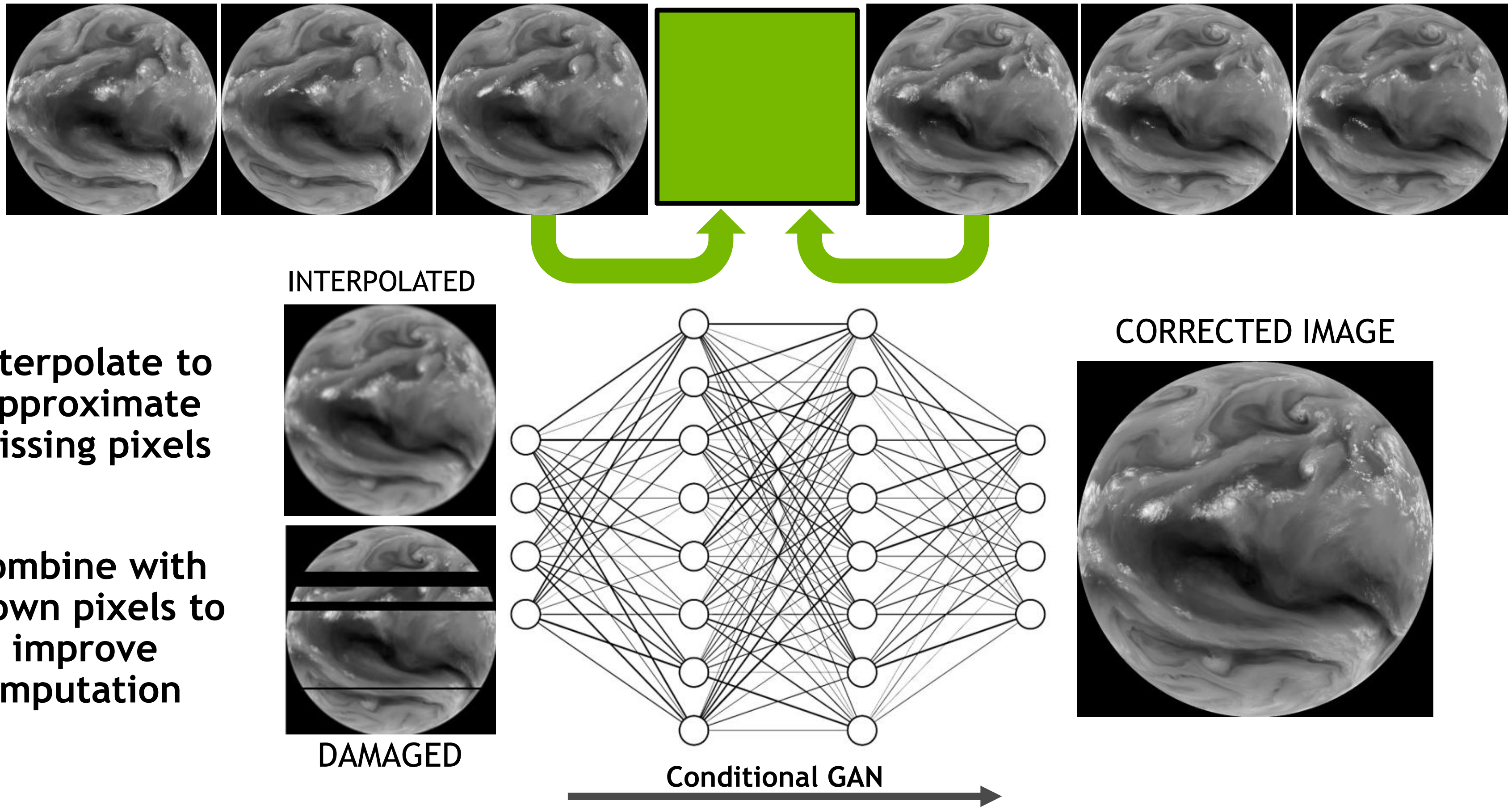
## Train a conditional GAN to reconstruct missing pixels




**Missing data can potentially be reconstructed from information in the other bands**

# INTERPOLATION + IMPUTATION

## Interpolate in time to provide additional information for imputation




**Interpolate to approximate missing pixels**

**Combine with known pixels to improve imputation**

(Or map from the interpolated images to the real images, to improve interpolated image quality)

# 4. ACCELERATION VIA NEURAL NET EMULATION

Deep neural networks can produce high fidelity approximations of expensive functions through supervised training on a large number of input-output data pairs. The resulting emulation can be multiple orders of magnitude faster than the original. It’s similar to a lookup table, but with feature-aware interpolation in high dimensional spaces. This approach enables GPU acceleration of arbitrarily complex functions without labor intensive code porting.

“Clockwork” by Mackenzie Bentley

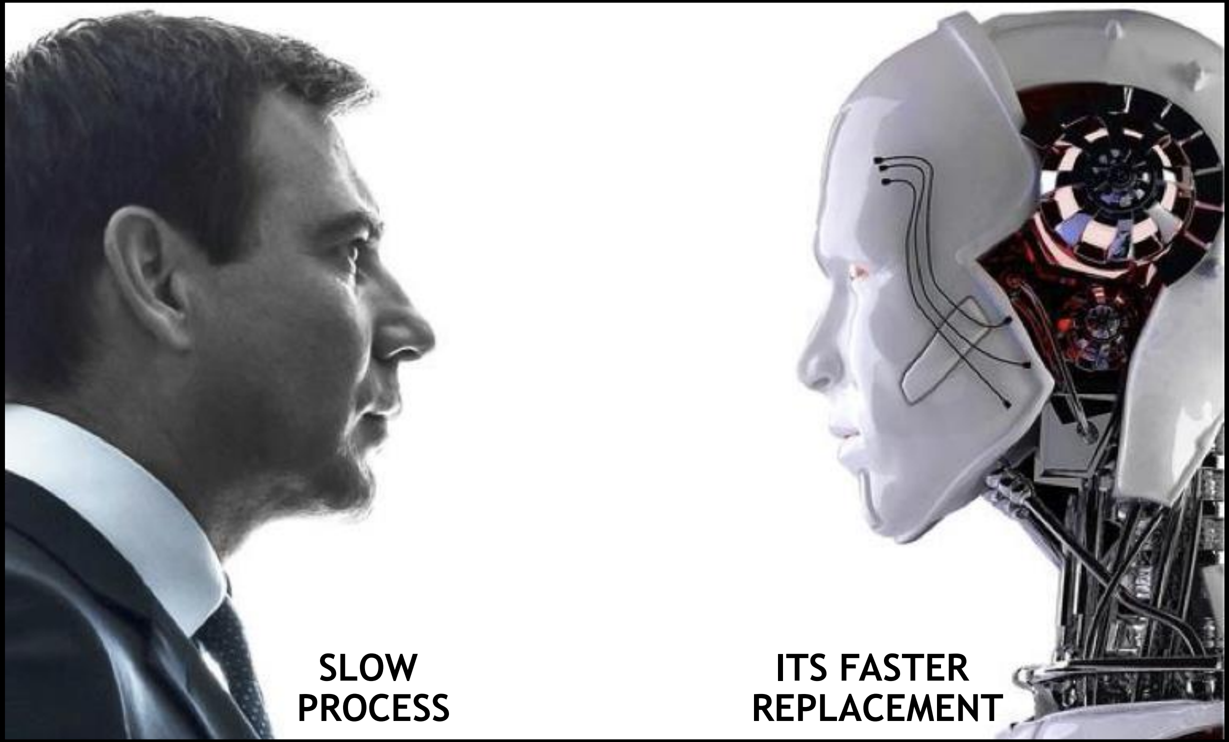
# ACCELERATION VIA EMULATION

## Do the same thing, but do it much *faster*

An alternate route to GPU acceleration
Accelerates conventional routines
Complimentary to OpenACC and CUDA
Replace expensive routines with DNNs
Train on 1000s of input/output pairs
No need to port original code to GPU
Orders of magnitude faster at runtime

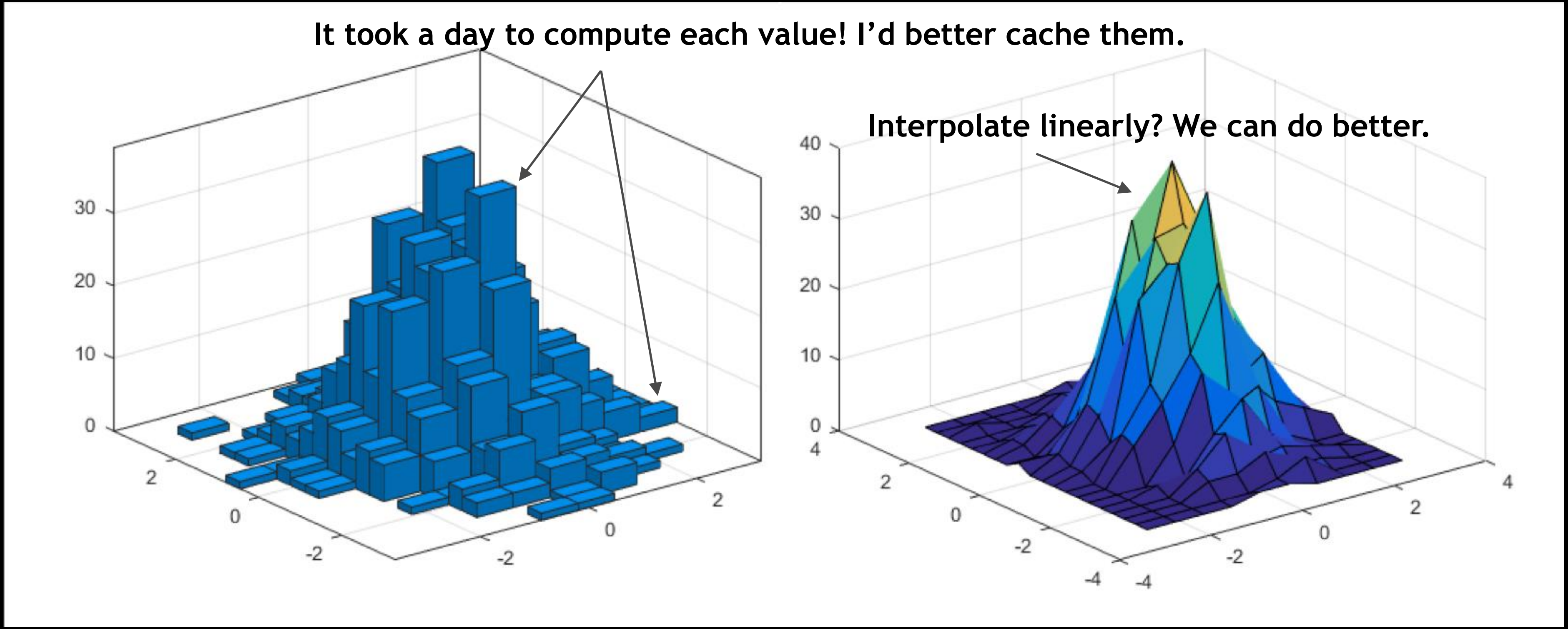
Examples:

- Ocean wave-wave interactions
- Radiation parametrization
- Cloud super-parametrization
- Particle collider simulations



# EMULATION: AN AI POWERED LOOKUP-TABLE

## Precompute expensive values, and interpolate intelligently

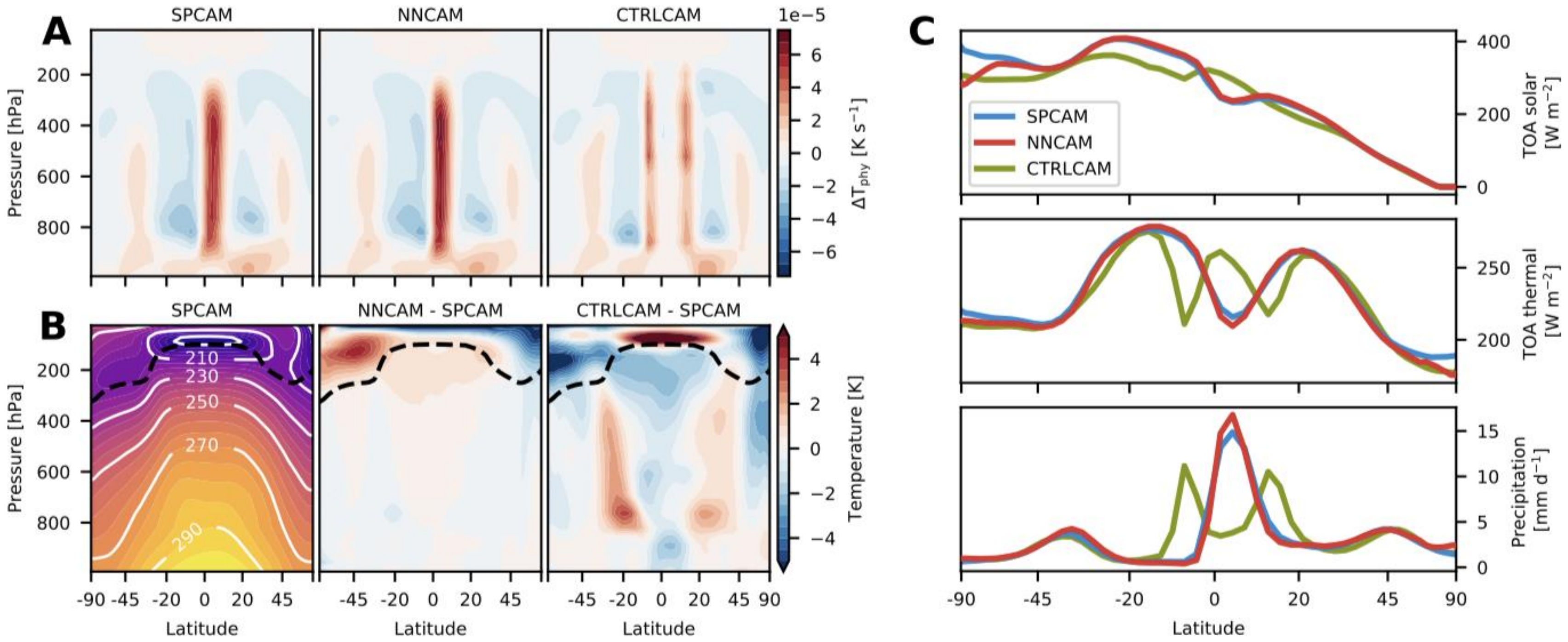



- Imagine it takes a day to compute a single value
- Do you ever want to repeat that calculation?
- What if you want a value that is *almost the same*?
- Deep learning emulation fits a custom curve comprised of features learned from your data.
- It interpolates but can't extrapolate.

# EMULATION: CAM SUPER-PARAMETRIZATION (20X)

## Deep learning to represent sub-grid processes in climate models

Stephan Rasp, Michael Pritchard, **UC Irvine** Pierre Gentine, **Columbia University**



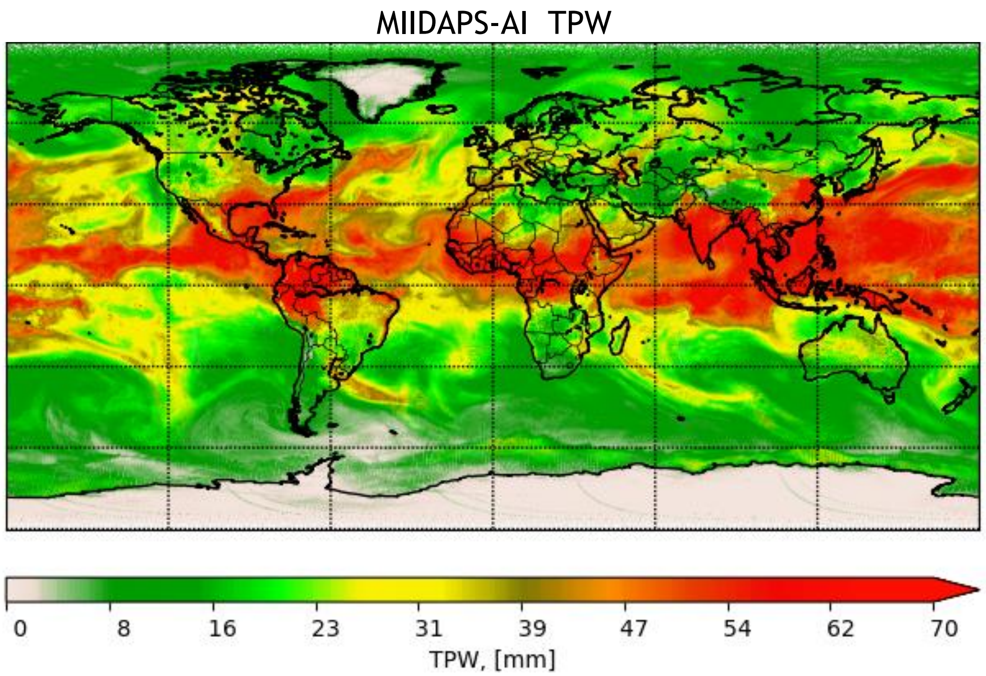

A) Mean heating rate B) Mean temp and biases C) Top of atmosphere fluxes, and precipitation

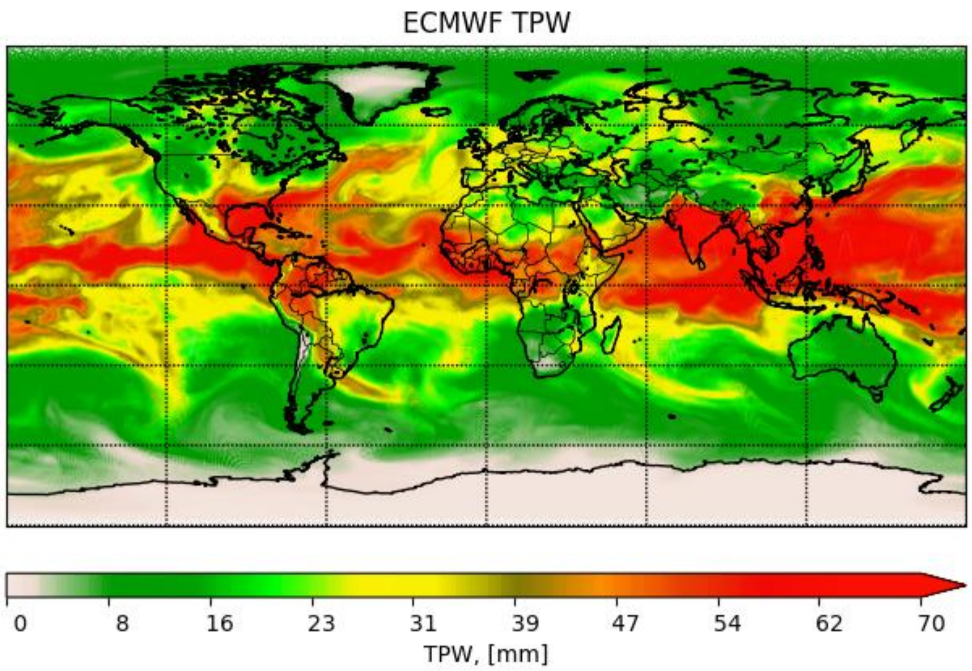
**SPCAM is a 2d cloud-resolving parameterization for greater accuracy**
**NNCAM emulates SP-CAM, with 20x speedup**
Details: 9 fully connected layers, 567k params, 8 hours training time on a single NVIDIA GTX 1080

# EMULATION: MIIDAPS-AI (1400X)

## Multi-Instrument Inversion and Data Assimilation Preprocessing System

Sid Boukabara **NOAA/NESDIS** Eric Maddy, Adam Neiss **Riverside Technology Inc**




**Inverse operator for multiple IR and microwave satellites. Iteratively uses CRTM radiative transfer model**

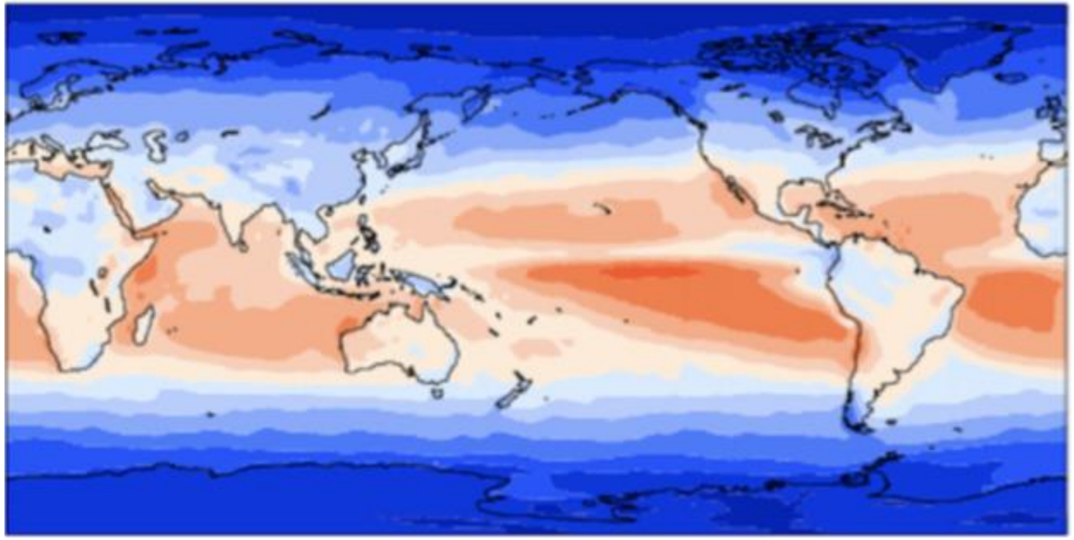
**5 seconds vs 2 hrs to process one day 1400x speedup.**

# EMULATION: RRTMG (10X)

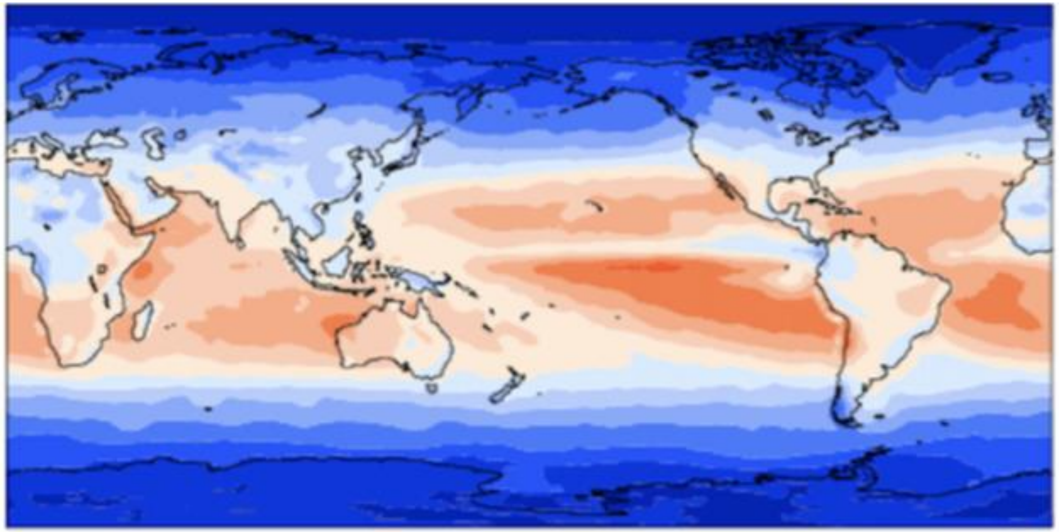
## Rapid Radiative Transfer Model for GCMs

**Matthew Norman, Pal Anikesh, ORNL**

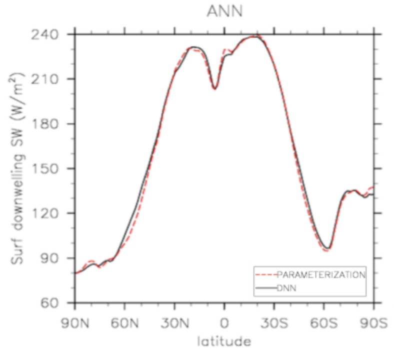
Surface Net SW Flux (RRTMG). Mean = 161.91 W/m$^2$



Surface Net SW Flux (Emulation). Mean = 161.91 W/m$^2$



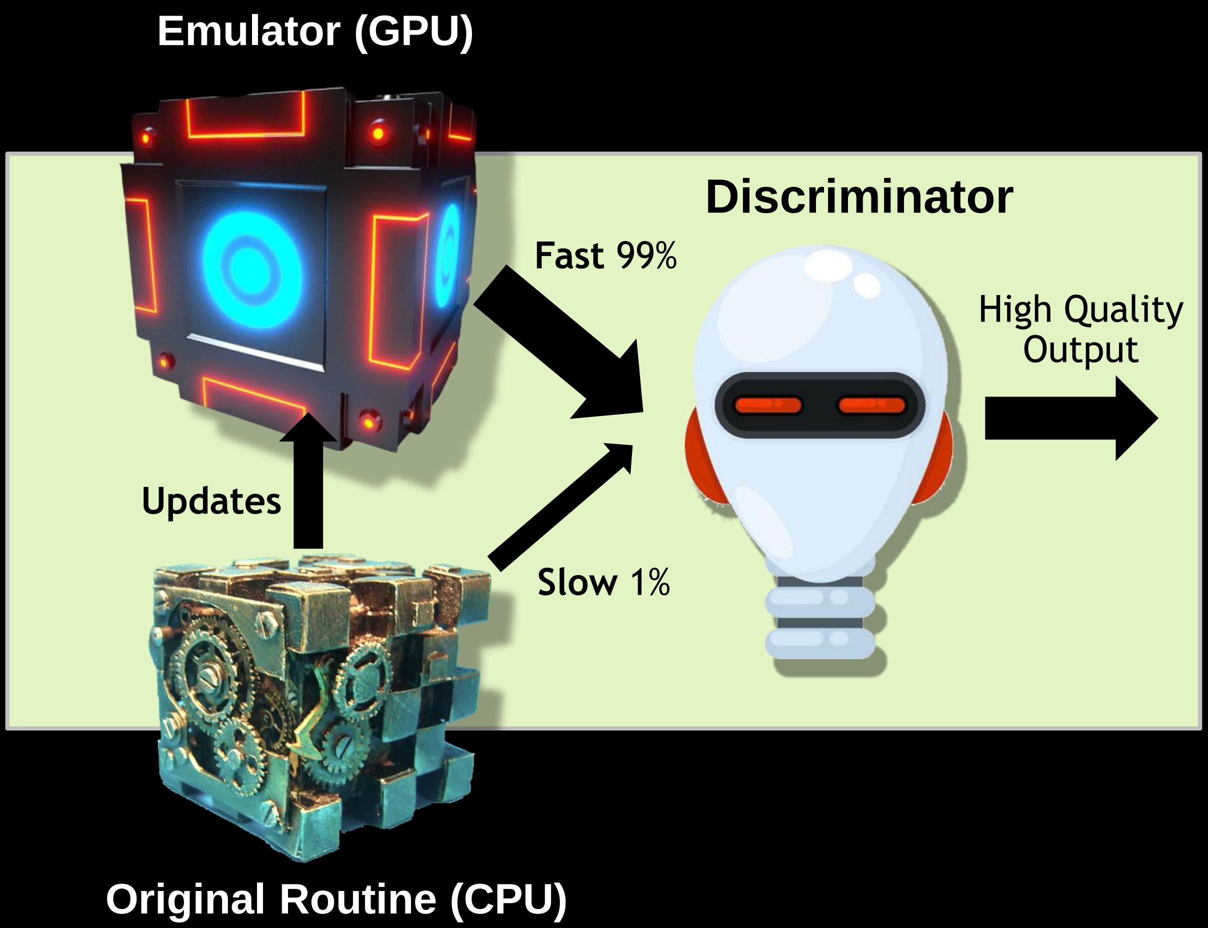
**Emulation of radiative transfer parametrization**
**E3SM global climate model**
**Speedup of 8-10x over original.**
**Details: 3778 inputs, fully connected, 3 hidden layers, 6million training samples**



# HYBRID EMULATION MODEL

## One approach to address the quality / coverage issues

- *Fast* emulation at run-time
- Discriminator ensures quality
- For new use cases:
- Discriminator flags errors
- Original routine is applied
- New output pairs cached
- Emulator weights are fine-tuned



# STOCHASTIC EMULATIONS

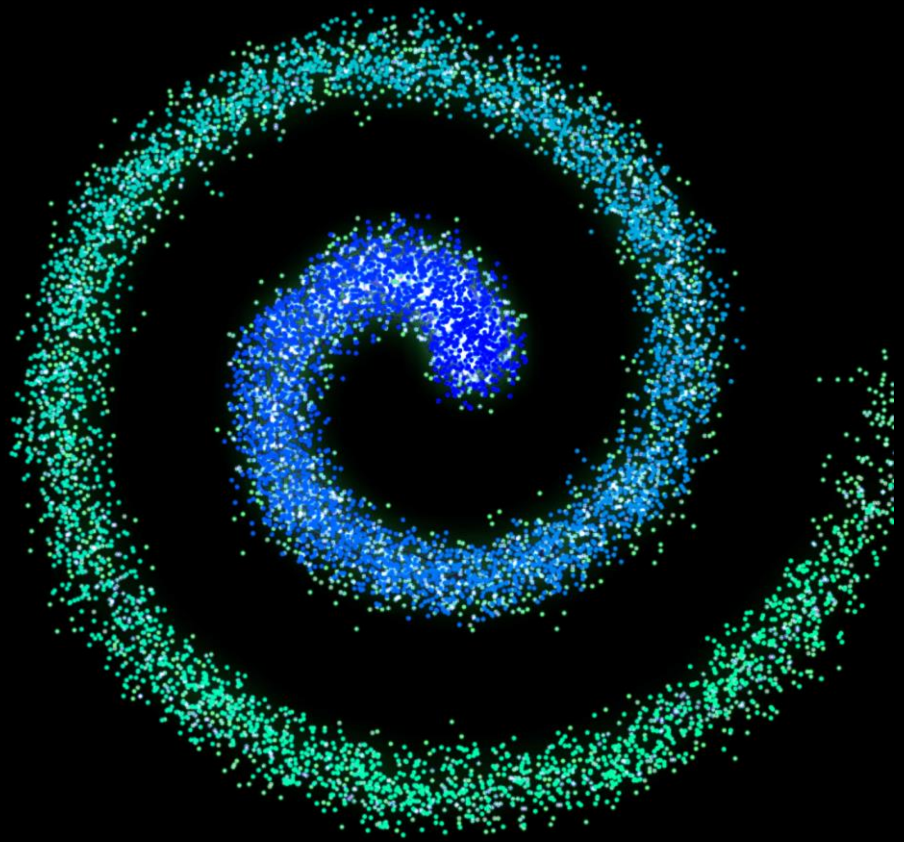
## Generative Adversarial Networks produce better emulations

Emulation via regression leads to artificially smoothed output (regression to the mean)

Use conditional GANs to stochastically sample the distribution of realizable states

More faithfully emulates the original function

Discriminator provides a natural mechanism for detecting errors



# 5. IMPROVED PHYSICAL PARAMETRIZATIONS FROM DATA

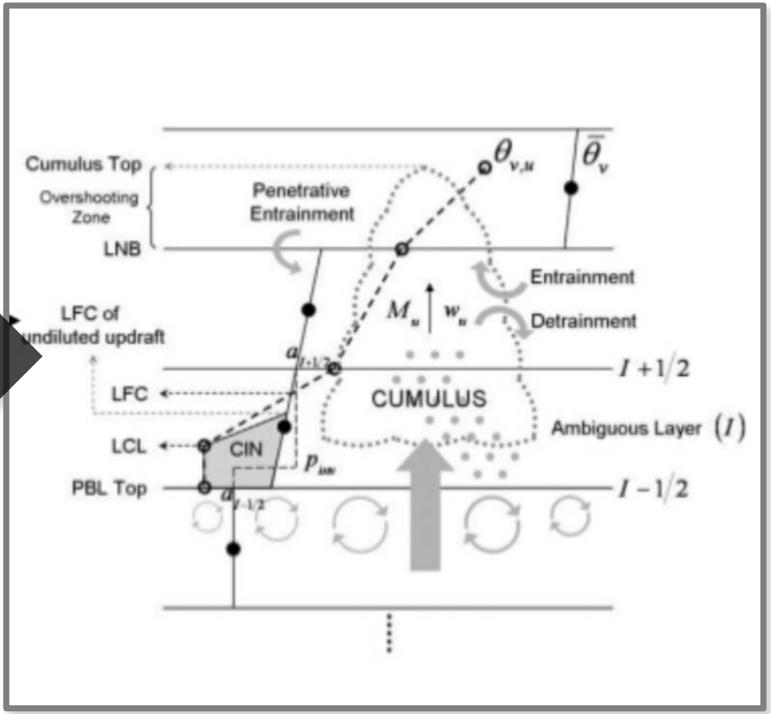
Physical parametrizations represent unresolved physics in climate and weather models. They need to be simple to be fast, and are often inaccurate approximations, hand coded by domain experts. Using deep learning, we can create more accurate parametrization directly from observational data, or from high resolution simulations.

# HOW WE USUALLY BUILD PARAMETERIZATIONS
## Expert guided physical approximation

**High resolution simulations or observations**

**Mad Scientist**

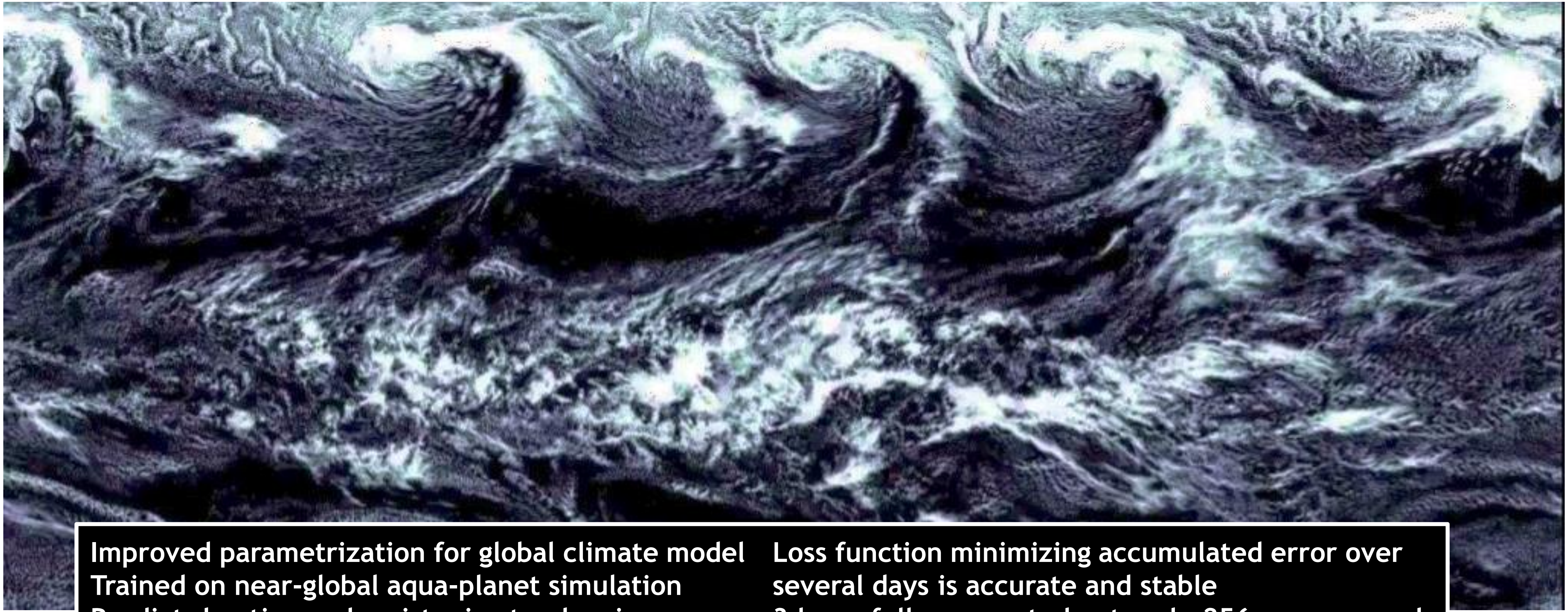
**Low Order Parametrization**



# UNIFIED PHYSICS PARAMETERIZATION

## Prognostic Validation of a Neural Network Unified Physics

Noah Brenowitz and Cristopher Bretherton, University of Washington, May 2018



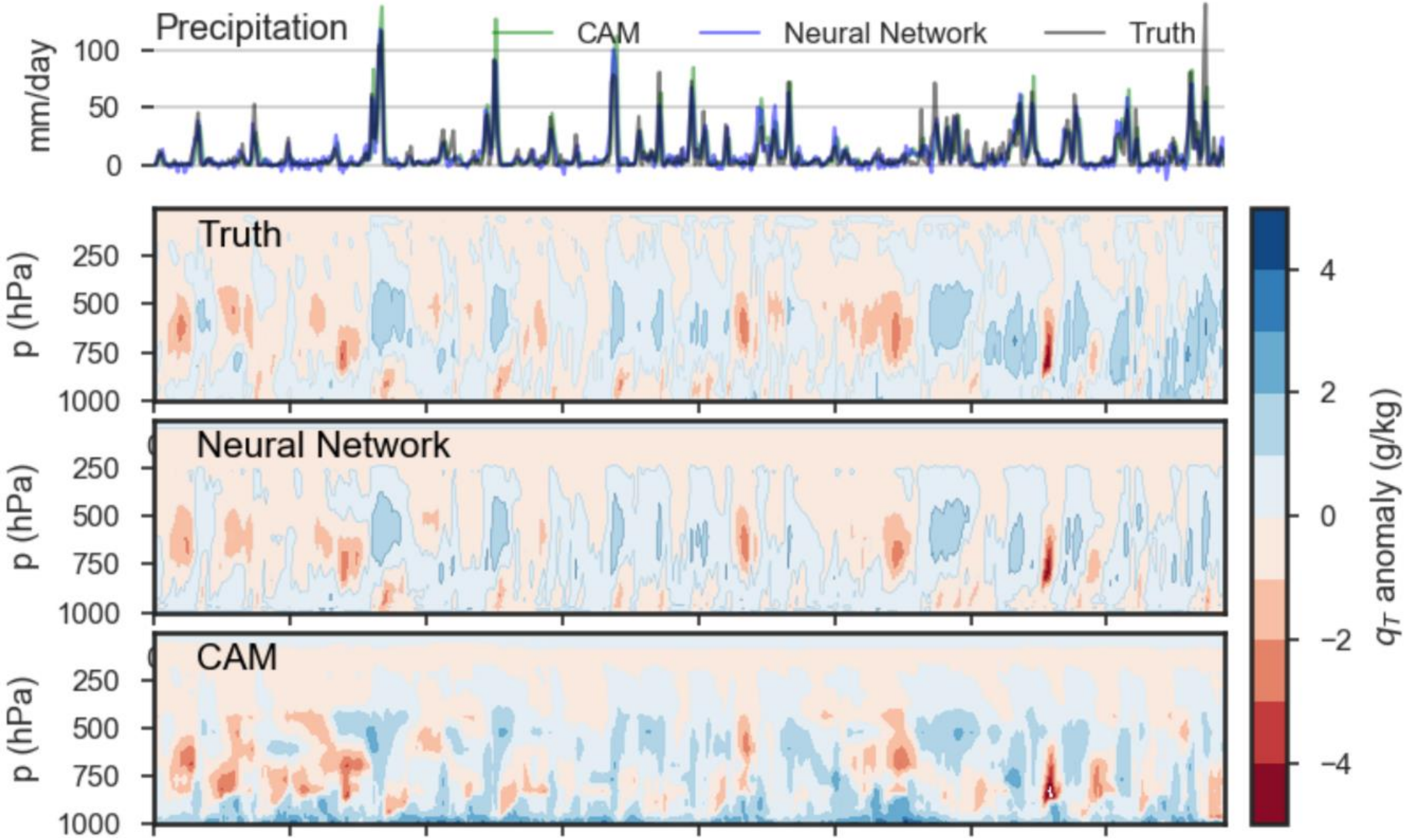
Improved parametrization for global climate model
Trained on near-global aqua-planet simulation
Predicts heating and moistening tendencies

Loss function minimizing accumulated error over several days is accurate and stable
3 layer fully connected network, 256 neurons each

# UNIFIED PHYSICS PARAMETERIZATION

## Prognostic Validation of a Neural Network Unified Physics

Noah Brenowitz and Cristopher Bretherton, University of Washington, May 2018

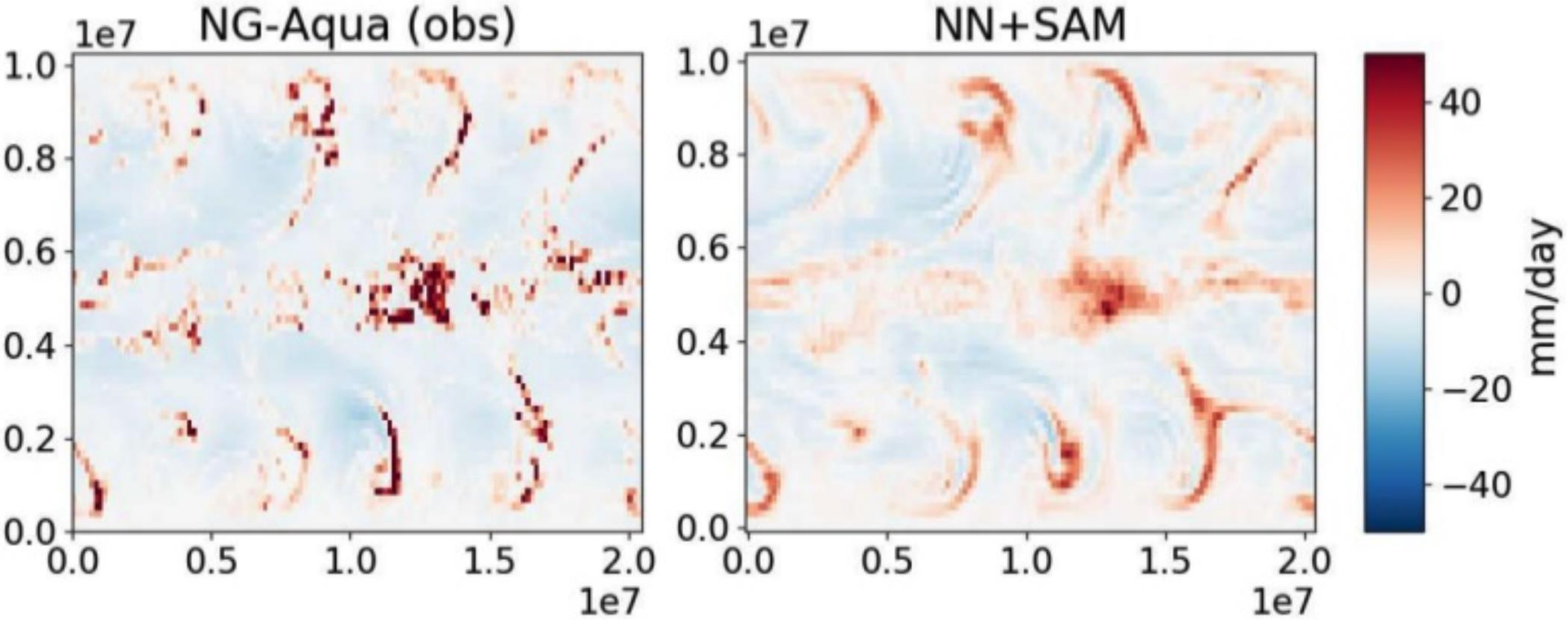



- **More accurate than CAM**
- **Improves forecast accuracy**

# UNIFIED PHYSICS PARAMETERIZATION

## Prognostic Validation of a Neural Network Unified Physics

Noah Brenowitz and Cristopher Bretherton, University of Washington, May 2018




Exhibits loss of stochasticity. (Fix using stochastic sampling based on conditional GAN)

# IMPROVED SOIL MOISTURE IN HRRR

Lidia Trailovik and Isadora Jankov NOAA ESRL

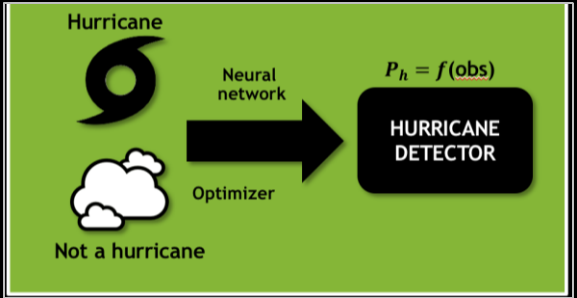
- Soil moisture is important for convection initiation
- Current parametrization in HRR is inadequate
- Create a better parametrization from field observations
- Use surface measurements to infer sub-surface state
- Mesonet weather station network provides ground truth

- Air temperature
- Barometric pressure
- Rainfall amount
- Relative humidity
- Soil moisture
- Soil temperature
- Solar radiation
- Wind direction
- Wind speed

# SUMMARY

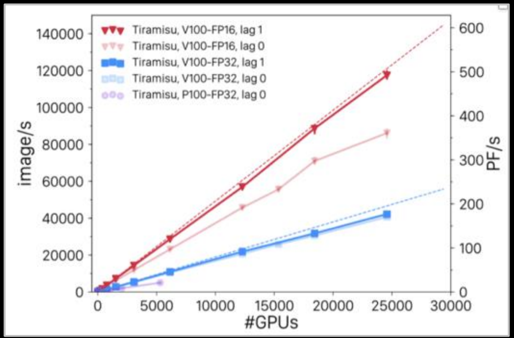
Using GPUs is critical to achieving performance gains on modern supercomputers.

Deep learning provides a new general purpose set of tools which are well suited for GPUs.

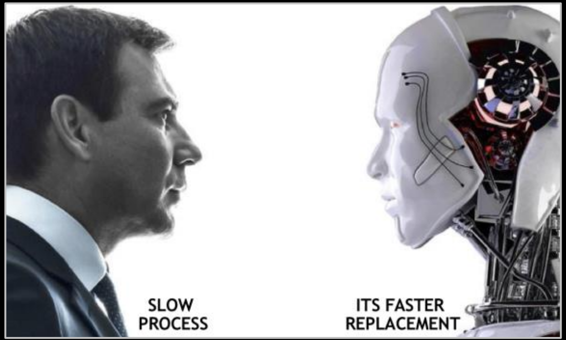



Use DL to construct functions by example, and freely mixed with traditional code.

Build software too complex or unintuitive to code by hand (like AlphaGo)

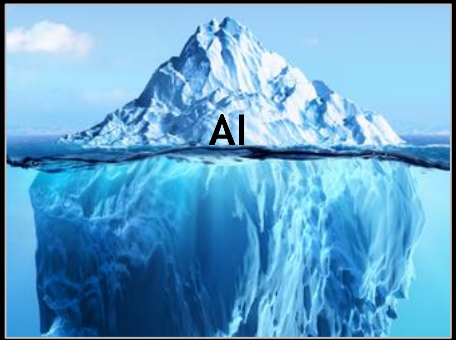



Scale trained networks up on very large systems, to analyze enormous data volumes




Emulate expensive routines, without porting code, to achieve 10x-1000x speedup (ex. inverse modeling)

Construct superior physical parameterizations directly from high resolution simulations or data

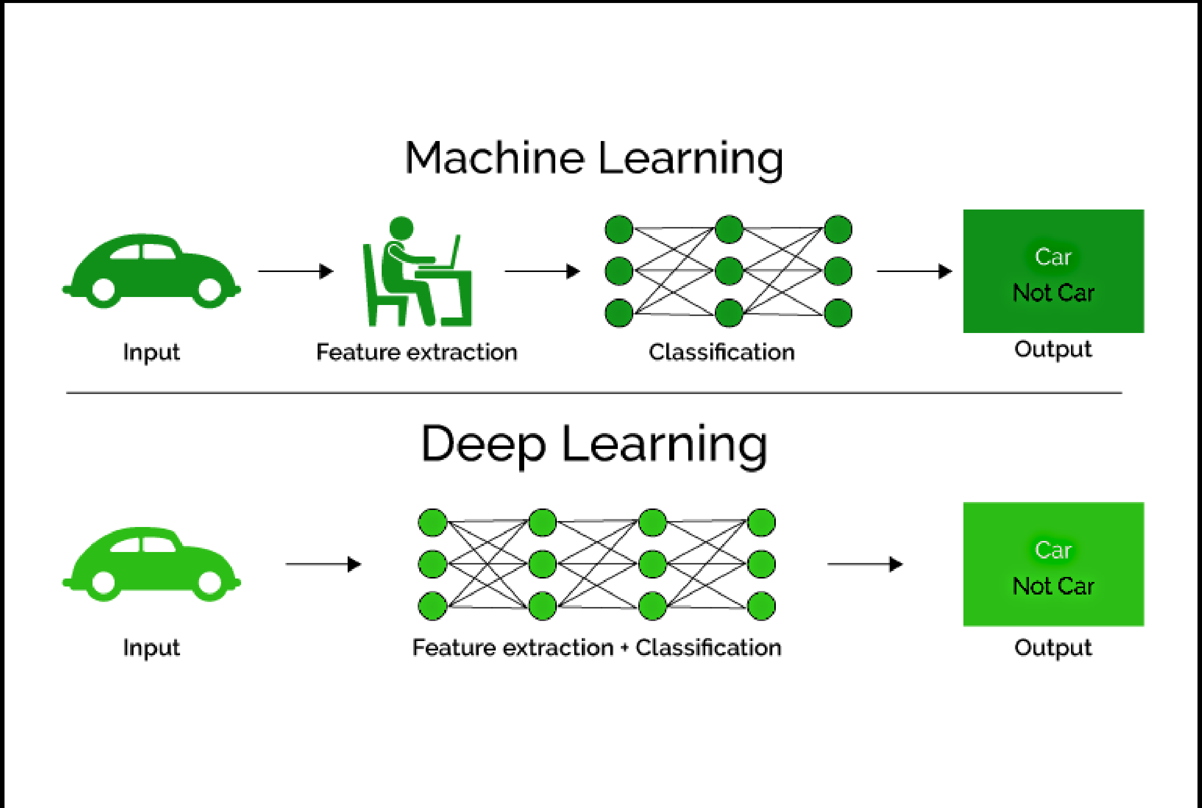



These examples are just the tip of the AI iceberg

dhall@nvidia.com

# DEEP LEARNING VS. MACHINE LEARNING

## When should I use deep learning vs classical ML?




### CLASSICAL ML

**Random forests, SVM, K-means, Logistic Regression**
Features hand-crafted by experts
Small set of features: 10s or 100s
Dataset is too small for deep learning
**NVIDIA RAPIDS:** orders of magnitude speedup

### DEEP LEARNING

**CNN, RNN, LSTM, GAN, Variational Auto-encoders**
Finds features automatically
High dimensional data: images, sounds, speech
Large set of training data (10k+ examples)
**NVIDIA CU-DNN:** accelerates DL frameworks

# SCIENTIFIC CHALLENGES

## Barriers to acceptance of deep learning as a tool for science

- **Interpretability:** **Can I understand what the neural-net is doing?** (Explainable AI)
- **Robustness:** **Will it always give me the right answer?** (GAN discriminator)
- **Conservation:** **Does it conserve mass, momentum, energy?** (Lagrange multiplier)
- **Coverage:** **How much training data do I need?** (Hybrid solution)
- **Convergence:** **How can I ensure that training will converge?** (regress then GAN)
- **Uncertainty:** **How certain can I be of the answers?** (Measure covariance)