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

Contractor's Name and Address:
Systems & Technology Research
600 West Cummings Park
Woburn MA 01801

UNSW
Climate Change
Research Centre

Milestone 6:
Preliminary software for the hybrid models/methods

Date of Report:

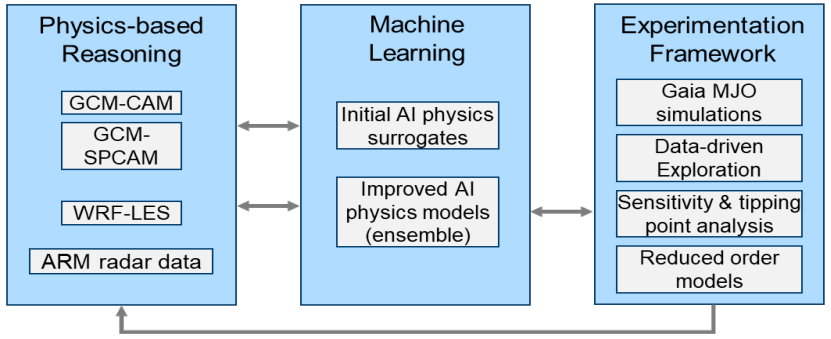
July 14, 2022

Gaia: Global AI Accelerator

UNSW Climate Change Research Centre

What's hard?

- GCMs are computationally expensive and lack the resolution needed to adequately model local convection and thus clouds
- This greatly increases GCM forecast errors and impedes propagation of large-scale wave phenomena such as the MJO



What will GAIA accomplish?

- Enable a GCM to accurately model local convection and predict self-organizing atmospheric wave phenomena (i.e. improved fidelity with 3X – 10X speedup)
- Exploit this GCM to explore possible future regimes and identify early warning signatures for MJO-related tipping points.

Progress

- Exported Pytorch AI CAM4 surrogate model into libtorch for integration with the Fortran-based GCM; completed the hybrid GCM-CAM model integration
- Began looking at ways to build in hybrid model stability, as this has been reported in literature to be a problem
- ***Next Steps***: Quantify hybrid model stability and skill and begin measures to improve the model using additional LES and ERA5 training data

M6 Update

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 are being 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 datasets

UNSW Climate Change Research Centre

Surrogate Inputs

With added variables

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 datasets

Surrogate outputs with added variables needed for hybrid model

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
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

UNSW Climate Change Research Centre

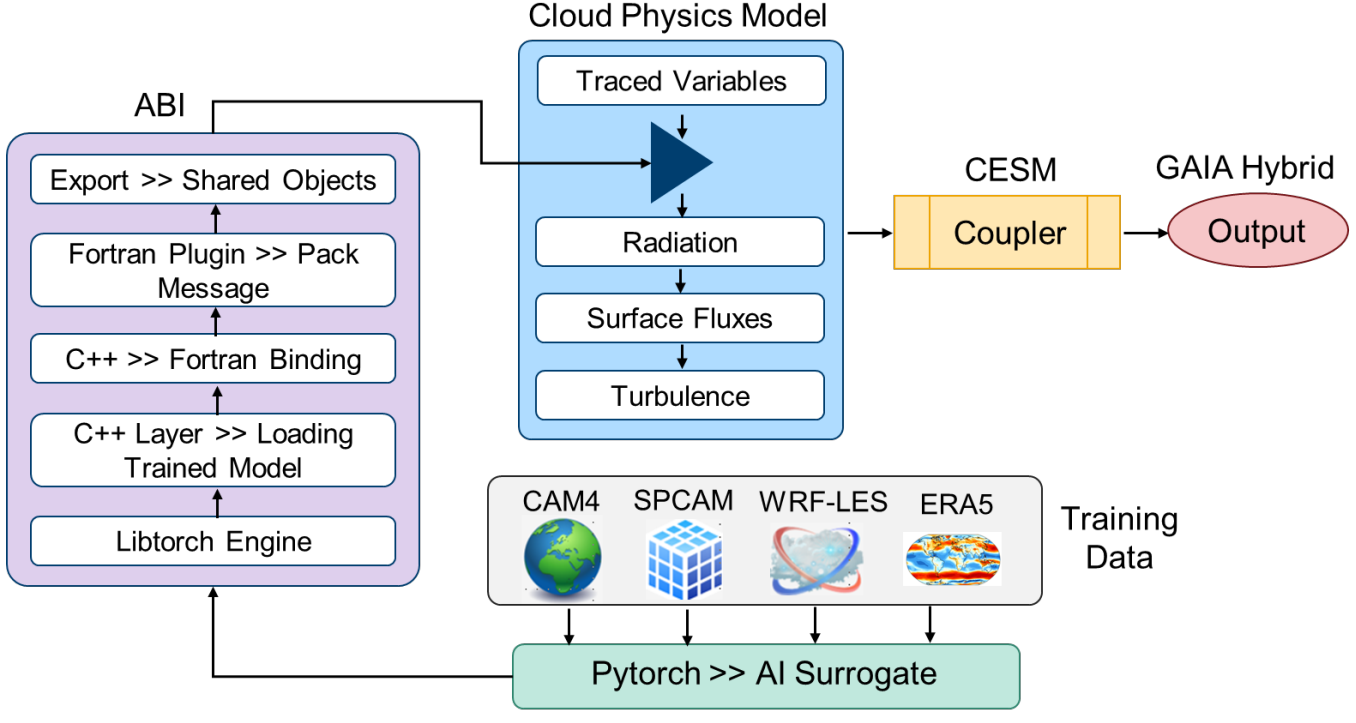
Hybrid model integration

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

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



Hybrid model characteristics

Architected for easy adoption by the community

Flexible

- Switches implemented to run hybrid and parameterized physics in parallel in a single time step

Generalizable

- Extendable to replace other physics parameterizations such as radiation, boundary layer and surface schemes

Modular & reliable

- 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; additional speedups may be possible

On hybrid model stability

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

Next Steps

UNSW Climate Change Research Centre

We are just beginning to test the hybrid models and should have results in a few days; key next steps:

- 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 looking at several approaches to address potential stability problems of the coupled DNN-GCM models; first, we will run CAM/SPCAM-GCM and DNN-GCM hybrid models side by side and track regions where energy blowup and model divergence occur:

- Assess if problem is triggered by DNN skill errors that become amplified, out-of-training behavior of the DNN, or other
- Assess local Jacobian of DNN input-to-output map in trouble areas, identify common features and potential spectral regularization methods
- If problem is one of insufficient DNN model skill, investigate an ensemble hybrid model conditioned on latitudinal zone, season, or both
- Assess impact of added memory terms
- Assess dimensional reduction regularization methods to improve generalizability
- Assess impact of data augmentation near trouble zones
- Look at larger local patches and add energy and mass conservation constraints within relevant time window as added DNN regularization

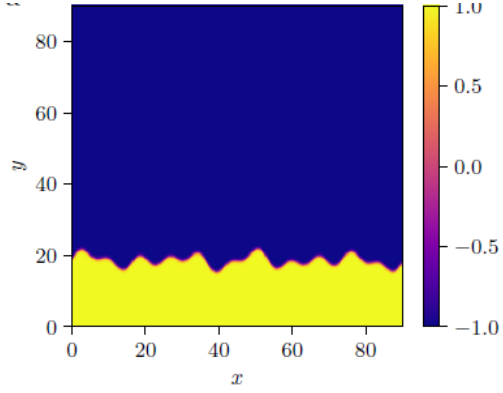
On data-driven corrective models

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

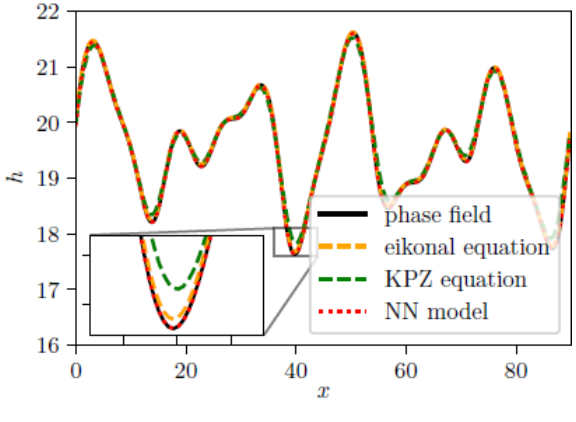
$$\frac{\partial \phi}{\partial t} = D\nabla^2\phi - (\phi - a)\left(\phi^2 - 1\right)$$



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

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

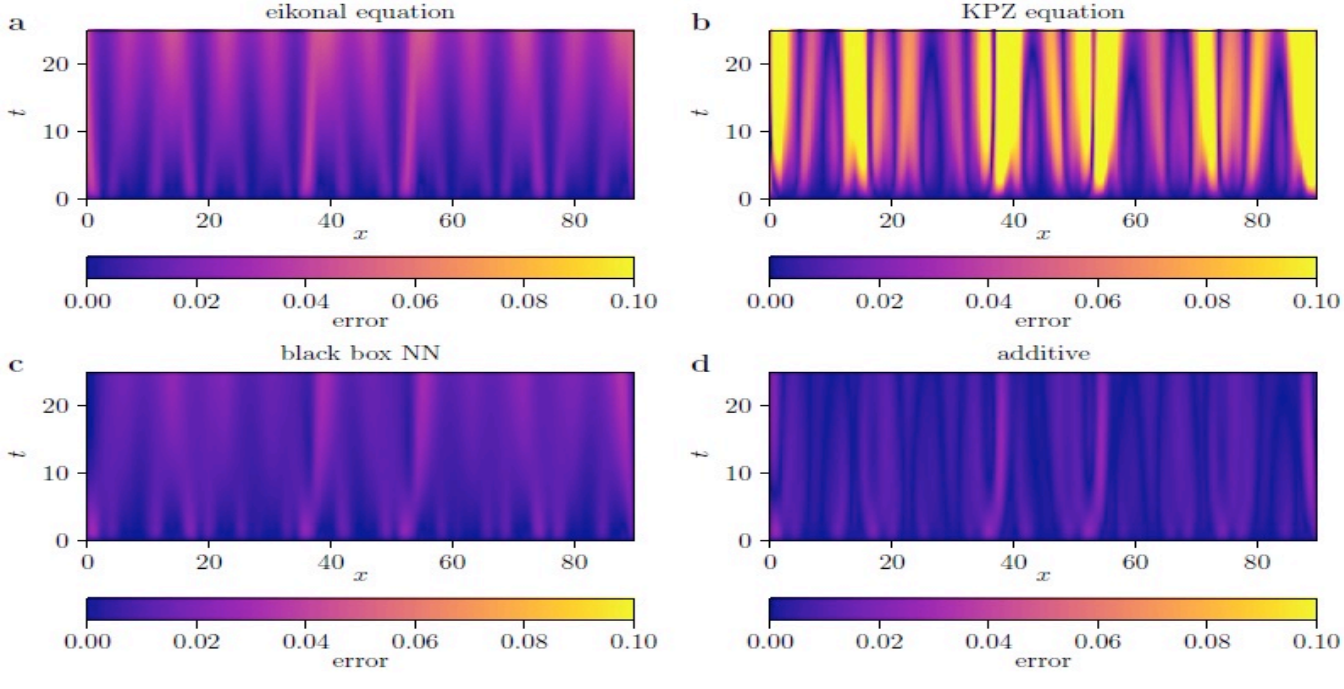
Functional Correction to KPZ

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

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

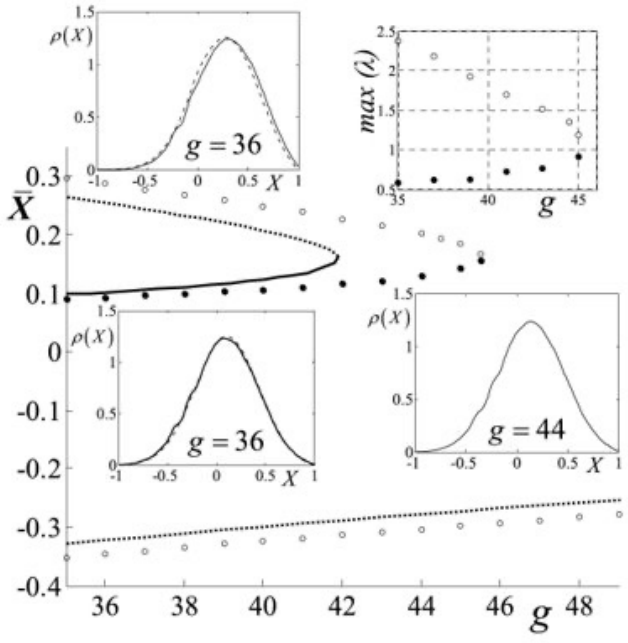


On tipping point analysis with reduced-dimension models

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)



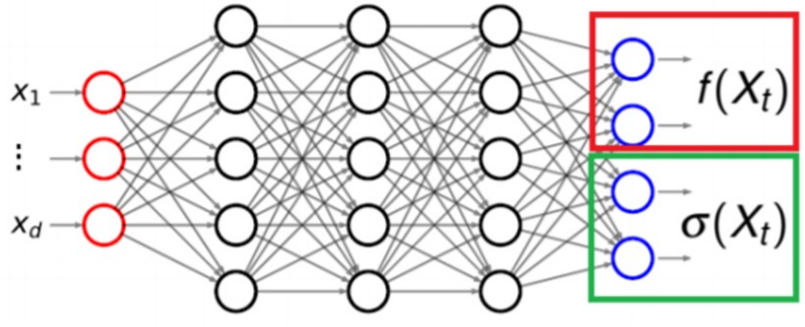
Agent distribution mean depends on parameter g, with tipping point at g ~ 45

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.

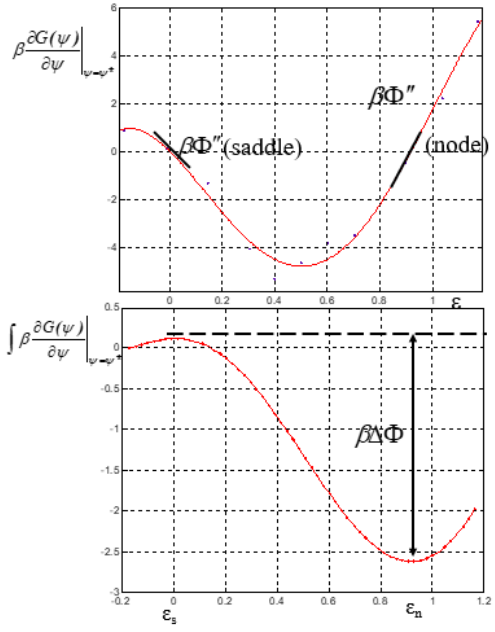


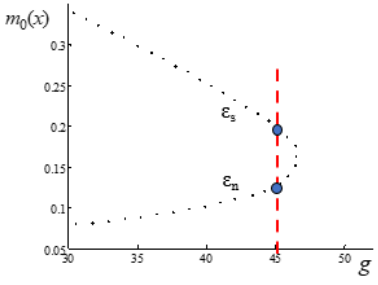
Use DNN to learn stochastic differential equation representation of behavior close to tipping point

Tipping point modeling, cont.

Kramers 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

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

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)