Ensemble Consistency Tests for ESM Development

Allison Baker

Applications Scalability and Performance Group
Computational Information Systems Laboratory, NCAR

with **Dorit Hammerling** (Colorado School of Mines),
Teo Price-Broncucia (CU Boulder),
Daniel Milroy, Stephen Molinari, Galen Vincent, Haiying Xu,
and many others!

March 4, 2024

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Need for Software Quality Assurance

CESM results are Bit-for-bit (BFB) reproducible if:

same software version,

same compiler and flags,

same MPI,

same parameters settings,

same initial conditions,

same hardware*,…

→ *not typically the case!*

Need for Software Quality Assurance

CESM results are Bit-for-bit (BFB) reproducible if:

same software version,

same compiler and flags,

same MPI,

same parameters settings*,*

same initial conditions*,*

same hardware*,*…*

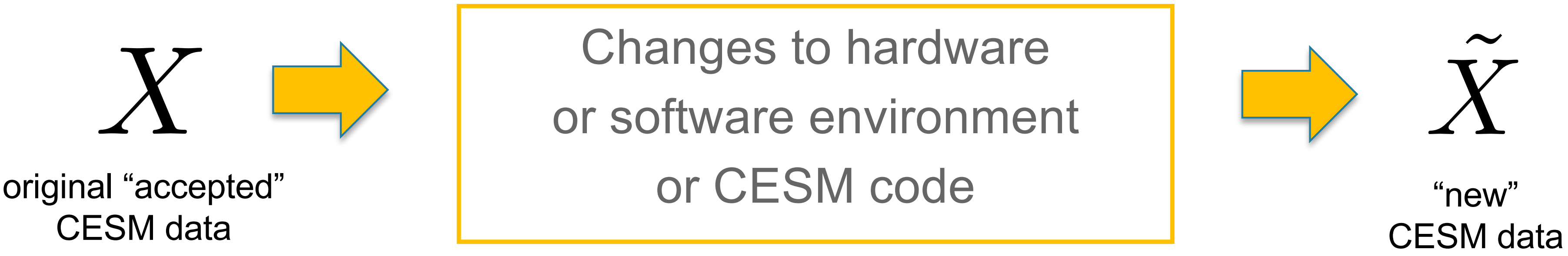
→ *not typically the case!*

Many changes during the CESM development cycle are not BFB:

- port to new environment (e.g., different institution)
- compiler changes
- code modifications (e.g., optimizations)
- heterogeneous computing (e.g., GPUs)

Ensure that changes during the CESM development life cycle do not adversely affect the results!

Motivation



X

original “accepted” CESM data

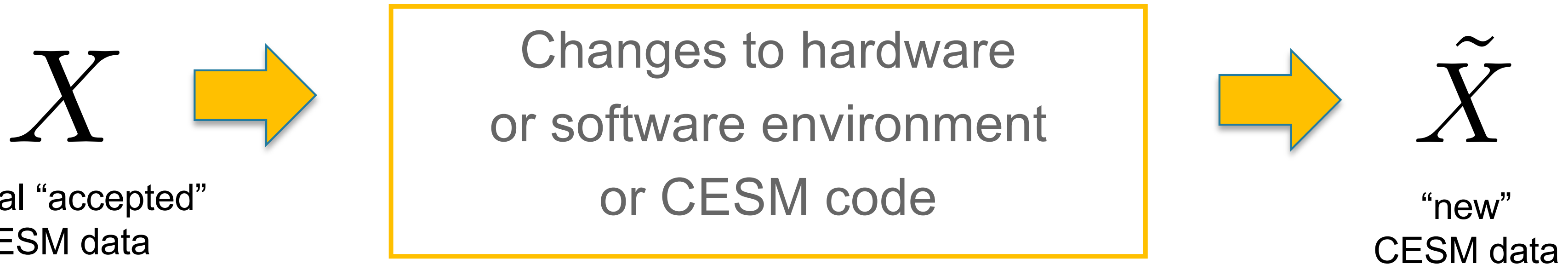
Changes to hardware or software environment or CESM code

$\tilde{X}$

“new” CESM data

Key question: If $X \neq \tilde{X}$ is the output still correct?

Motivation



X

original “accepted” CESM data

Changes to hardware or software environment or CESM code

$\tilde{X}$

“new” CESM data

Key question: If $X \neq \tilde{X}$ is the output still correct?

Does the new data still represent the same climate?
Or is it “climate-changing”?

Evaluating the difference

Question: How can we assess whether the difference between X and $\tilde{X}$ is climate-changing?

Challenge: there is no clear definition of “climate-changing”

Past approach: compare long simulations (~400 years)

- climate expertise required
- subjective
- computationally expensive
- time consuming

Need an automated tool!

- easy-to-use
- objective

Towards an easy-to-use, objective, automated tool …

Question: How can we assess whether the difference between X and $\tilde{X}$ is climate-changing?

Let's reframe the problem!

Our New Approach: Ensemble Consistency Test

New question: Is the new data *statistically distinguishable* from the original data?

Approach: evaluate in the context of the climate model's ***variability***

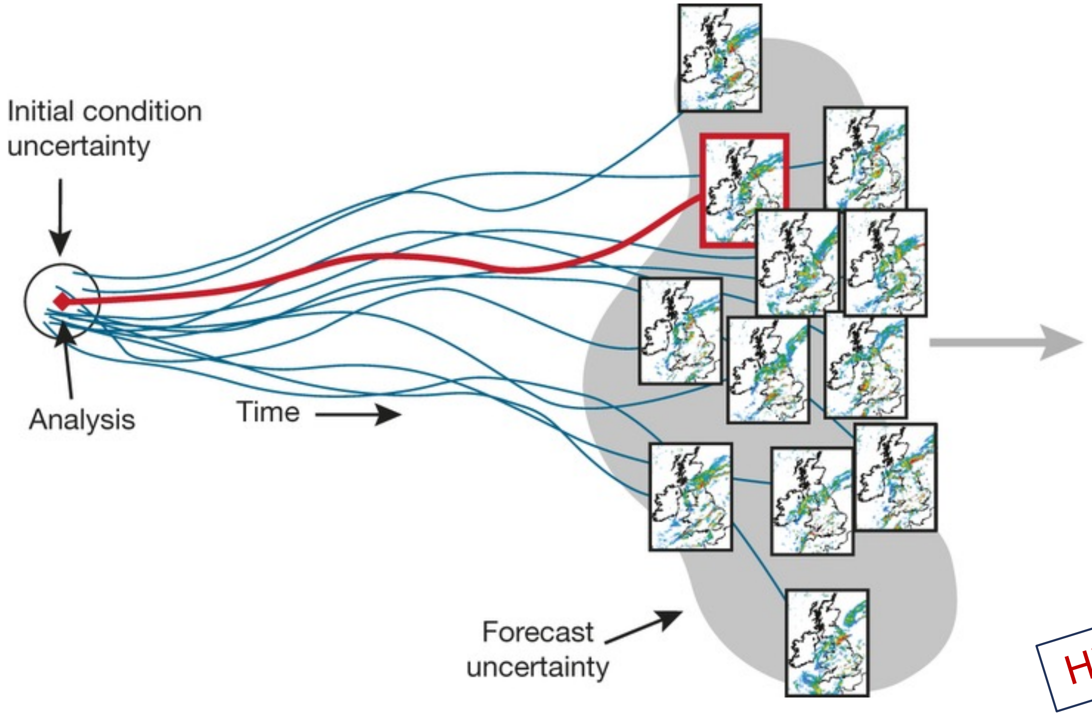


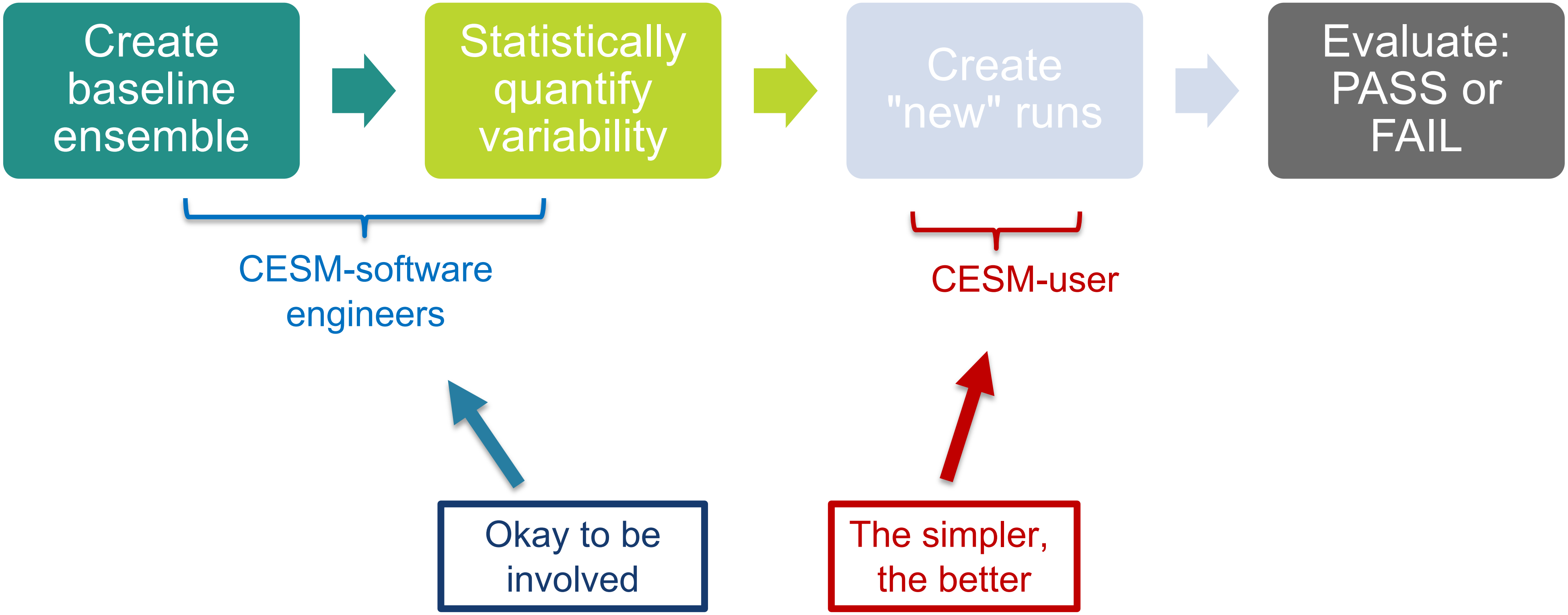
Image from G. Danabasoglu

i.e., evaluate new data in the context of an ensemble of "accepted" CESM runs

High dimensionality is a key issue …

Ensemble Consistency Test (ECT)

Overview:



Create baseline ensemble

Statistically quantify variability

Create "new" runs

Evaluate: PASS or FAIL

CESM-software engineers

CESM-user

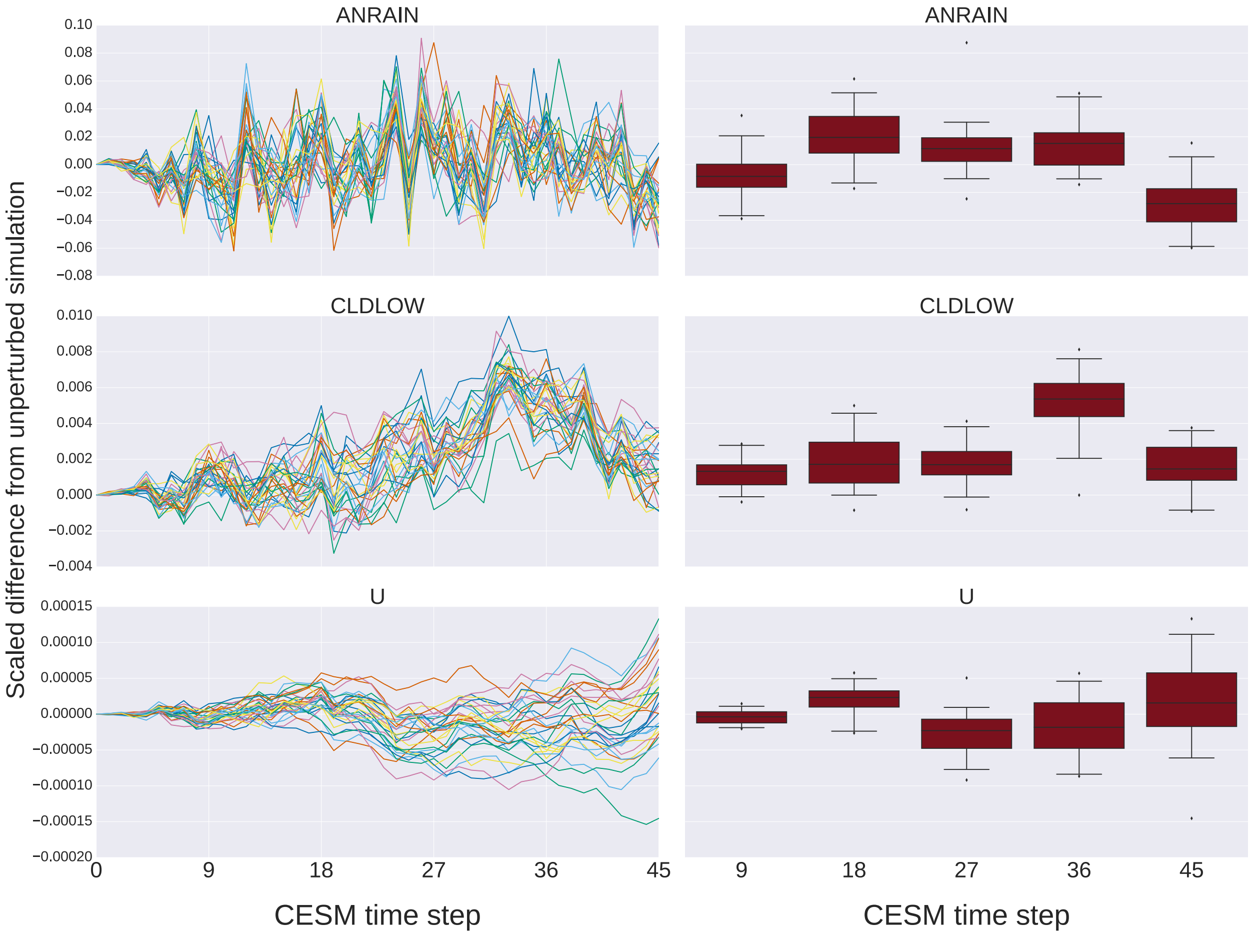
Okay to be involved

The simpler, the better

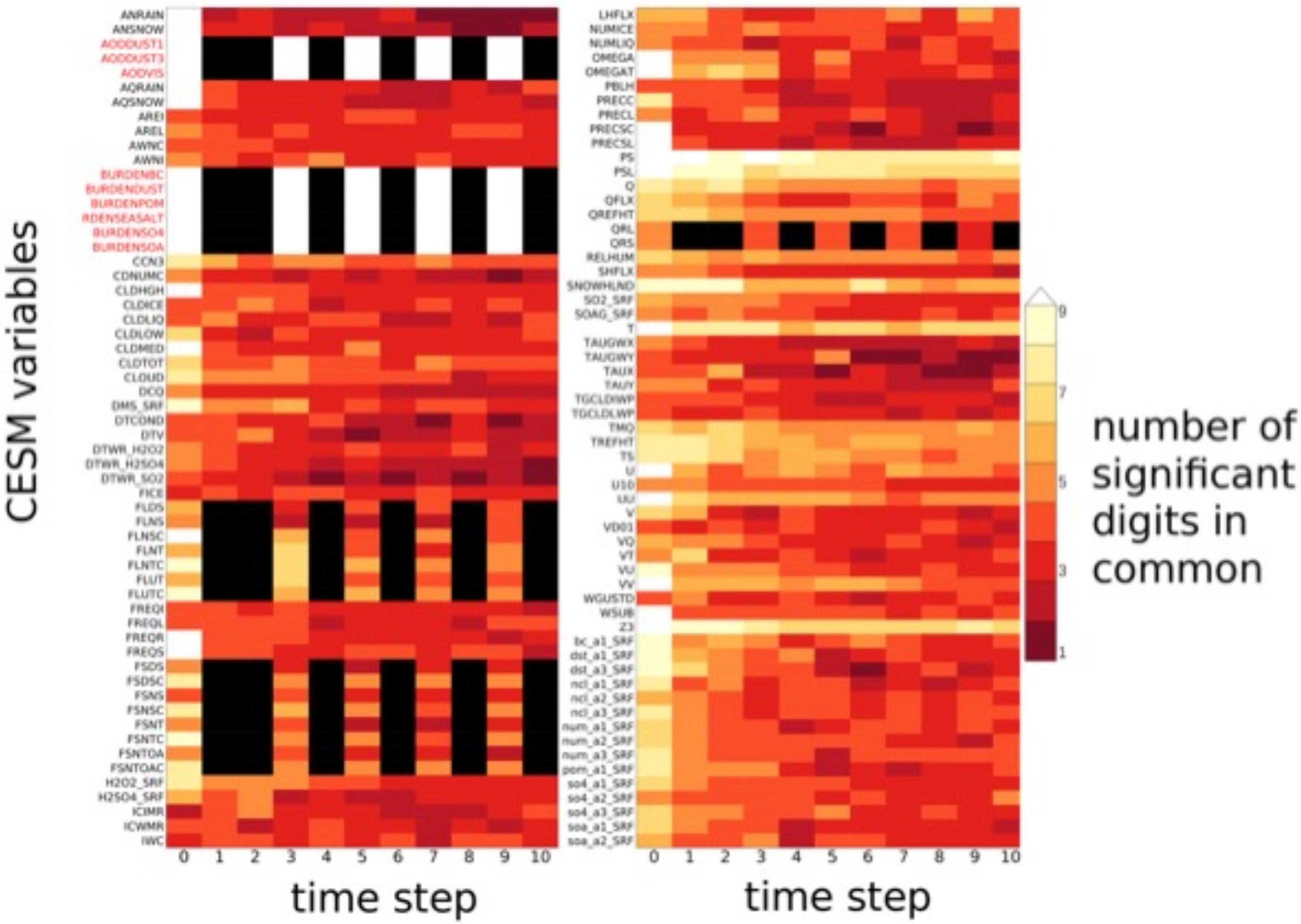
Ensemble and testing idea

Create baseline ensemble of CESM runs:

- “accepted” machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- 200+ variables
- short simulations



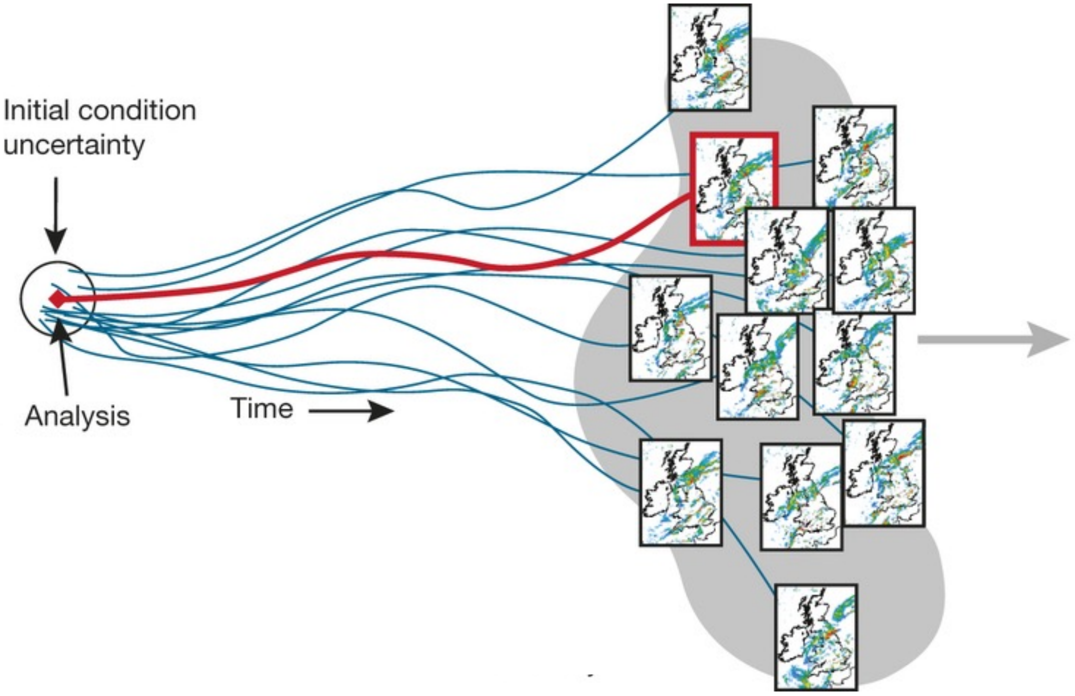
First 10 time steps: original vs. perturbed



Ensemble and testing idea

Create baseline ensemble of CESM runs:

- “accepted” machine/software stack
- 1-deg atmosphere and land
- *O* (10^{-14}) perturbations in initial temperature
- 200+ variables
- **9 time step simulations** (*4.5 hours model time - initially it was 1 year*!)



*The **mean** of each field is all that is needed…*

Ensemble and testing idea

Create baseline ensemble of CESM runs:

- "accepted" machine/software stack
- 1-deg atmosphere and land
- O (10^{-14}) perturbations in initial temperature
- 200+ **globally averaged** variables
- 9 time step simulations

Compare variable value in "new" run to the ensemble distribution:

- many variables are highly correlated!
- difficult to make pass/fail choices based on variables

→ use principal component analysis!

Quantify ensemble variability

New testing tool based on Principal Component Analysis (PCA):

- standardize variables (different scales)
- project data into orthogonal space
  (orthogonalize data in the direction of maximized variability…)
- resulting linear combinations of variables (scores) are used for the ensemble distribution
- use enough scores to represent most of the variance

compare scores from new runs to distribution of scores from ensemble

Hypothesis Testing based on Principal Components

Key: picks up "correctness" of relationships between variables

null hypothesis (H_0): the new climate simulations come from the same distribution as the ensemble simulations.

ECT issues a pass or fail, and must balance:

- false positive rate: probability of falsely rejecting H_0 when it is true
- power: the probability of correctly rejecting H_0 when it is false

Ideally:

- false positive rate is as low as possible
- power is as high as possible

©Johnny Sajem · illustrationsOf.com/1048884

How well does ECT work?

Works extremely well in practice

- modifications *expected* to be climate-changing *fail*
  - e.g. relative humidity, dust emissions, CO_2 levels
- modifications *not expected* to be climate-changing *pass*
  - e.g., threads, -O0, compiler version, code rearrangement
- *hard to find any “real error” it doesn’t catch!*

Currently in-use:

- CESM port verification and code optimization (e.g., GPUs)
- automated Python tool in CESM release
- uncovered errors in code and hardware
- CAM, CLM, POP
- *climate-modeling expertise is not required!*

How can we apply ECT to other models?

Recent work: Developing a “recipe” for applying the ECT to new (or updated) models

Determining the test parameters:

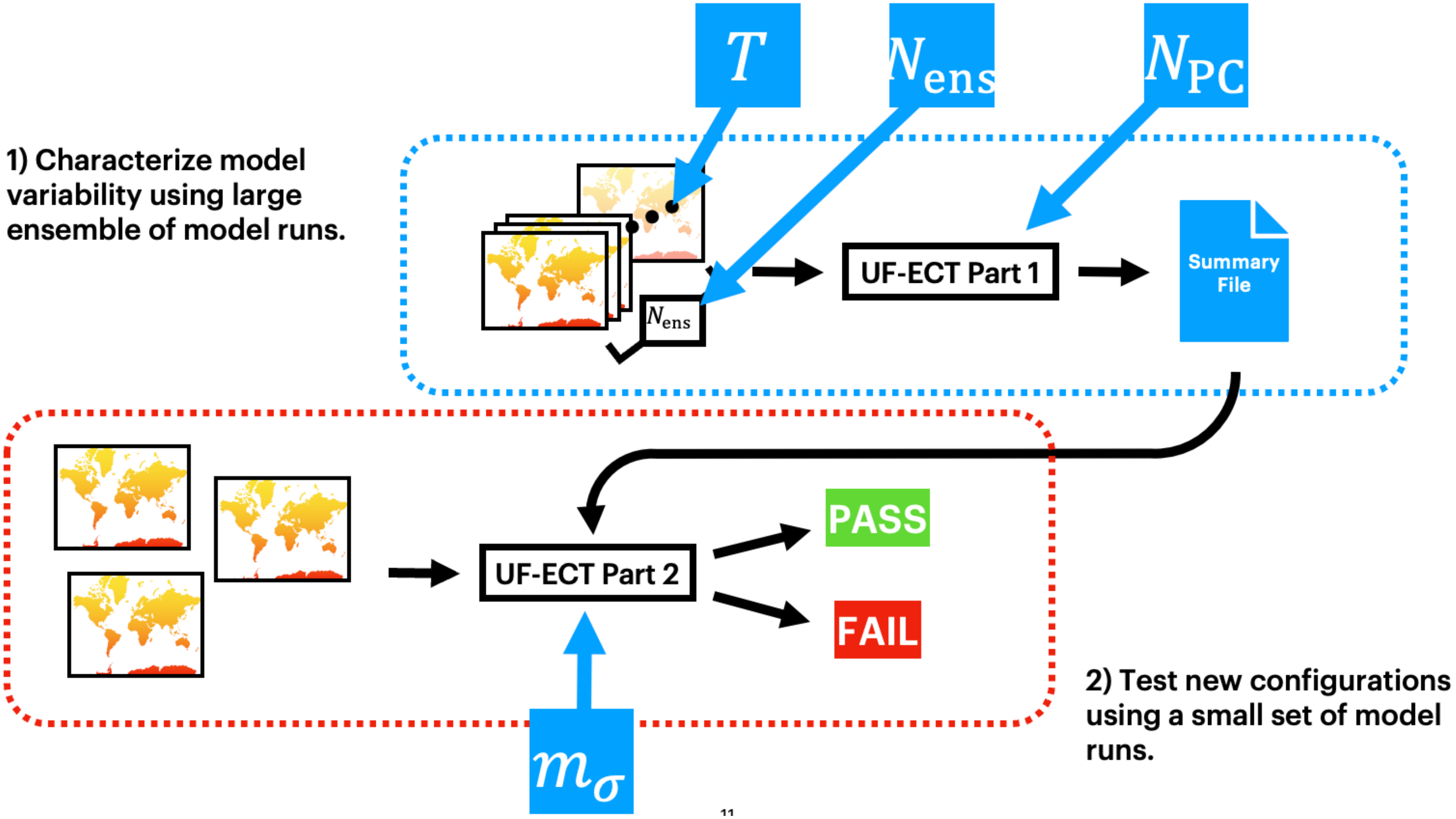
- How long to run model?
- How many PC dimensions to use?
- How large of an ensemble?
- How to set the failure cutoff?

Demonstrate our approach on:

- a new model: MPAS-A
- CAM 6.3 (originally designed for CAM 5.3)

How can we apply ECT to other models?

ECT Overview: identify correct test parameters for given model



1) Characterize model variability using large ensemble of model runs.

2) Test new configurations using a small set of model runs.

Overview of Approach

1. Initial: generate a sufficiently large ensemble to experiment with
2. Determine model run length: $\boldsymbol{T}$
   - *makes sure perturbations have propagated through the model.*
3. Determine which variables to exclude
4. Determine the number of PC dimensions : $\boldsymbol{N_{PC}}$
   - *to capture most of the variance of the model .*
5. Determine the acceptance region: $\boldsymbol{m_\sigma}$ and ensemble size: $\boldsymbol{N_{ens}}$
   - *to keep a low false positive rate*

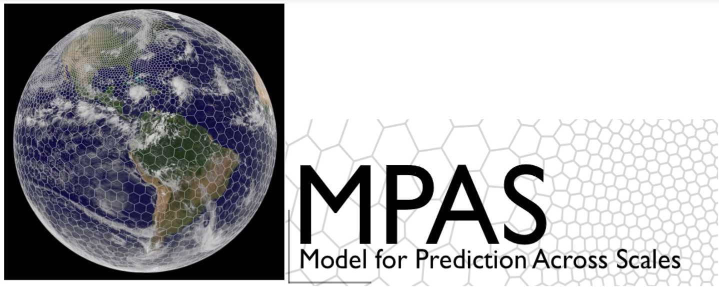
Overview of Approach

1. Initial: generate a sufficiently large ensemble to experiment with
2. Determine model run length: ***T***
   - *makes sure perturbations have propagated through the model.*
3. Determine which variables to exclude
4. Determine the number of PC dimensions : N_{PC}
   - *to capture most of the variance of the model .*
5. Determine the acceptance region: m_{σ} and ensemble size: N_{ens}
   - *to keep a low false positive rate*

	CAM 5.3	MPAS	CAM 6.3
# output vars (after exclusions)	108	43	275
T (timesteps)	9	26	7
N_{PC}	50	2.0	128
m_{σ}	2.0	2.0	2.23
N_{ens}	350	200	1650

A change in test parameters may be required when model significantly changes!

Example of types of testing (for MPAS)



Test Title	Test Description	Test Result (EET Failure Rate)
Compiler	Change from Intel’s Fortran Compiler to GNU	0.12%
Core Count	Change from 36 cores to 96 cores	0.2%
Compiler Optimizations	Change from Intel -03 compiler optimizations to -01	0.12%
Order of Operations	Change part of MPAS convection code to do a set of operations in a different, but mathematically equivalent, order.	1.67%
Precision	Change from double to single precision	100%
New Cluster	Run on default Derecho configuration (Intel compiler)	37.91%
New Cluster (No FMA)	Run on default Derecho configuration (Intel compiler) but without FMA.	0.15%

Concluding remarks

Ensemble consistency test approach (ECT):

- good option when bit-for-bit reproducibility is not possible
- objective, user-friendly
- works surprisingly well for climate models in practice!

Current/future work:

- finalizing MPAS-A test and generalization/automation for other models (new release soon!)
- finding root cause of a failure (hard!)

Using structured hypothesis testing with PCA is useful in the context of numerical models with many output variables.

Thanks!!!
abaker@ucar.edu

References…

A.H. Baker, D.M. Hammerling, M.N. Levy, H. Xu, J.M. Dennis, B.E. Eaton, J. Edwards, C. Hannay, S.A. Mickelson, R.B. Neale, D. Nychka, J. Shollenberger, J. Tribbia, M. Vertenstein, and D. Williamson, A new ensemble-based consistency test for the community earth system model (pyCECT v1.0). GMD, 2015.

A.H. Baker, Y. Hu, D.M. Hammerling, Y. Tseng, X. Hu, X. Huang, F.O. Bryan, and G. Yang, Evaluating Statistical Consistency in the Ocean Model Component of the Community Earth System Model (pyCECT v2.0). GMD, 2016.

D.J. Milroy, A.H. Baker, D.M. Hammerling, J.M. Dennis, S.A. Mickelson, and E.R. Jessup, Towards characterizing the variability of statistically consistent Community Earth System Model simulations. Procedia Computer Science (ICCS 2016), 2016

A.H. Baker, D.J. Milroy, D.M. Hammerling, and H. Xu. Quality assurance and error identification for the Community Earth System Model. Proceedings of Correctness ’17, 2017

D.J. Milroy, A.H. Baker, D.M. Hammerling, and E.R. Jessup, Nine time steps: ultra-fast statistical consistency testing of the Community Earth System Model (pyCECT v3.0), GMD, 2018.

D.J. Milroy, A.H. Baker, D.M. Hammerling, Y. Kim, T. Hauser, and E.R. Jessup, Making root cause analysis feasible for large code bases: a solution approach for a climate model, HPDC19, 2019

D. Ahn, A.H. Baker, M. Bentley, I. Briggs, G. Gopalakrishnan, D.M. Hammerling, I. Laguna, G.L. Lee, D.J. Milroy, M. Vertenstein, Keeping Science on Keel When Software Moves, Communications of the ACM, January 2021.

A PCA-based hypothesis testing framework for large ensembles. *In preparation.*