NCAR UCAR

air • planet • people

Tackling the Simulation and Analysis Frontiers of Atmospheric and Earth System Science

Dr. Rich Loft
Director, Technology Development
Computational and Information Systems Laboratory
National Center for Atmospheric Research

ECMWF HPC Workshop
September 26, 2018

NSF UCAR NSF

Outline

- The Challenges
- Accelerators
- Big Data
- Machine Learning

Optimizing CESM-2 for Intel Xeon performance

IPCC-WACS RoI: higher efficiency = more science

CESM Configuration	Atmos Resolution (km)	Ocean Resolution (km)	Speedup (%)
Low-res IPCC	100	100	13%
WACCM chemistry	100	100	11%
High-res IPCC	25	100	25%
Ultra-high Ocean eddy permitting	25	10	35%

100 km CESM on Cheyenne: greater capability

NCAR System	Intel Xeon Processor	CESM Version	Capability (sim yr/day)	Cost (core-hr per sim yr)
Cheyenne	18c (v4)	CESM2	30	3500
Yellowstone	8c (v2)	CESM2	19.6	5167
Yellowstone	8c (v2)	CESM1	10.6	1521

100 km Take-aways

- **CESM is 48% more efficient on Cheyenne compared to Yellowstone.**
- **CESM-2 on Cheyenne can deliver 2.8x the capability, compared to CESM1 on Yellowstone.**

- **$285K:** Estimated total cost to provision 1% more climate computing over Cheyenne's 4-year life.
- **$3.1M to $9.9M:** Total valuation of CESM improvements at NCAR, depending on the use case. Total valuation to the climate community of this work is likely even greater.

Great success, but…

- To improve earth system model integration rate by **10x** following this trajectory, will take **22 years**.
- **Additional bottlenecks:**
  - Long verification runs consume computer resources
  - Models are becoming harder to tune.
  - Too much data!
  - Analytics: serial, inefficient, cumbersome to adapt
- Frontiers of meteorological prediction (on sub-seasonal to seasonal timescales) are becoming earth system models.

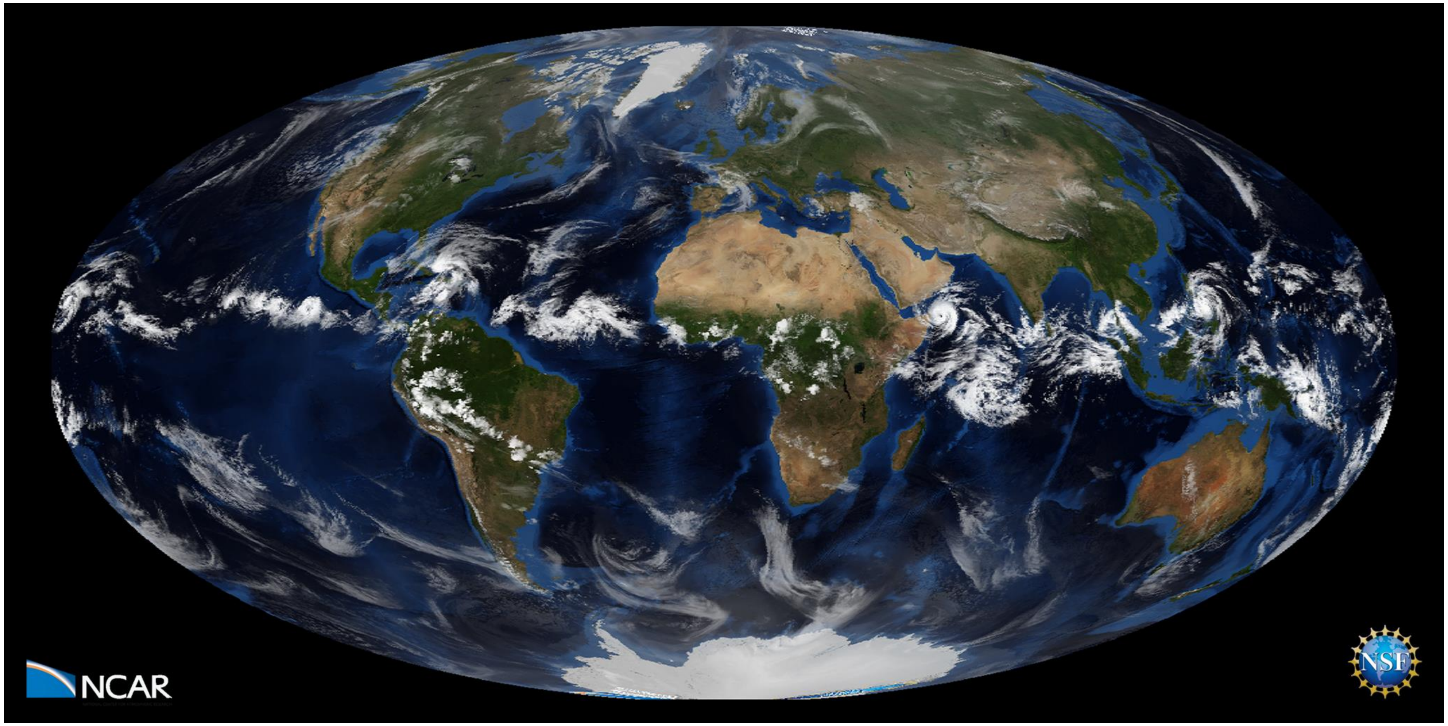
Talk Summary

- **Science 3.0: HPC + ML**
- NCAR's Computing Lab (CISL) is betting on the merging of accelerated HPC and Machine Learning (e.g. through GPUs) to accelerate the modeling enterprise.
- On the data side, our acceleration strategy is to pursue lossy compression, machine learning-enabled parallel analytics software, all with supporting high-IOPS-enabled SSD storage infrastructure.
- Initial results are encouraging…
- But much more needs to be done to prove these ideas out!

Outline

- The Challenges
- Accelerators
- Big Data
- Machine Learning

Model for Prediction Across Scales (MPAS)
A Global Meteorological & Future Climate Model Component

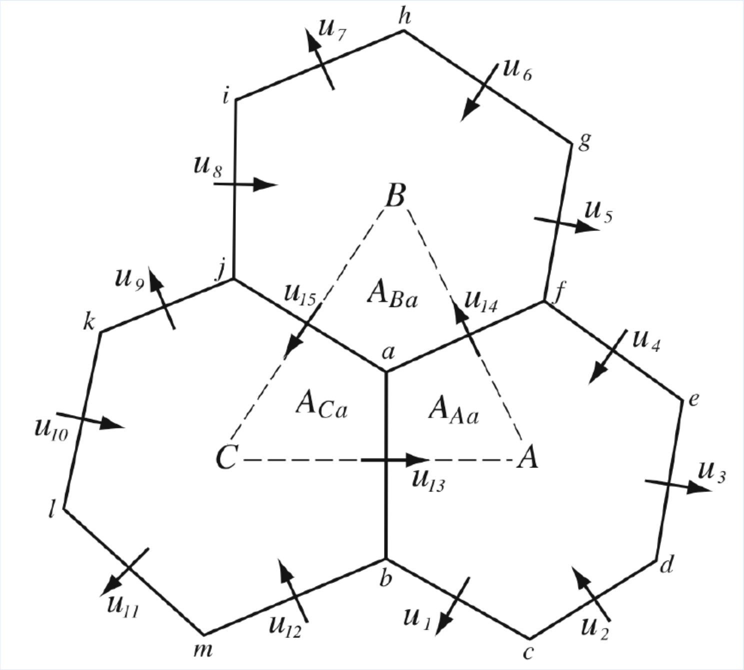


Simulation of 2012 Tropical Cyclones at 4 km resolution – Courtesy of Falko Judt, NCAR

MPAS: the algorithmic description

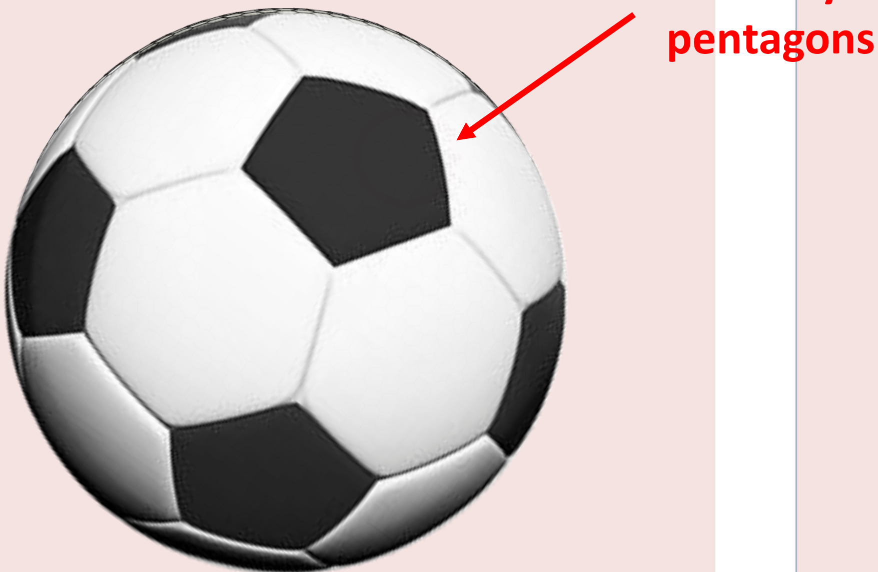
- **Fully compressible non-hydrostatic equations written in flux form**
- **Finite Volume Method on staggered grid**
  - The horizontal momentum normal to the cell edge (u) is sits at the **cell edges**.
  - Scalars sit at the **cell centers**
- **Split-Explicit timestepping scheme**
  - Time integration 3rd order Runge-Kutta
  - Fast horizontal waves are sub-cycled

MPAS is based on unstructured centroidal Voronoi (hexagonal) meshes using C-grid staggering and selective grid refinement.



MPAS: the grids…

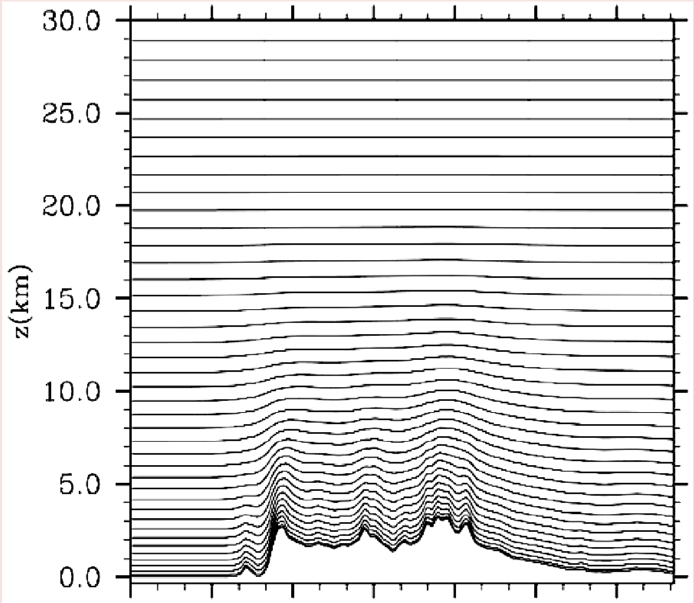
Horizontal



MPAS
Unstructured Voronoi (hexagonal) grid

- Good scaling on massively parallel computers
- No pole problems

Vertical



MPAS
Height-based hybrid smoothed terrain-following vertical coordinate

- Improved numerical accuracy

MPAS: The Code inventory

- **Dynamics Solver** ~10,000 SLOC
- **Physics** ~ 100,000 SLOC
  - Radiative Transport: ~37,000 SLOC
  - NOAH Land Surface Model: ~21,000 SLOC
  - Other physics code: ~42,000 SLOC
- **Time evenly split between dynamics and physics**

Goals of MPAS-GPU Portability Project

- Achieve portability across Xeon, Xeon Phi* and GPU architectures without sacrificing CPU performance
- Minimize use of architecture-specific code:

```
#ifdef _GPU_
   :
#elseif _CPU_
   :
#endif
```

- Manage porting/optimization costs
  - Use OpenACC to enable CPU-GPU portability
- Use all the hardware (CPU & GPU) available
  - After all we pay for it!

* I know…

MPAS refactoring strategy

- **Use all the hardware (CPU & GPU) available**
- **CPU resident**
  - NOAH LSM is large, branchy and inexpensive -> CPU
  - RT is large, expensive but can run asynchronously -> CPU
  - I/O should be asynchronous -> CPU
- **GPU resident**
  - Dry/moist dynamics
  - All other physics

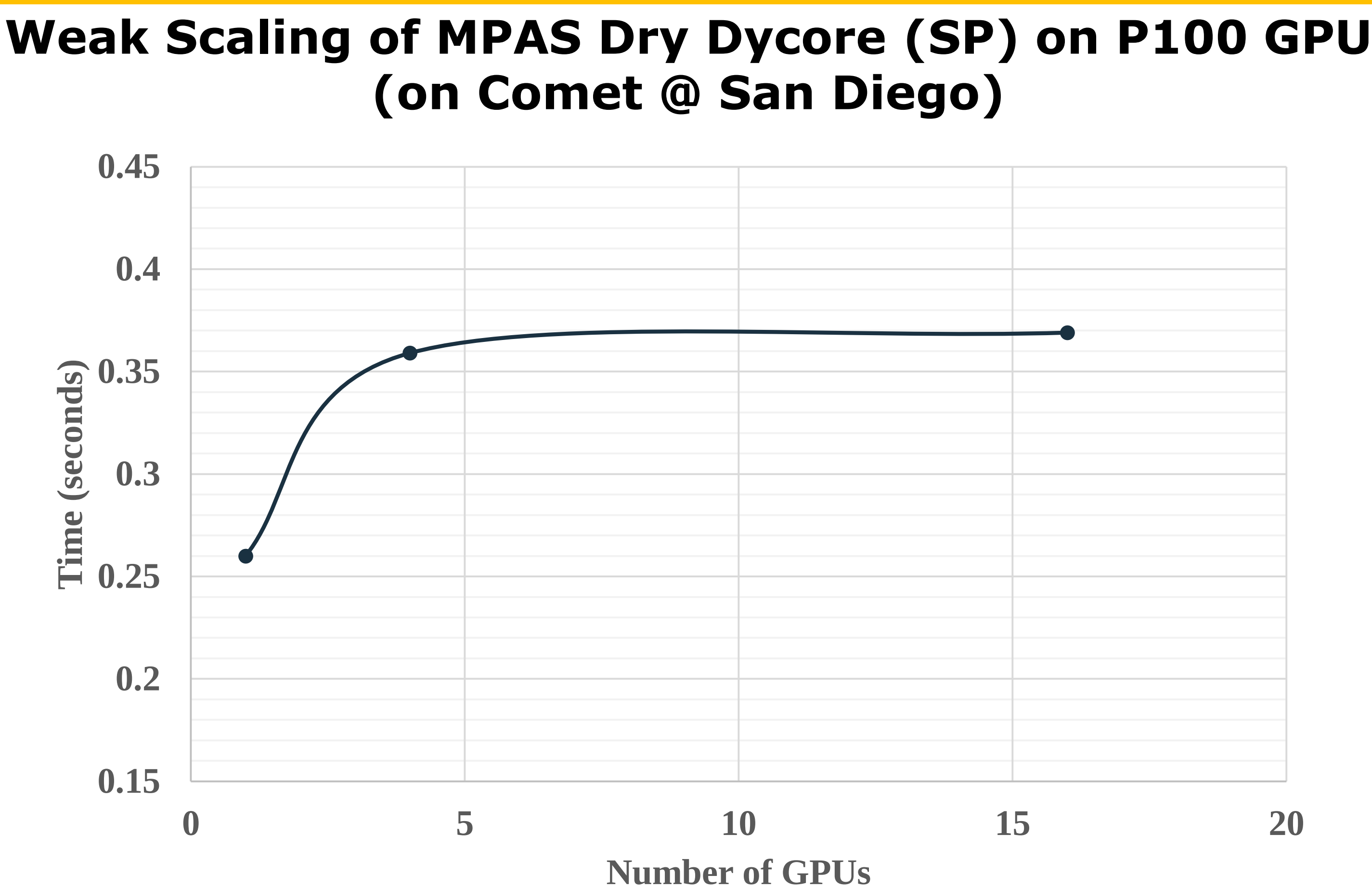
MPAS Dycore: Single Node Performance

- **Timers**
  - MPAS GPTL timers reported in log files
- **GPU Timing : Has no updates from device to host**
  - Host updates maybe needed for printing values on screen
  - Host updates maybe needed for netcdf file output

Dataset		Broadwell (Fully Subscribed, OpenMP Enabled, Intel compiled, Base code)	P100 with Haswell(1 GPU, PGI compiled, OpenACC code)	V100 with Skylake (1 GPU, PGI compiled, OpenACC code)	Speedup Broadwell vs P100	Speedup Broadwell vs V100	Speedup P100 vs V100
120 Km (40K)	SP	0.46	0.25	0.16	1.84	2.95	1.60
	DP	0.94	0.42	0.23	2.24	4.09	1.83

Broadwell Node vs one GPU device

MPAS Dycore: Multi-GPU Performance



Time per timestep, 4 GPUs per node, 1 MPI rank per GPU, Max of 4 MPI ranks per node, Intranode Affinity for MPI ranks, Uses OpenMPI, PCIe no NVLink, PGI 17.10

MPAS Software Stack: Physics

- **Software**
  - MPAS 6
  - Intel Compiler 17.0, PGI Compiler 17.10
- **Full physics suite**
  - Scale-aware Ntiedtke Convection, WSM 6 Microphysics, Noah Land surface, YSU Boundary Layer, Monin-Obhukov Surface layer, RRTMG radiation, Xu Randall Cloud Fraction
  - Radiation interval: 30 minutes
  - Single precision (SP)
  - Verification in progress, **performance measured in wall clock seconds per timestep**

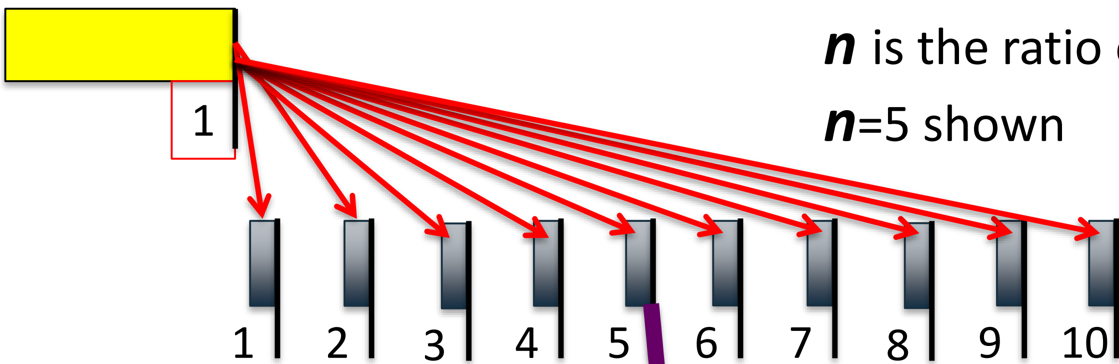
MPAS Physics: Single Node Performance

- **Timers**
  - MPAS GPTL timers reported in log files
- **GPU Timing : Excludes Host-Device-Host data transfer time**
  - Physics is yet to be verified, hence data copy directives are not removed

	Single node, Integrated Code Time step Time (seconds)			
Physics Module	**Broadwell**	**P100**	**V100**	**Broadwell vs V100**
WSM6	0.255	0.096	0.080	3.2
YSU	0.013	0.008	0.007	2.0
Monin Obukhov	0.001	0.001	0.001	0.9

Broadwell Node vs one GPU device

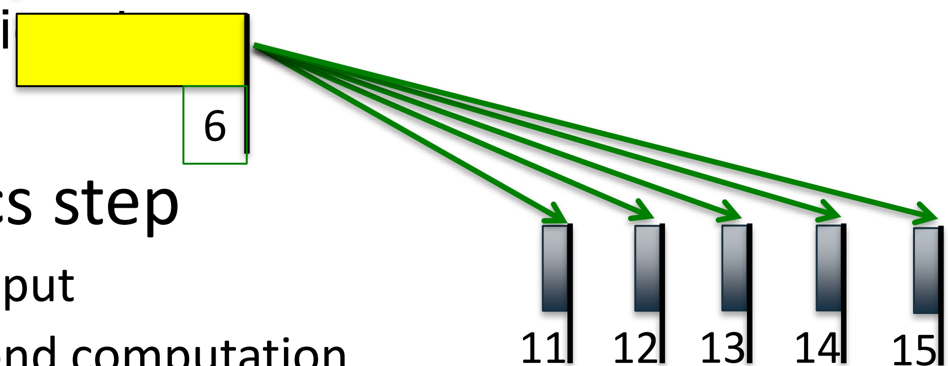
Scheme to Overlap Radiation with Dynamics Solver Execution



n is the ratio of rad to dyn timesteps

n=5 shown

- First radiation step provides information to first ***2n*** dynamics and physi



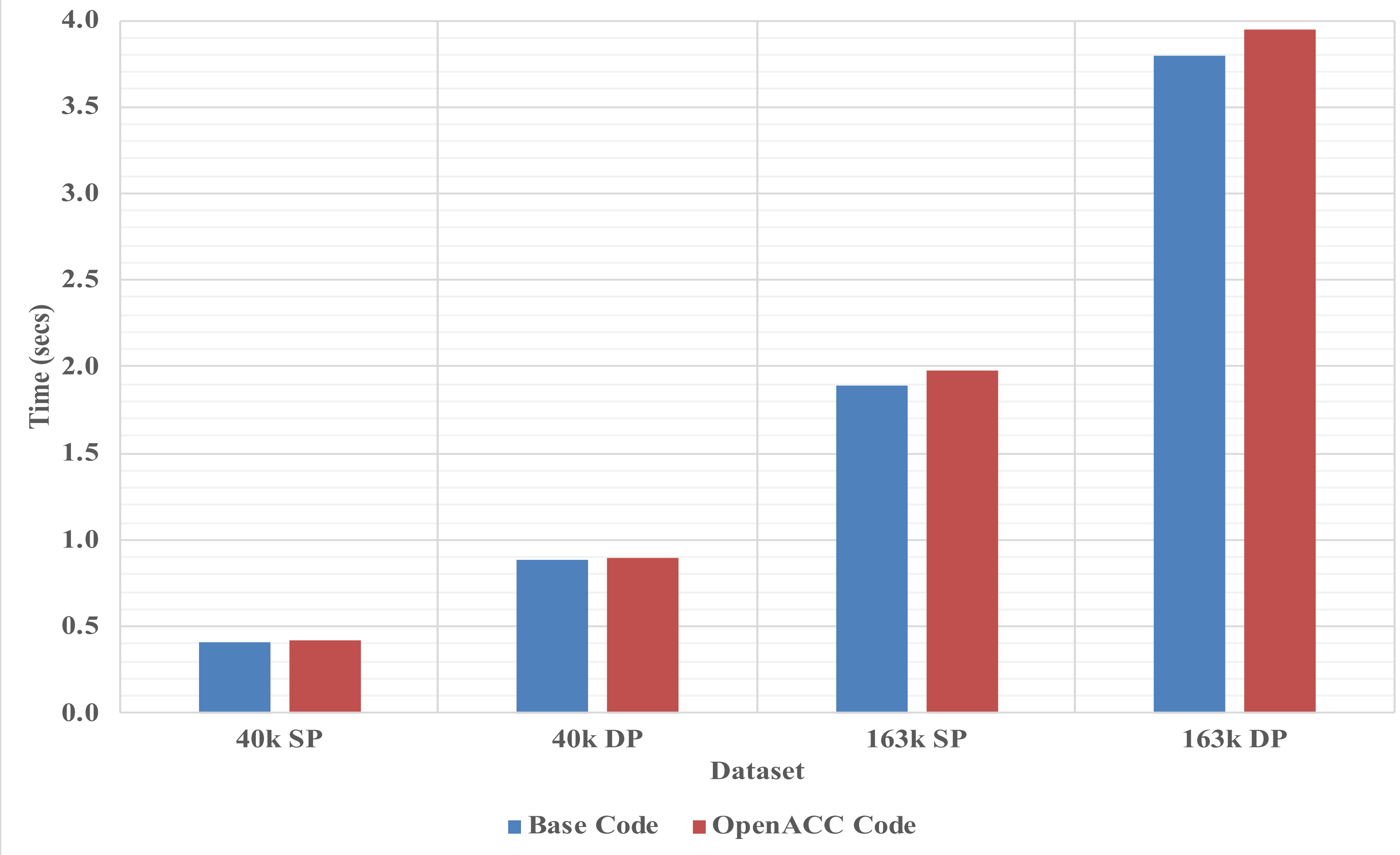
- After ***n*th** dynamics step
  - Radiation receives input
  - Radiation starts second computation
  - Dynamics keeps processing

Full MPAS Multi-GPU: State of Play

Model component	BW node (sec)	P100 optimized & verified (sec)	P100 optimized (sec)	P100 Ported (sec)	Best time (sec)	Comments
Dry dynamics	0.40	0.36			0.36	With MPI overhead
Moist dynamics	0.05	0.030	0.025		0.025	
WSM6	0.23		0.096	0.18	0.096	
YSU	0.012			0.0082	0.0082	
Monin Obukhov	0.0004			0.00094	0.0004	Review port
RRTM	0.35	N/A	N/A	N/A	0.0405	est. PCIe oh
New Tiedtke (scale ins.)	0.04				0.04	Port in progress
NOAH LSM	0.0005	N/A	N/A	N/A	0.0005	On CPU
CPU sec/step Total	1.19			**GPU sec/step**	0.57	@120 km
BW node/P100 (with RT)	2.08	**BW node/P100 (without RT)**	1.58			

Where does MPAS-GPU stand regarding *Performance* Portability?

Performance Portability: Have we “broken” CPU performance?



For datasets up to 40k per node, the execution time is identical.
For 40k & 163k, the variation is <1% & <4% respectively.

MPAS GPU team

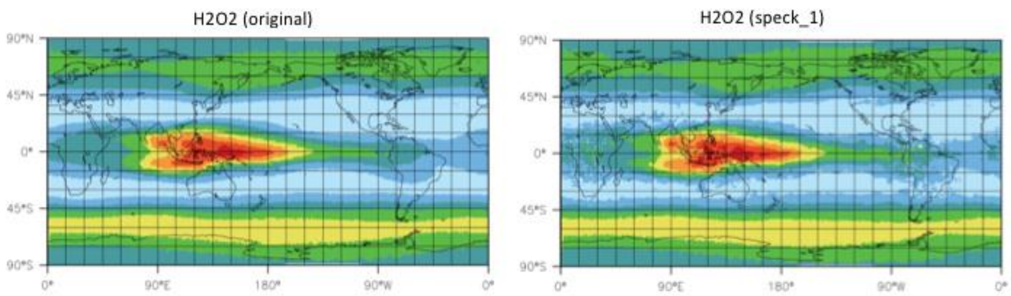
- **NCAR**
  - Dr. Raghu Raj Kumar, CISL
  - Supreeth Suresh, CISL
  - Michael Duda, Software Engineer, MMM
  - Dave Gill, Software Engineer, MMM
- **NVIDIA/PGI**
  - Dr. Carl Ponder, Senior Applications Engineer
  - Brent Leback, PGI Compiler Engineering Manager
  - Craig Tierny, Applications Engineer
- **University of Wyoming**
  - Prof. Suresh Muknahallipatna
  - GRAs: Pranay Reddy, Sumathi Lakshmiranganathan, Cena Miller, Bradley Riotto, Clint Olsen
  - Undergrads: Aisha Mohamed, Brett Gilman, Briley James, Suzanne Piver
- **Korean Institute of Science and Technology Information**
  - Jae Youp Kim, GRA
- **IBM/TWC**
  - Constantinos Evangelinos, IBM Researcher

Outline

- The Challenges
- Accelerators
- Big Data
- Machine Learning

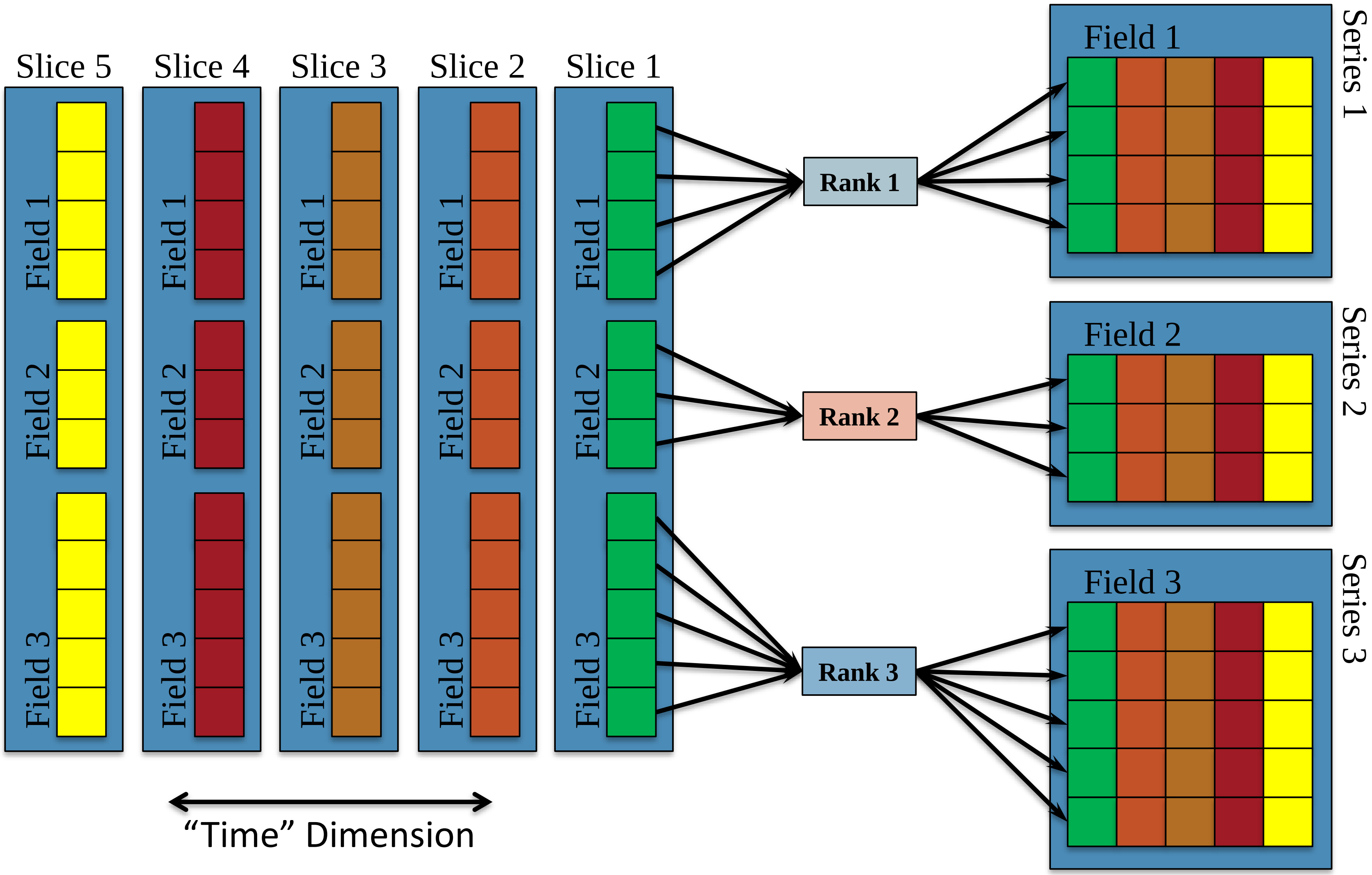
Lossy Data Compression

- **Motivation**
  - Increasing resolution and computational power lead to more and more data. *And there is no end in sight!*
  - Can we use lossy compression to reduce climate storage needs *…quickly, and without (negatively) impacting science results?*
- **Breakthrough**
  - An average 5x history file compression factor observed using ***fpzip***
- **Complications**
  - Max compressibility characteristics of variables differ a lot.
  - Different compression algorithms better suited to certain variables.
  - Ideal to use a set of methods tailored to each variable.



Compression result using Speck multi-wavelet method.

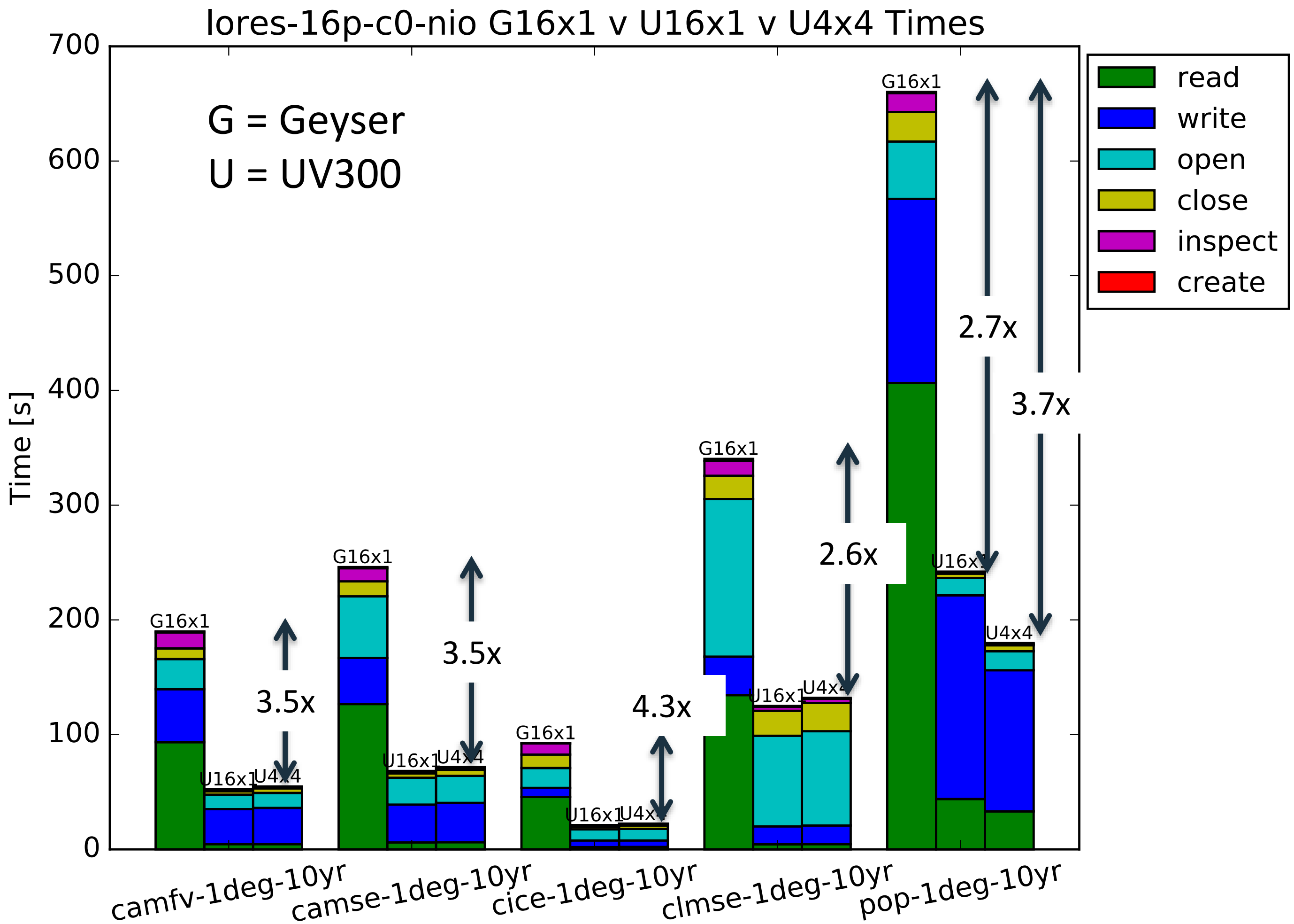
PyReshaper: parallel history files to timeseries converter



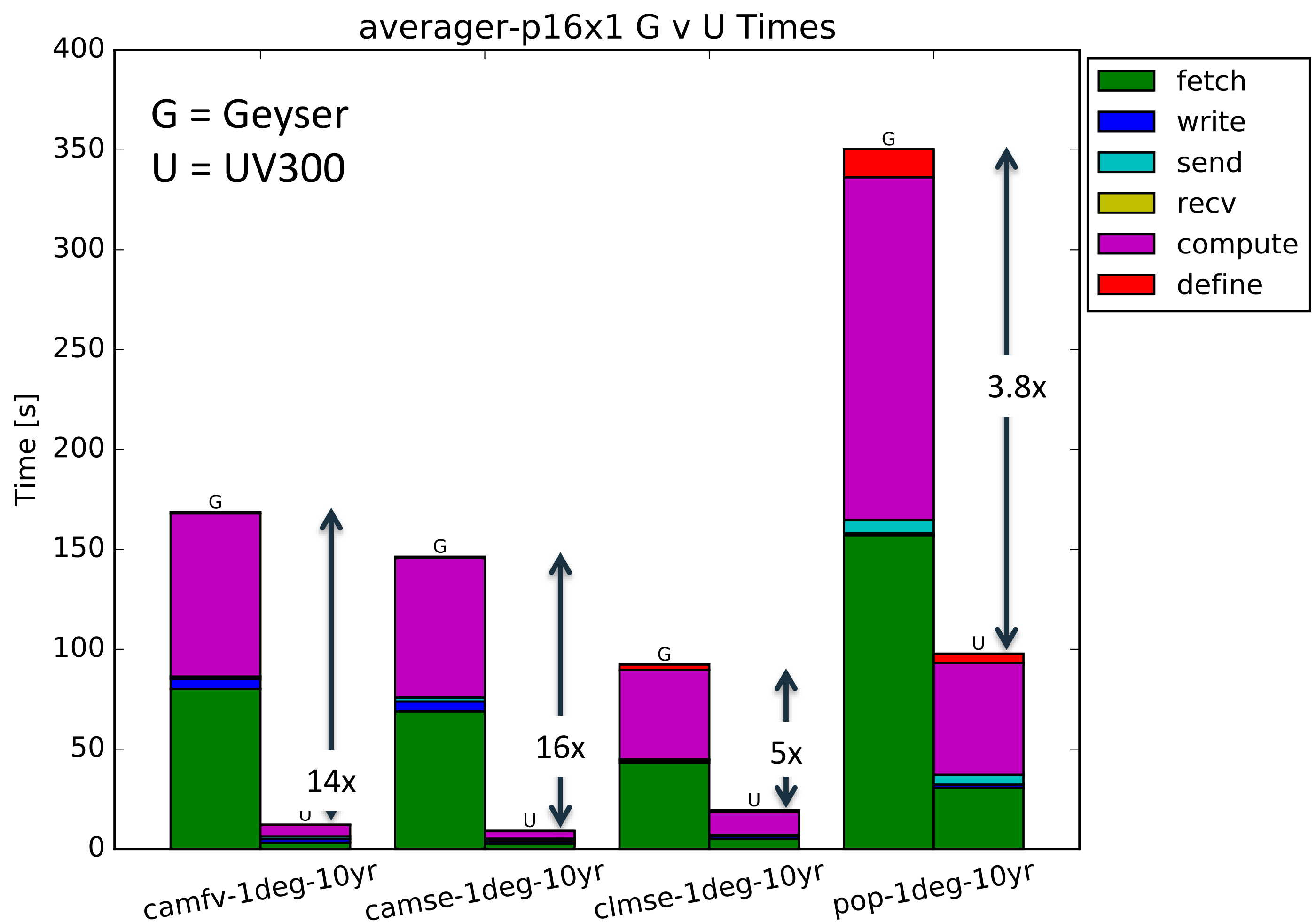
PyReshaper: parallel speedups relative to NCO operators

- 10x speedup on 0.25° CAM-SE dataset
  - NCO 4.75 hours
  - Pyreshaper 28 minutes
- 8x speedup on 0.10° POP dataset
  - NCO 14.5 hours
  - PyReshaper 1.7 hours

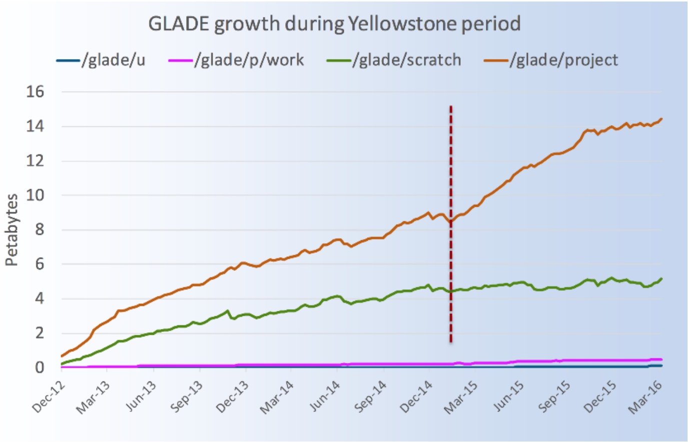
Multiplicative PyReshaper Speedup on SSD



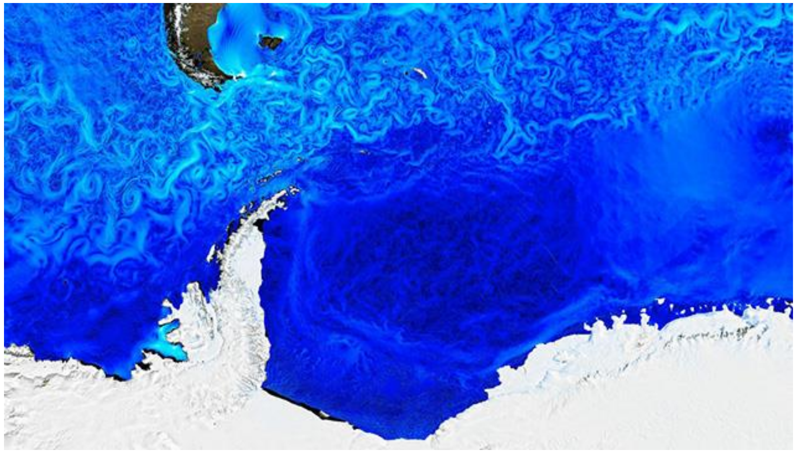
PyAverager Speedup on SSD



Pangeo Data: Toward a Big Data Analysis Platform



Petabyte-scale data volumes are straining CISL's infrastructure



Scalable analytics solutions are required to work with large datasets

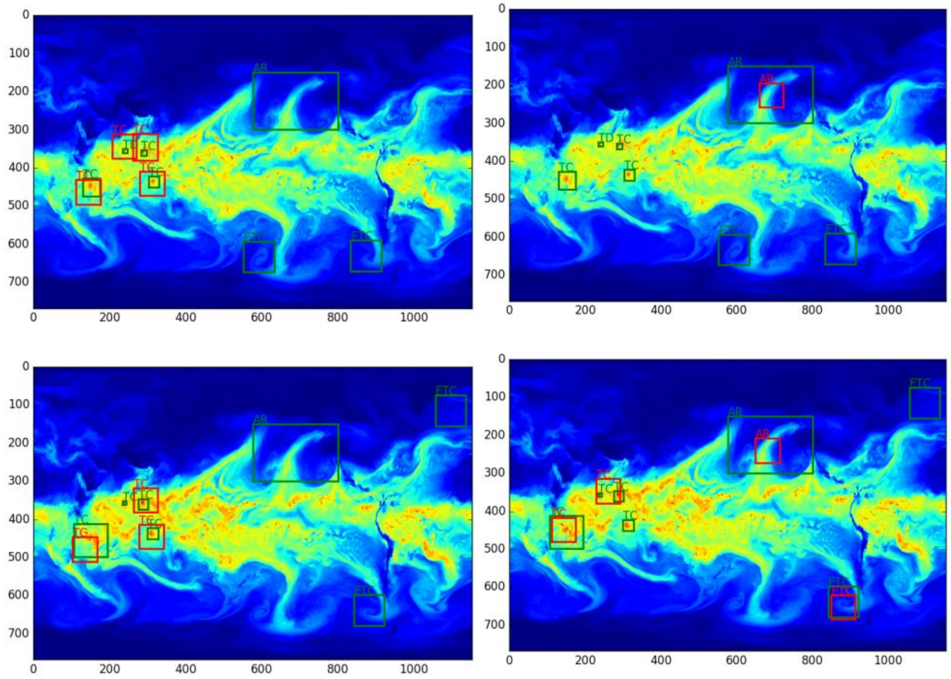
Pangeo Goal: create an open-source toolkit for the analysis of climate datasets, built on the **Python** language ecosystem, **Xarray** multi-dimensional array tools, and **Dask** parallel analytics system.

Parallelism is key: single device performance is falling behind!

Outline

- The Challenges
- Accelerators
- Big Data
- Machine Learning

ML Applications: Feature Recognition and Tracking



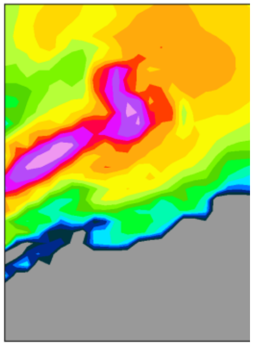
From Racah et al. 2017. https://arxiv.org/abs/1612.02095

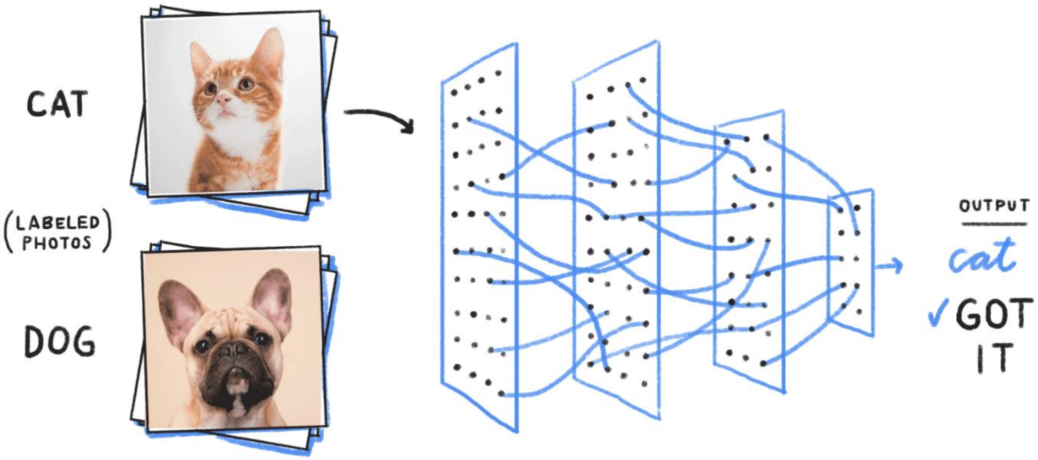
Slide courtesy of D.J. Gagne, NCAR

- **Goal: identify and track weather features at different scales**
- **Examples**
  - Tropical cyclones
  - Fronts
  - Supercells
  - Atmospheric rivers
- **Group at LBNL/NERSC**
  - Trained convolutional neural networks to identify tropical cyclones and the area covered by them
  - Used semi-supervised learning approach to encode spatial data
- **Requires hand-labeled data for training**

Model Post-Processing

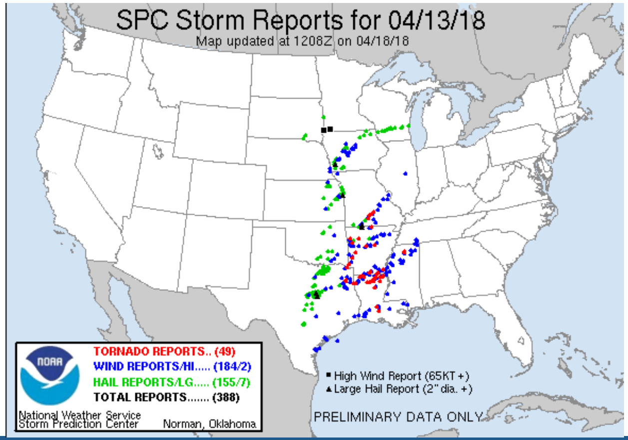
Numerical
Model Out





ility
il

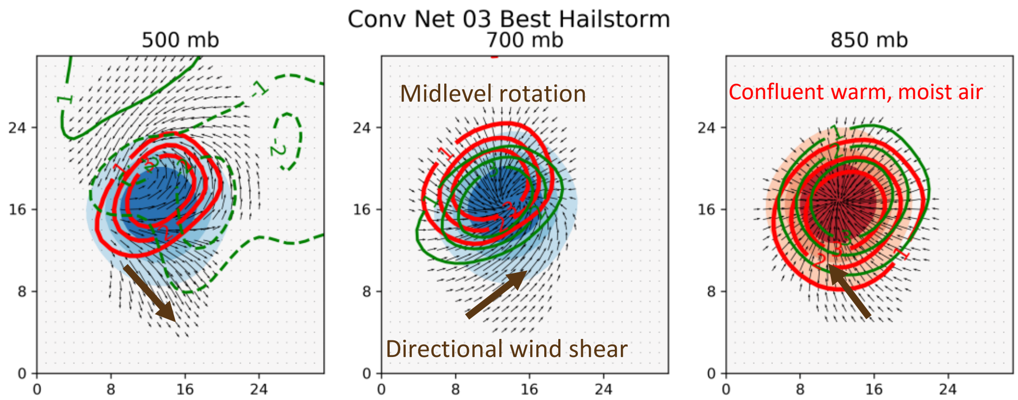
Idea: use neural networks to enhance the prediction of damaging hail



Slide courtesy of: D.J. Gagne, NCAR

Interpretability: Do Neural Nets dream of electric hail storms?

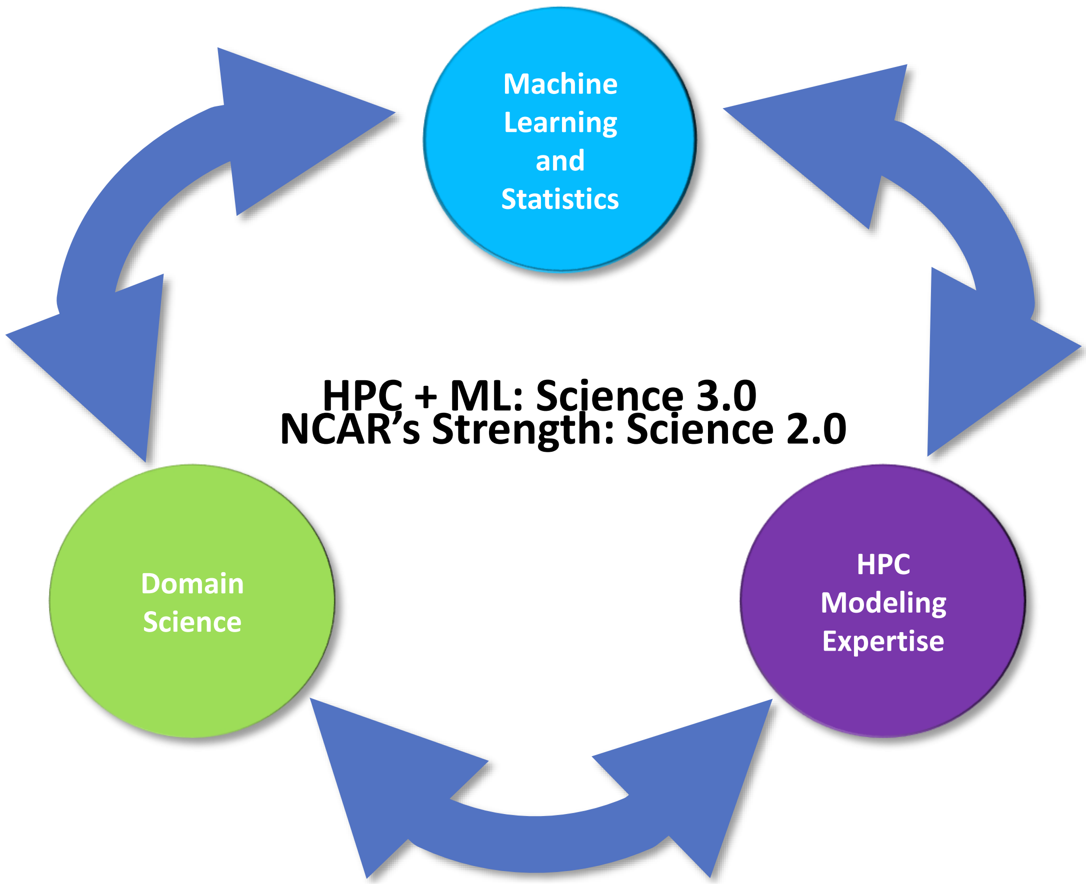
Neural network identify physically relevant features for hailstorm prediction from core weather fields. Running the network in reverse reveals these features.



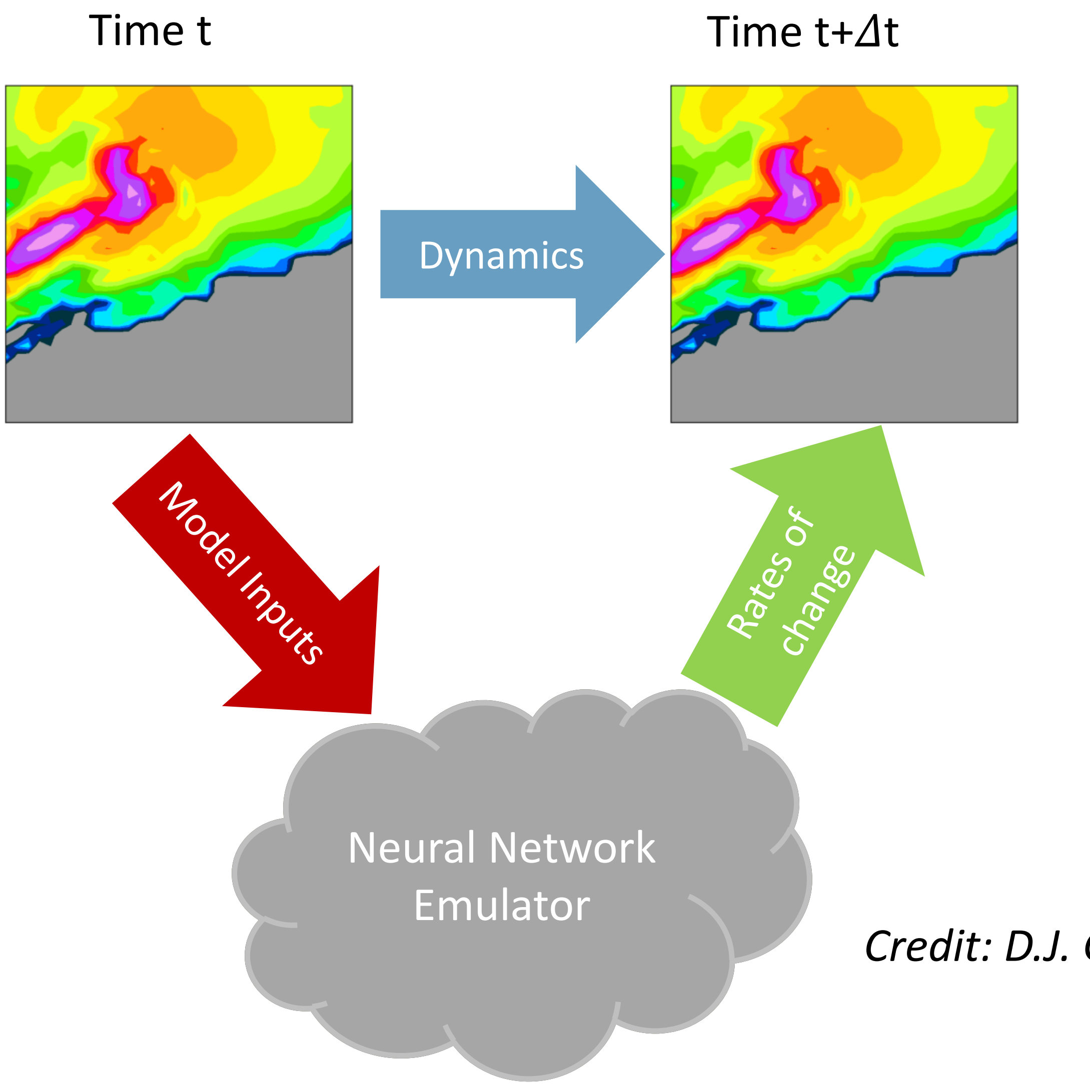
Red: Temperature
Green: Dewpoint
Filled: Height

Courtesy of Dr. D.J. Gagne, NCAR

Science 3.0: Blending Machine Learning and Traditional HPC



Replacing Models with Emulation

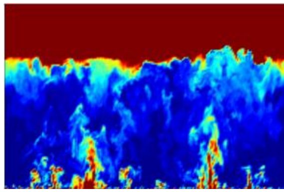


Credit: D.J. Gagne, NCAR

AIML: New Machine Learning Group in CISL

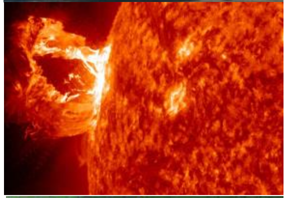
AIML Founding Research Focus: model emulation

Why machine-learned emulation? The *per-core performance* of conventional computer architectures has stagnated, and models are getting *increasingly complex*. Replacing human-crafted parameterizations with machine learning algorithms may simplify, accelerate and improve models.



- **Sub-grid-scale turbulence -Drs. Kosovic & Haupt (RAL), Gagne (AIML)**
  - **improved representation of the surface layer in meteorological models**

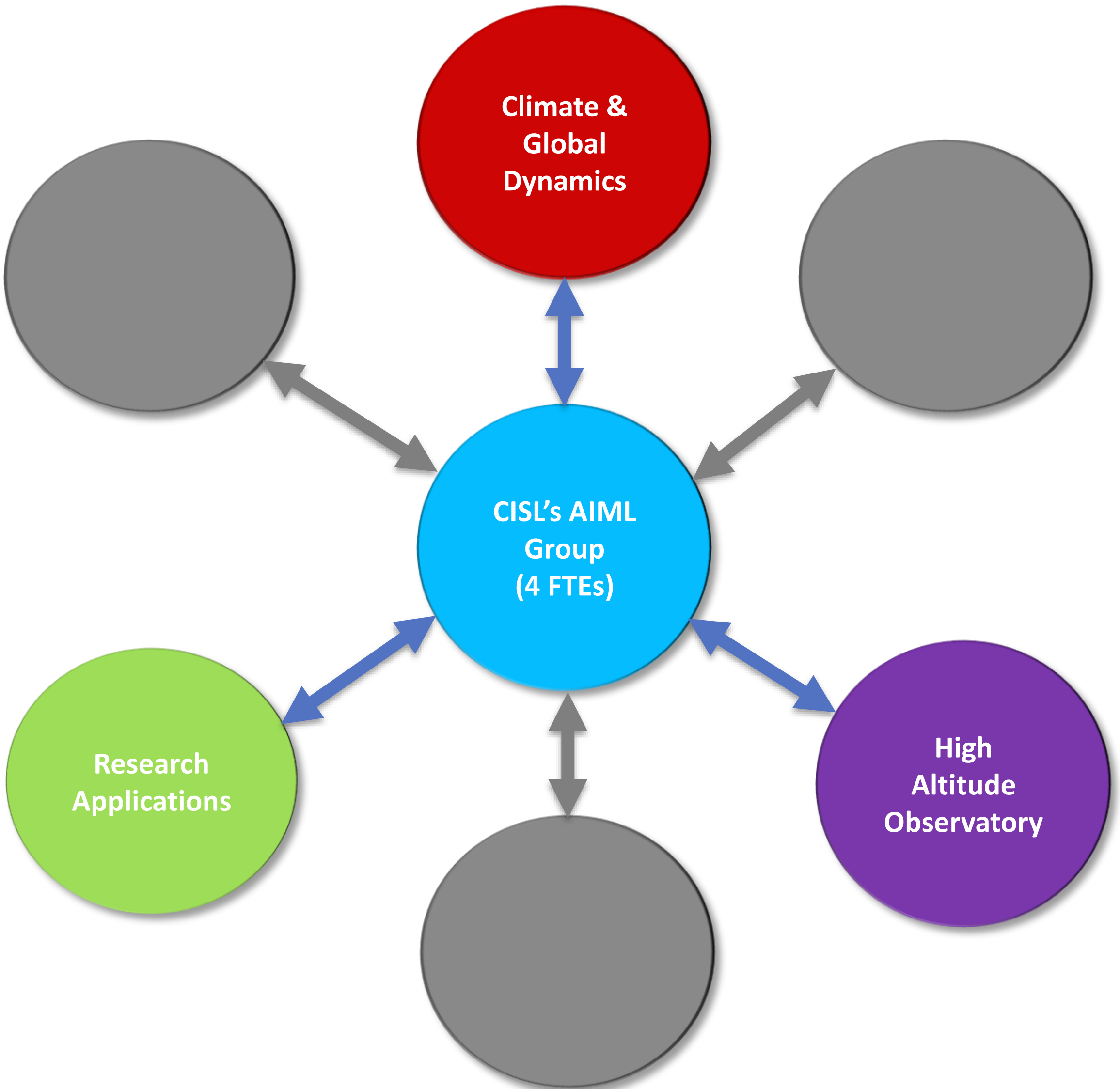
- **Cloud microphysics - Drs. Gettelman (CGD), Gagne & Sobhani (AIML)**
  - **improved weather and climate modeling**



- **Interplanetary coronal mass ejection (CME) - Drs. Gibson (HAO), Flyer (AIML)**
  - **space weather prediction**

- **Seasonal weather patterns - Drs. Sobhani (AIML) & DelVento (CISL)**
  - **Seasonal prediction of dangerous hot weather in the Eastern U.S.**

AIML: An ML+HPC Research Hub for NCAR

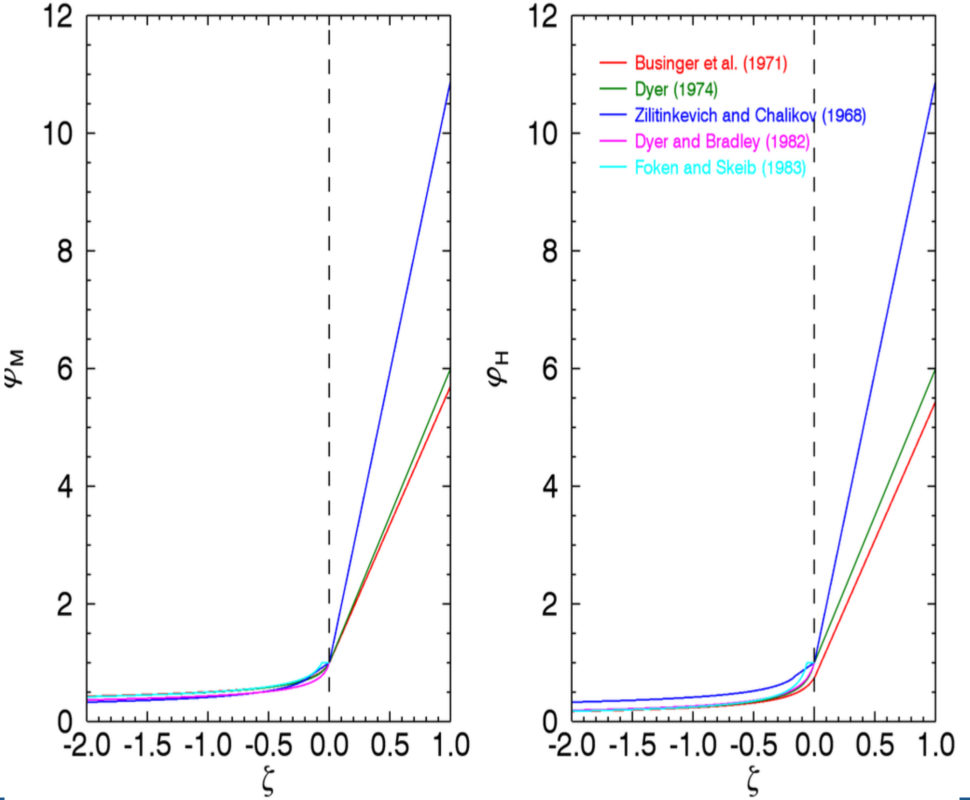


Atmospheric Surface Layer Parameterizations: Based on Monin-Obukhov Similarity

In atmospheric models Monin-Obukhov similarity relations are used to determine surface fluxes and stresses.

Stability functions Φ_M and Φ_H must be determined experimentally.

Stability functions are determined from field studies under nearly ideal atmospheric flow conditions characterized by horizontally homogeneous flat terrain and stationarity.



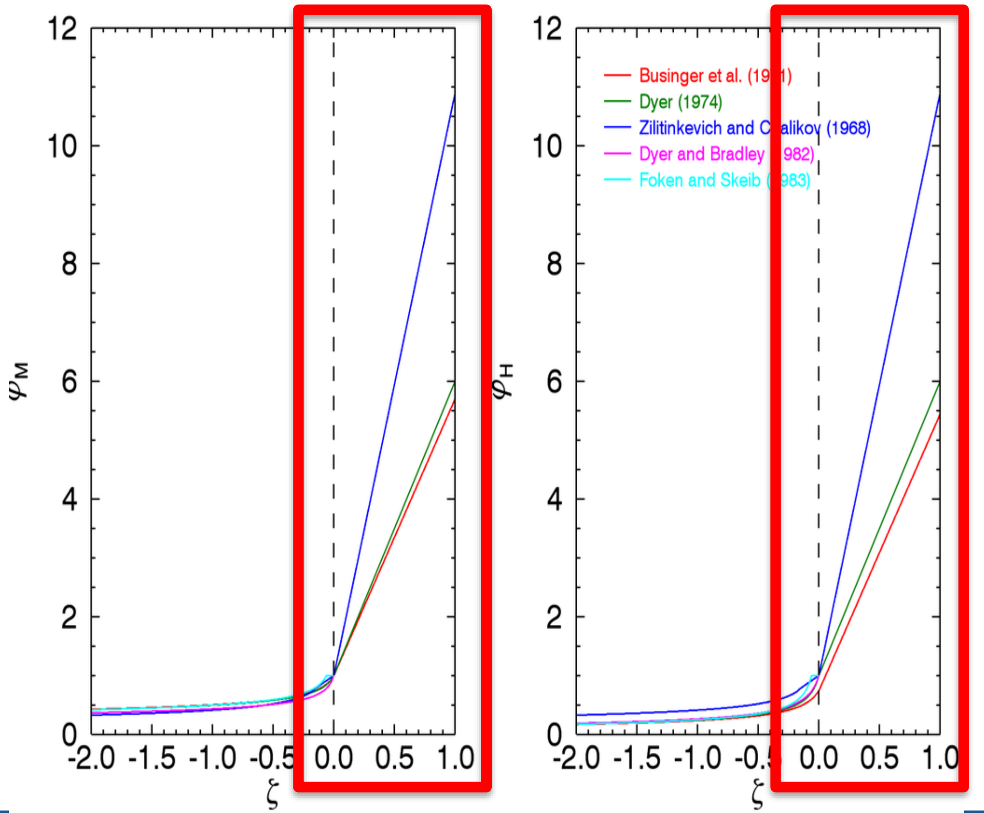
Atmospheric Surface Layer Parameterizations: Based on Monin-Obukhov Similarity

In atmospheric models Monin-Obukhov Similarity relations are used to determine surface fluxes and stresses.

Stability functions Φ_M and Φ_H must be determined experimentally.

Stability functions are determined from field studies under nearly ideal atmospheric flow conditions characterized by horizontally homogeneous flat terrain and stationarity.

Even under such idealized conditions, in particular under stable stratification, there is large variation in stability functions determined from different field studies.



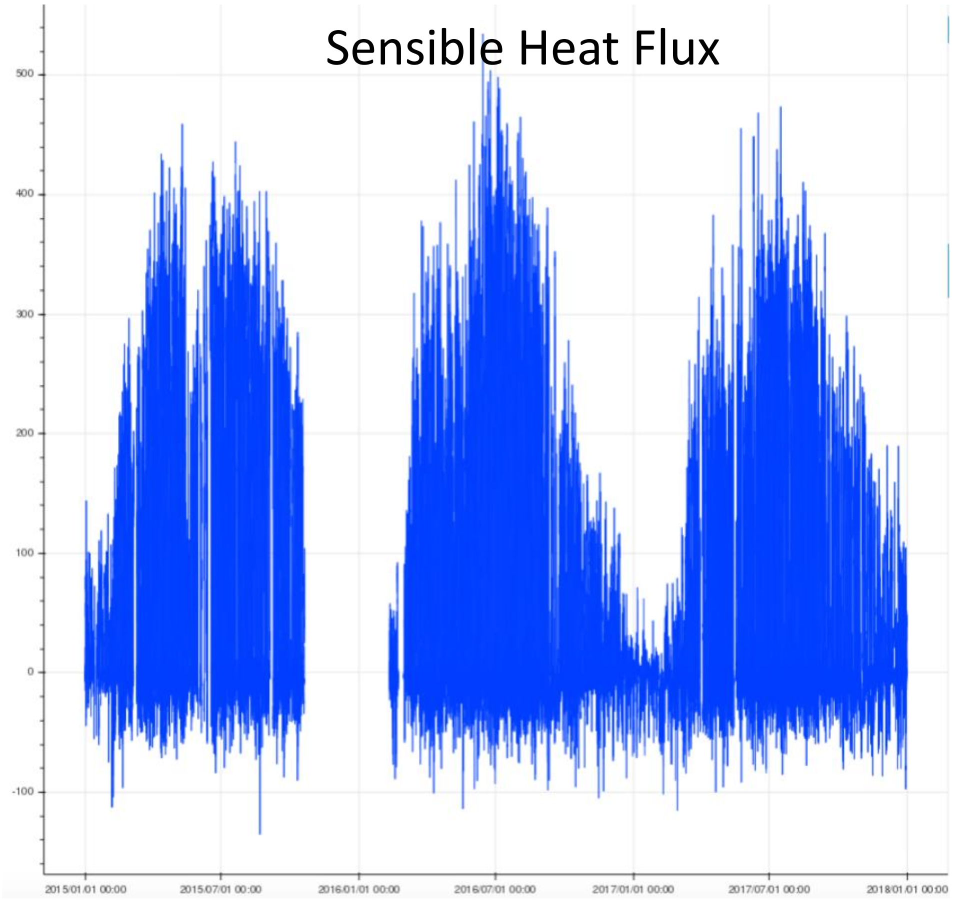
We Use Machine Learning Algorithms to Estimate Surface Stresses and Fluxes from Wind and Temperature Profiles

- Regression is commonly used to estimate stability functions and thus relationship between surface stresses and fluxes and wind and temperature profiles.
- Instead, we use machine learning algorithms to develop models relating surface stresses and fluxes to wind and temperature profiles.
- Most of the previous field studies used to determine stability functions were process studies of episodic nature - a few months in length.
- To develop machine learning models we need long observational records.
- We have therefore selected three data sets that provide multiyear records:
  - KNMI-mast at Cabauw (Netherlands), 213 m tower, 2000 - 2017,
  - FDR tower in Idaho – measurements from 2015 - 2017, and
  - National Renewable Energy Laboratory (NREL) National Wind Technology Center (NWTC) M5 tower, 130 m, 2012 – 2018.

Selected Data Sets Were Analyzed for Quality and Split into Training and Test Data Sets

Idaho data pre-processing:

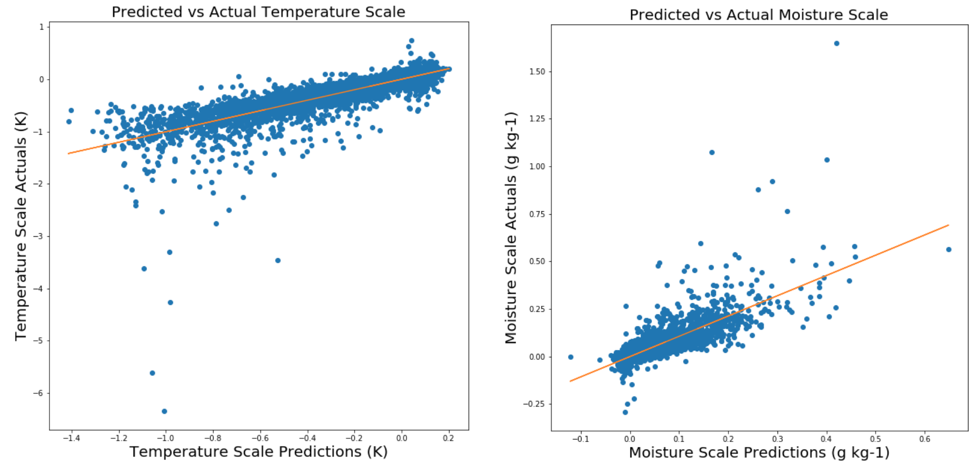
- Initial flux and met tower datasets had 52,608 flux/stress and 315,648 wind/temperature data points
- Removing NaNs reduced the data set to 42,637 and 295,101 data points
- Bad data points were removed
- Flux data was every 30 minutes, met tower data every 10 minutes
- Data set merge completed → 40,684 rows x 60* columns

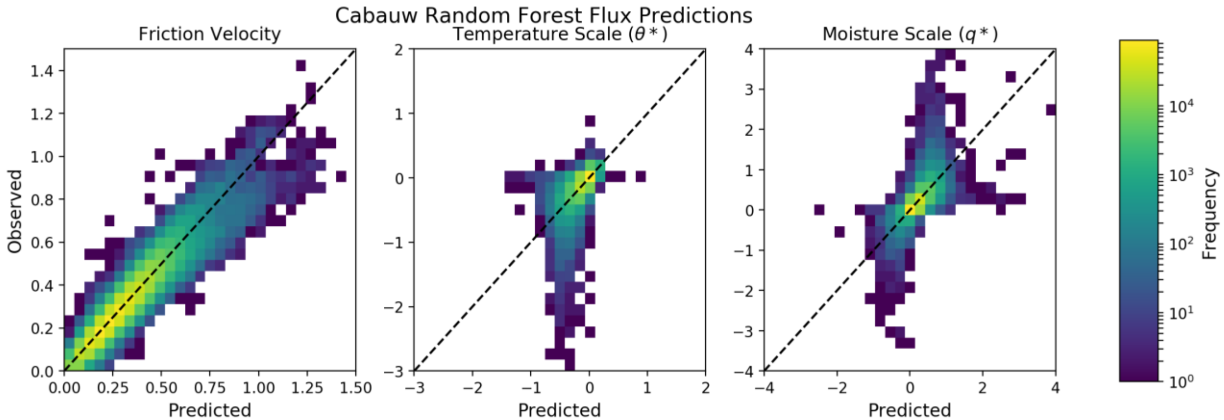


Initial Surface Layer Machine Learning Results

- Trained random forest to predict turbulent temperature and moisture scales
- Model captures general trend of scales for Idaho and Cabauw
- Model is under-forecasting large magnitude temperature and moisture scales in Cabauw
- Solar irradiance and zenith angles were most important variables because of strong diurnal and seasonal signals in data
- Temperature, wind and moisture data also contributed

Idaho Random Forest Surface Flux Predictions



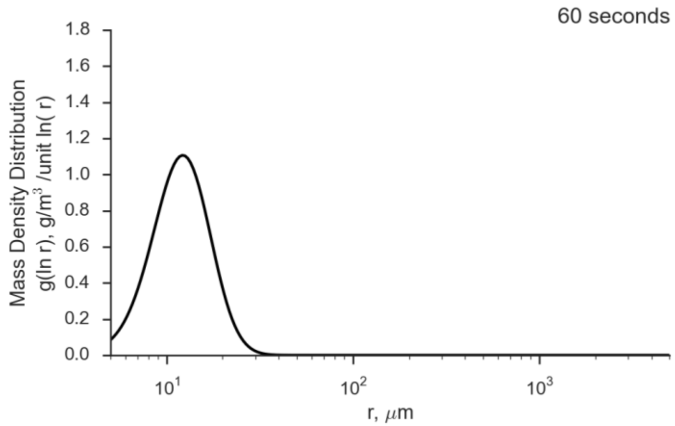


Atmospheric Surface Layer: Next Steps

- Need to drop irrelevant variables
  - Time variables that are not needed (hour, minute, etc.)
  - Variables not needed for flux predictions (i.e. “Top Temp”)
- Convert variables to appropriate units (if necessary)
- Compute derived variables used in atmospheric models (i.e. virtual potential T)
- Evaluate predictor importance for each flux
  - Correlations between predictor and predictand
  - Random forest for predictor importance

Emulating Cloud Microphysics: Motivation

- Precipitation formation is a critical uncertainty for weather and climate models.
- Different sizes of drops interact to evolve from small cloud drops to large precipitation drops.
- Detailed codes (right) are too expensive for large scale models, so empirical approaches are used.
- Let's emulate one (or more)
- Goal: put a detailed treatment into a global model and emulate it using ML techniques.
- Good test of ML approaches: can they reproduce a complex process, but with simple inputs/outputs?

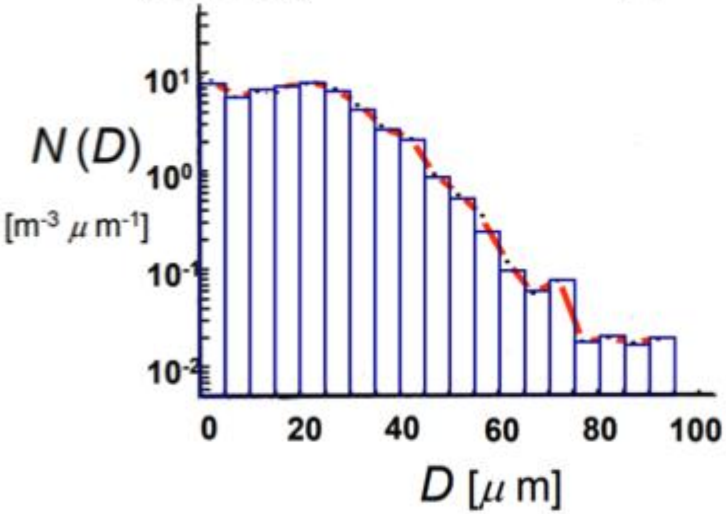


Sd-coal model output animation

Credit: Daniel Rothenberg

Ultimate Goal: Predict evolution of hydrometeor size distributions

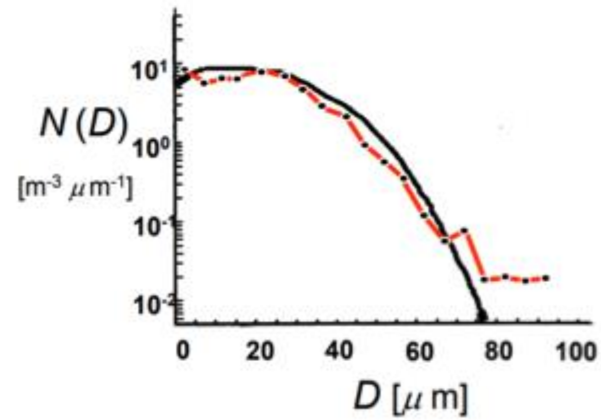
Bin-resolving: (spectral) $N(D) = \sum_{i=1}^{I} N_i$

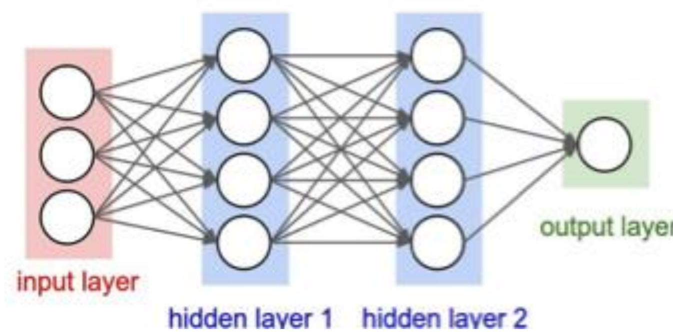


Divide microphysical particles into bins for different sizes, and compute evolution of each bin separately
Accurate but too expensive.

Bulk schemes calculate with a semi-empirical particle size distribution (PSD).

Bulk: $N(D) = N_0 D^{\alpha} e^{-\lambda D}$





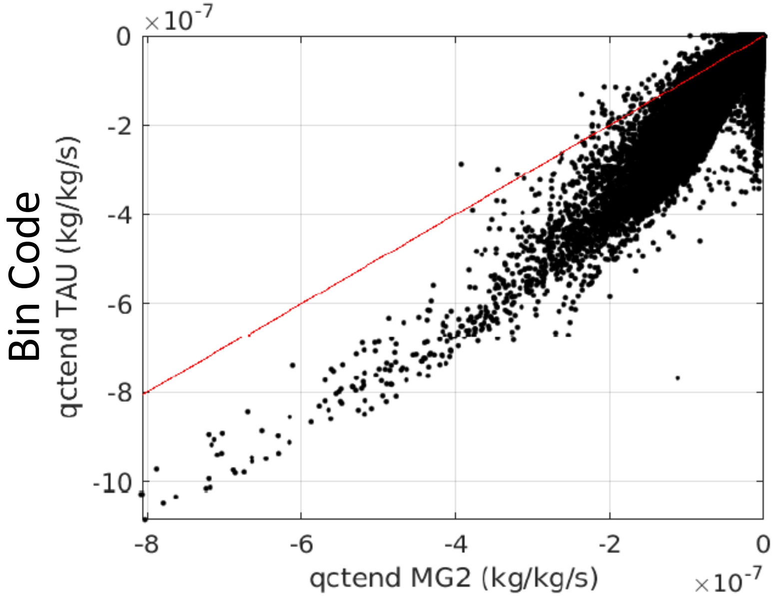
Machine Learning Approach

- Replace the bulk precipitation formation process in clouds with a detailed process model (Stochastic Collection equation).
  - Community Atmosphere Model (CAM)
- Advantages: we can run CAM as long as necessary to generate training data, and we can train everywhere on the planet!
- All auto-conversion events extracted from CAM run.
- To enhance training, data were log-transformed and re-scaled.
- Deep fully-connected neural network trained to predict the auto-conversion budget tendencies for bulk scheme.
- Neural network validated globally against withheld days.

CAM Results with Bin Model

- Implemented a detailed stochastic collection process in CAM (from a bin microphysical model: 'TAU')
- Translate size distributions (functions) to bins, run the detailed code, translate back.
- Testing now. (diagnostic first, then feeding back on the model)
- Compare to existing approach in model (empirical fit): see Fig.

CAM Cloud Water Tendency
Control v. Detailed Bin Code

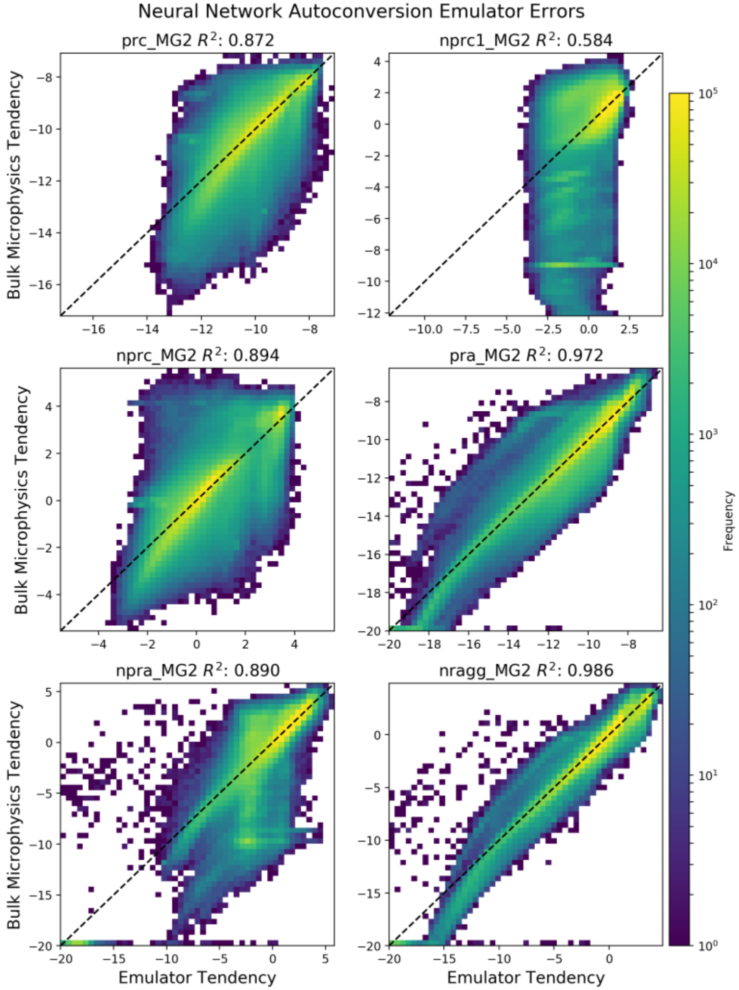


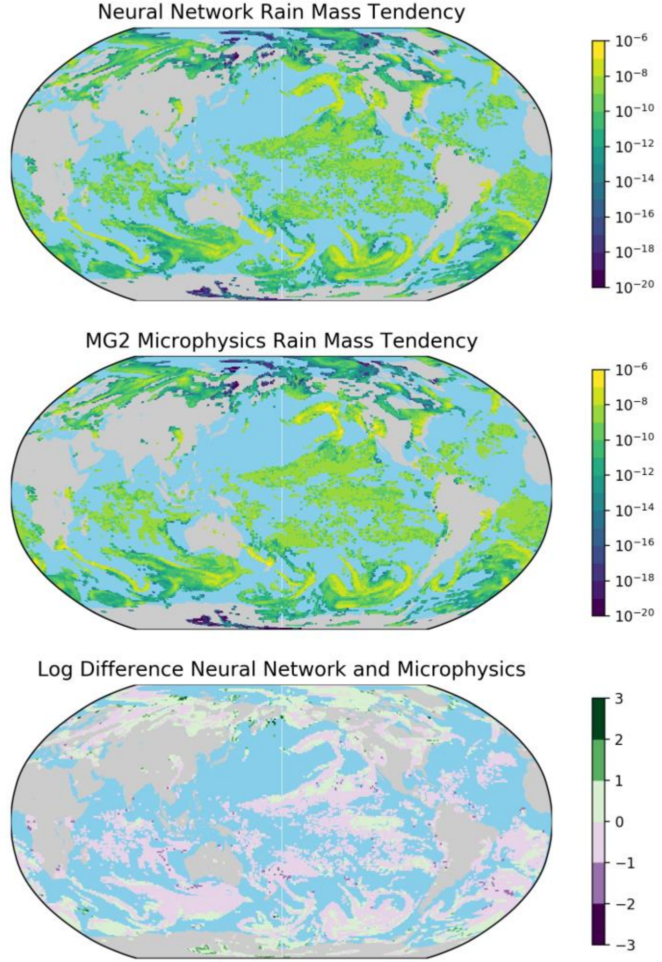
Control

Detailed 'TAU' bin code produces similar changes in cloud water to existing approach, but different structure.

Will this matter for climate?
Can we emulate it?

Results: Machine Learning-based Microphysics



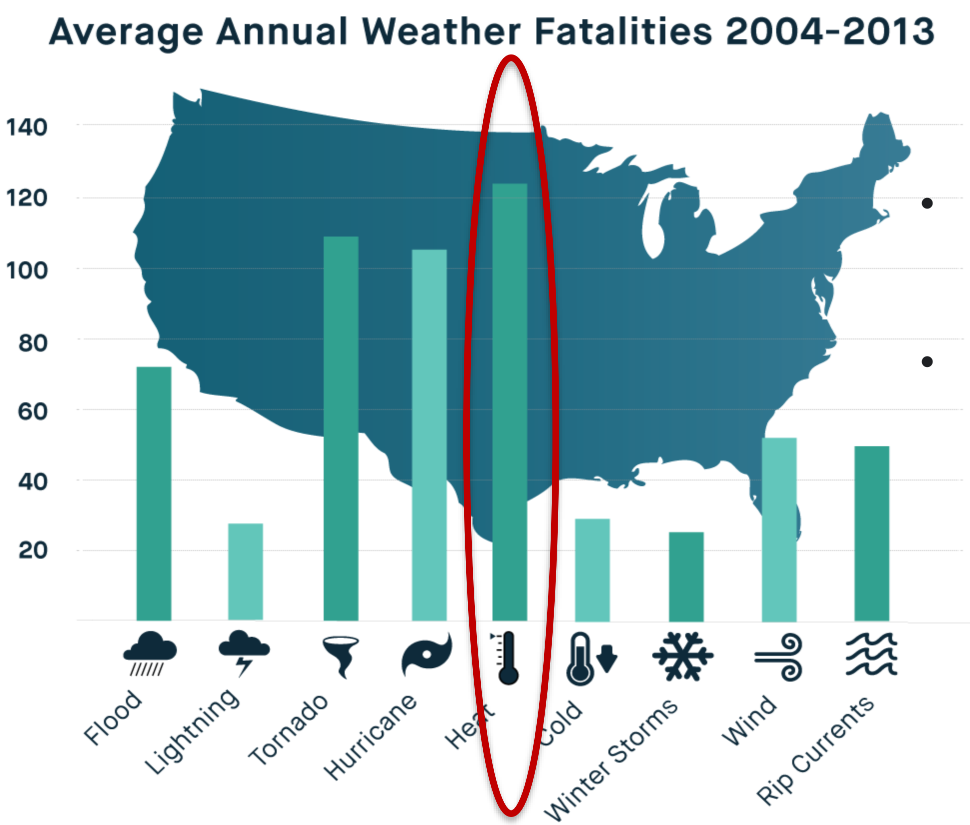


Neural net emulates bulk processes closely for most cases.

Neural network correctly captures the spatial structures of tendencies.

Seasonal Forecast of Hot Days Using Machine Learning

HEAT: THE SILENT KILLER!



Data from NOAA

- Heat is the **deadliest weather event** in the U.S.
- **~ 30 % of all weather-related fatalities** in the US for the period of 1988-2017.

Billion Dollar Losses from Disasters (2004-2013)

$392 Billion
Hurricanes

$78 Billion
Heat Waves/Droughts

$46 Billion
Tornadoes/Severe Storms

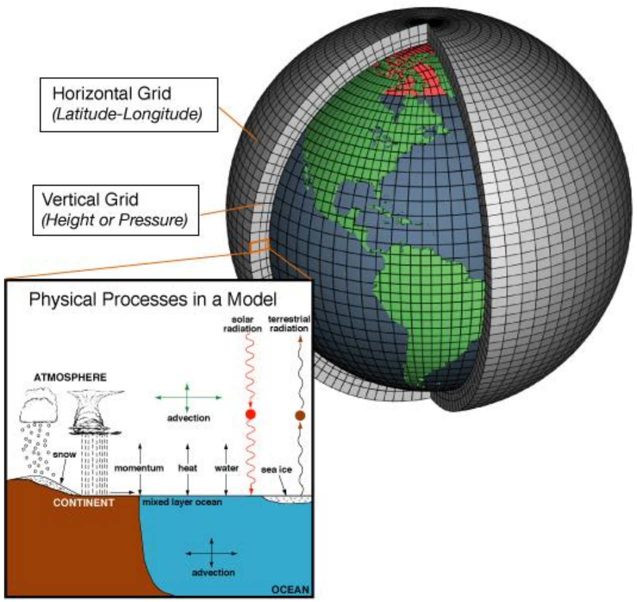
$30 Billion
Flooding/Severe Storms

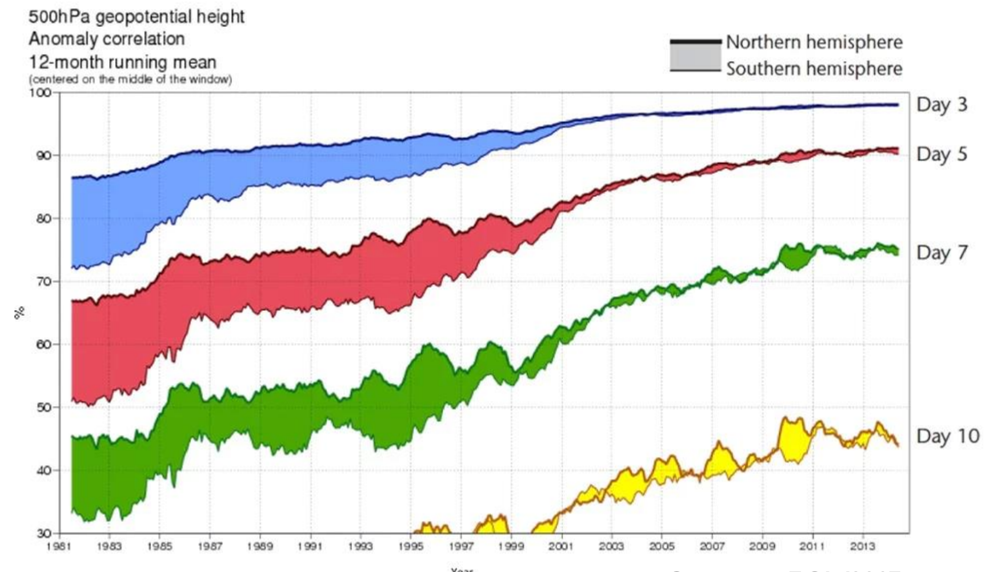
US Global Change Research Program, 2016

Economic losses and mortality rates of heat events can be significantly mitigated with improved forecasts.

Numerical Weather Prediction Models

Can machine learning do better?





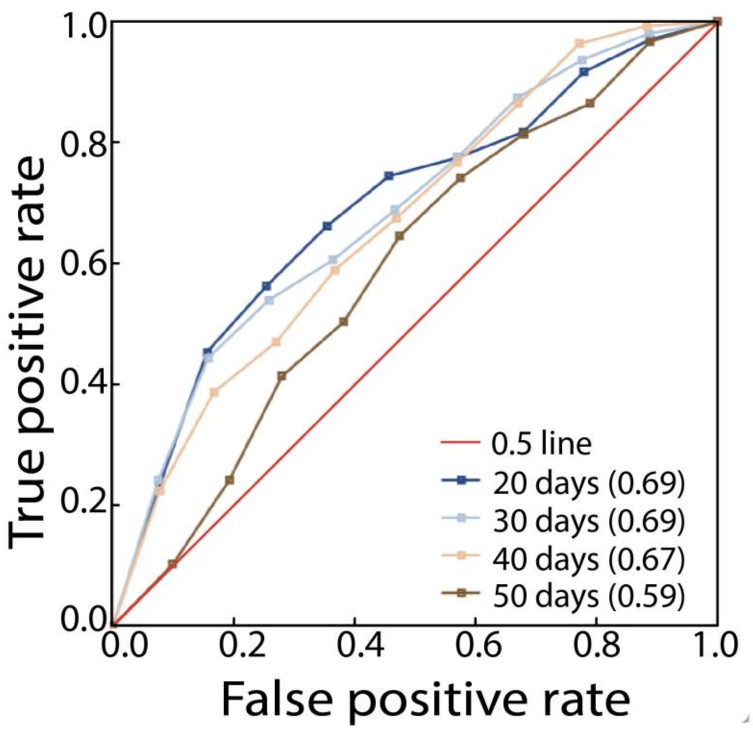
Source: ECMWF

Atmosphere only: Effectively no skill for forecasts longer than 10 days.

Long-range forecasts are computationally expensive!

- **Require a coupled Earth system prediction model**
- **CESM example: 25 km/10 km ~200,000 core-hours/simulated year**

McKinnon et al. 2016 showed correlation between anomalously warm SSTs and anomalously hot days in the Eastern US.

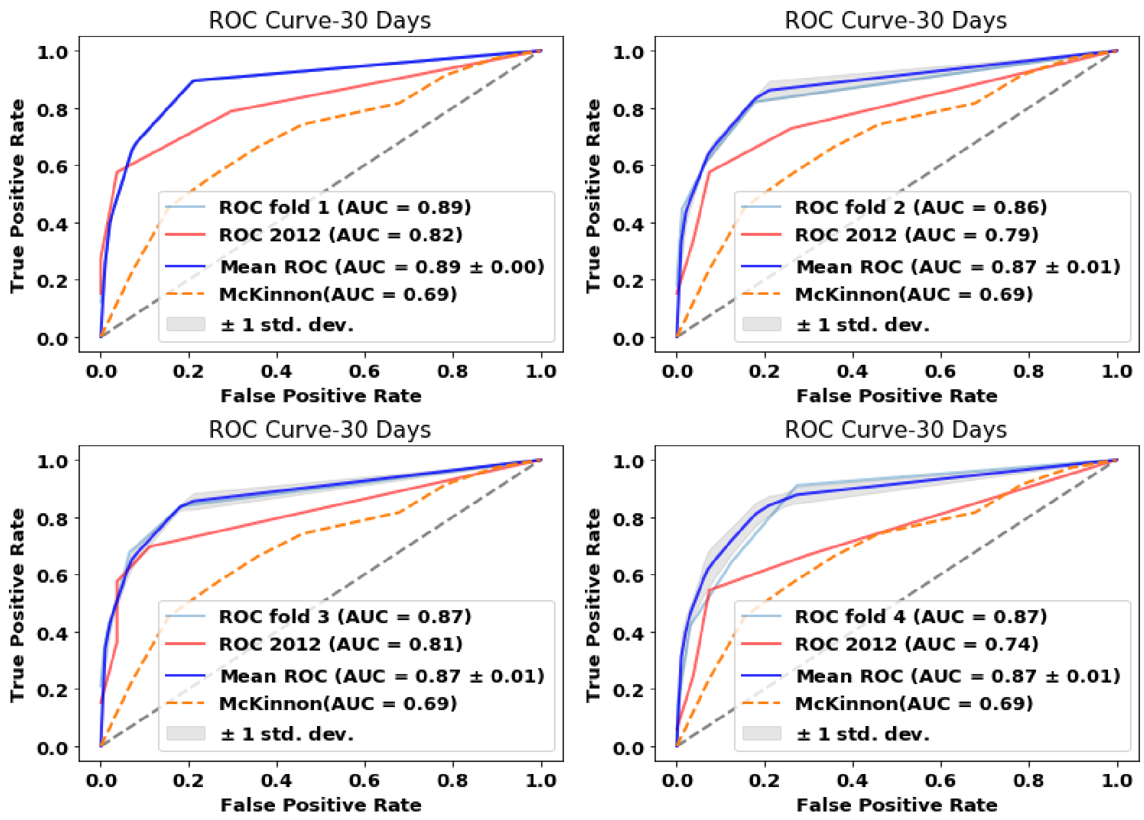


Source: McKinnon et al. 2016

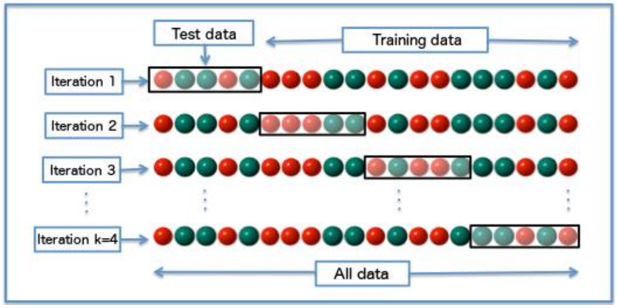
- McKinnon et al. 2016 uses correlations between SST anomalies and previous SST anomalies associated with heat events to predict 'hot' days. ROC score between 0.59-0.69.
- Here, we aim to improve on these results using Machine Learning or Deep Learning approaches.

Machine Learning Preliminary Results

Lead time = 30 Days



K-fold Cross-Validation



Outstanding emulator challenges

- Scientific concerns about “black box” science, e.g. interpretability & reproducibility.
- How the inputs to an emulator are conditioned/scaled and the architecture of the NN setup (a.k.a. hyper-parameters) are critical to the successful formulation of a successful emulator. Research question, involves domain knowledge.
- NN’s do poorly if extreme events are underrepresented in the training data.
- Having NNs respect constraints and conservation laws (e.g. monotonicity, energy conservation) so they don’t crash the code is a real bugaboo. Basic research question.
- NNs appear to be emulating but when their forcing is iteratively mapped back into a model, failures occur.

Talk Summary

- **Science 3.0: HPC + ML**
- NCAR's Computing Lab (CISL) is betting on the merging of accelerated HPC and Machine Learning (e.g. through GPUs) to accelerate the modeling enterprise.
- On the data side, our acceleration strategy is to pursue lossy compression, machine learning-enabled parallel analytics software, all with supporting high-IOPS-enabled SSD storage infrastructure.
- Initial results are encouraging…
- But much more needs to be done to prove these ideas out!