UNSW
Climate Change
Research Centre

Gaia: Global AI Accelerator:
Modeling MJO structures and tipping point analysis

Milestone 6
Preliminary software for the hybrid models/methods

Date of Report:

Initially submitted Jul 13, 2022

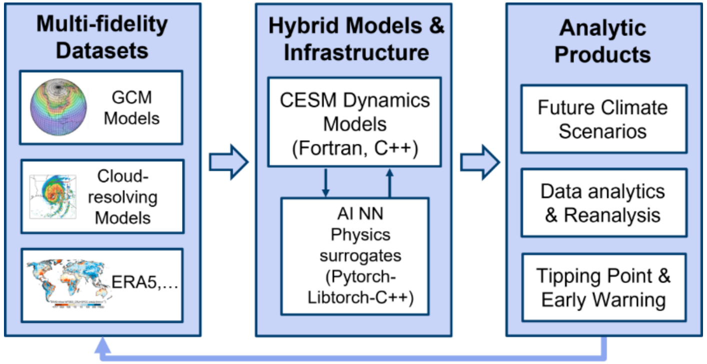
Updated: Aug 13, 2022

Gaia: Global AI Accelerator

UNSW Climate Change Research Centre

Goals:

- Improve speed & skill of atmospheric models using hybrid AI cloud physics surrogates:
  - Accurately model local convection
  - Predict self-organizing atmospheric phenomena, e.g. MJO
- Exploit hybrid models to explore future climate regimes and identify early tipping point signatures



Why is this Hard?

- Global climate models (GCMs) are computationally expensive and lack the resolution required to model local convection and clouds
- This restricts our ability to model convection processes including wave phenomena such as MJO, and limits forecasting skill and the ability to explore future climate regimes

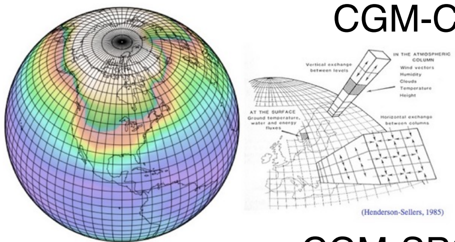
Key Gaia ideas and approach:

- Use AI surrogates to model cloud physics to substantially increase computational efficiency
- Use LES and reanalysis data to learn corrections to the AI surrogate models with the goal of improved MJO modeling while retaining computational efficiency
- Use improved hybrid models and reduced order models to explore tipping point regimes (such as predicted “superMJO”) and early warning signatures

Gaia Hybrid Model Building Approach

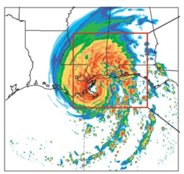
UNSW Climate Change Research Centre

Datasets for AI Training

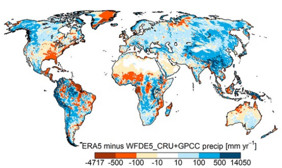


CGM-CAM4

CGM-SPCAM



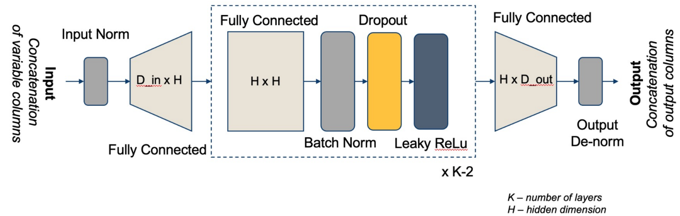
WRF-LES
Cloud-resolving Models



Reanalysis Datasets
(e.g. ERA5)

AI NN Surrogate Cloud Physics Model(s)

(e.g. FC, CNN, AE, Resnet)



AI CAM-trained
NN Surrogate Model(s)

AI NN
Corrective Model(s)

+

Hybrid Model(s)

GCM Dynamics
(Fortran, C++)

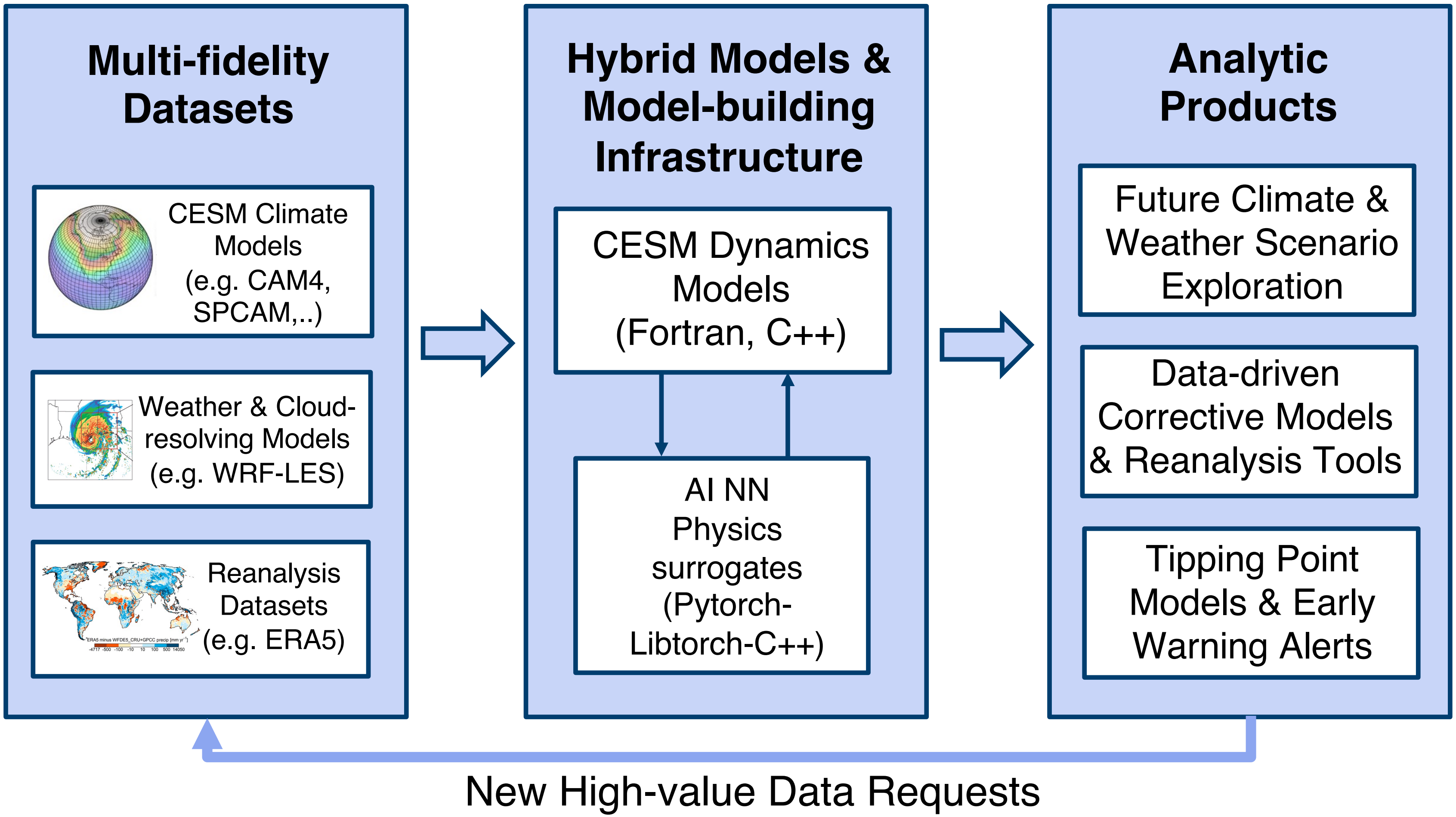
AI NN
Cloud Physics
(Pytorch-
Libtorch-C++)

Predicted
Trajectory

Gaia Building Blocks & Products

str

UNSW Climate Change Research Centre



M6 Update

str

UNSW Climate Change Research Centre

Description of datasets, data preprocessing, python modeling & analytics, and data products can be found at:

https://github.com/stresearch/gaia

Training of initial AI cloud physics model surrogates is currently based on 3 and 4-year runs from two NCAR community atmospheric models (CAM4 and SPCAM); these datasets have been reproduced with *additional variables** identified as useful for the hybrid model, and will later be extended to 10 years of simulation time:

GCM

- Community Atmospheric Model (CAM4)
- 30 minute time-step
- 2.5-degree grid (144x96)
- 30 altitude levels
- Four year run (1979 SST; Time Varying) which will be extended to ten years.
- Outputs every 3 hours + additional model time-step (memory)

CRM

- SPCAM (super parameterized CAM)
- 20 minute time-step
- 16 SAM (The System for Atmospheric Modeling) Columns
- 26 levels
- Year 2000 SST (Climatology)
- Three year simulations:
  - Morrison Microphysics + Conventional parameterization for moist convection and large-scale condensation.
  - Morrison Microphysics + Higher-order turbulence closure scheme, Cloud Layers Unified By Binormals (CLUBB)
- Outputs every 3 hours + additional model time-step (memory)

** See following slides*

Gaia AI inputs (state info passed from GCM)

UNSW Climate Change Research Centre

Surrogate Inputs

Name	Long Name	shape	unit
Q	Specific humidity	(T, L, 96, 144)	kg/kg
T	Temperature	(T, L, 96, 144)	K
U	Zonal wind	(T, L, 96, 144)	m/s
V	Meridional wind	(T, L, 96, 144)	m/s
OMEGA	Vertical velocity (pressure)	(T, L, 96, 144)	Pa/s
Z3	Geopotential Height (above sea level)	(T, L, 96, 144)	m
PS	Surface pressure	(T, 96, 144)	Pa
SOLIN	Solar insolation	(T, 96, 144)	W/m2
SHFLX	Surface sensible heat flux	(T, 96, 144)	W/m2
LHFLX	Surface latent heat flux	(T, 96, 144)	W/m2
FSNS	Net solar flux at surface	(T, 96, 144)	W/m2
FLNS	Net longwave flux at surface	(T, 96, 144)	W/m2
FSNT	Net solar flux at top of model	(T, 96, 144)	W/m2
FLNT	Net longwave flux at top of model	(T, 96, 144)	W/m2
FSDS	Downwelling solar flux at surface	(T, 96, 144)	W/m2

** Note that at this stage we are not adding 4D radiative inputs (e.g. SOLL)*

Gaia AI outputs (state info passed to GCM)

UNSW Climate Change Research Centre

Surrogate Outputs

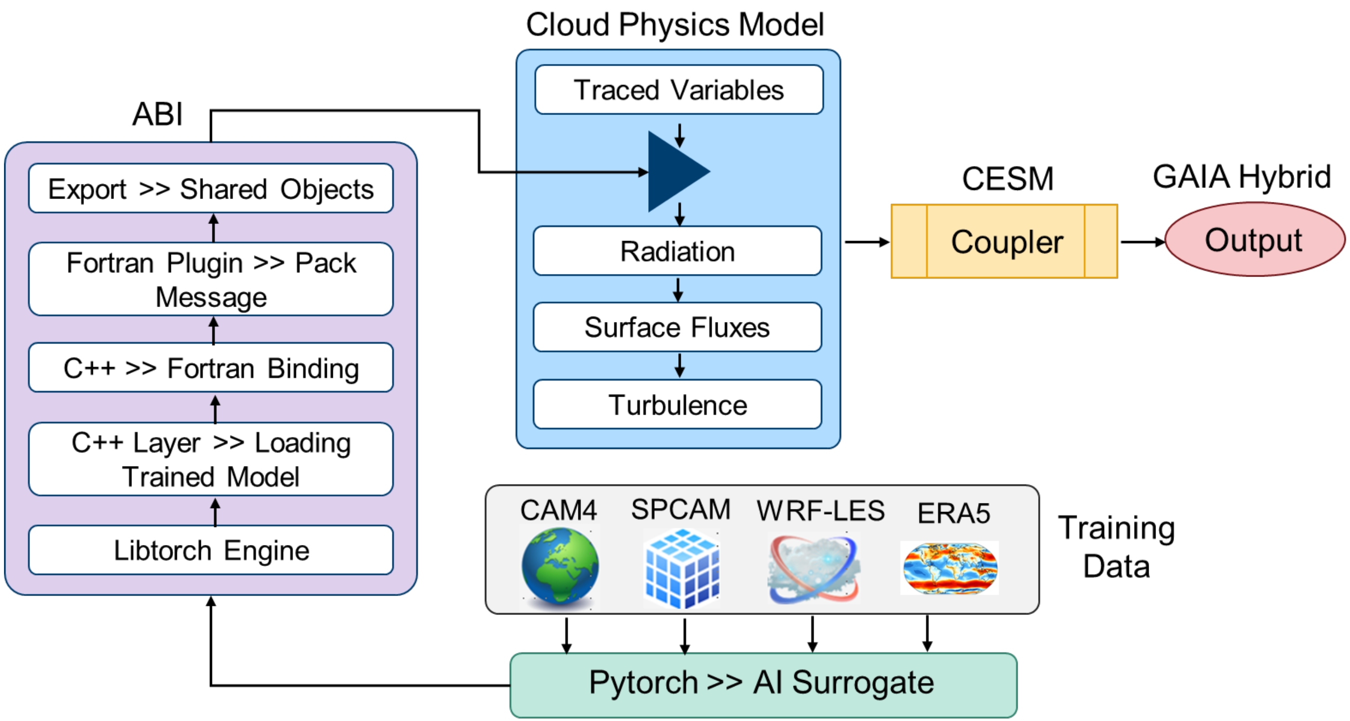
Name	Long Name	shape	unit
PTEQ	Q total physics tendency	(T, L, 96, 144)	kg/kg/s
PTTEND	T total physics tendency	(T, L, 96, 144)	k/s
DCQ	Q total tendency moist processes	(T, L, 96, 144)	kg/kg/s
DTCOND	T total tendency moist processes	(T, L, 96, 144)	k/s
QRS	Shortwave heating rate	(T, L, 96, 144)	k/s
QRL	Longwave heating rate	(T, L, 96, 144)	k/s
CLOUD	Total cloud cover	(T, L, 96, 144)	fraction
CONCLD	Convective cloud cover	(T, L, 96, 144)	fraction
FSNS	Net solar flux at surface	(T, 96, 144)	W/m2
FLNS	Net longwave flux at surface	(T, 96, 144)	W/m2
FSNT	Net solar flux at top of model	(T, 96, 144)	W/m2
FLNT	Net longwave flux at top of model	(T, 96, 144)	W/m2
FSDS	Downwelling solar flux at surface	(T, 96, 144)	W/m2
FLDS	Downwelling longwave flux at surface	(T, 96, 144)	W/m2
SRFRAD	Net radiative flux at surface	(T, 96, 144)	W/m2
SOLL	Solar downward near infrared direct to surface	(T, 96, 144)	W/m2
SOLS	Solar downward visible direct to surface	(T, 96, 144)	W/m2
SOLLD	Solar downward near infrared diffuse to surface	(T, 96, 144)	W/m2
SOLSD	Solar downward visible diffuse to surface	(T, 96, 144)	W/m2
PSL	Sea level pressure	(T, 96, 144)	W/m2
PRECT	Total precipitation rate (liquid+ice)	(T, 96, 144)	m/s
PRECC	Convective precipitation rate (liquid+ice)	(T, 96, 144)	m/s
PRECL	Large-scale precipitation rate (liquid+ice)	(T, 96, 144)	m/s
PRECSC	Convective snow rate	(T, 96, 144)	m/s
PRECSL	Large-scale snow rate	(T, 96, 144)	m/s

Hybrid model integration

- Completed integrating the pytorch-based AI surrogate back into the Fortran-based GCM models
- Both the traditional parameterized physics model and the AI/ML surrogate physics model integrations can run side by side in the same run for comparison
- Currently debugging binding problem in the C++-Fortran binding code

Application Binary Interface (ABI)

- Start with total T,Q physics tendencies and track them back
- Export Pytorch Model with Torchscript
- Bypass Python by using C++
- Call C++ within Fortran



Code components

UNSW Climate Change Research Centre

- initindx.F90
- machine_learning_model.F90
- machine_learning_model_config.F90
- ml_solin.F90
- ml_srfxfer.F90
- physpkg.F90
- tphysac.F90
- tphysac_param.F90
- tphysbc.F90
- tphysbc_ml.F90
- tphysbc_param.F90

- CMakeLists.txt
- torch-plugin.f90
- torch-wrap-cdef.f90
- torch-wrap.cpp

- *.config.F90 file provides configurations to switch on/off different physical processes (e.g. radiation)
- *solin.F90 takes out solar insolation calculations out of radiation (makes easy to bypass complex radiative calculations)
- tphysac* and tphysbc* provide the gateway for the ABI to function
- Machine_learning_model.F90 reads in data from the ABI

- ABI Wrappers
  - Torch-wrap* contains wrappers and definitions for the Libtorch and C++ binding to function
  - *plugin loads the Pytorch model into the CESM code base

Hybrid model characteristics

str

UNSW Climate Change Research Centre

Architected as a set of modular building blocks for easy adoption and easy extension by the community

Flexibility

- Switches enabled to run hybrid & parameterized physics in a single time step.

Generalizability

- Extendable to replace other physics parameterizations such as radiation, boundary layer and surface schemes due to modular code structure.

Modularity & Reliability

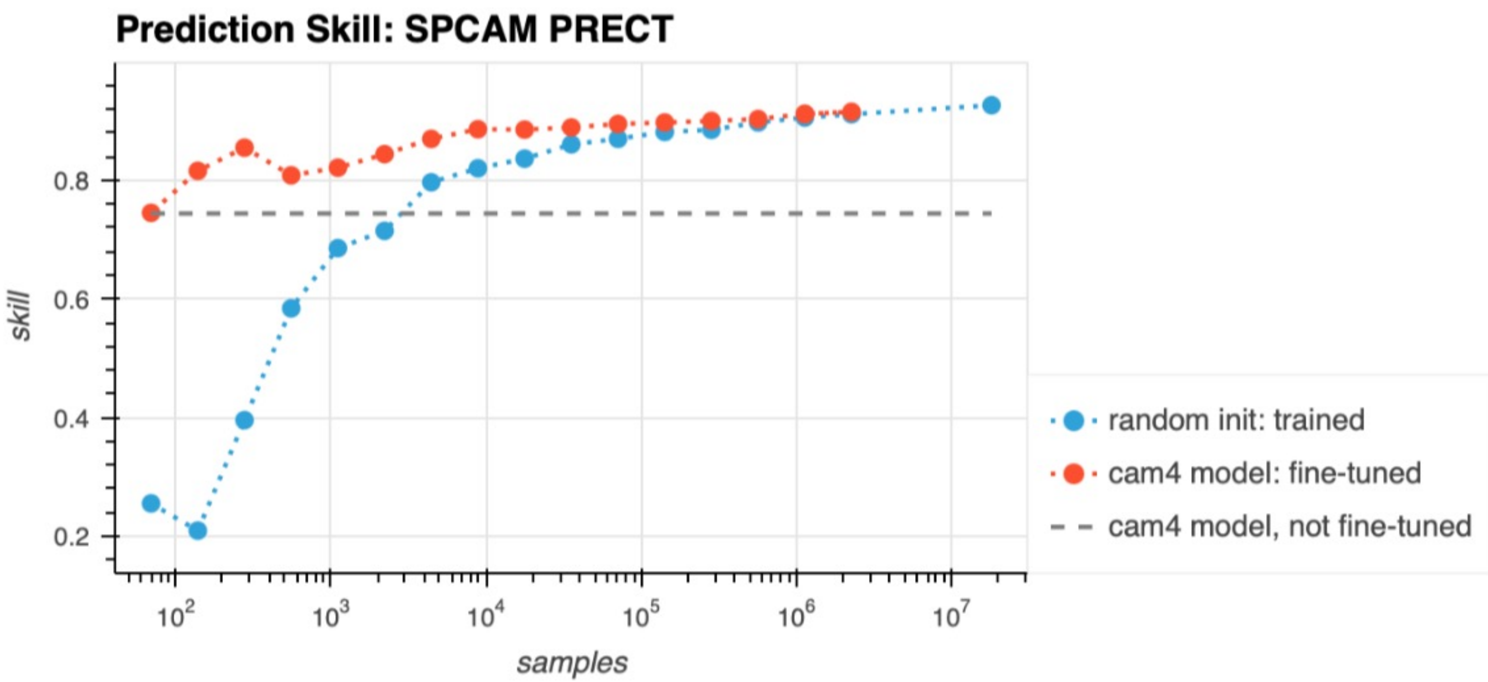
- Implementation fits nicely with CESM and can be joined with the CESM source tree efficiently.

Performance

- Faster to run by > 3X than initial parametrized runs depending on the complexity of the ML trained model.

Exploring fine-tuning a model for a subset of output variables

- Fine-tune CAM4-trained model to predict SPCAM total precipitation
- Compare to SPCAM-trained model trained from scratch
- SPCAM testset used, looking at total precipitation output variable only
- Prediction skill reaches ~92%, (CAM4-trained model on CAM4 testset reaches 96% skill)



Gaia Current Status (Aug 13, 2022)

str

UNSW
Climate Change
Research Centre

Completed

- Initial CAM4 and SPCAM datasets generated (4 years, 30/20 min timestep)
- Developed & optimized several AI surrogates for both CAM4 and SPCAM
  - Validated AI skill for several architectures (FC, CNN, bottlenecked FC, transformer)
  - Dimensional analysis to quantify model complexity; also assessed impact of memory terms
  - Jacobian analyses as potential new skill metric and means of regularization
- Hybrid model machinery developed
  - Integrates pytorch AI surrogates into NCAR's Fortran GCM models
  - ~ 3X speedups
  - New CAM4 and SPCCAM datasets generated to accommodate additional variables

Ongoing

- Run hybrid models and evaluate for stability, skill, and MJO modeling

Next Steps

- Stability enforcing methods (if needed)
- AI retraining and augmentation (using WRF/LES model data and ERA5 reanalysis data)
- Explore forcings (e.g. high SST state regimes)
- Reduced order tipping point model development

Hybrid model stability

str

UNSW Climate Change Research Centre

Stability of deep neural net surrogates coupled into global atmospheric climate models has been an issue reported in the literature, e.g.:

- N. D. Brenowitz, T. Beucler, M. Pritchard, C. S. Bretherton, "Interpreting and Stabilizing Machine Learning Parameterizations of Convection", J. Atmos. Sci., 2020
- X. Wang, Y. Han, W. Xue, G. Yang, G. J. Zhang, "Stable climate simulations using a realistic general circulation model with neural network parameterizations for atmospheric moist physics and radiation processes", GeoSci. Model. Dev., 2022

- Problems manifest as rapid blowup in time of the total energy of the system; this can happen even for high-skill DNN surrogates
- Problem is not fully understood
- Solutions are mostly heuristic, and include:
  - Alternative network architectures, e.g. ResNets
  - Input variable ablation
  - Denoising architectures
  - Optimizing for wave propagation response in a linearized model
- Problem raises the concern that retrained AI's (e.g. using LES or ERA datasets) might also exhibit stability issues when integrated back into GCM

Approaches to hybrid model stability

We are just beginning to test the hybrid models; first steps will be to:

- Quantify hybrid model stability
- Quantify hybrid model skill on mean properties
- Quantify hybrid model skill in reproducing MJO-type structures
- Compare SPCAM and CAM4 hybrid models

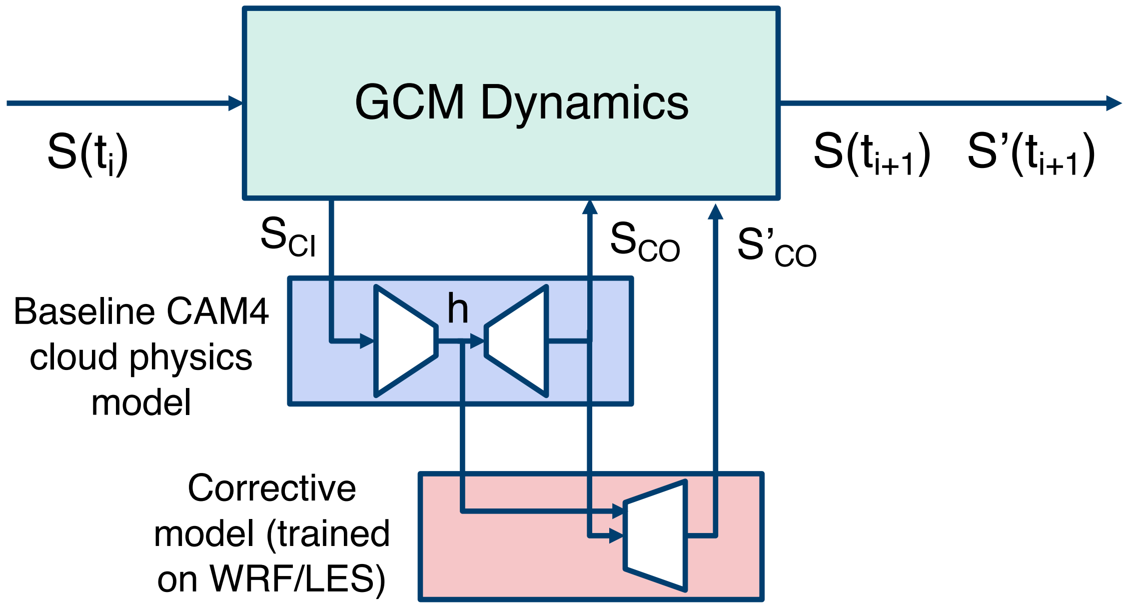
We are considering several approaches to address potential stability problems of the coupled AI-GCM hybrid models:

- First assess if problem is triggered by amplified skill errors, out-of-training behavior, non-causal AI behaviors, or other
- Assess local Jacobian of DNN input-to-output map in trouble areas, identify common features and potential spectral regularization methods
- Assess if problems are agnostic to AI architecture and/or AI model skill
- Assess impact of added memory terms or other approaches to encouraging causality
- Assess dimensional reduction regularization methods to improve generalizability
- Assess if AI surrogates conditioned on latitude and/or season perform more reliably
- Assess impact of data augmentation near trouble zones
- Assess impact of alternative normalizations

Another idea for hybrid model(s) training

UNSW Climate Change Research Centre

Consider a 2-step process: (1) train CAM physics model surrogate as before; (2) train corrections to this baseline model based on performance of hybrid model



Training of corrective model based on cost function looking at various skill metrics, stability, and S' residuals

Another approach to gray-box corrections

- Constrain the NN to possess the desired coarse bifurcation diagram
- The coarse steady states as well as their coarse stabilities
- Prescribe this as part of the loss ---
- OPTIMIZING NNs with algebraic constraints (KKT optimization)

- Separately LIFTING in multiscale problems:
- construct full model IC consistent with given coarse features:
- Initializing on slow manifolds / Umbrella Sampling / GANs
- Coarse: e.g. Majda skeletal Madden-Julian States; Fine: current models

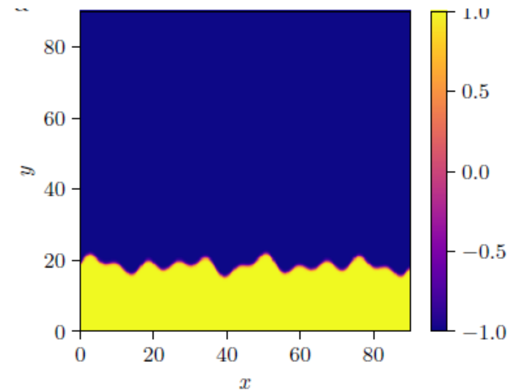
CFD Example: data-driven corrective models from simulation data

Goal: Hardwire partially known physics in the structure of the ML algorithm (e.g. ANN) that learns the PDE from data

(a) as an additive correction to a known, approximate equation

(b) as a functional correction

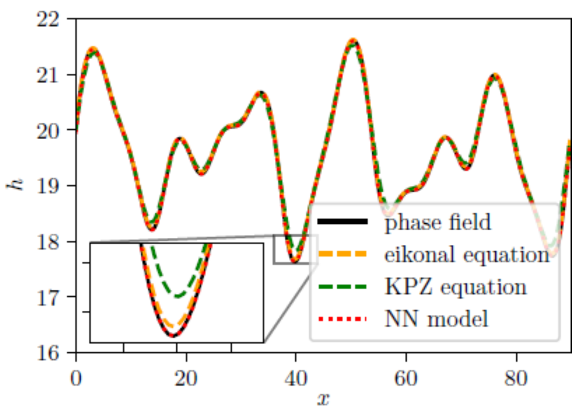
$$\frac{\partial \phi}{\partial t} = D\nabla^2\phi - (\phi - a)\left(\phi^2 - 1\right)$$



An illustrative example: a phase field equation, with 2D vector field ϕ and 1D boundary h for which two levels of approximate interface equations can be derived:

Eikonal $$\frac{\partial h}{\partial t} = \frac{D}{1+\left(\frac{\partial h}{\partial x}\right)^2}\frac{\partial^2 h}{\partial x^2} - \sqrt{2D}a\sqrt{1+\frac{1}{2}\left(\frac{\partial h}{\partial x}\right)^2}$$ and KPZ $$\frac{\partial \tilde{h}}{\partial t} = D\frac{\partial^2 \tilde{h}}{\partial x^2} - a\sqrt{\frac{D}{2}}\left(\frac{\partial \tilde{h}}{\partial x}\right)^2$$

Black box NN model performance:



Additive Correction to KPZ

$$\begin{aligned}\frac{\widehat{\partial h}}{\partial t} &= f_{KPZ}\left(h, \partial h/\partial x, \partial^2 h/\partial x^2\right)\\ &+ NN_{\Theta}\left(h, \partial h/\partial x, \partial^2 h/\partial x^2\right)\\ &= f_{add}\left(h, \partial h/\partial x, \partial^2 h/\partial x^2\right).\end{aligned}$$

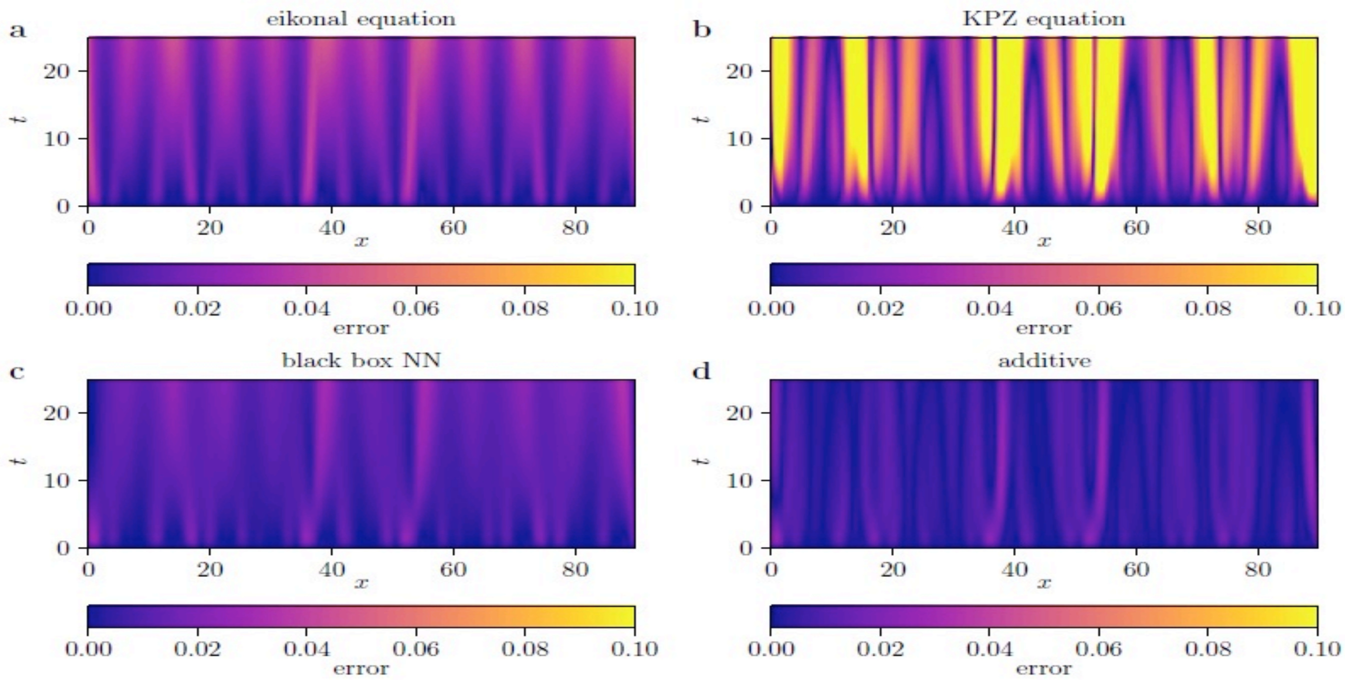
Functional Correction to KPZ

$$\begin{aligned}\frac{\widehat{\partial h}}{\partial t} &= NN_{\Theta}\left(f_{KPZ}, \partial f_{KPZ}/\partial x, \partial^2 f_{KPZ}/\partial x^2\right)\\ &= f_{fun}\left(f_{KPZ}, \partial f_{KPZ}/\partial x, \partial^2 f_{KPZ}/\partial x^2\right).\end{aligned}$$

Comparison of space-time errors for Eikonal, KPZ, black box NN, and corrective NN

Note: while black box NN provides a better mapping than either Eikonal or KPZ approximations, learning a NN correction to the KPZ approximation does better yet!

We plan to exploit this corrective gray box approach to improve our AI surrogate model performance

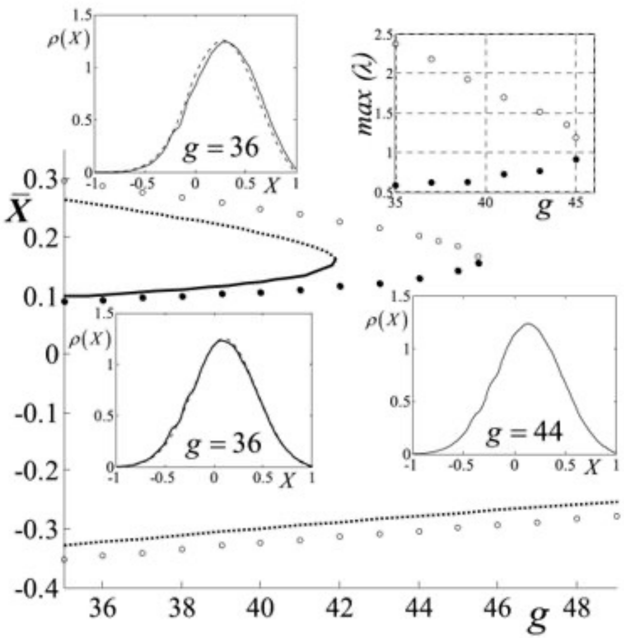


On tipping point analysis with reduced-dimension models

Goal: build targeted bifurcation surrogates close to tipping points

Example: complex economic stochastic agent-based model with 50,000 agents

Type of tipping point: fold (turning point)



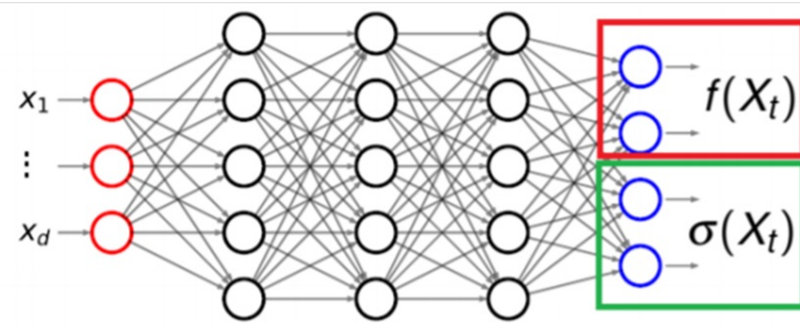
Agent distribution mean depends on parameter g, with tipping point at g ~ 45

To *learn* an SDE

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t$$

We need to approximate the drift and diffusivity.

Using a Deep Neural Network.

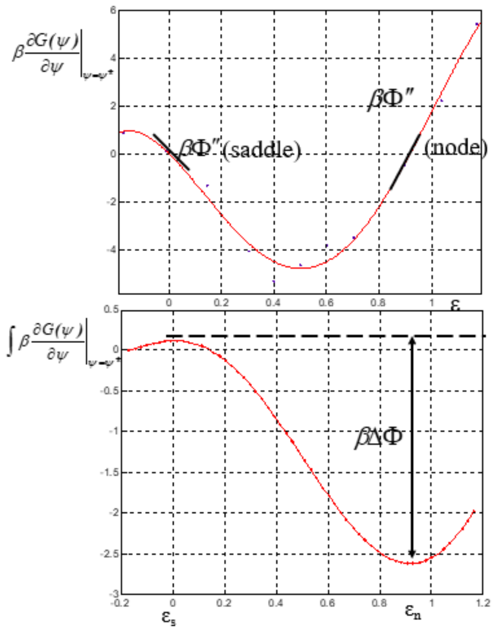


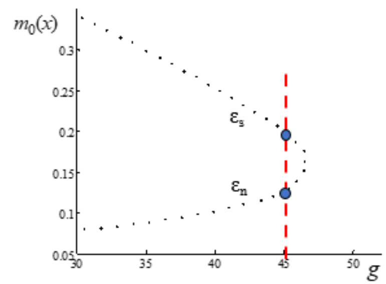
Use DNN to learn stochastic differential equation representation of behavior close to tipping point

Tipping point modeling, cont.

Kramer’s theory for modeling Brownian escape times

System properties are obtained from short-scale nonequilibrium simulations (at g = 45.2 in this example) and the learned stochastic differential equation





$$\frac{\partial \overline{\psi(t;\psi_0)}}{\partial t} \approx -D\beta \frac{\partial G(\psi)}{\partial \psi}\Big|_{\psi=\psi^*}$$

$$\frac{d\, var[\psi(t;\psi_0)]}{dt} \approx 2D(\psi^*)$$

mean time to escape

$$\tau \approx \frac{2\pi}{\overline{D}\sqrt{\beta\Phi''(N_{min})|\beta\Phi''(N_{saddle})|}} e^{\beta\Delta\Phi}$$

Key idea: for many complex systems, the tipping point dynamics becomes low (even 1D) dimensional close to a tipping point; we will attempt to learn a reduced dimensional tipping point model from our hybrid model at strong forcing (e.g., for elevated sea surface temperatures)

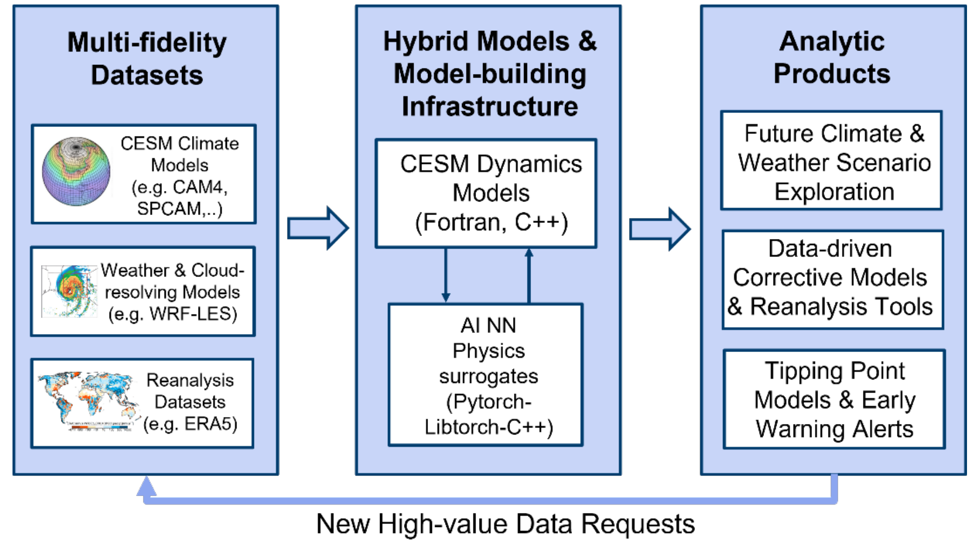
Summary

Completed

- Generated several training datasets
- Trained and validated an ensemble of AI surrogates
- Completed large engineering task of hybrid model infrastructure build

Next Phase I Steps

- Evaluate hybrid models
- Test new approaches to guarantee model stability and skill
- Test new approaches to build in model corrections using high fidelity model and observational data
- Develop reduced dimensional models, regularization on skeletal models, tipping point models



Phase 2 Analytic Products

- Fast “what if” trajectory analysis under forcing conditions
- Early warning tipping point signatures
- Data-driven model corrections
- Quantify value of new data