Machine Learning and the Post-Dennard Era of Climate Simulation

Séminaire IA et Climat, LIP6

V. Balaji

Laboratoire des Sciences du Climat et de l'Environnement (LSCE)
Institut Pierre-Simon Laplace (IPSL)
NOAA/Geophysical Fluid Dynamics Laboratory (GFDL) and Princeton University

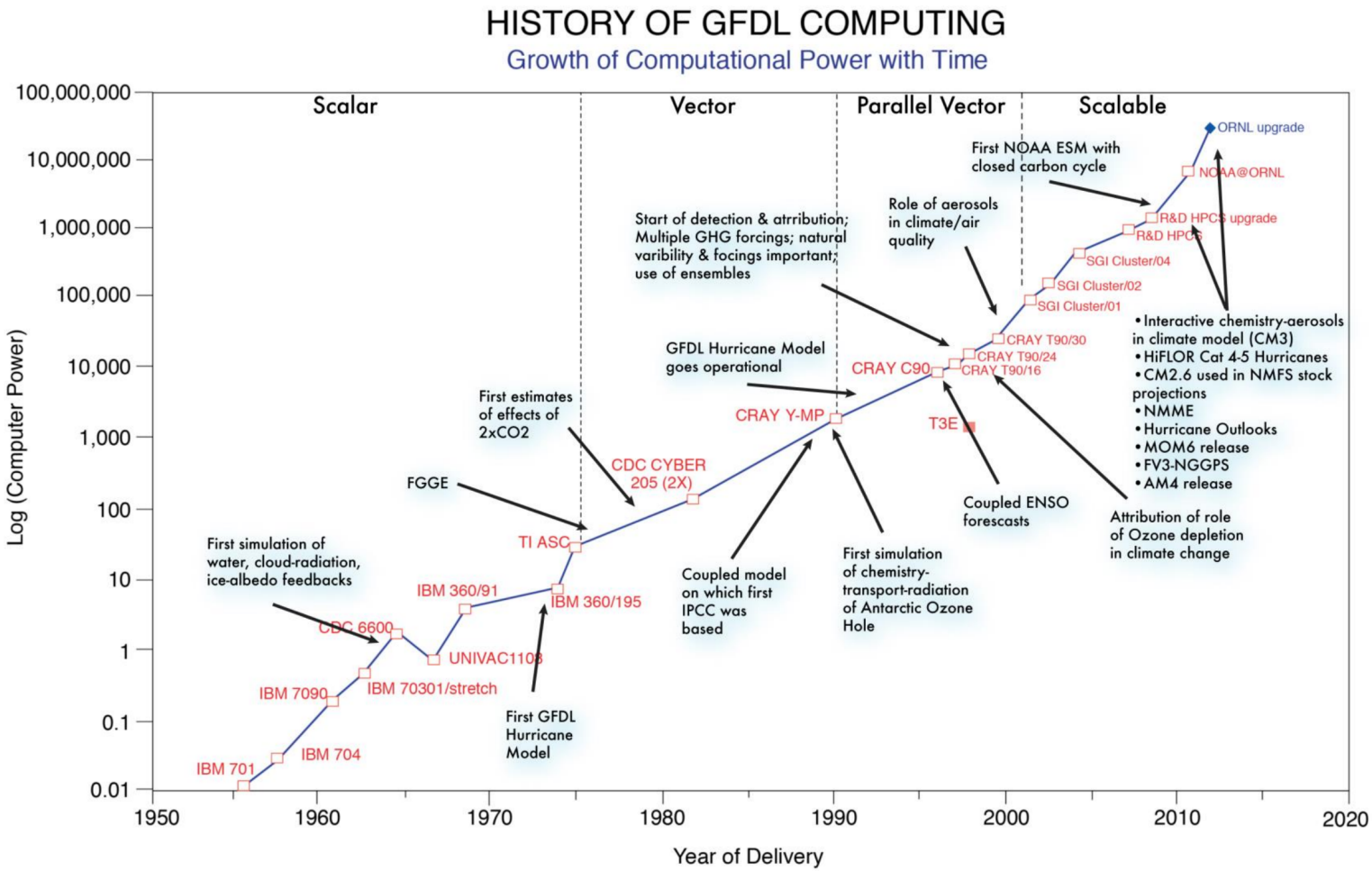
20 February 2019

Outline

1. Hardware evolution at the end of Dennard scaling
   - The end of Dennard scaling
   - Specialized and commodity computing
   - Deep learning is an industry driver
2. Challenges in model development
   - No separation of scales
   - Calibration of coupled systems
3. Applications of machine learning
   - Learning parameterizations from high-resolution simulations
   - Learning low-order manifolds for uncertainty exploration
4. Ideas and challenges

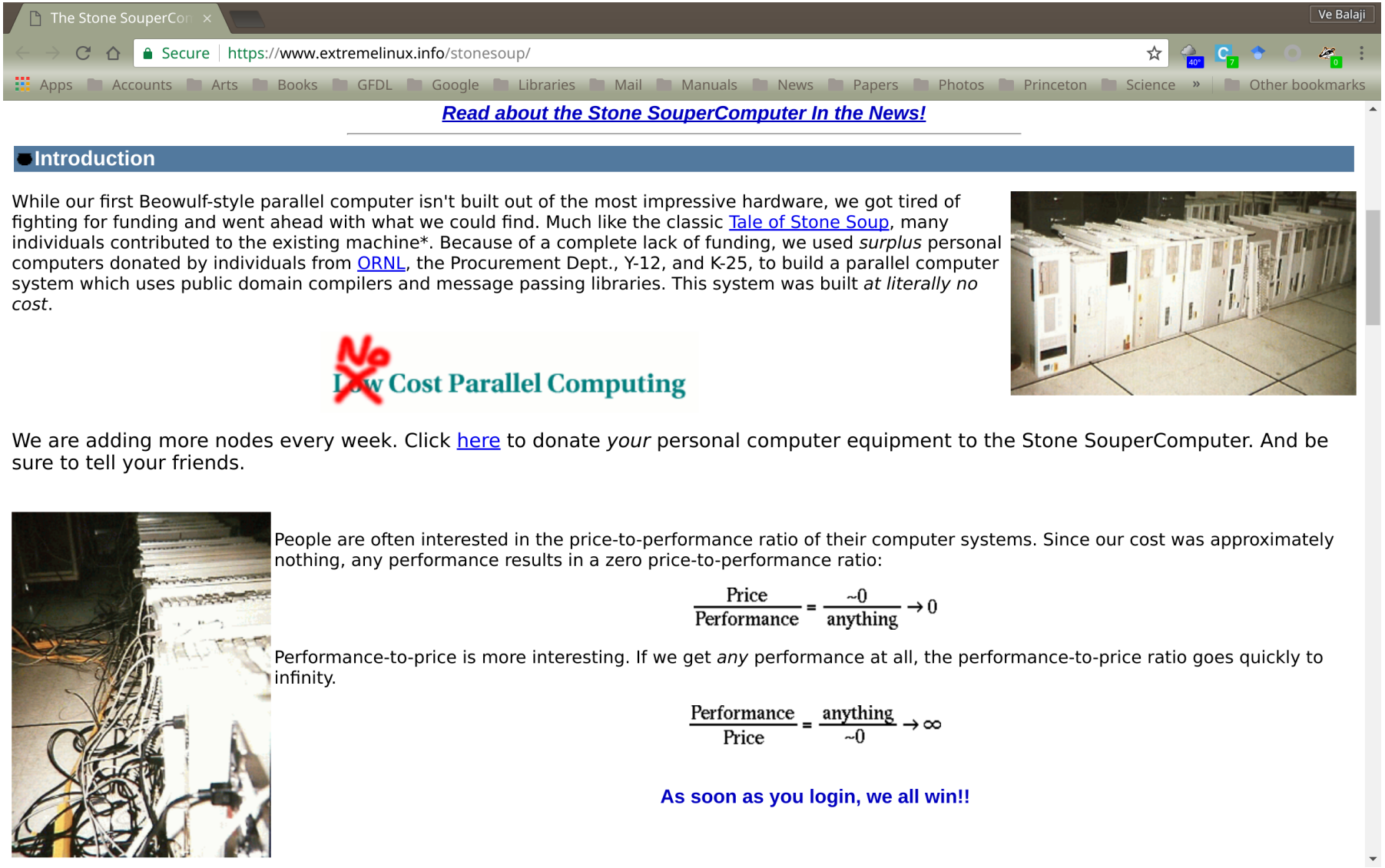
Outline

History of GFDL Computing



Courtesy V. Ramaswamy, NOAA/GFDL.

Beowulf clusters



Dennard Scaling

TABLE I

SCALING RESULTS FOR CIRCUIT PERFORMANCE

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/\kappa$
Doping concentration N_a	κ
Voltage V	$1/\kappa$
Current I	$1/\kappa$
Capacitance $\epsilon A/t$	$1/\kappa$
Delay time/circuit VC/I	$1/\kappa$
Power dissipation/circuit VI	$1/\kappa^2$
Power density VI/A	1

Table 1 from Dennard (1974). Shows scaling of various quantities when transistor dimension is reduced by factor κ.

Moore's Law and End of Dennard scaling

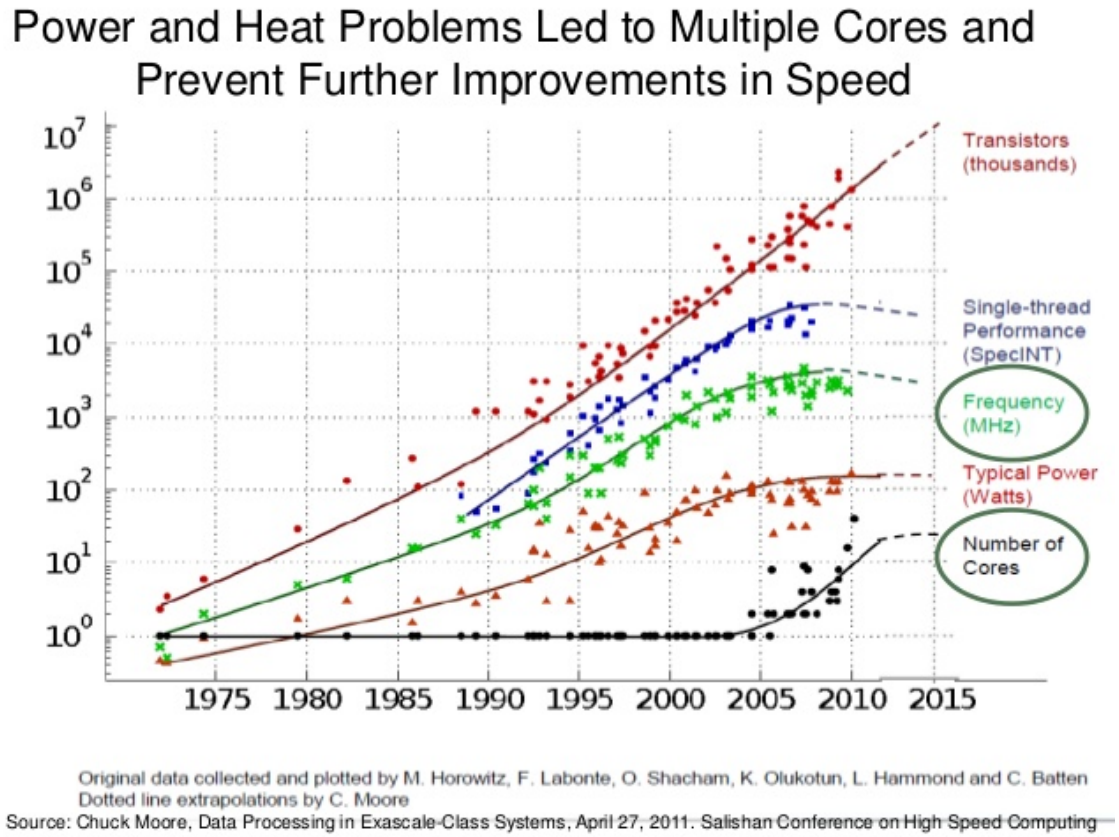


Figure courtesy Moore 2011: *Data processing in exascale-class systems.*

- Processor concurrency: Intel Xeon-Phi.
- Fine-grained thread concurrency: Nvidia GPU.

Theoretical peak performance

- Peak flops: computer vendors like! clock speed $\times$ number of ALUs $\times$ clock-cycle concurrency (e.g FMA, AVX).

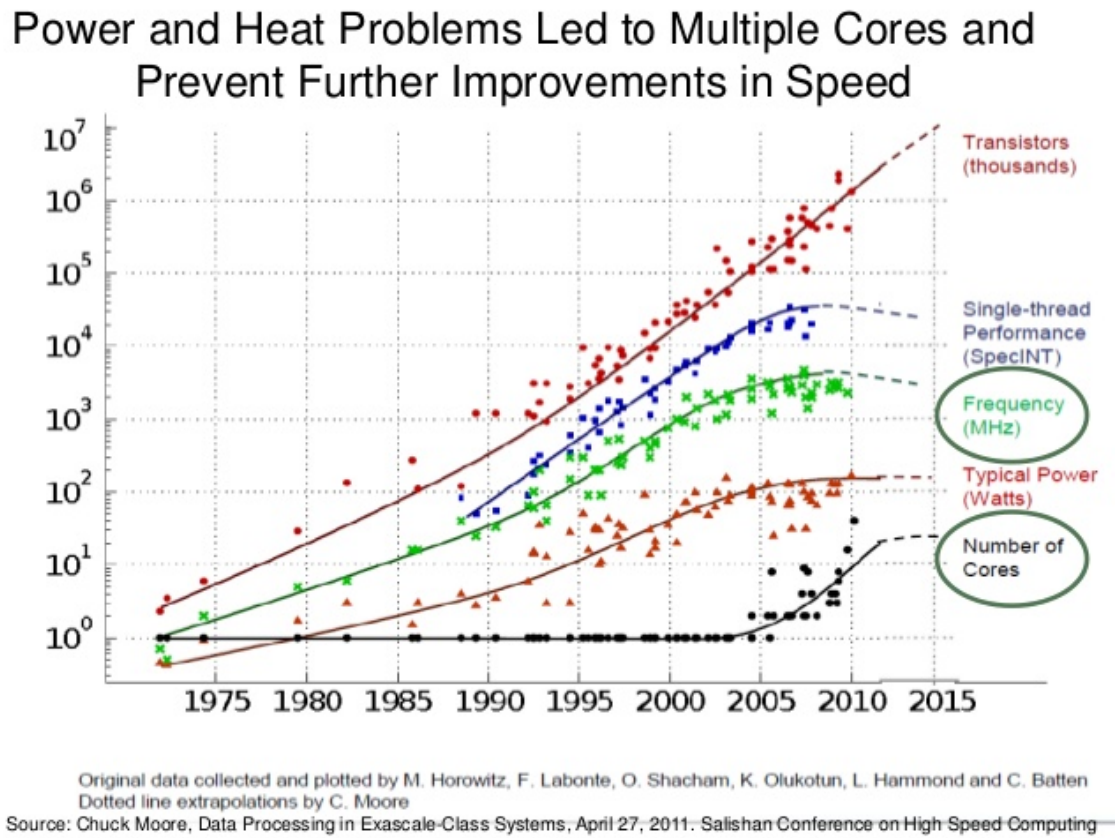
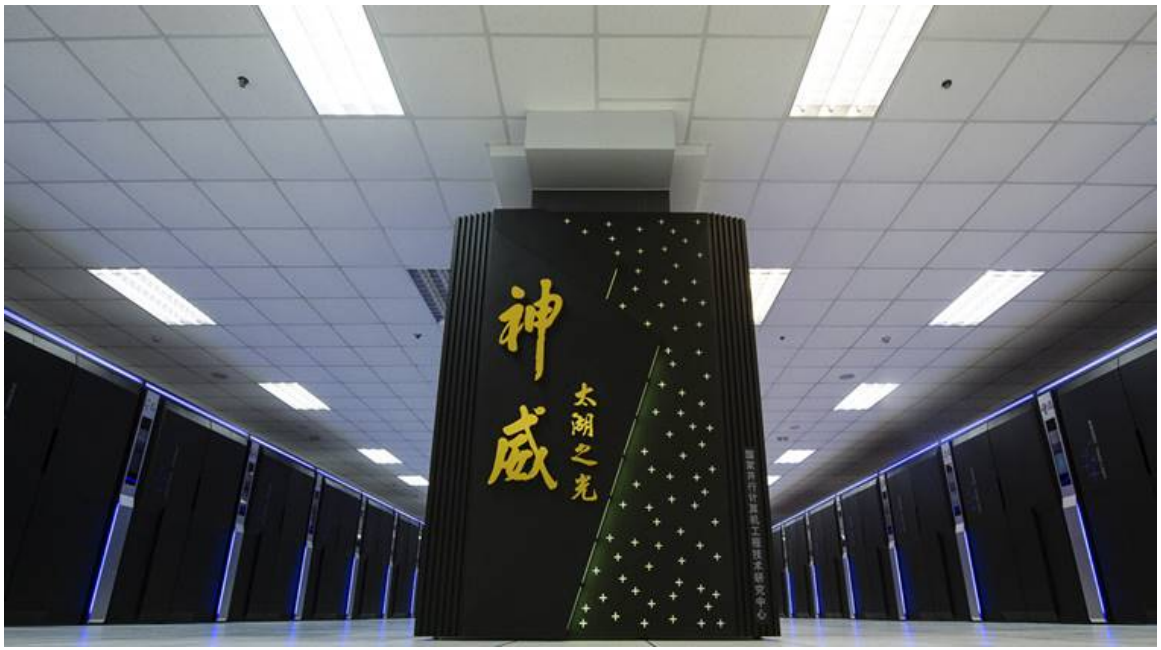


Figure courtesy Moore 2011: *Data processing in exascale-class systems.*

Sustained performance

- All ALUs cannot be kept active all the time, and theoretical peak is unachievable in practice. Measure actually achieved performance on actual computations.
- Linpack (Dongarra 1998) linear algebra package; maximum achievable performance. Basis for Top500 (http://www.top500.org). Current top machines: Sunway TaihuLight (China), Tianhe-2 (China), Titan (US).



Even Linpack is misleading...

- Real codes often gated by memory bandwidth.
- Roofline model:

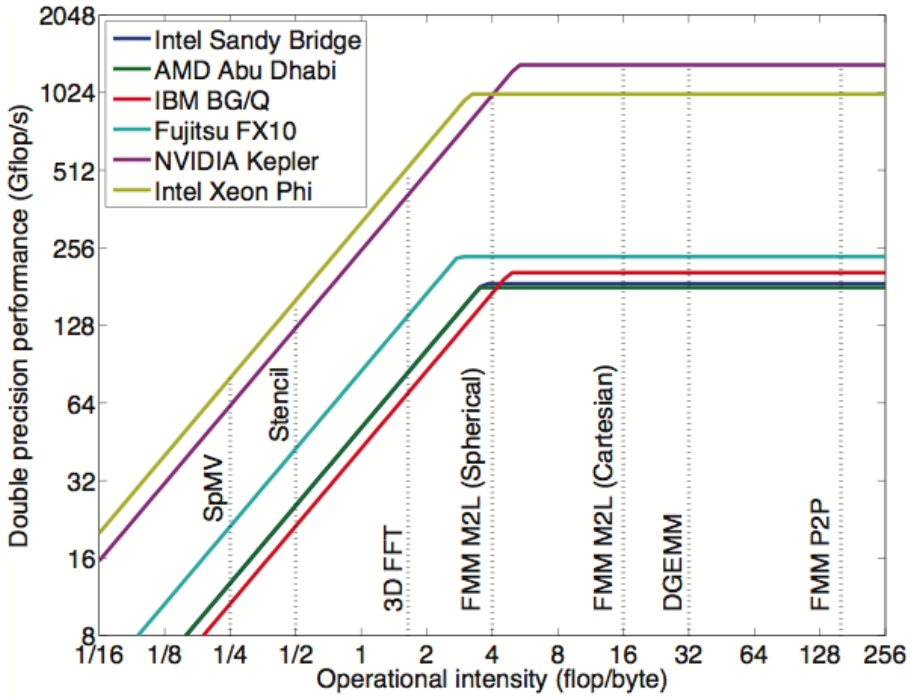


Figure courtesy Barba and Yokota *SIAM News* 2013.

Top500 revisited

- HPCG/HPL ratio is a measure of “percent of peak” (Dongarra and Heroux 2013).
- All recent HPC acquisitions in climate/weather have been on conventional Intel Xeon (see Balaji et al 2017).

Site	Computer	Cores	HPL Rmax (Pflops)	HPL Rank	HPCG (Pflops)	HPCG/ HPL	% of Peak
NSCC / Guangzhou	Tianhe-2 NUDT, Xeon 12C 2.2GHz + Intel Xeon Phi 57C + Custom	3,120,000	33.9	1	.632	1.8%	1.1%
RIKEN Advanced Inst for Comp Sci	K computer Fujitsu SPARC64 VIIIfx 8C + Custom	705,024	10.5	4	.461	4.4%	4.1%
DOE/OS Oak Ridge Nat Lab	Titan, Cray XK7 AMD 16C + Nvidia Kepler GPU 14C + Custom	560,640	17.6	2	.322	1.8%	1.2%
DOE/OS Argonne Nat Lab	Mira BlueGene/Q, Power BQC 16C 1.60GHz + Custom	786,432	8.59	5	.167	1.9%	1.7%
Swiss CSCS	Piz Daint, Cray XC30, Xeon 8C + Nvidia Kepler 14C + Custom	115,984	6.27	6	.105	1.7%	1.3%
Leibniz Rechenzentrum	SuperMUC, Intel 8C + IB	147,456	2.90	14	.0833	2.9%	2.6%
DOE/OS LBNL	Edison, Cray XC30, Xeon, 12c, 2,4GHz + Custom	133,824	1.65	24	.0786	4.8%	3.1%
GSIC Center TiTech	Tsubame 2.5 Xeon 6C, 2.93GHz + Nvidia K20x + IB	76,032	2.78	15	.073	2.6%	1.3%
Max-Planck	iDataPlex Xeon 10C, 2.8GHz + IB	65,320	1.28	34	.061	4.8%	4.2%
CEA/TGCC-GENCI	Curie tine nodes Bullx B510 Intel Xeon 8C 2.7 GHz + IB	77,184	1.36	33	.051	3.8%	3.1%
Exploration and Production Eni S.p.A.	HPC2, Intel Xeon 10C 2.8 GHz + Nvidia Kepler 14C + IB	62,640	3.00	12	.0489	1.6%	1.2%

The "Navier-Stokes Computer" of 1986

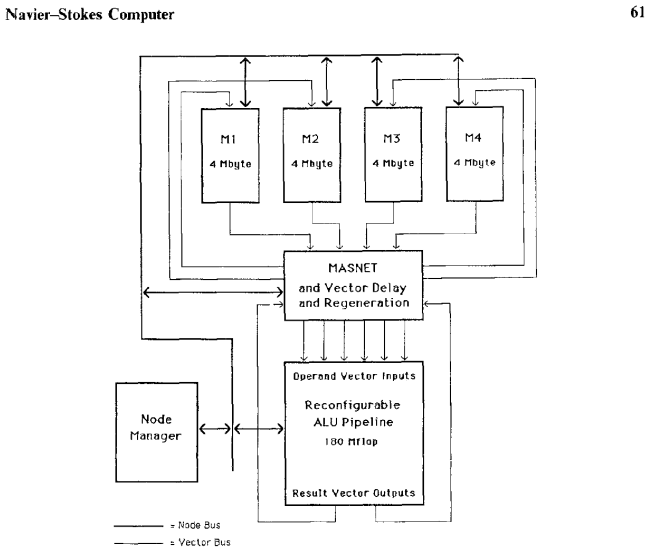


Fig. 5. Block diagram of miniNode.

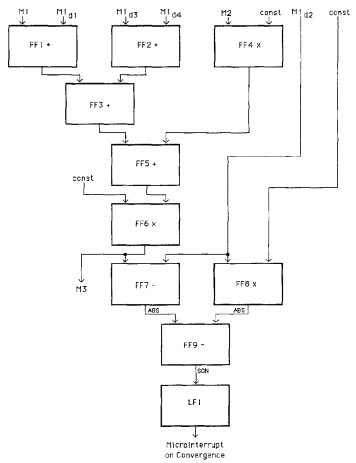


Fig. 6a. Block diagram of pipeline configuration 1.

"The Navier-Stokes computer (NSC) has been developed for solving problems in fluid mechanics involving complex flow simulations that require more speed and capacity than provided by current and proposed Class VI supercomputers. The machine is a parallel processing supercomputer with several new architectural elements which can be programmed to address a wide range of problems meeting the following criteria: (1) the problem is numerically intensive, and (2) the code makes use of long vectors."
Nosenchuck and Littman (1986)

Irreproducible Computing, Inexact Hardware

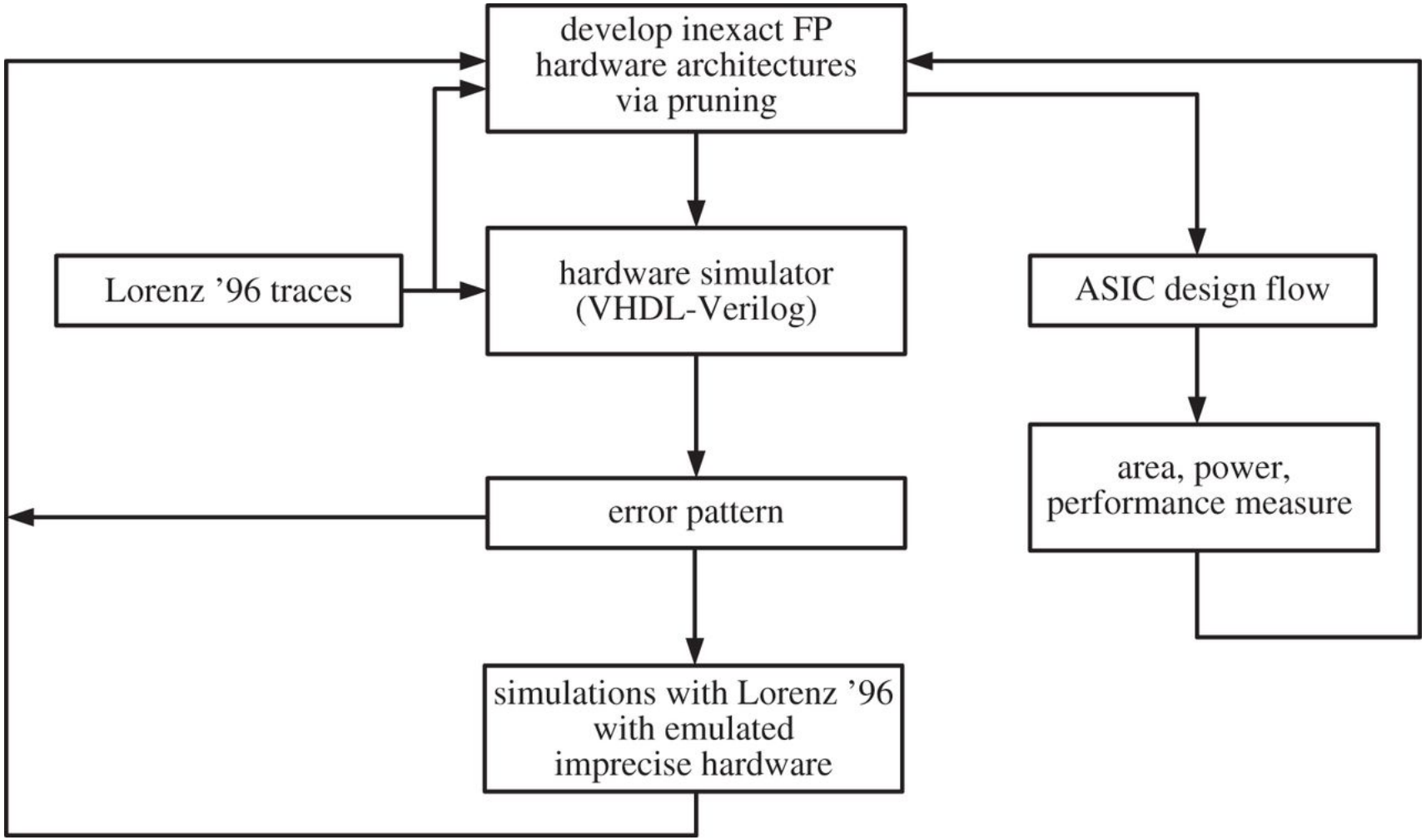
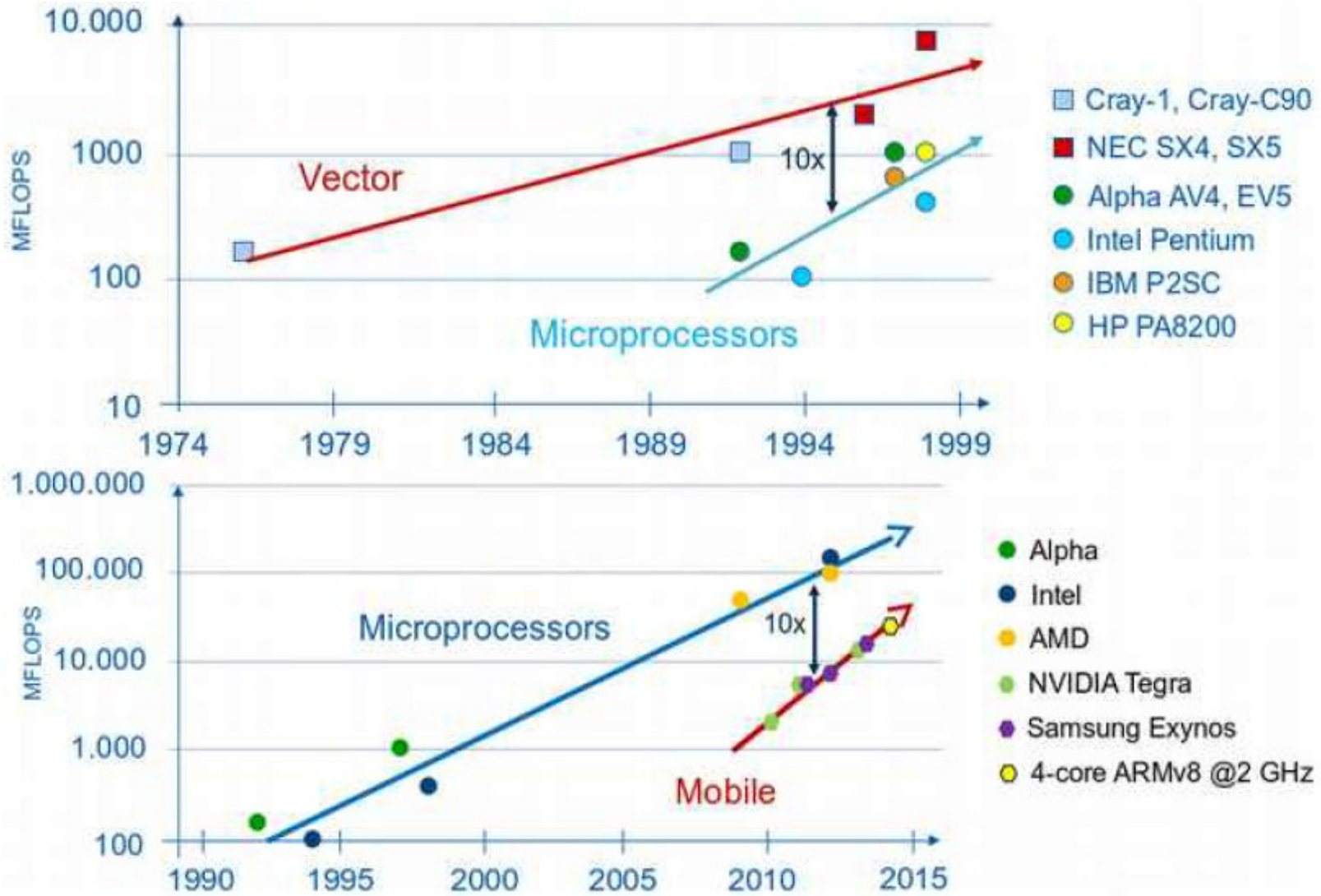


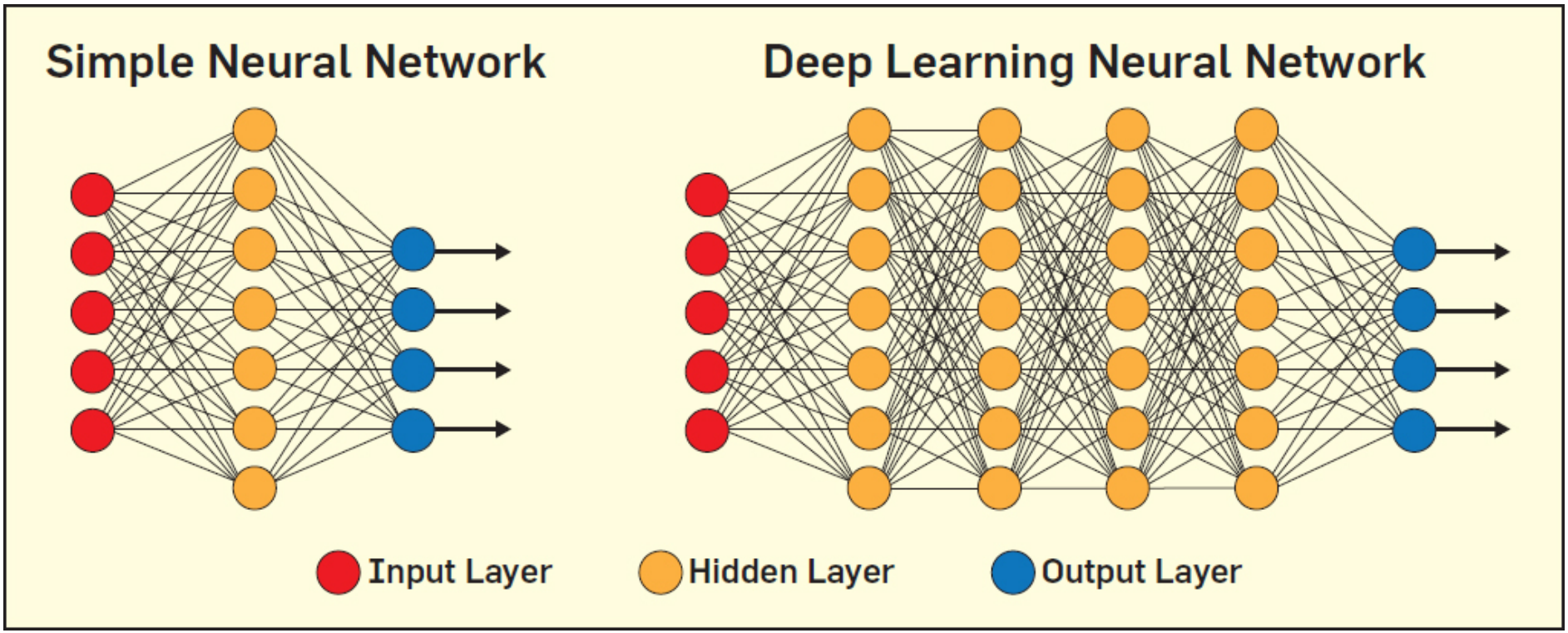
Figure 2 from Düben et al, *Phil. Trans. A*, 2016.

The inexorable triumph of commodity computing



From *The Platform*, Hemsoth (2015).

Deep Learning



From Edwards (2018), ACM.

Low precision arithmetic for Deep Learning

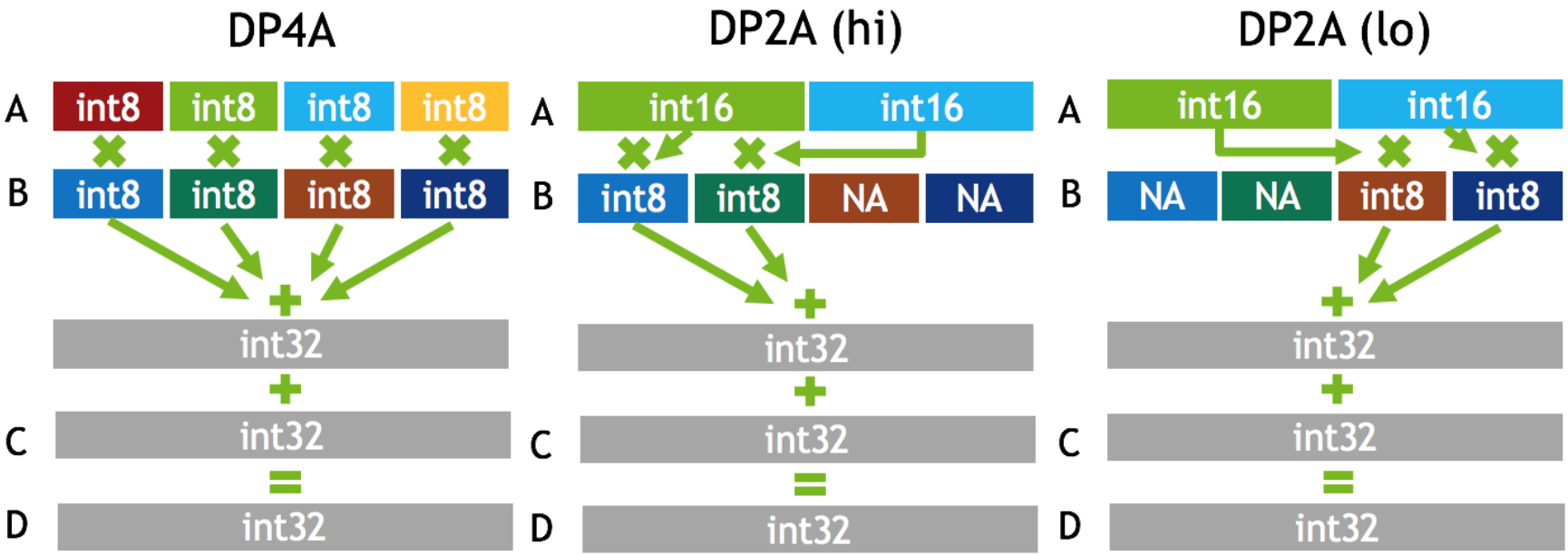
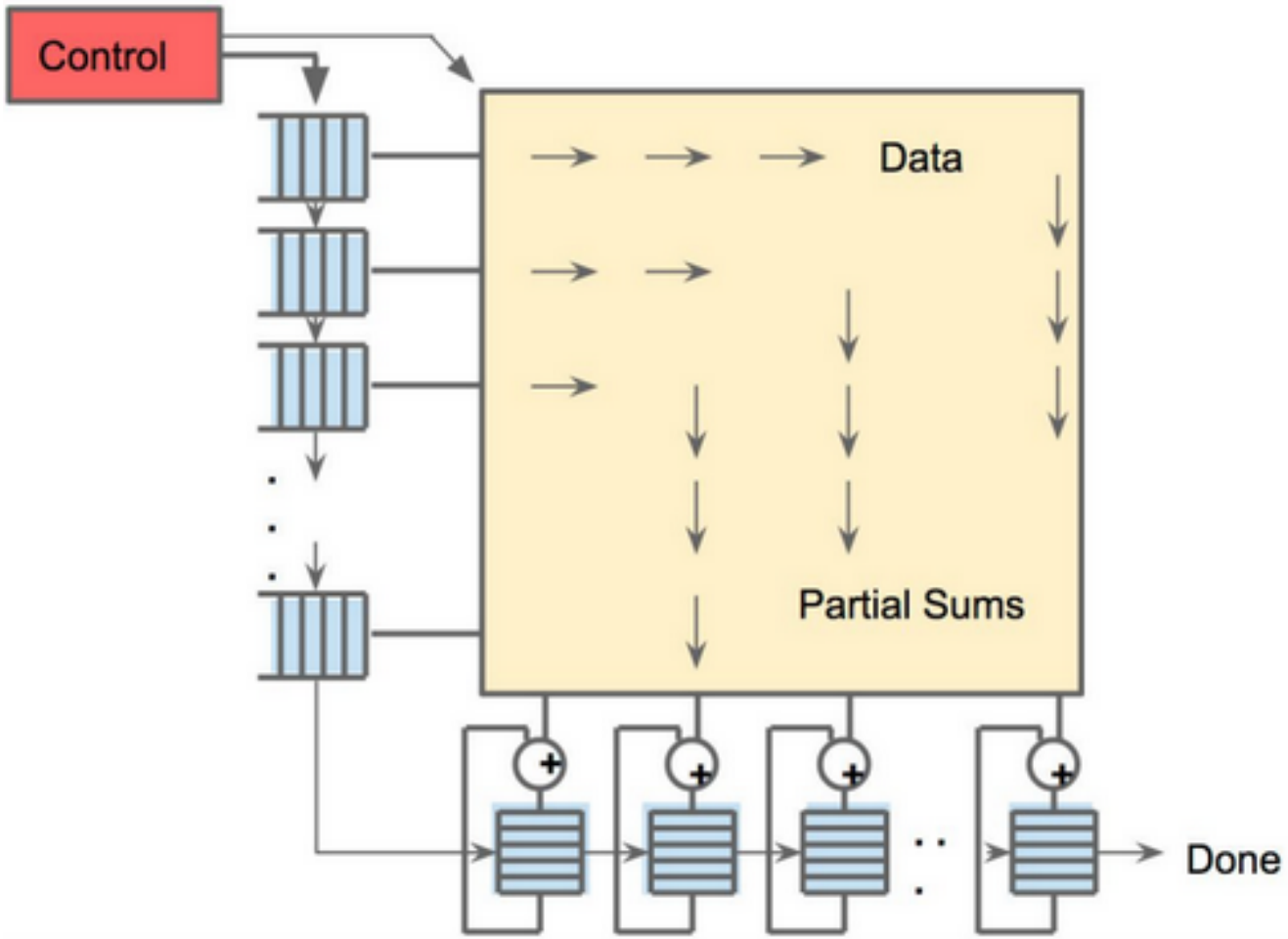


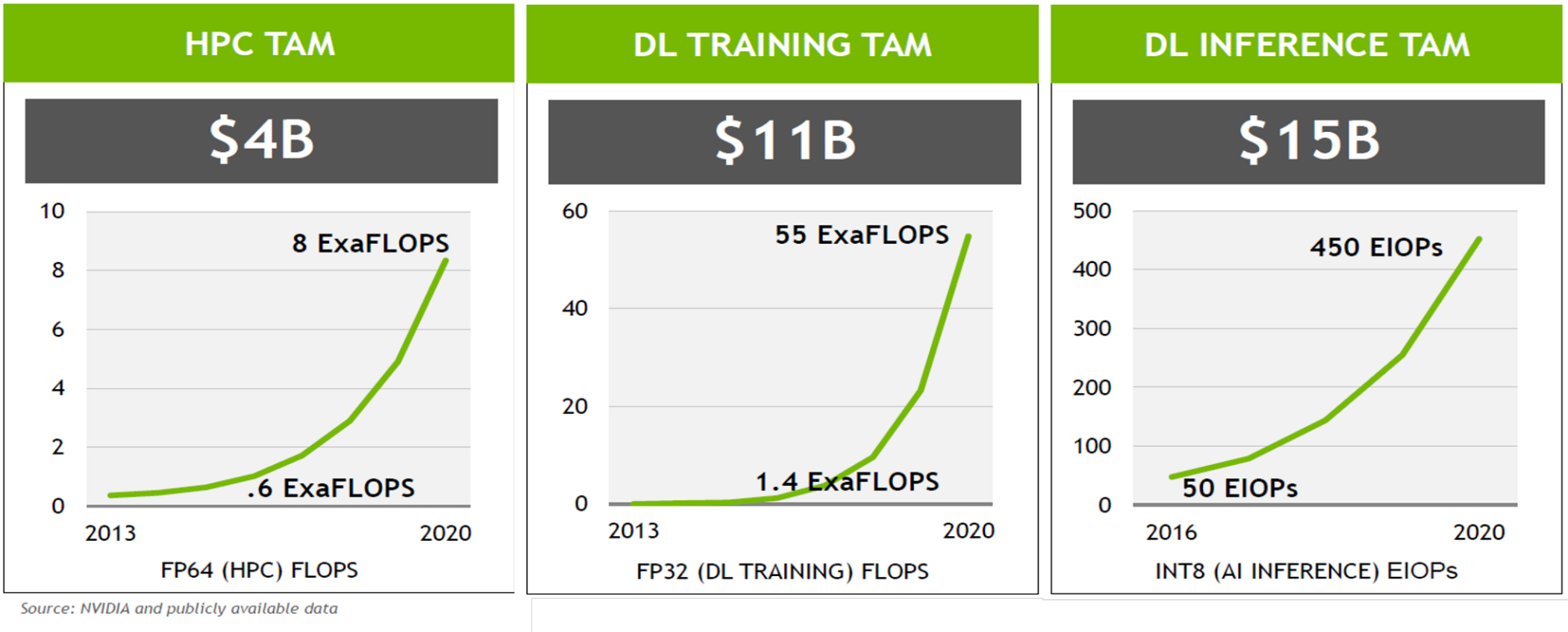
Figure courtesy NVidia. Google TPU also uses low precision.

Google TPU (Tensor Processing Unit)



Hardware pipelining of steps in matrix-multiply. Figure courtesy Google.

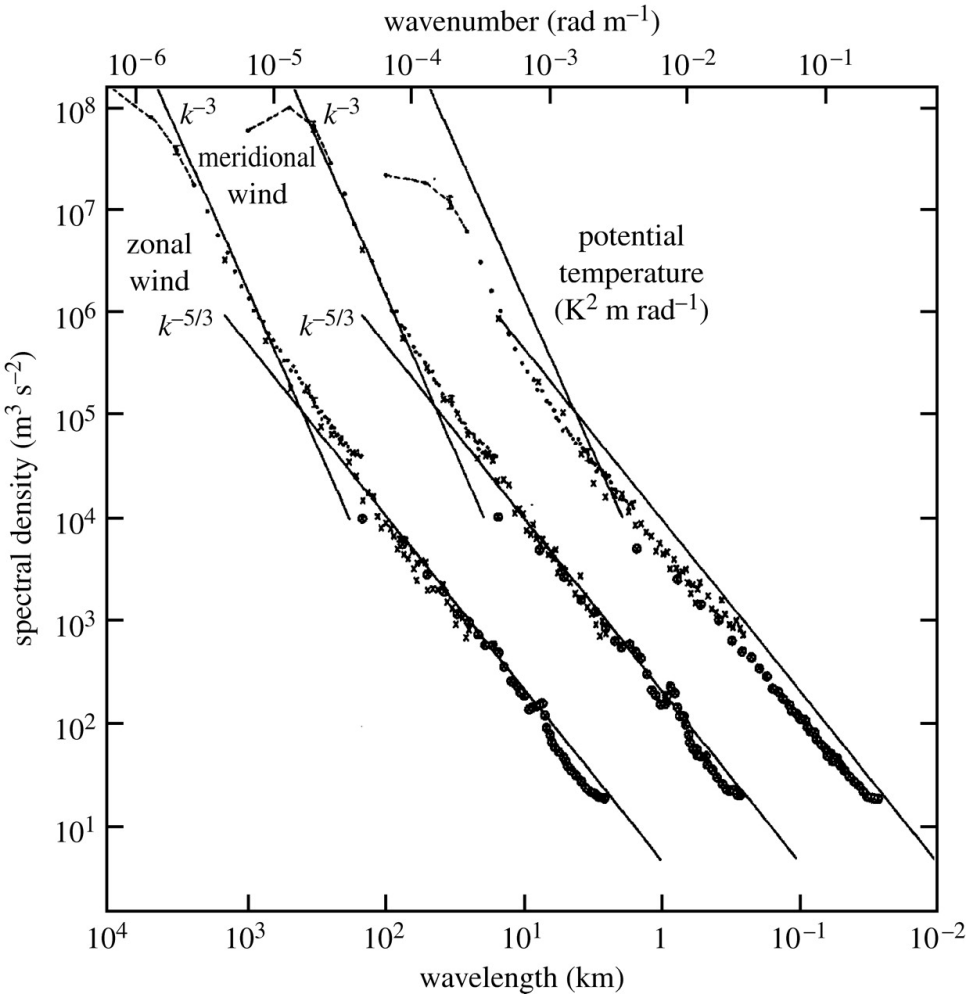
ML is subverting the HPC market



Courtesy NVidia, via Seeking Alpha.

Outline

No separation of "large" and "small" scales



Nastrom and Gage (1985). More fidelity, more complexity over time in small scales ("physics").

Atmospheric process scales

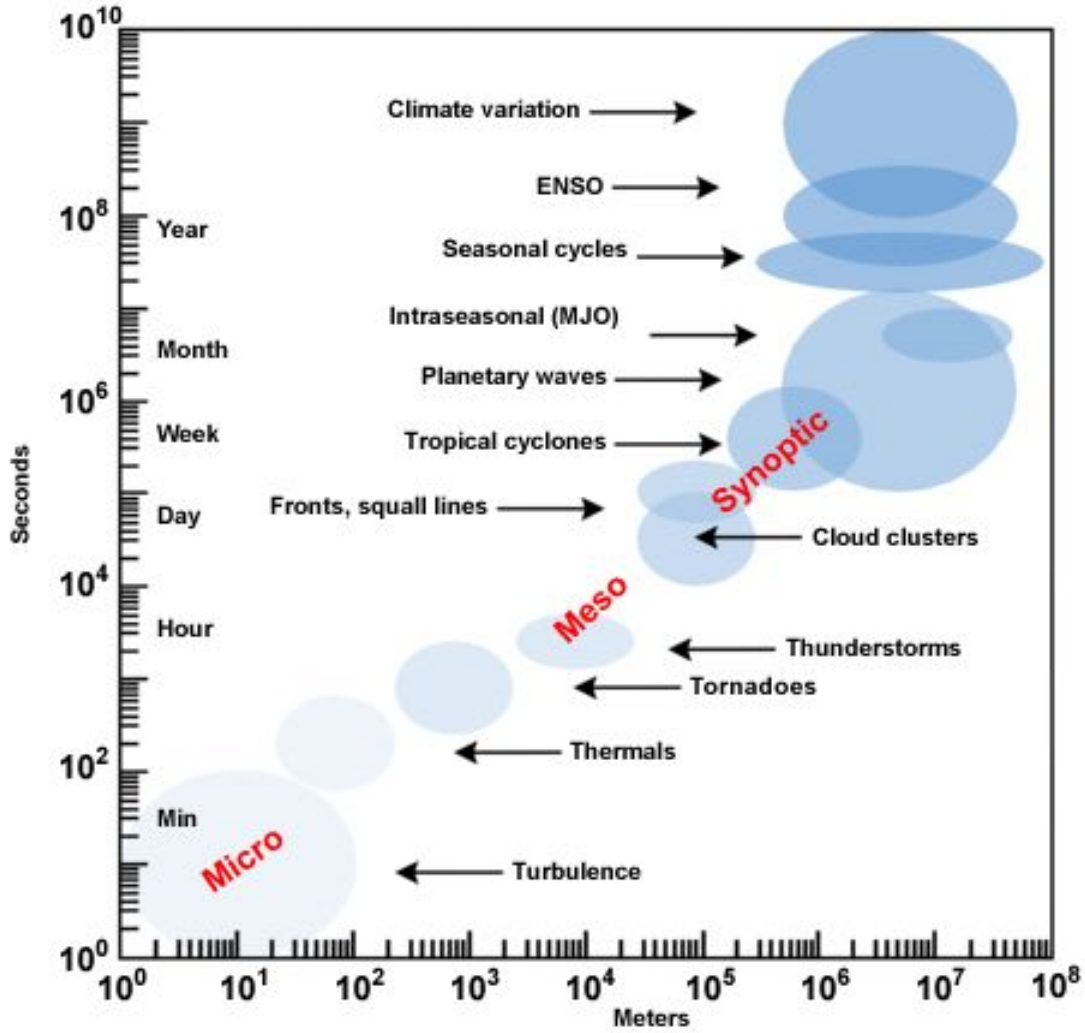


Figure courtesy UCAR.

Oceanic process scales

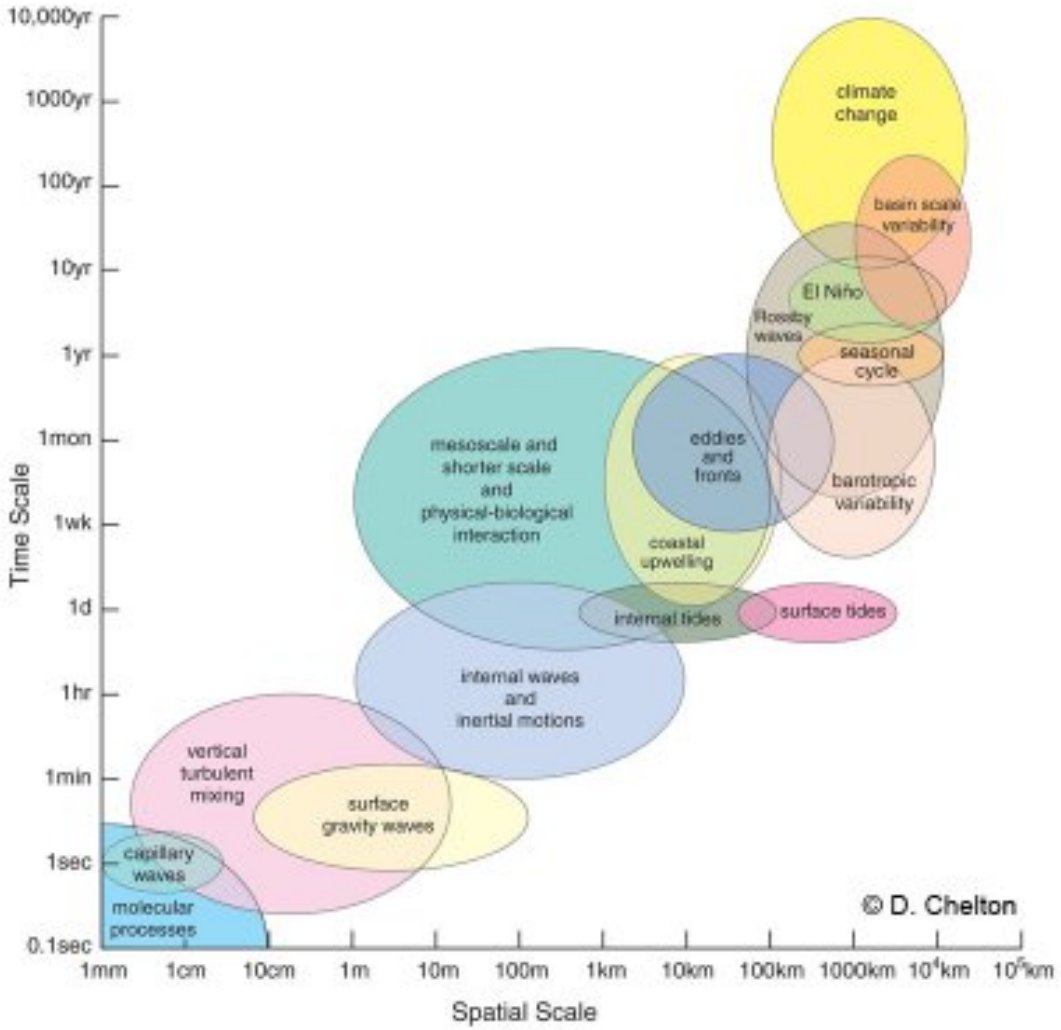
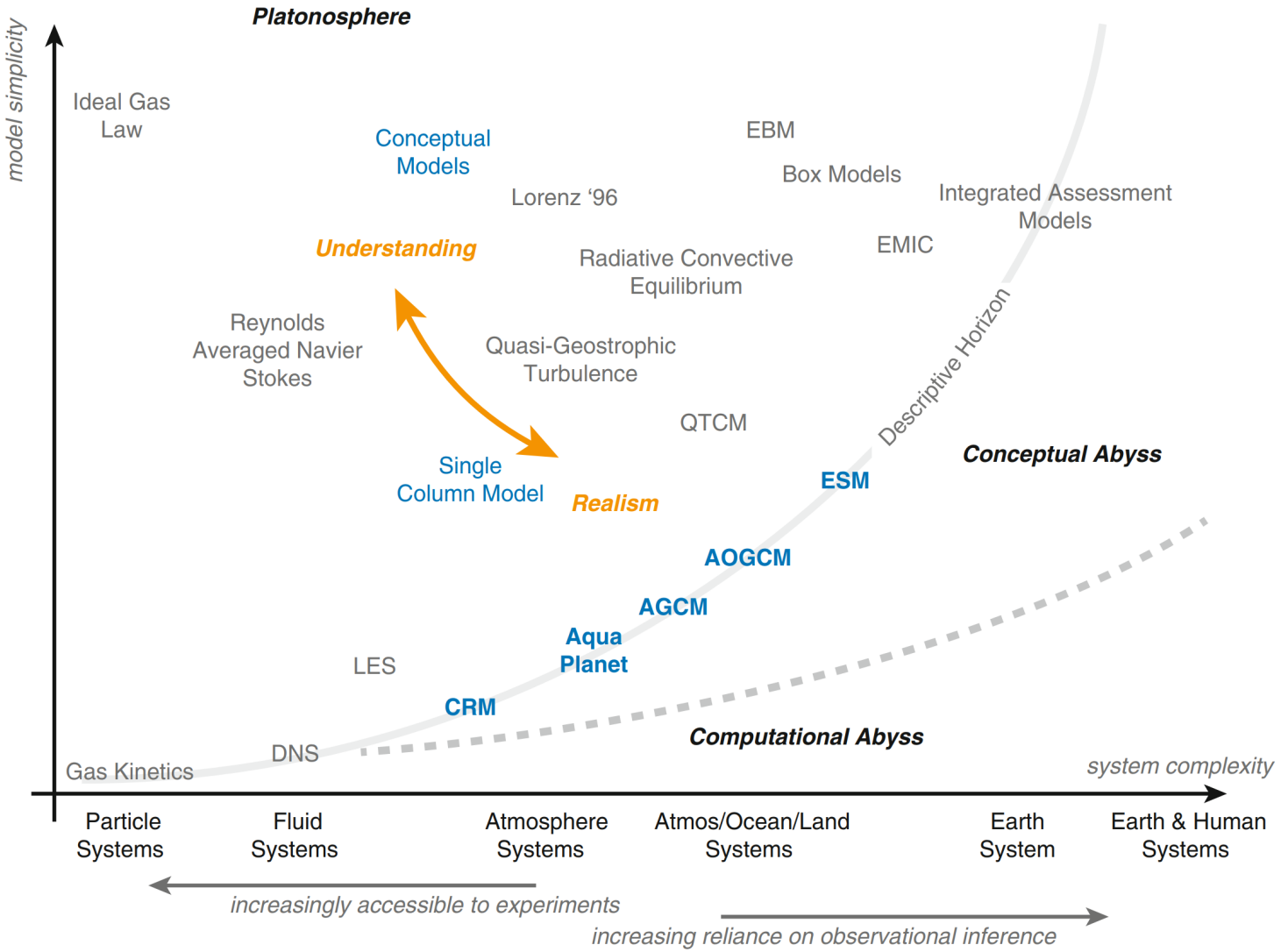


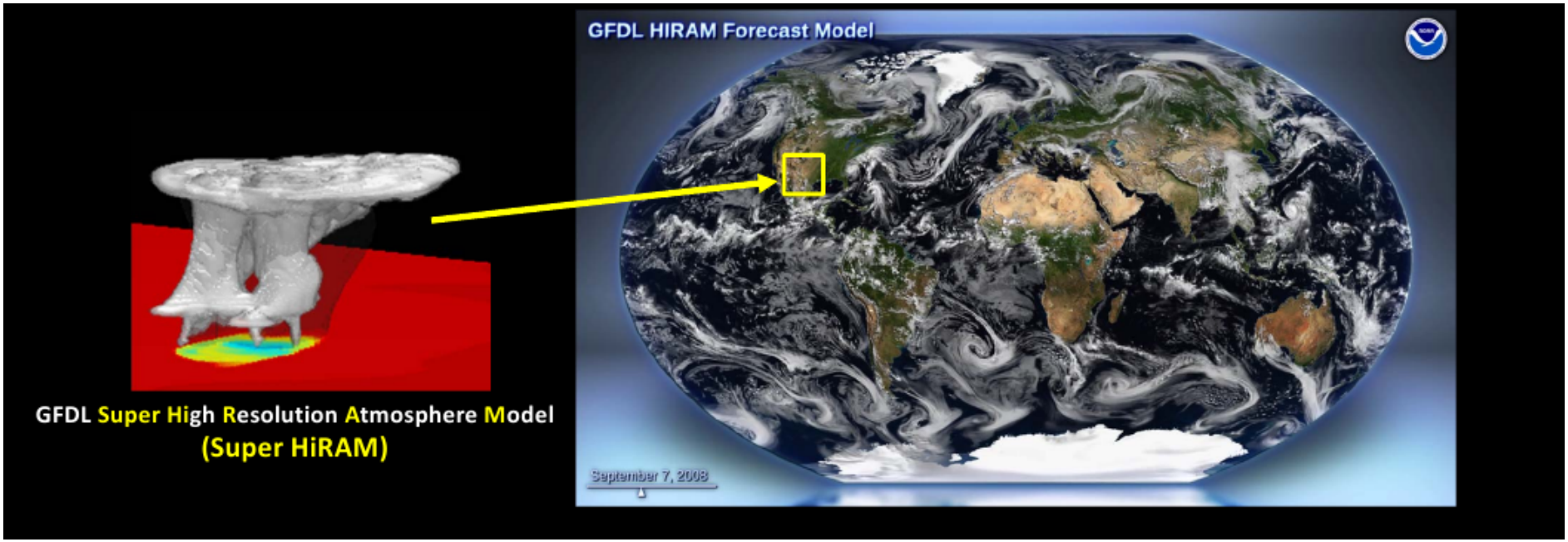
Figure courtesy Oregon State University.

The model zoo



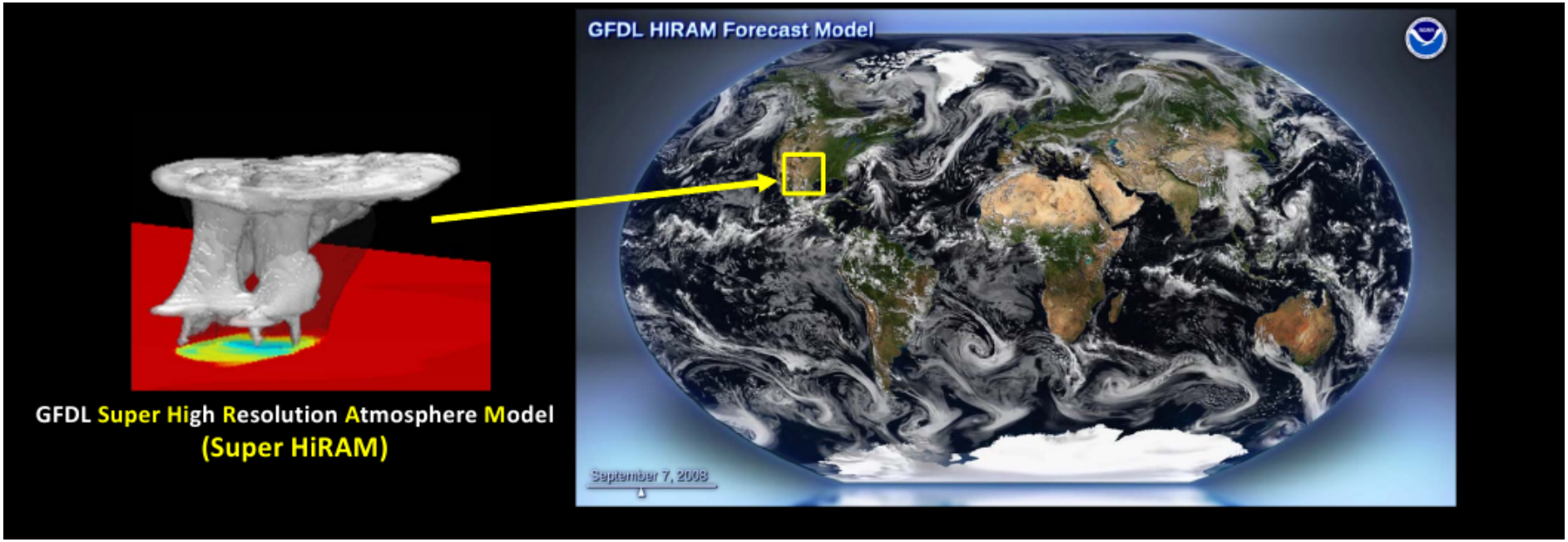
From Bony et al (2013).

NGGPS: Next-Generation Global Prediction System



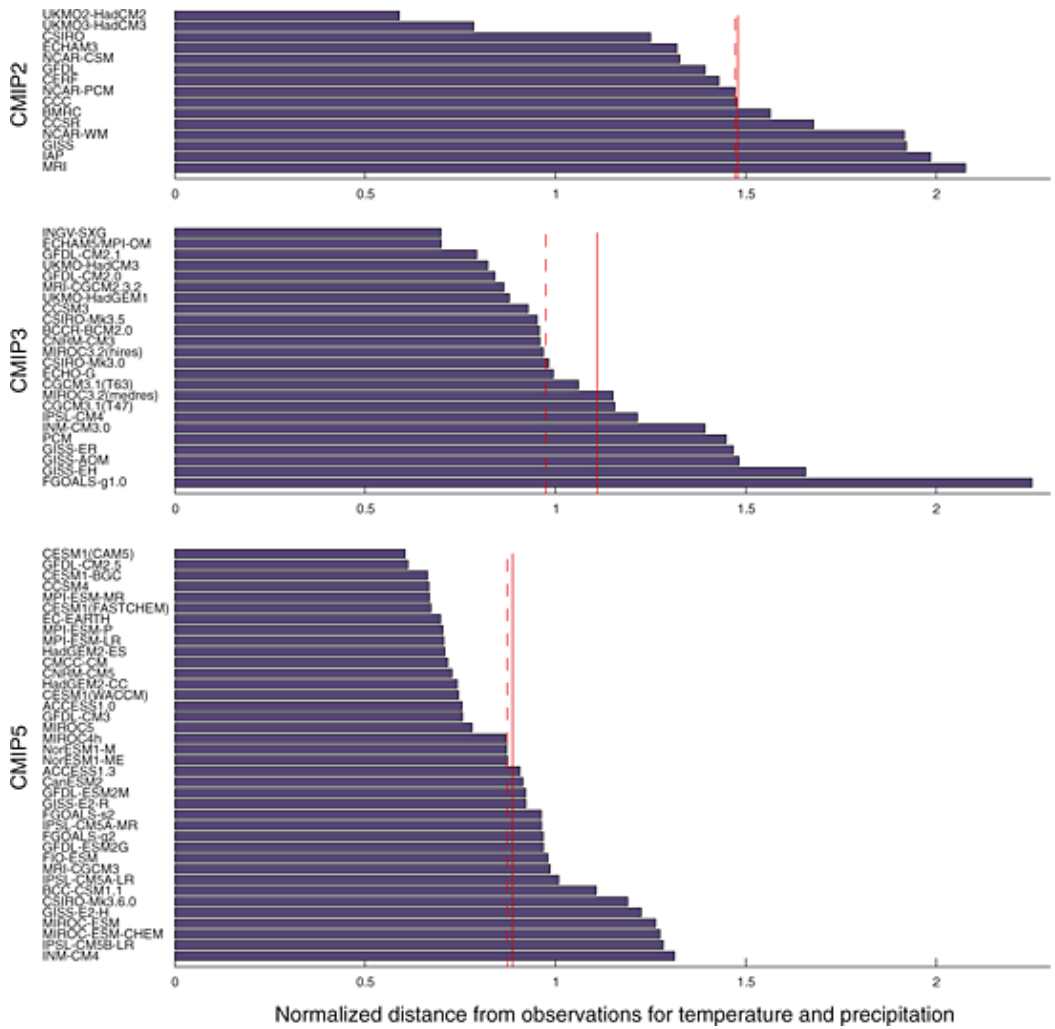
FV3 dynamical core from GFDL for the next-generation forecast model (target: 3 km non-hydrostatic in 10 years running at $\sim$ 200 d/d)

Passing the climate Turing test?



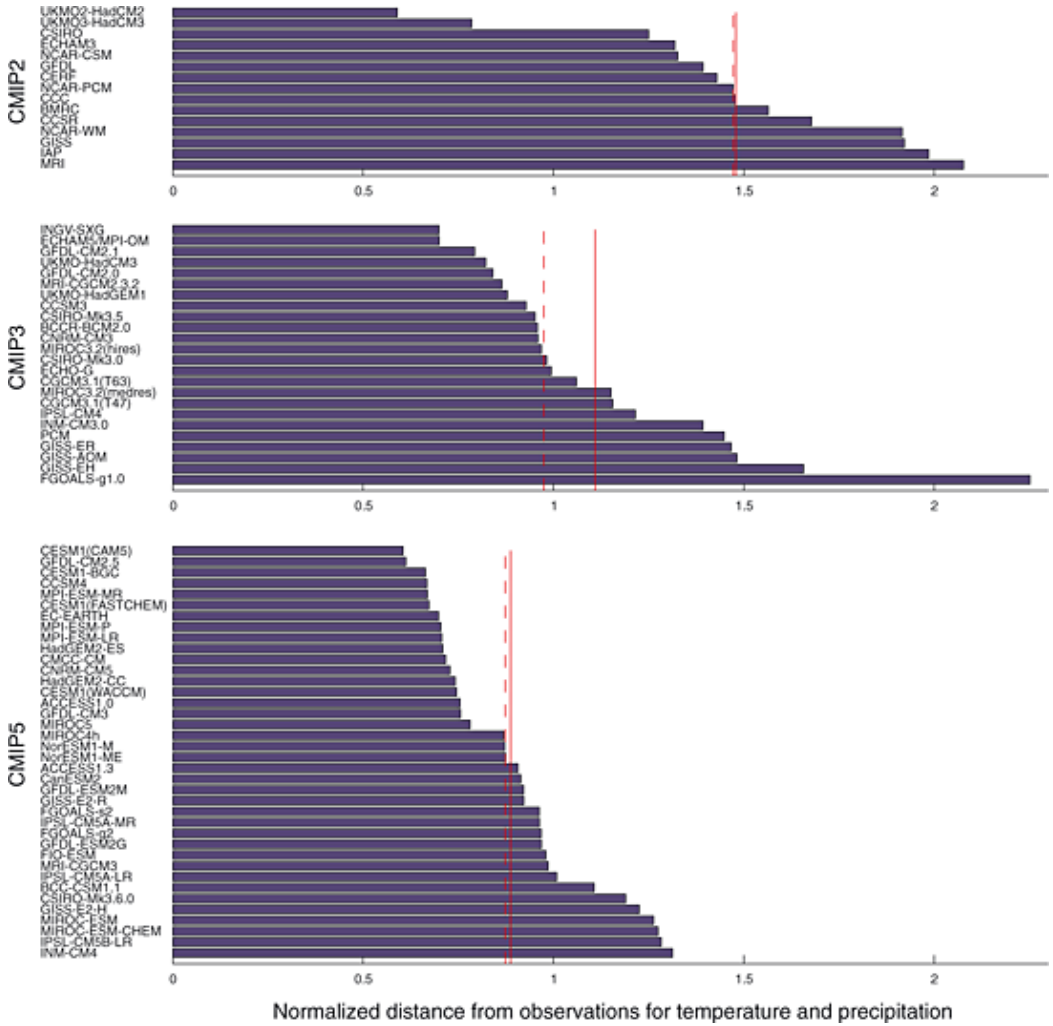
We may be able to simulate everything in great detail, but do we understand how it works?

Multi-model “skill scores”



Based on RMS error of surface temperature and precipitation. (Fig. 3 from Knutti et al, GRL, 2013).

Multi-model skill scores?



More complex models that show the same skill represents an "advance"!

The model hierarchy

- Molecular biology uses a hierarchy of "models": *E. Coli*, *C. Elegans*, fruit fly, mouse, *H. Sapiens*, ...
- We have a similar hierarchy of equations: PG, QG, PE, Boussinesq, anelastic, compressible...
- and model types: LES, CRM, AOGCM, ESM, ...
- and a hierarchy of idealized experiments: turbulent flow, radiative-convective equilibrium, aquaplanet, AMIP, OMIP, control, historical, ...
- Community must run common experiments at all levels of the hierarchy ("idealized MIPs")...
- "Verification" (or falsification) of idealized planet Earth? analysis must isolate underlying mechanisms even in complex models.

Adapted from Held (2005, 2014). Model Hierarchies Workshop, November 2016 in Princeton.

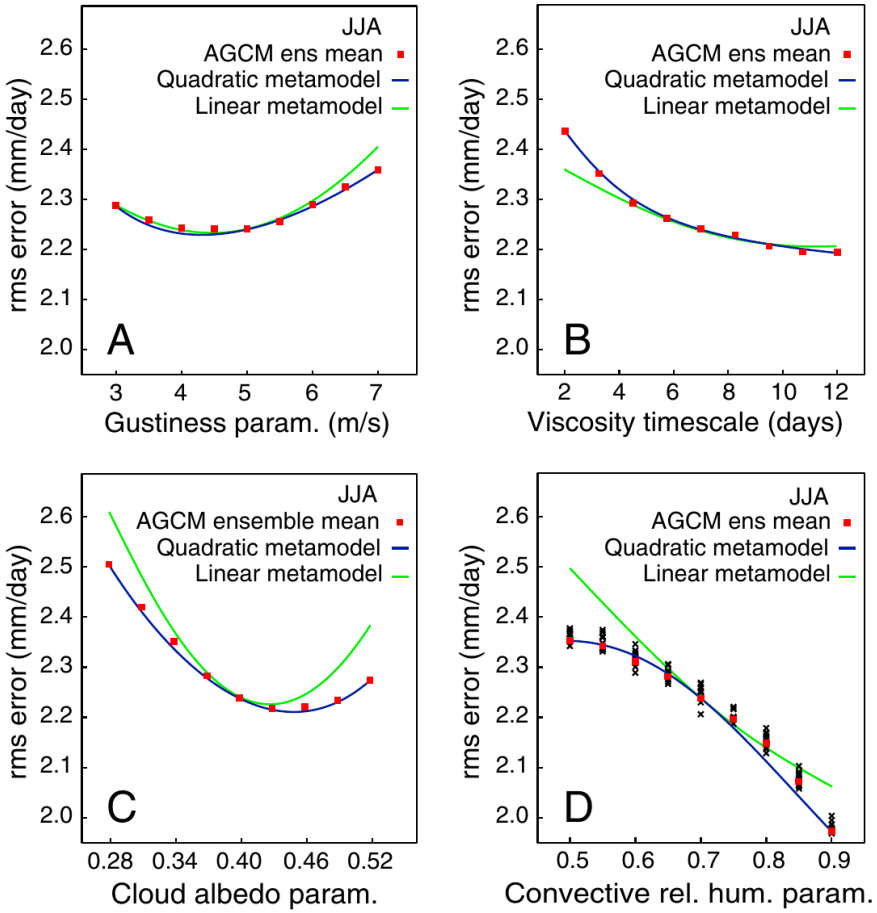
Model calibration

Model calibration or "tuning" consists of reducing overall model bias (usually relative to 20th century climatology) by modifying parameters. In principle, minimizing some cost function:

$$C(p_1, p_2, ...) = \sum_{1}^{N} \omega_i \| \phi_i - \phi_i^{obs} \|$$

- Usually the p must be chosen within some observed or theoretical range $p_{min} \leq p \leq p_{max}$.
- "Fudge factors" (applying known wrong values) generally frowned upon (see Shackley et al 1999 discussion on history of "flux adjustments". More on that later...)
- The choice of ω_i is part of the lab's "culture"!
- The choice of ϕ_i^{obs} is also troublesome:
  - overlap between "tuning" metrics and "evaluation" metrics.
  - "Over-tuning": remember "reality" is but one ensemble member!

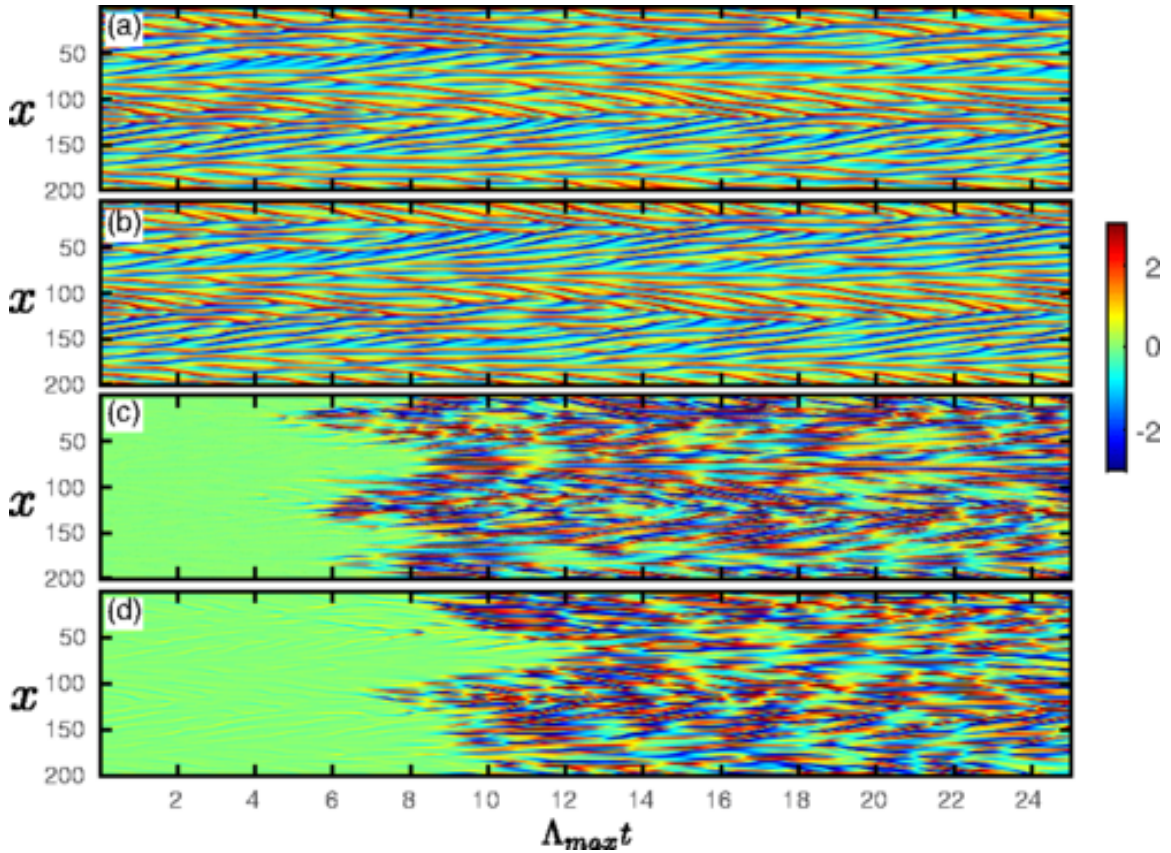
Objective methods of tuning



Neelin et al (2010) construct “metamodels” to aid in multi-parameter optimization. See also Zamboni et al.

Outline

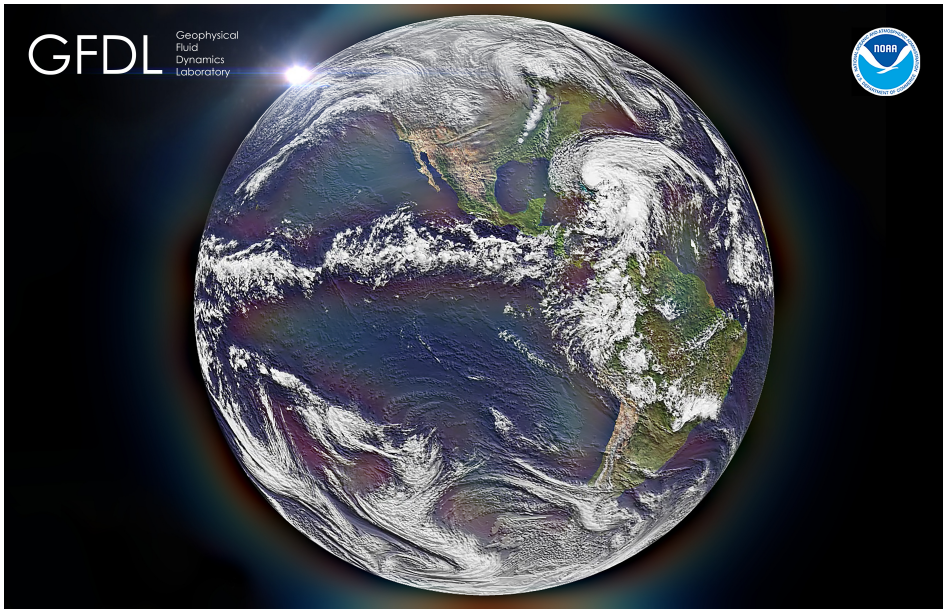
Model-free prediction vs model augmentation



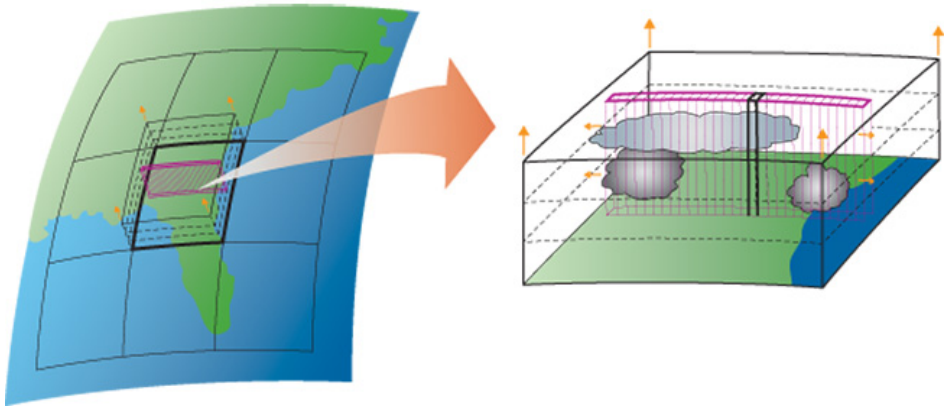
From Pathak et al, PRL (2018), *Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach*

Movie: Pathak's flame front in *Quanta*.

Learn parameterizations from observations



(Courtesy: S-J Lin, NOAA/GFDL).



(Courtesy: D. Randall, CSU; CMMAP).

- Global-scale CRMs (e.g 7 km simulation on the left) and even super-parameterization using embedded cloud models (right) remain prohibitively expensive.
- Can we learn the statistical aggregate of small scales? See Schneider et al 2017, Gentine et al (2018), O'Gorman and Dwyer (2018), Bolton and Zanna (2018), ...

Learning sub-gridscale turbulence

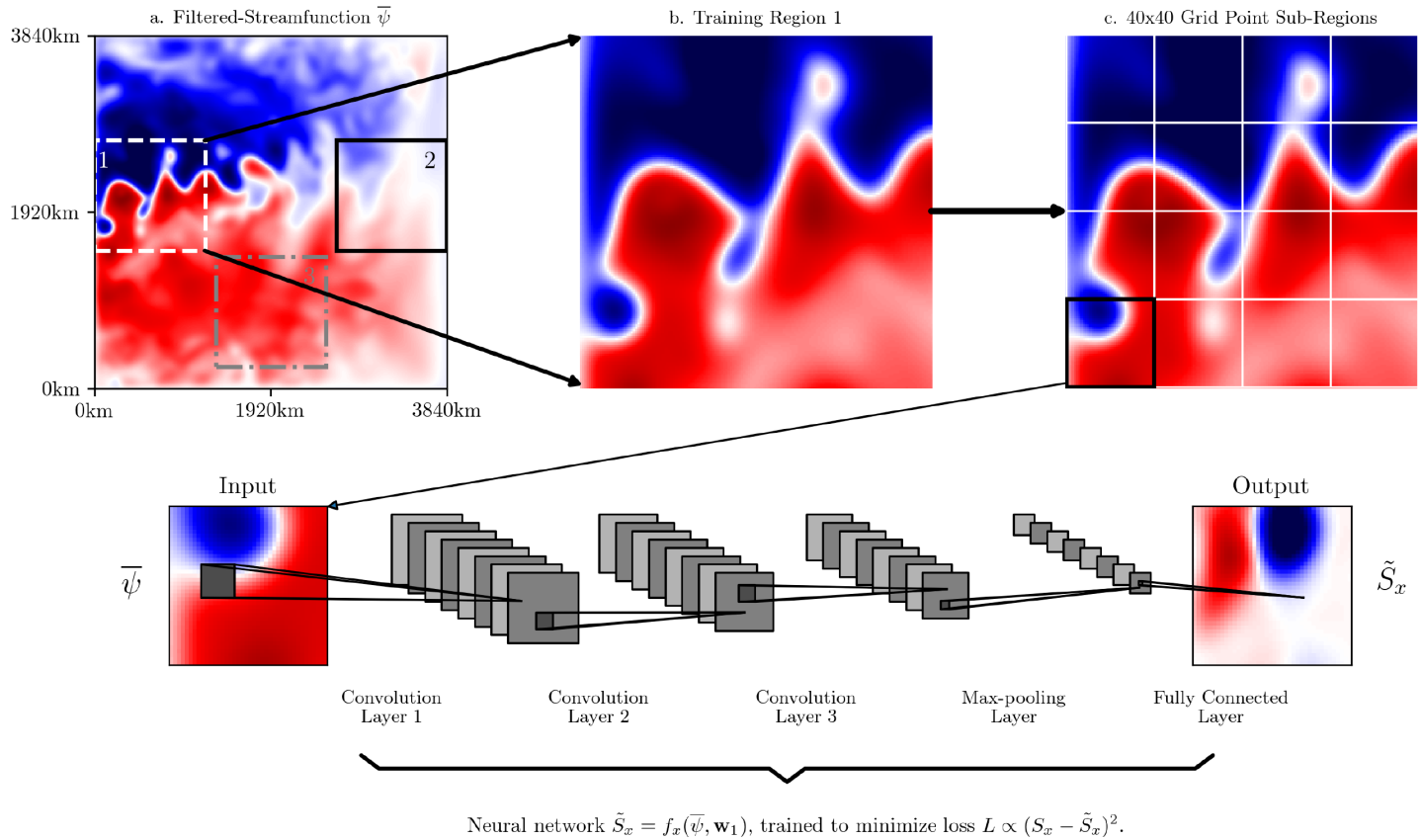
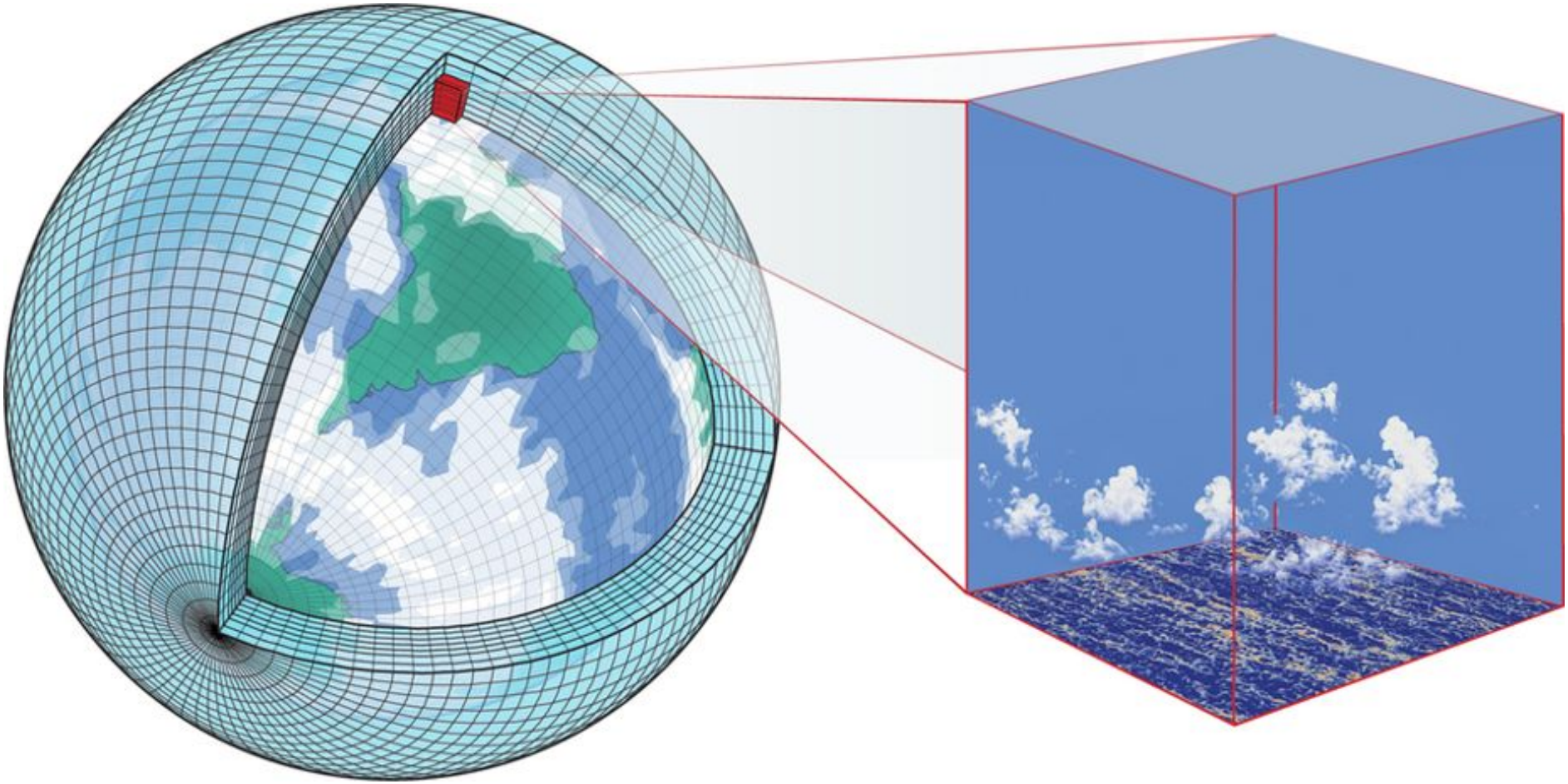


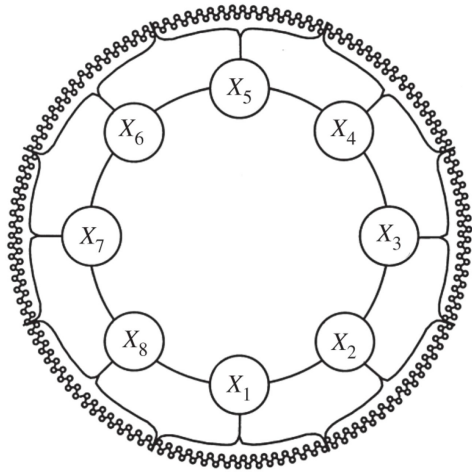
Fig 1 from Bolton and Zanna (2018), in review for JAMES.

Caltech/MIT Earth Machine



From Schneider et al 2017.

Lorenz 96, a nice abstraction

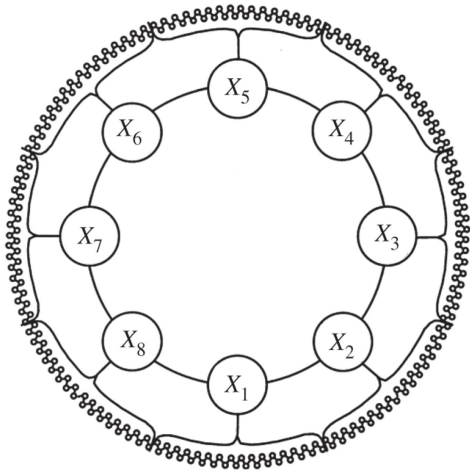


$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F - \frac{hc}{b}\sum_{j=1}^{32} Y_{j,k} + f \qquad (1)$$

$$\frac{dY_{j,k}}{dt} = -cbY_{j+1,l}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + \frac{hc}{b}X_k \qquad (2)$$

A nice abstraction of a system with fast and slow modes, whose coupling strength can be varied... maybe too interesting? See metastability issues in Schneider et al, GRL (2017).

Lorenz 96 with slow and fast forcing

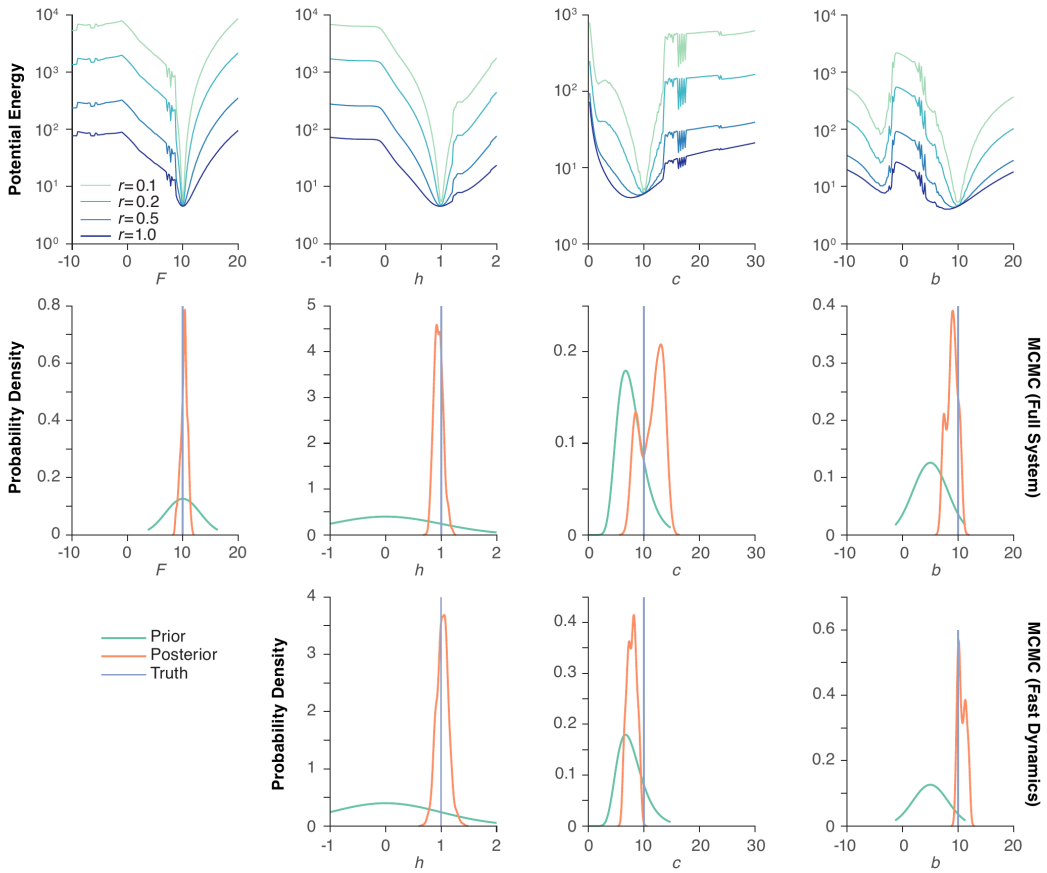


$$\frac{dX_k}{dt} = -X_{k-1}(X_{k-2} - X_{k+1}) - X_k + F + \delta F_k - \frac{hc}{b}\sum_{j=1}^{32} Y_{j,k} + f \quad (3)$$

$$\frac{dY_{j,k}}{dt} = -cbY_{j+1,l}(Y_{j+2,k} - Y_{j-1,k}) - cY_{j,k} + \frac{hc}{b}X_k \quad (4)$$

F can be time-varying on a "slow" timescale (e.g GHG forcing) or "fast" (aerosols). Loosely based on Christensen and Berner (2018).

Lorenz96 in perfect model setting



From Schneider et al 2017. Learn Lorenz96 parameters F, h, c, b from prior run.

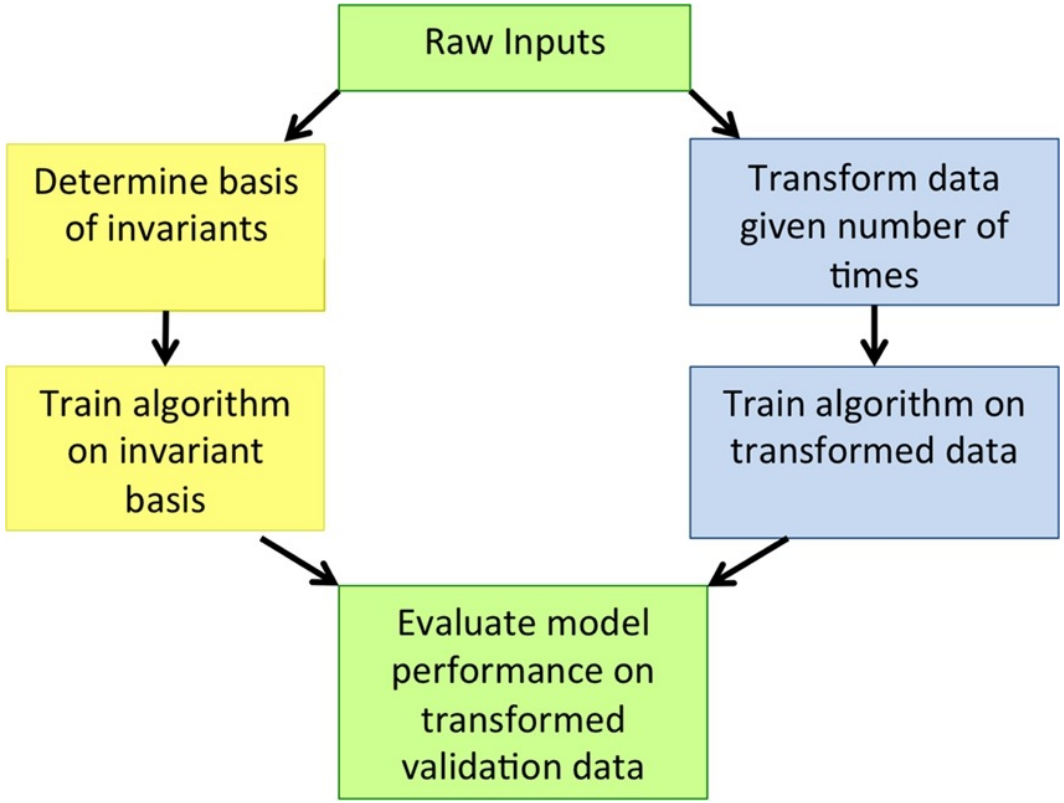
The slow manifold

$$\frac{dU}{dt} = -VW + bVZ - aU \tag{5}$$
$$\frac{dV}{dt} = -UW - bUZ - aV + aF \tag{6}$$
$$\frac{dW}{dt} = -UV - aW \tag{7}$$
$$\frac{dX}{dt} = -Z - aX \tag{8}$$
$$\frac{dZ}{dt} = bUV + X - aZ \tag{9}$$

U, V, W are derived from vorticity ("Rossby waves") and X, Z are derived from divergence and departures from geostrophy ("gravity waves").
From Lorenz (1992), *The slow manifold – what is it?*

Making ML respect known physical constraints

See momentum conservation discussion in Bolton and Zanna (2018), *Applications of Deep Learning to Ocean Data Inference and Sub-Grid Parameterisation*.



From Ling et al, JCP (2016), *Machine learning strategies for systems with invariance properties*

Distilling Free-Form Natural Laws from Experimental Data

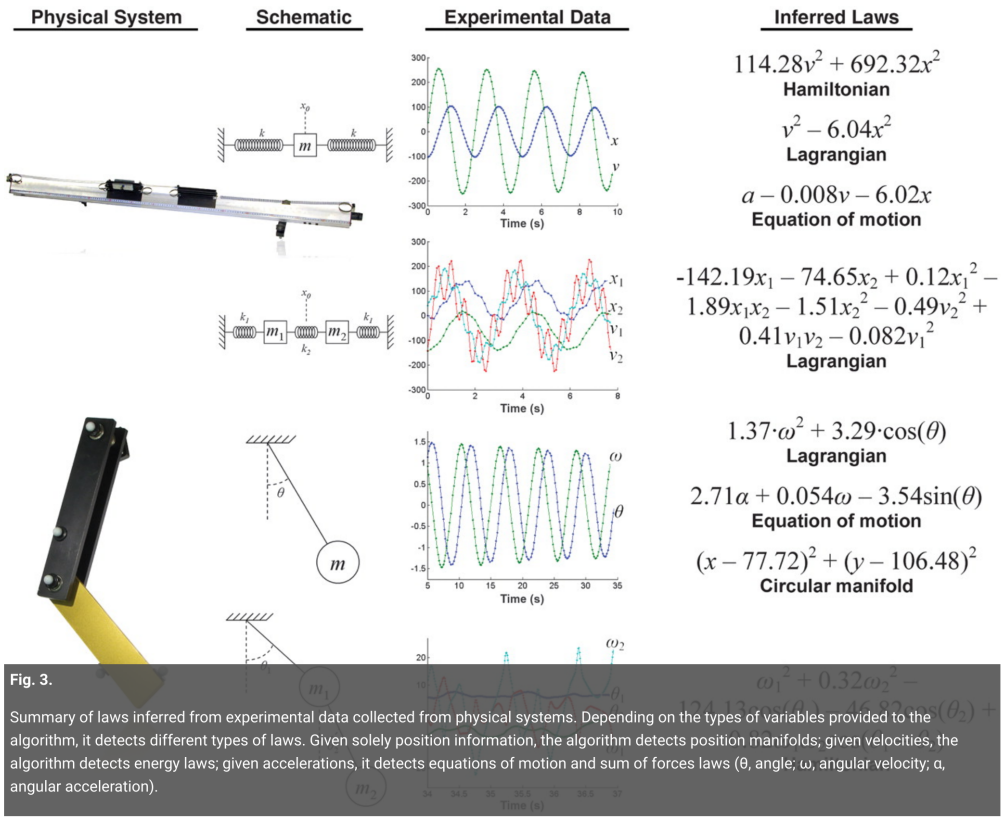


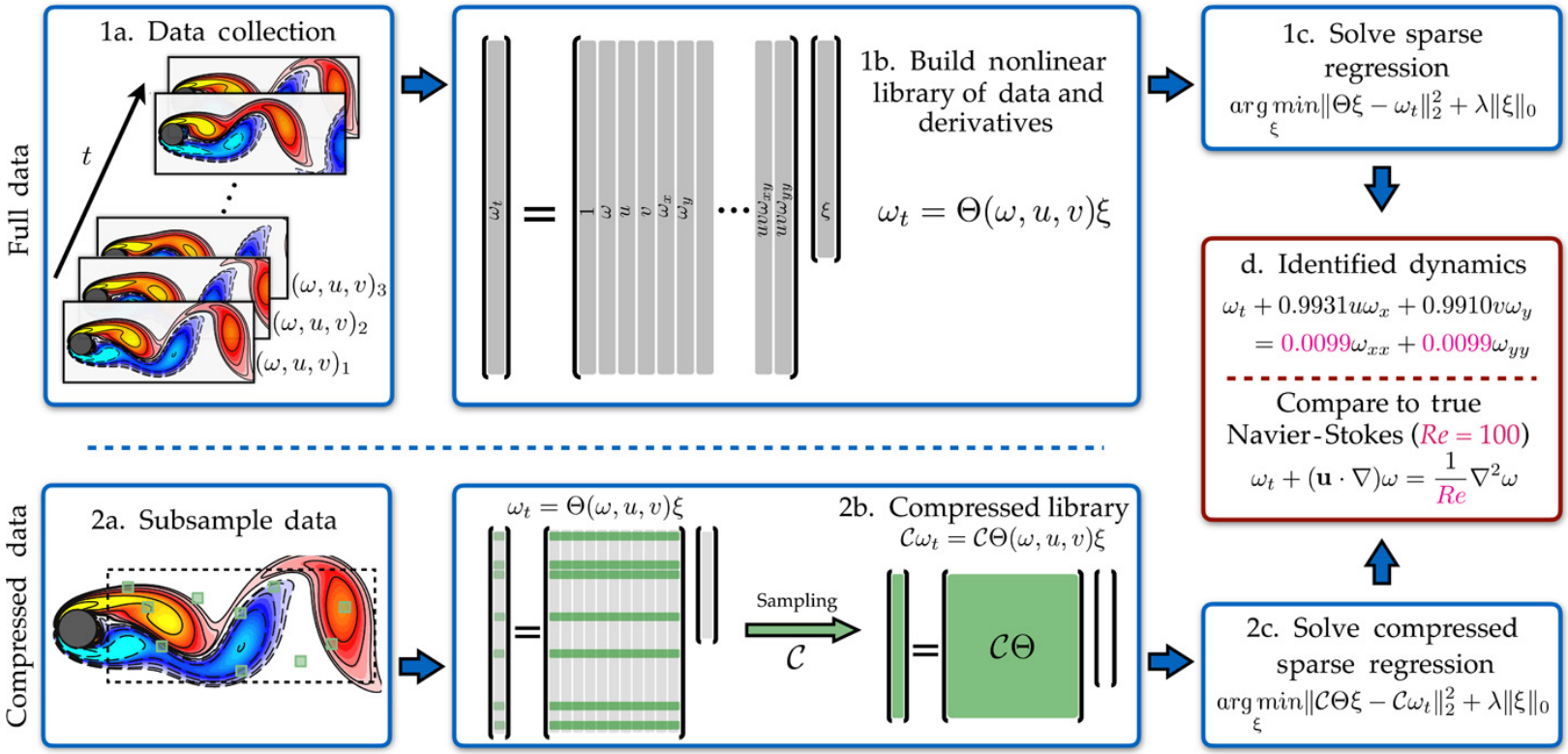
Fig. 3.

Summary of laws inferred from experimental data collected from physical systems. Depending on the types of variables provided to the algorithm, it detects different types of laws. Given solely position information, the algorithm detects position manifolds; given velocities, the algorithm detects energy laws; given accelerations, it detects equations of motion and sum of forces laws (θ, angle; ω, angular velocity; α, angular acceleration).

3 of 4

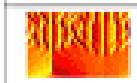
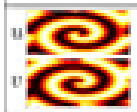
From Schmidt and Lipson, *Science*, 2009. My little *hommage*, Gaitán et al (2016), *Can we obtain viable alternatives to Manning's equation using genetic programming? Eureqa* software available under license.

Navier-Stokes from data



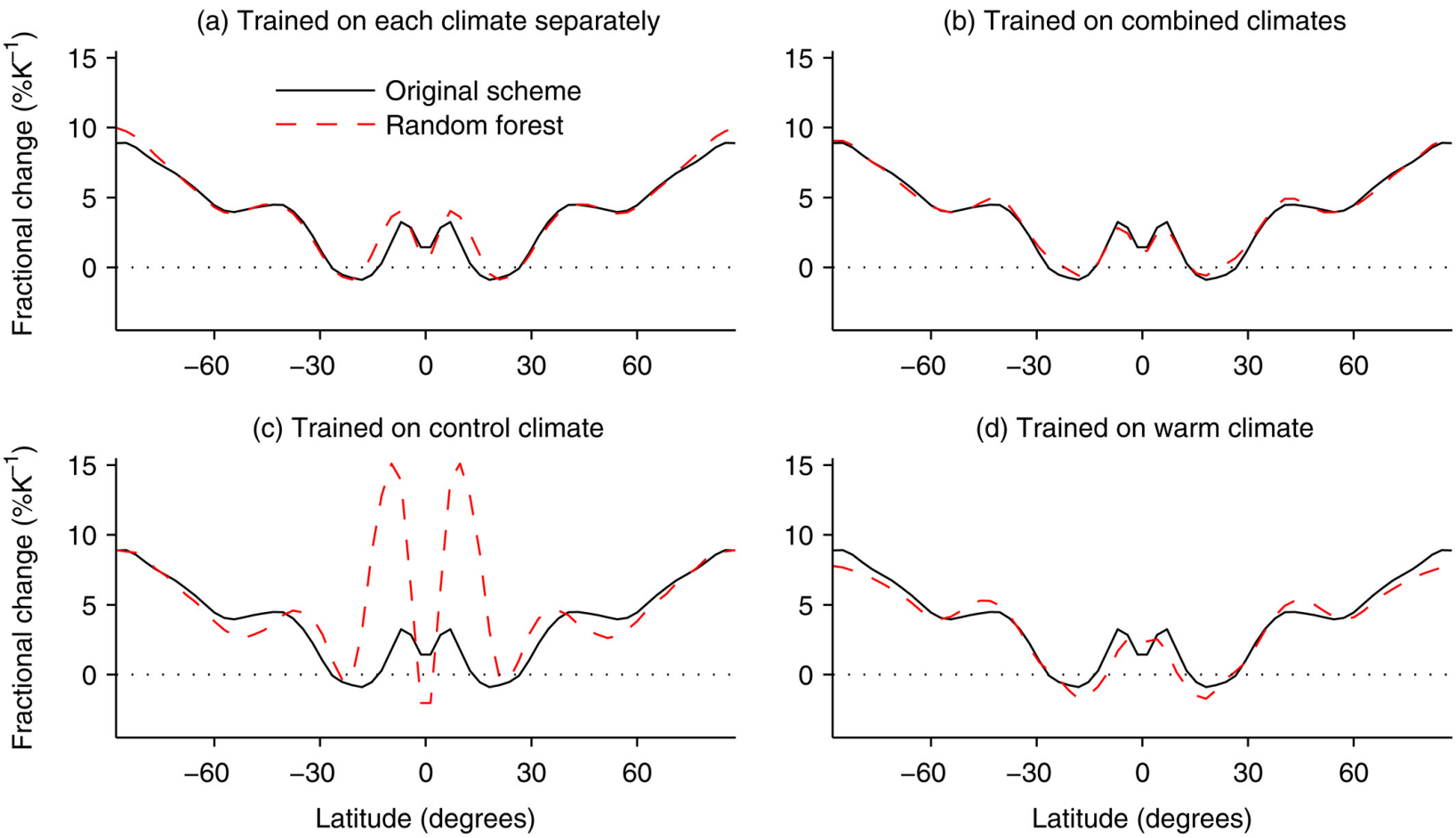
From Rudy et al (2017).

Other canonical PDEs from data

PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	$1 \pm 0.2\%, 7 \pm 5\%$	$x \in [-30, 30], n = 512,\ t \in [0, 20], m = 201$
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	$0.15 \pm 0.06\%, 0.8 \pm 0.6\%$	$x \in [-8, 8], n = 256,\ t \in [0, 10], m = 101$
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	$0.25 \pm 0.01\%, 10 \pm 7\%$	$x \in [-7.5, 7.5], n = 512,\ t \in [0, 10], m = 401$
 NLS	$iu_t + \frac{1}{2}u_{xx} + \lvert u\rvert^2 u = 0$	$0.05 \pm 0.01\%, 3 \pm 1\%$	$x \in [-5, 5], n = 512,\ t \in [0, \pi], m = 501$
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	$1.3 \pm 1.3\%, 52 \pm 1.4\%$	$x \in [0, 100], n = 1024,\ t \in [0, 100], m = 251$
 Reaction Diffusion	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	$0.02 \pm 0.01\%, 3.8 \pm 2.4\%$	$x, y \in [-10, 10], n = 256,\ t \in [0, 10], m = 201$ subsample 1.14%
 Navier-Stokes	$\omega_t + (\mathbf{u} \cdot \nabla)\omega = \frac{1}{Re}\nabla^2\omega$	$1 \pm 0.2\%\ ,\ 7 \pm 6\%$	$x \in [0, 9], n_x = 449,\ y \in [0, 4], n_y = 199,$ $t \in [0, 30], m = 151,$ subsample 2.22%

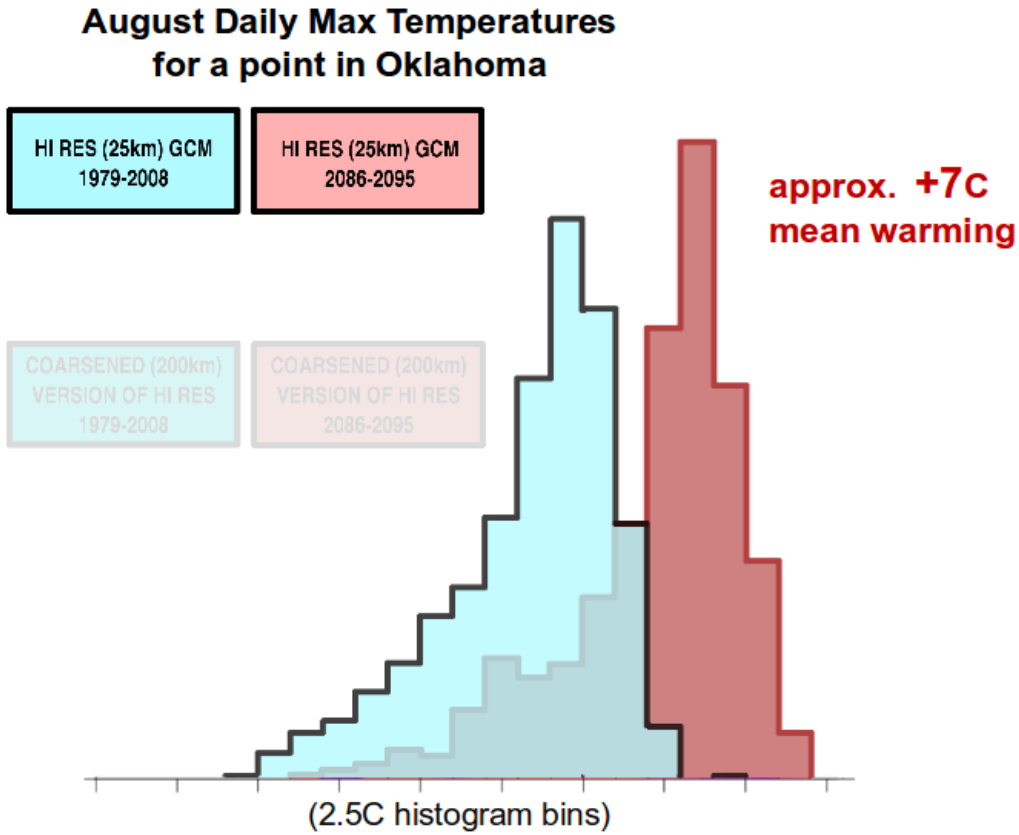
From Rudy et al (2017). Works for other canonical PDE systems as well! (including conservative and non-linear dissipative chaotic systems.

Limitations of training data



From O'Gorman and Dwyer, JAMES, 2018. Limitations of training on short non-stationary time series.

Error patterns associated with stationarity assumption



Errors can be traced with warming outside the temperature distribution of the training period. Caution needed at distribution tails (“extreme events”). Dixon et al (2016).

Where models and data are both weak...

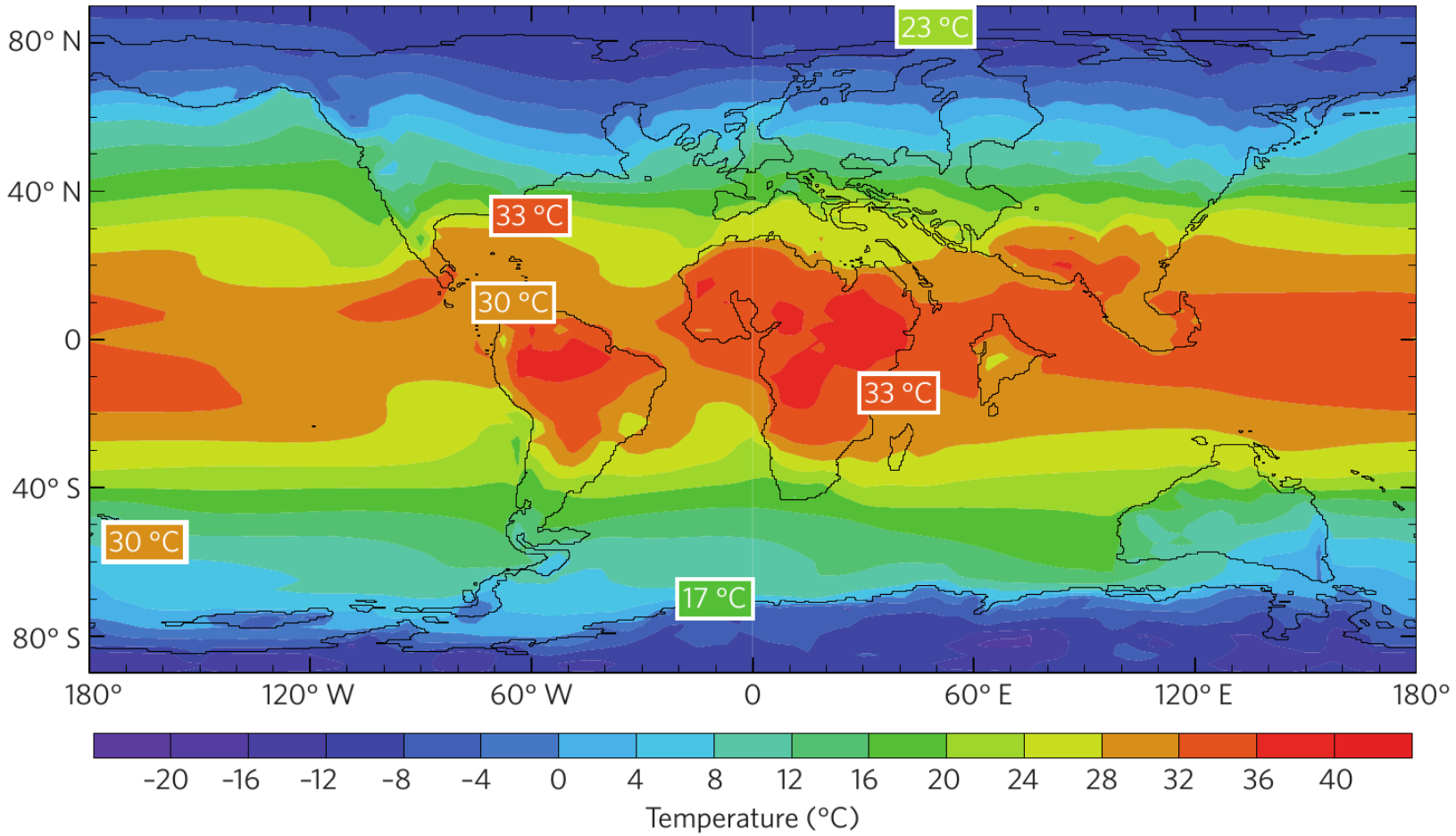


Fig 1 from Valdes (2011). GCMs are unable to simulate the Paleocene-Eocene climate of 55 My ago.

Outline

Questions: metamodels and supermodels

- Machine learning and "AI" still is in the positive phase of a hype cycle (publication bias, reproducibility crisis) but it isn't *all* hype. It is subverting the HPC hardware market.
- Huge variety of methods!
- Supermodels: some components replaced by learning agents. Metamodels: low-dimensional emulators.
- Fundamental questions still unanswered:
  - Are model-free methods useful?
  - How do we derive the invariant basis of a complex system?
  - Can we use ML to derive the functional form of a slow manifold?
  - Can we derive a useful model hierarchy?
  - Can this metamodel be used for parameter uncertainty exploration?
  - How much physical knowledge (e.g conservation laws) must be embedded in the ML? What if the embedded knowledge is incorrect? ("*It's not what you don't know, it's what you know for sure that just ain't so*", Mark Twain never said.)
  - What happens to supermodels as the features of the training data evolve?