Exascale Climate: Can Machine Learning Deliver the Goods?

Rich Loft
Director, Technology Development
CISL/NCAR
loft@ucar.edu

Collaborators: David John Gagne*, Tyler McCandless*,
Andrew Gettelman, Jack Chen, Daniel Rothenberg, Alma Hodzic,
Siyuan Wang, **Keeley Lawrence***

***NCAR AIML Group**

Computer improvement slowing, data volumes growing

CPU performance slowing

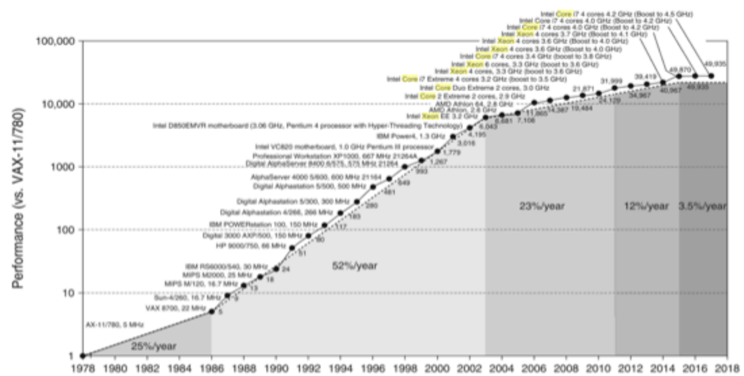
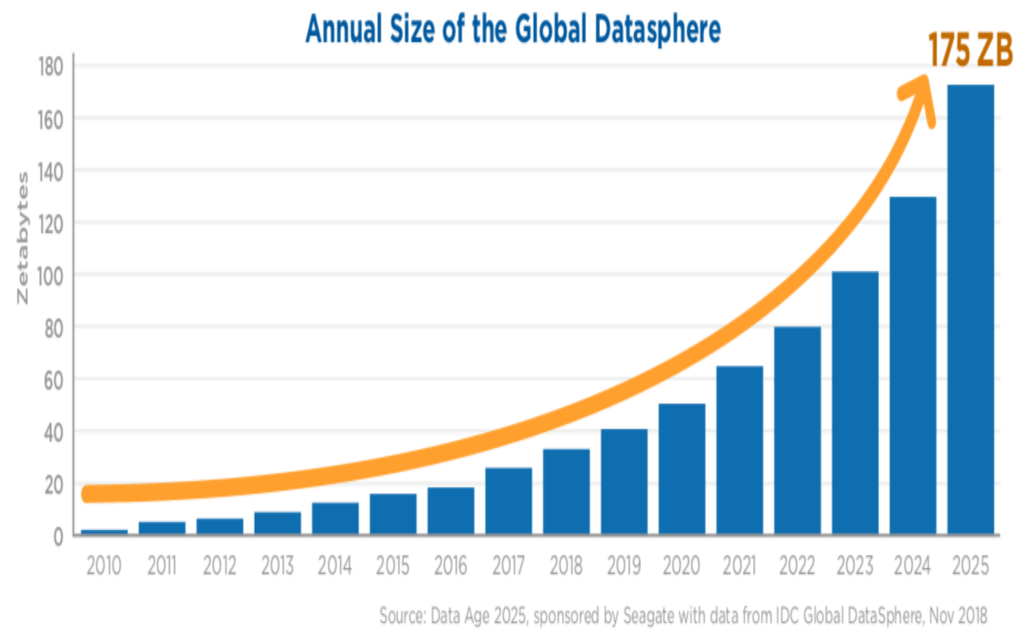


Figure 1.1 Growth in processor performance over 40 years. This chart plots program performance relative to the VAX 11/780 as measured by the SPEC integer benchmarks (see Section 1.8). Prior to the mid-1980s, growth in processor performance was largely technology-driven and averaged about 22% per year, or doubling performance every 3.5 years. The increase in growth to about 52% starting in 1986, or doubling every 2 years, is attributable to more advanced architectural and organizational ideas typified in RISC architectures. By 2003 this growth led to a difference in performance of an approximate factor of 25 versus the performance that would have occurred if it had continued at the 22% rate. In 2003 the limits of power due to the end of Dennard scaling and the available instruction-level parallelism slowed uniprocessor performance to 23% per year until 2011, or doubling every 3.5 years. (The fastest SPECintbase performance since 2007 has had automatic parallelization turned on, so uniprocessor speed is harder to gauge. These results are limited to single-chip systems with usually four cores per chip.) From 2011 to 2015, the annual improvement was less than 12%, or doubling every 8 years in part due to the limits of parallelism of Amdahl's Law. Since 2015, with the end of Moore's Law, improvement has been just 3.5% per year, or doubling every 20 years! Performance for floating-point-oriented calculations follows the same trends, but typically has 1% to 2% higher annual growth in each shaded region. Figure 1.11 on page 27 shows the improvement in clock rates for these same eras. Because SPEC has changed over the years, performance of newer machines is estimated by a scaling factor that relates the performance for different versions of SPEC: SPEC89, SPEC92, SPEC95, SPEC2000, and SPEC2006. There are too few results for SPEC2017 to plot yet.

Source: Hennesey & Patterson, Computer Architecture: A Quantitative Approach, 6th Edition

Data volumes growing



Source: Data Age 2025, sponsored by Seagate with data from IDC Global DataSphere, Nov 2018

Drowning in a Sea of Complexity

- Due to insufficient sustained computing power Earth system models can't resolve key phenomena and timescales.
- Scientists try to describe the unresolved scales using human-crafted physics parameterizations (equations that approximate the processes).
- Model *software complexity* grows, driven by the increasing complexity of these parameterizations.
- Growing *architectural complexity* further hinders the ability to port and optimize complex Earth system model codes on new architectures.
- Due to insufficient computing power models can't resolve key phenomena and timescales.

Why machine learning?

Traditional models

- Models are implemented in complex “one-off” code.
- Model algorithms are at odds with computer architectural trends.
- Data is a problem.

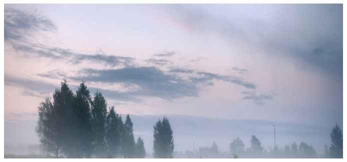
Machine learning

- Machine learning software implemented in reusable code.
- Machine learning is well aligned with architectural trends.
- Data is still a problem, but with machine learning it is also an opportunity.

Three candidate processes studied

Goal: Evaluate how machine learning models perform both physically and computationally at representing physical processes.

- **Surface Layer**: machine learning parameterization trained from observations to minimize assumptions required by Monin-Obukhov Similarity Theory (MOST)



- **Microphysics**: machine learning emulator trained on simulation data from a bin microphysics process is inserted into bulk microphysics scheme

- **Secondary Organic Aerosols:** can we use ML to emulate the incredibly complex chemistry of SOA formation?

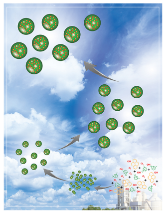
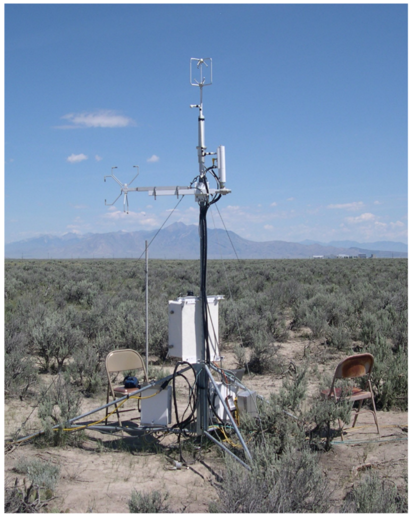


Image Credits
Surface Layer Image: UK Met Office
Macroburst: Pete Mangione's Pinpoint Weather Blog, August 5, 2015
SOAs: *Years of results regarding secondary organic aerosols reduce uncertainty in climate projections,* May 12, 2015 physics.org.

Motivation: Surface Layer Methods

- Regression is commonly used to estimate the stability functions used in M-O theory.
- 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 only a few months in length.
- To develop robust machine learning models, we need long observational records.
- We found only two data sets that provide suitable, multiyear records
- Fit random forests and neural networks to each site to predict friction velocity, sensible heat flux, and latent heat flux

Cabauw, Netherlands
KNMI Mast
213 m tower
Data from 2003-2017

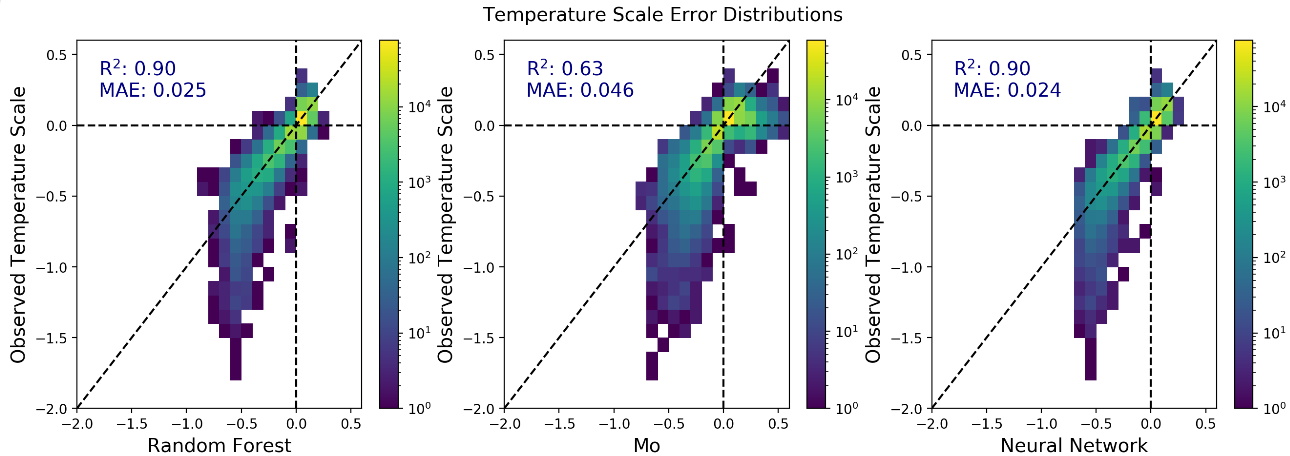


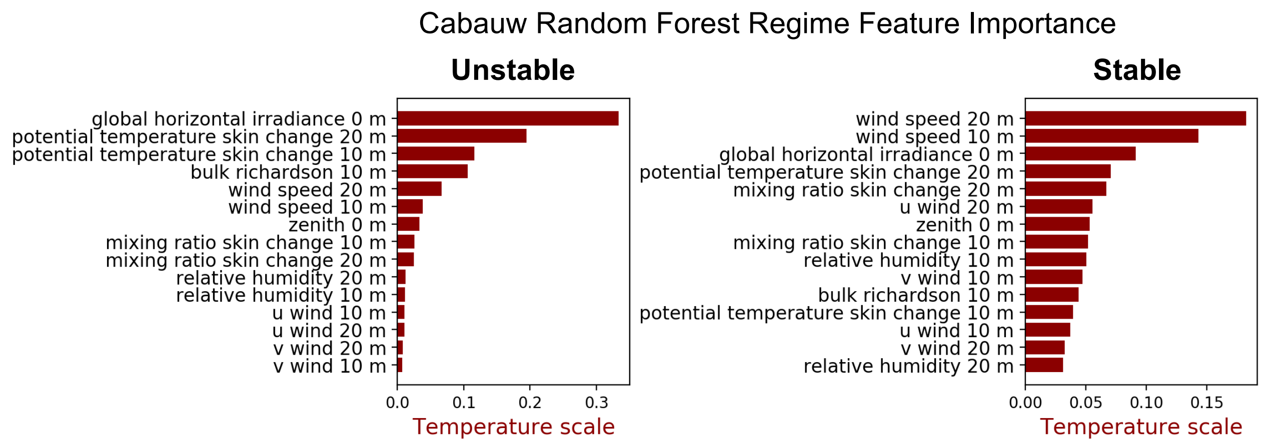
Scoville, Idaho, USA
FDR Tower
Flux tower
Data from 2015-2017

Surface Layer Results

Key Updates

- Trained with 30-minute-averaged data
- Evaluated different subsets of predictors
- Added neural network surface layer parameterization to WRF
- Calculated variable importance rankings for different stability regimes





Surface Layer Conclusions

- Machine learning surface layer models can improve on estimating surface flux information over Monin-Obukhov
- Random forests and neural networks have similar amounts of error offline but perform differently within WRF
- Training at multiple sites improves generalization compared with training at one site
- Multi-site training challenge: inconsistencies in variables measured and heights of measurements

Pilot Project 2: Microphysics Emulator

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 (right).

Detailed codes are too computationally expensive for large scale models, so empirical approaches are used.

Goal: Put increasingly detailed treatments into CAM6 physics and emulate them using ML techniques.

- Tel Aviv University scheme (35 bins)
- Superdroplet (Rothenberg) (~300 bins)

Question: Can ML approaches reproduce the effects of binned schemes without adding significant computational cost?

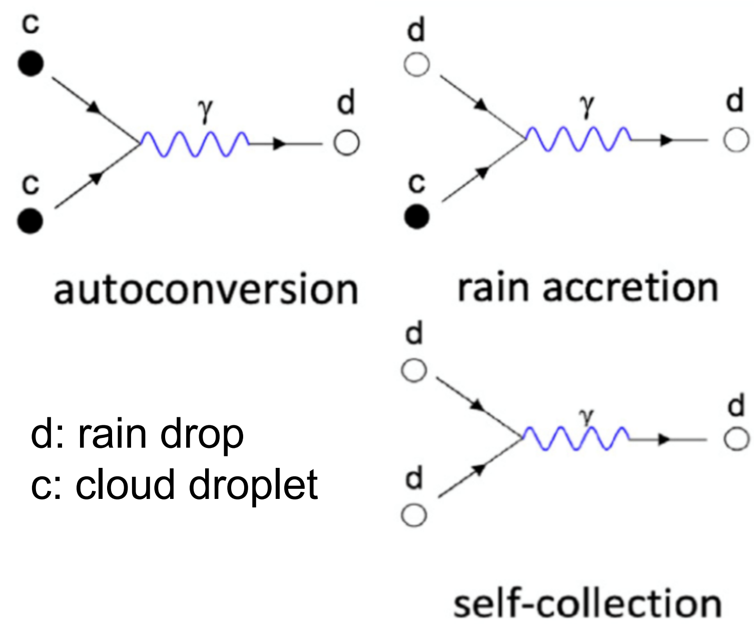


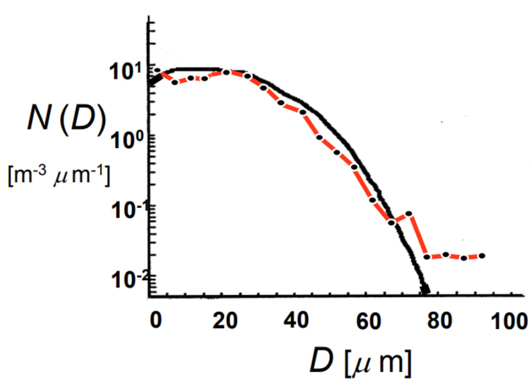
Image credit: Tapiado, et al., *Empirical values and assumptions in the microphysics of numerical models*, Atm. Res. 215, 2019, p 214-238.

Bulk vs. Bin vs. Emulator Microphysics

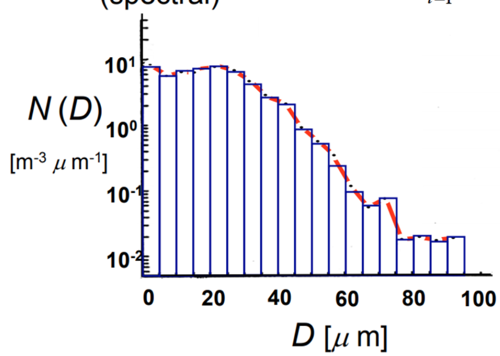
Bulk scheme (MG2 in CAM6):
Calculate with a semi-empirical particle size distribution (PSD). Gamma distribution often used.

Bin Scheme Divide particle sizes into bins and calculate evolution of each bin separately. Better representation of interactions but much more computationally expensive.

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



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



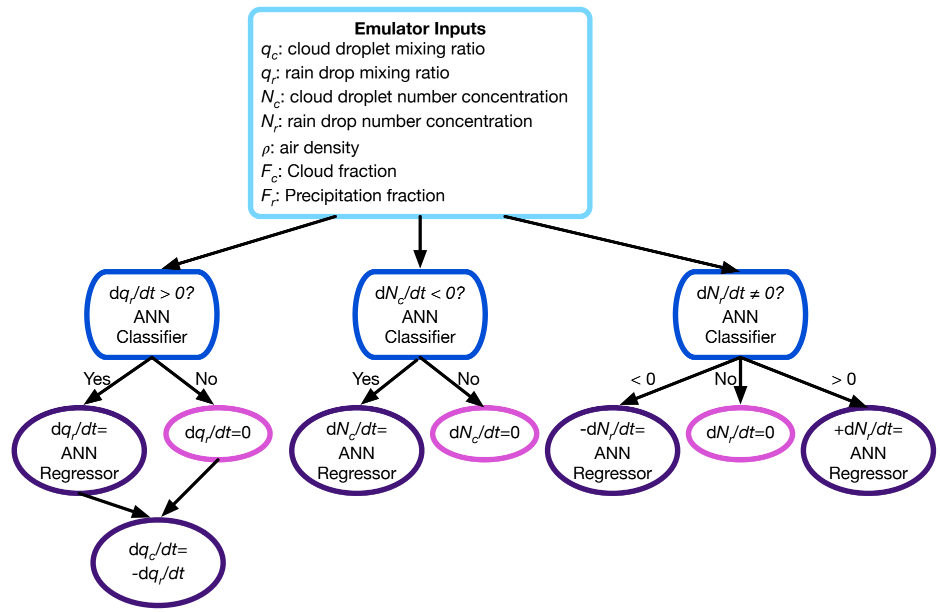
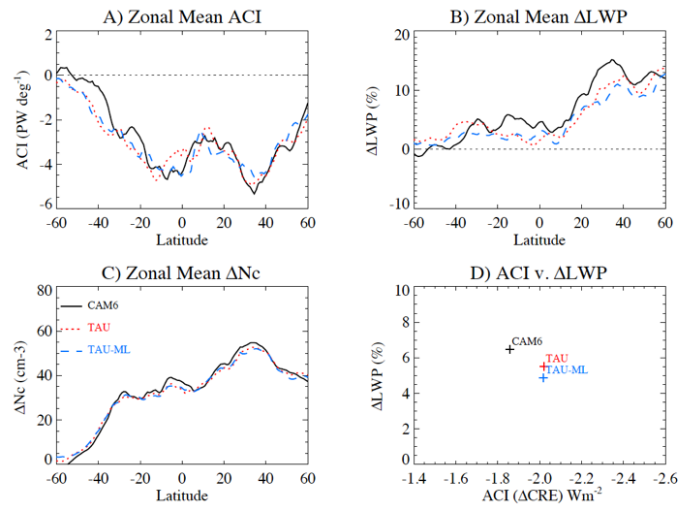
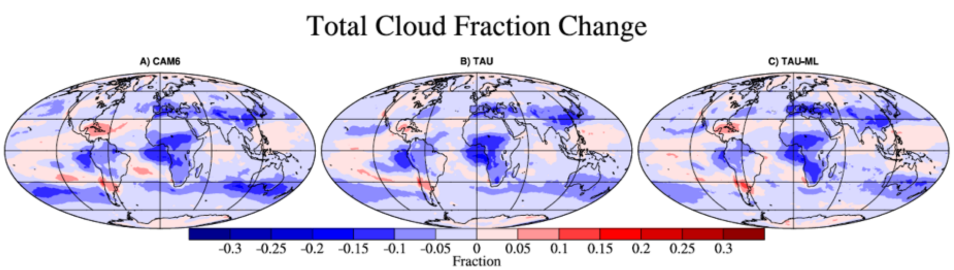


Image credit: Andrew Gettelman, NCAR

CAM6 Feedback Comparison

- Examined emergent properties in CAM6 for MG2, TAU and TAU ML emulator
- Aerosol-Cloud Interactions are similar between TAU and TAU ML
- Shortwave cloud radiative feedbacks are higher in the southern hemisphere, especially for emulator
- Cloud fraction not being reduced as fast in TAU and emulator





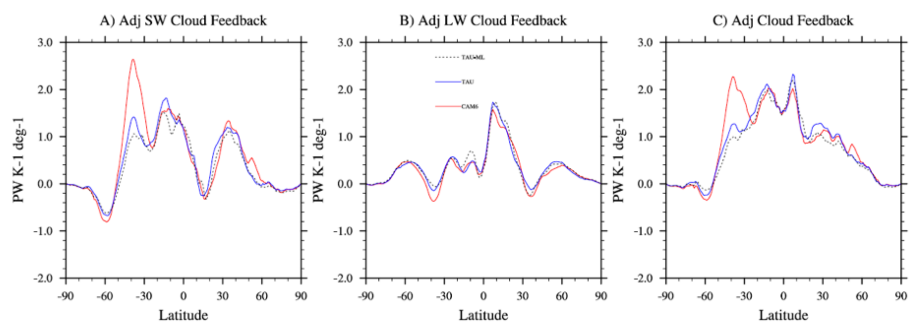


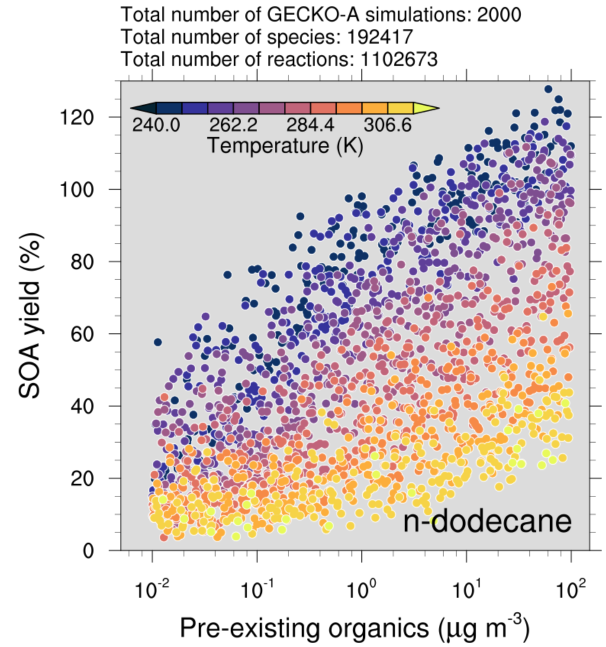
Image credit: Andrew Gettelman, NCAR

Microphysics Emulation Conclusions

- The TAU microphysics emulator largely replicates the climate effects of the original TAU code
- Some feedback effects observed from use of emulator related to thickness of clouds
- Optimized TAU neural network CAM only runs about 8% slower than control CAM run with MG2; TAU run 300% slower

ML 2020: GECKO-A Project

- **Goals**
  - Build catalog of GECKO-A chemistry model runs under a diverse set of atmospheric conditions
  - Train neural network emulator from catalog
  - Run emulator in NWP model
- **Accomplishments**
  - Created catalog of GECKO-A runs for different molecules
  - GECKO-A run as box model with fixed atmospheric conditions and fixed initial amount of precursor
  - Evaluated large set of neural network hyperparameters.
  - Devised performance metrics for total gas, aerosol, and precursor species.
- **Remaining Tasks**
  - Training more complex neural networks
  - Completing GECKO-A catalog
  - Integration with an NWP model



Secondary Organic Aerosol yields for n-dodecane from GECKO-A simulation

*Student Assistant-led project
Ms. Keely Lawrence, CU

Thanks!