D Watson-Parris | October 1st 2026

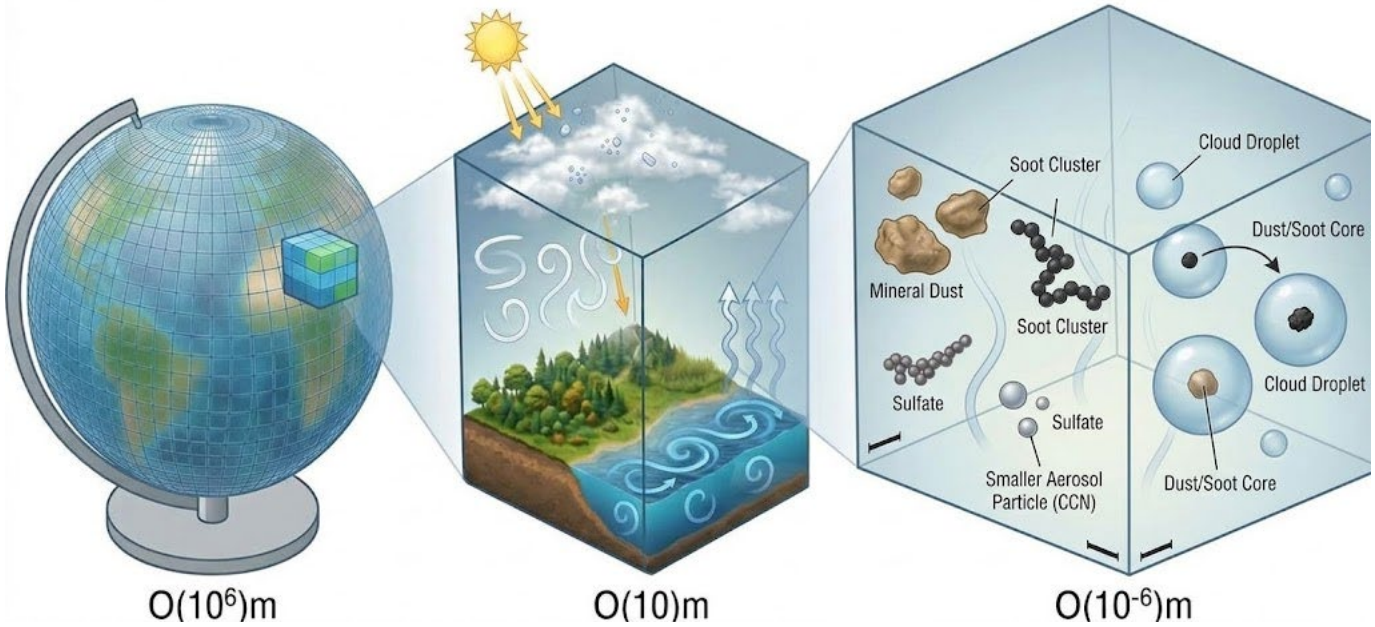


Gradients, benchmarks, and hybrids: a new toolkit for an old uncertainty

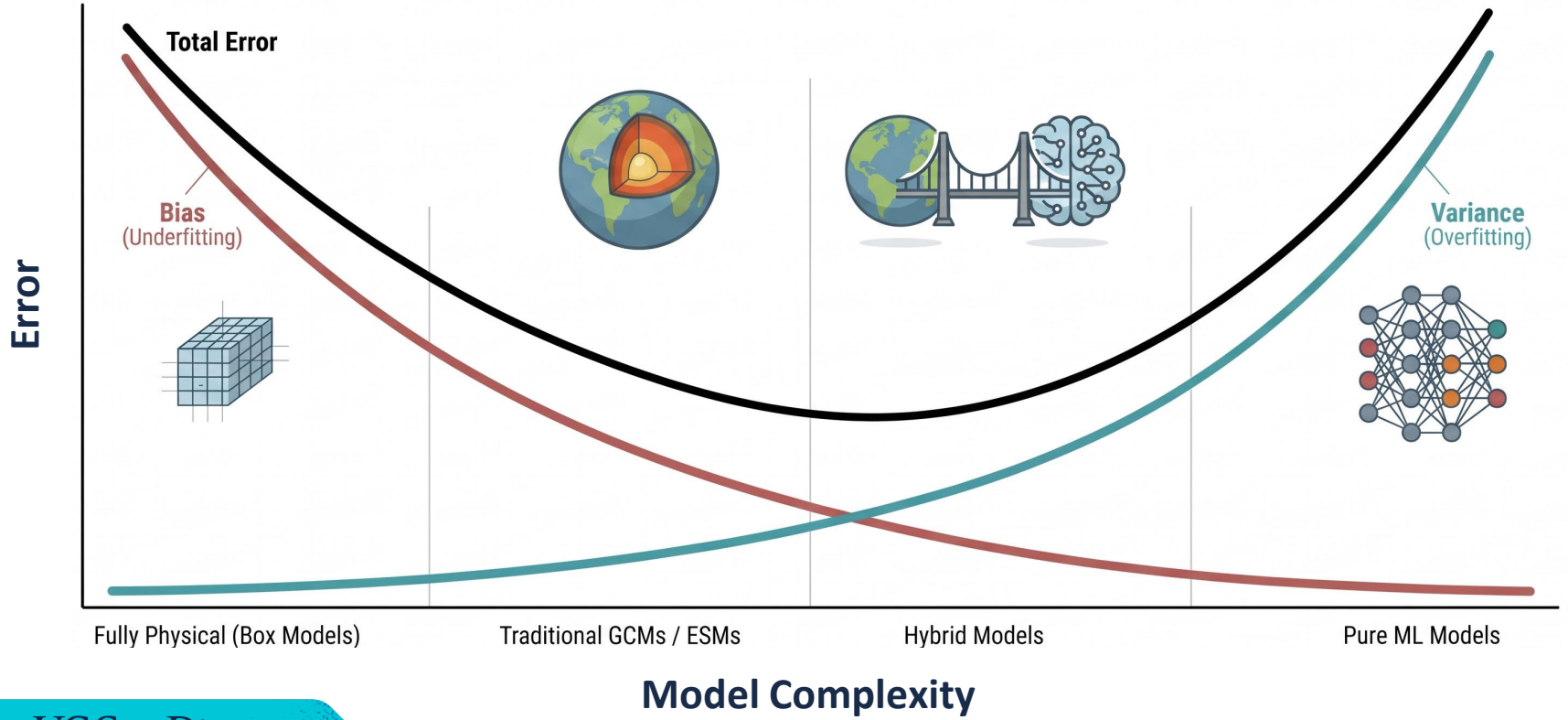
Unpacking AI Weather Models, IMSI, Chicago

UC San Diego

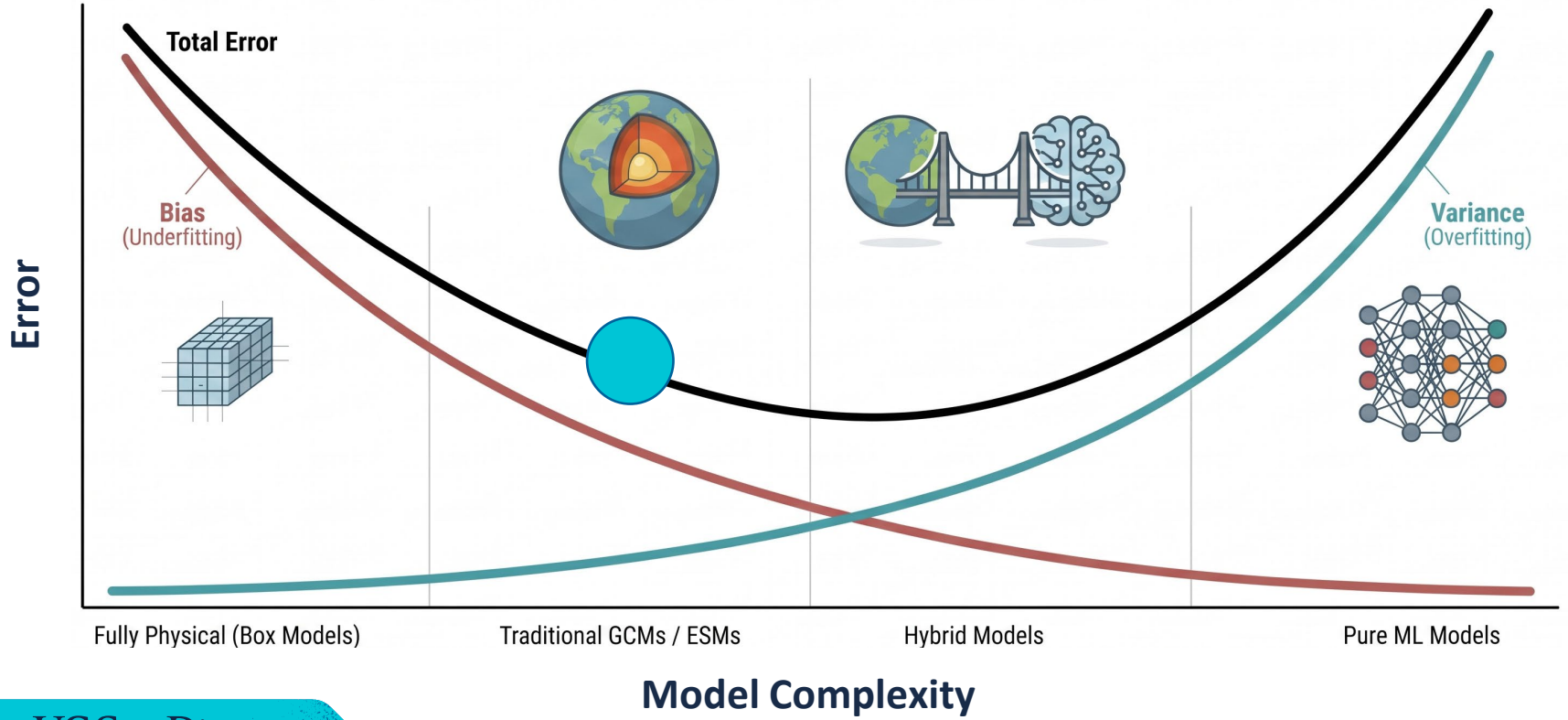
Atmospheric aerosol are a truly multi-scale physical problem



Finding the sweet spot



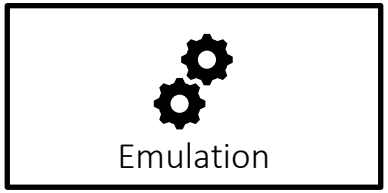
Finding the sweet spot



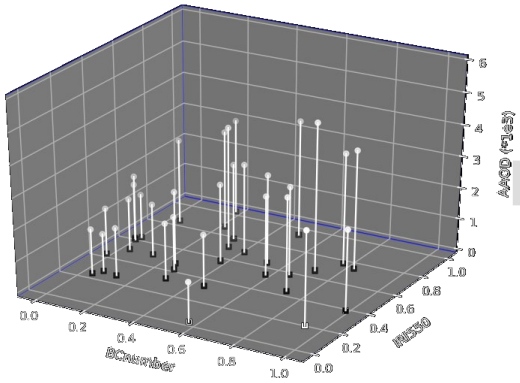
Climate model emulation for calibration

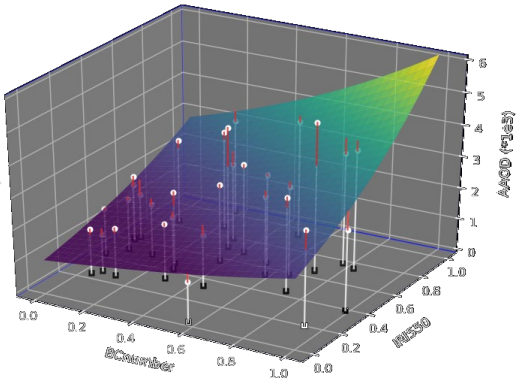
Emulator augmented PPEs are currently the best approach for calibration

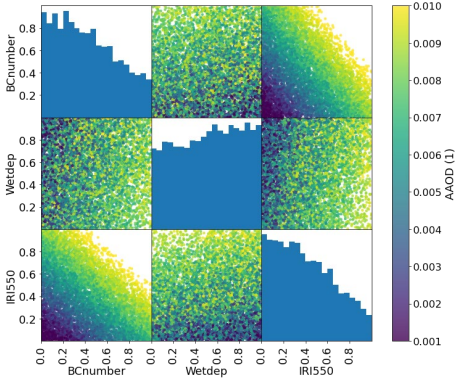












…but they are expensive and don't scale to coupled models

UC San Diego

Introducing JAX-GCM (JCM)

- **Fully differentiable physical** atmospheric Python model with gradients from JAX for calibration and sensitivity analysis out-of-the-box

```python
from jcm.model import Model

# Initialize a default (aquaplanet) model
model = Model()
predictions = model.run(total_time=30)
model.predictions_to_xarray(predictions)
```

Davenport, Maddan et al. *(GMD 2026)*

Introducing JAX-GCM (JCM)

- **Fully differentiable physical** atmospheric Python model with gradients from JAX for calibration and sensitivity analysis out-of-the-box
- **Plug-and-play** parameterizations enable flexible experimentation and online learning

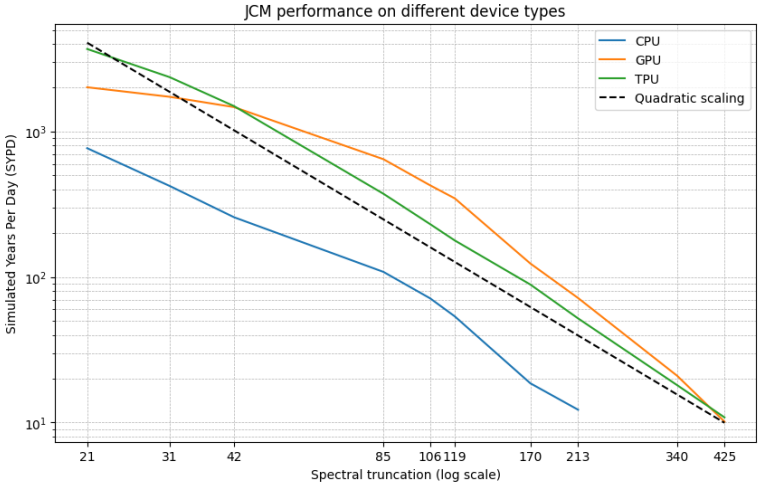
```
from jcm.model import Model
from jcm.physics.icon import IconPhysics

# Initialize a default (aquaplanet) model
model = Model(physics=IconPhysics())
predictions = model.run(total_time=30)
predictions.to_xarray()
```

Davenport, Maddan et al. *(GMD 2026)*

Introducing JAX-GCM (JCM)

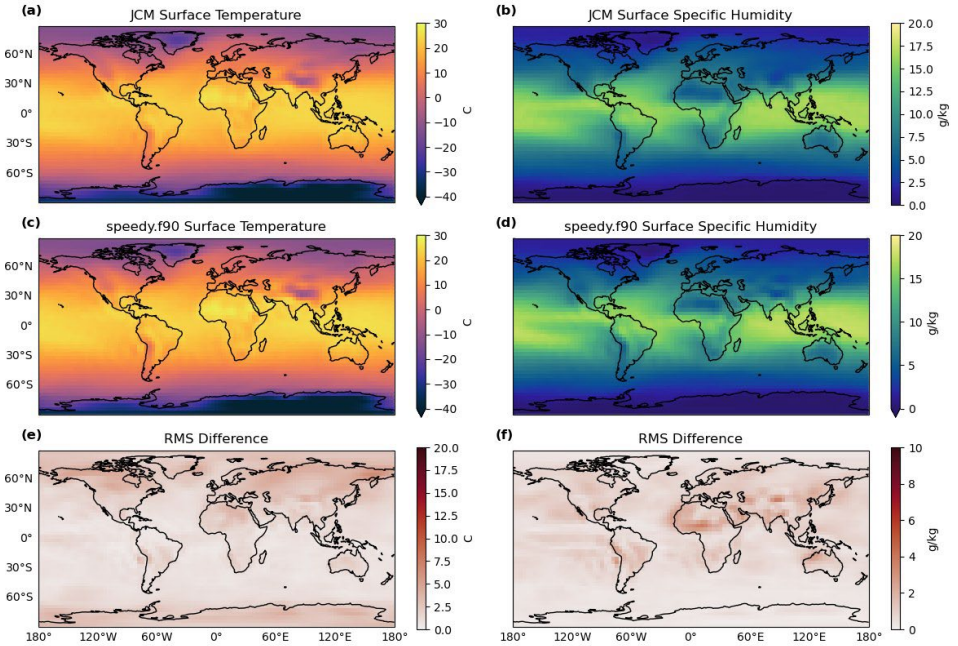
- **Fully differentiable physical** atmospheric Python model with gradients from JAX for calibration and sensitivity analysis out-of-the-box
- **Plug-and-play** parameterizations enable flexible experimentation and online learning
- **Compatible** with CPUs, GPUs, and TPUs, with automatic parallelization for scalability



Davenport, Maddan et al. *(GMD 2026)*

Introducing JAX-GCM (JCM)

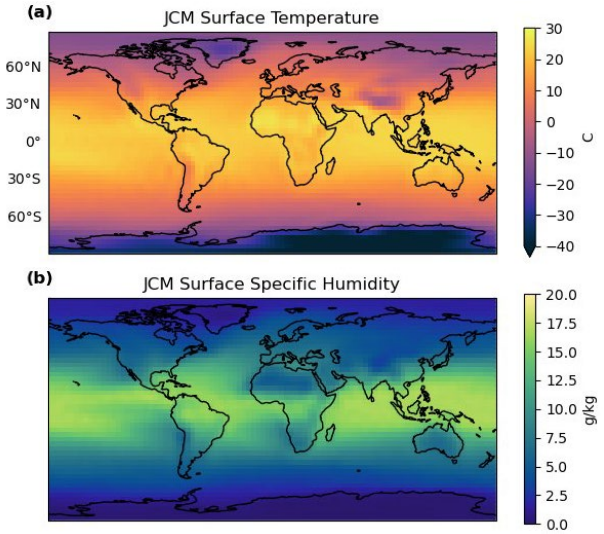
- **Fully differentiable physical** atmospheric Python model with gradients from JAX for calibration and sensitivity analysis out-of-the-box
- **Plug-and-play** parameterizations enable flexible experimentation and online learning
- **Compatible** with CPUs, GPUs, and TPUs, with automatic parallelization for scalability
- **Validated** against Fortran baseline, with example optimization notebooks for tuning and experimentation



Davenport, Maddan et al. *(GMD 2026)*

Gradient based calibration of JCM

Stratiform cloud albedo
Relative humidity threshold
Maximum entrainment

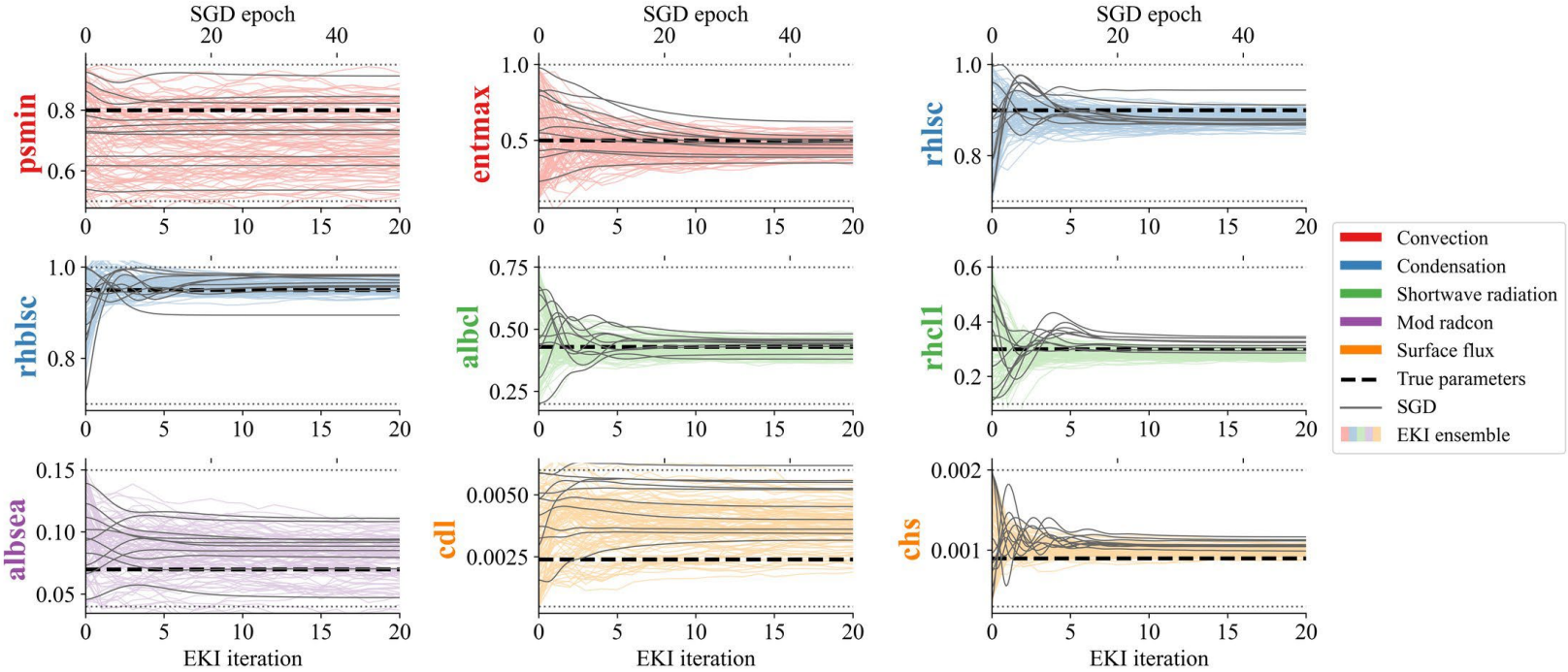


Shortwave radiation
Longwave radiation
Condensation
Convection
Humidity
Surface fluxes
Surface radiative properties

Gjini et al. *(in prep.)*

UC San Diego

Comparison with Ensemble Methods



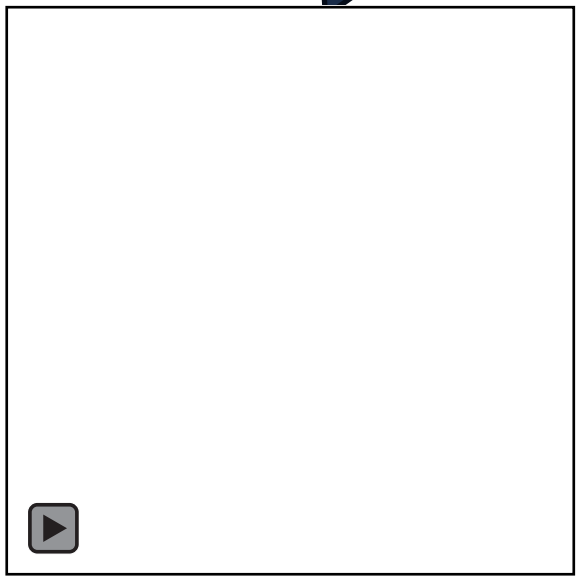
Gjini et al. *(in prep.)*

UC San Diego

Introducing JAX Aerosol Module (JAM)
Coming in JCM v3

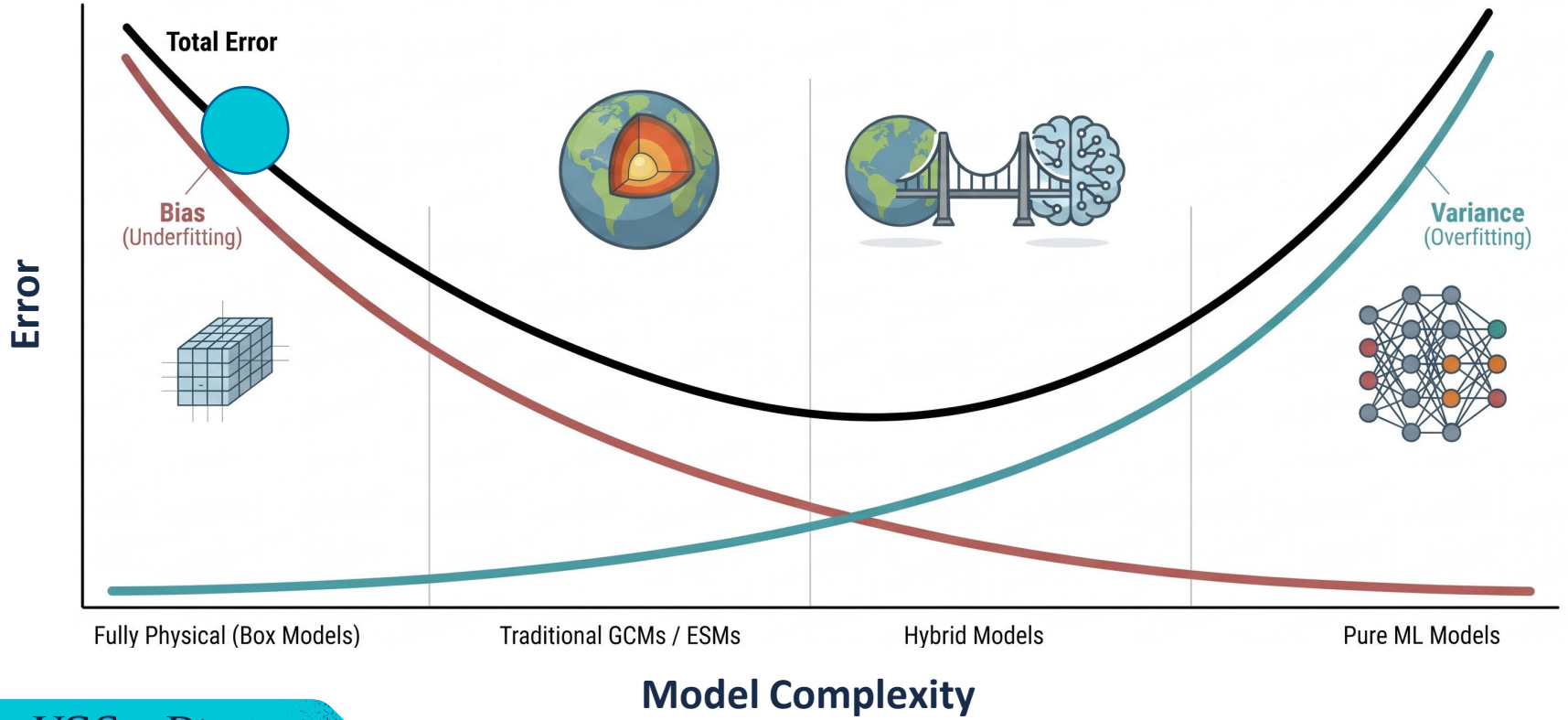
New Version!

Microphysics	MAM4-JAX (swappable), 4 modes, 9 species, 25 interstitial tracers
Chemistry	gas-phase + aqueous sulfur, prescribed oxidants
Sources	CEDS + biomass burning; interactive sea salt, dust, DMS
Sinks	dry dep, sedimentation, in-cloud + below-cloud scavenging
Radiation	per-band Mie optics over modal lognormals
Clouds	2-moment, ARG activation, het. ice nucleation on dust/BC

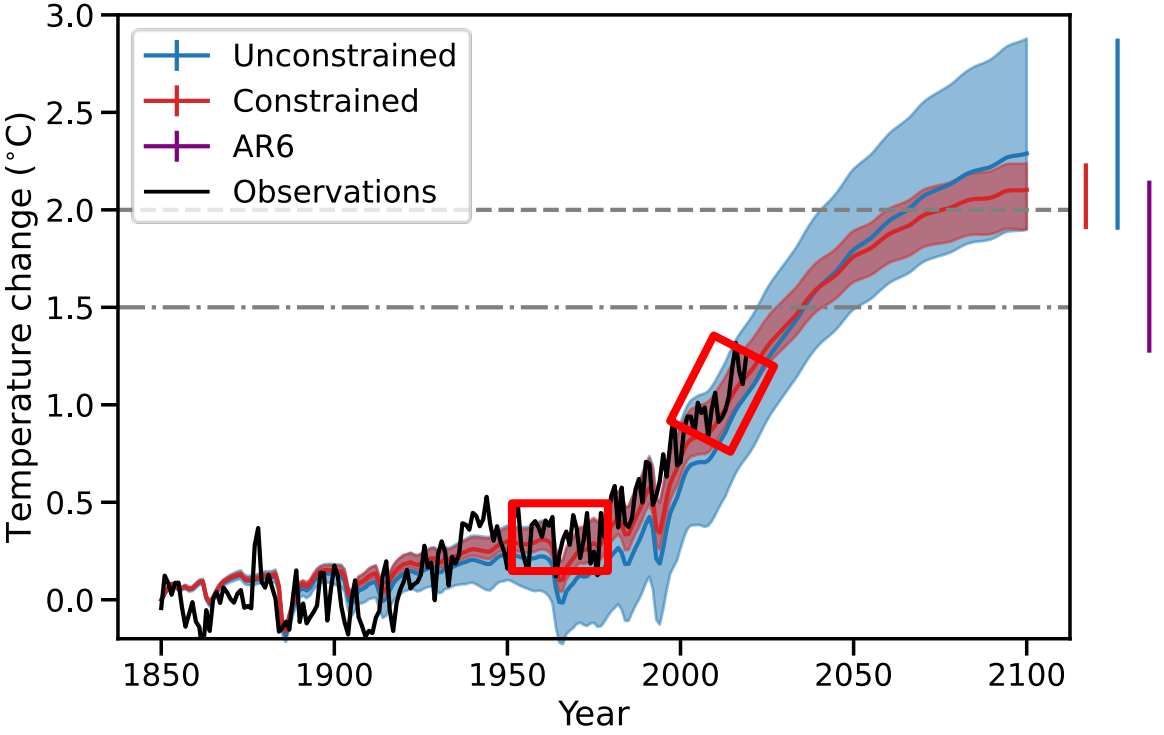


8 simulated years / day on 1× A100 (!)

Finding the sweet spot

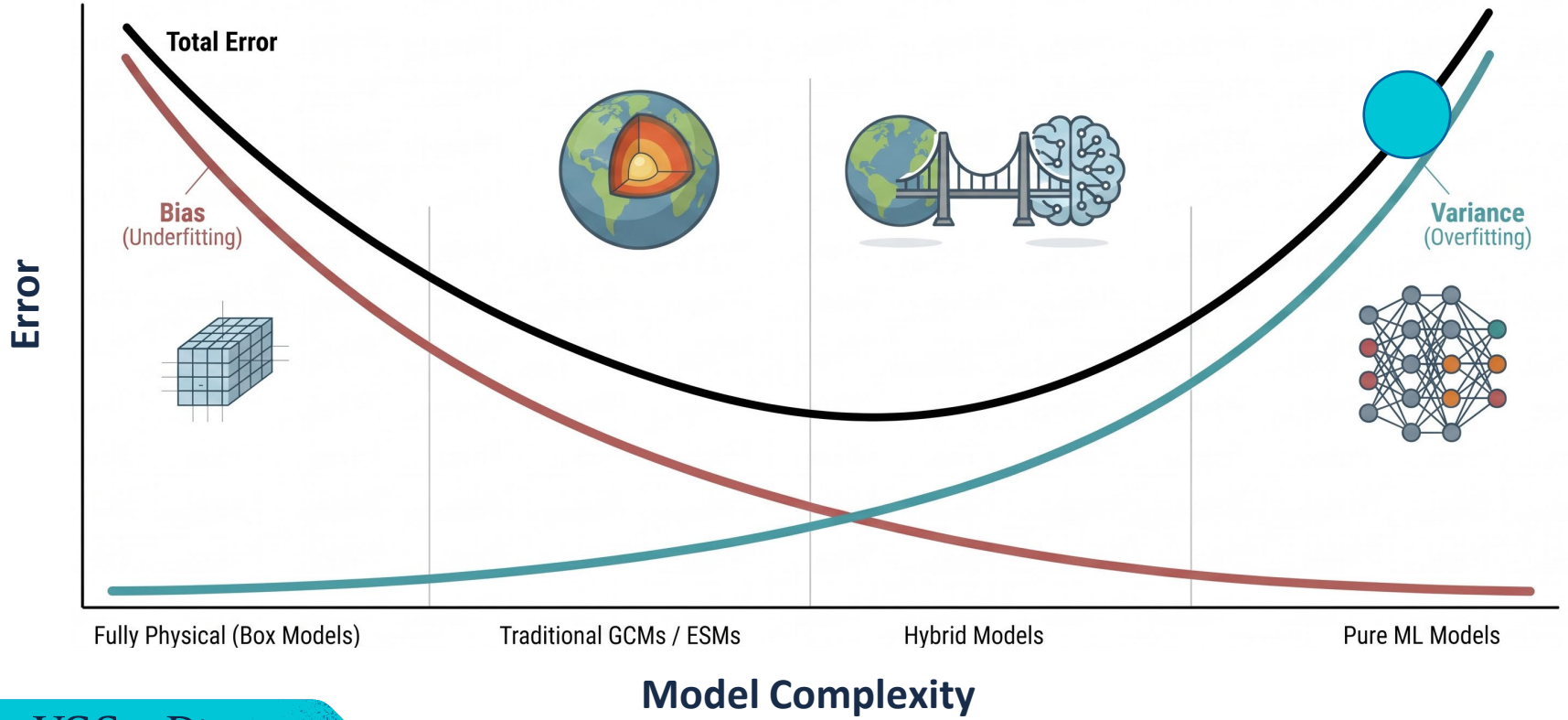


Rate of warming



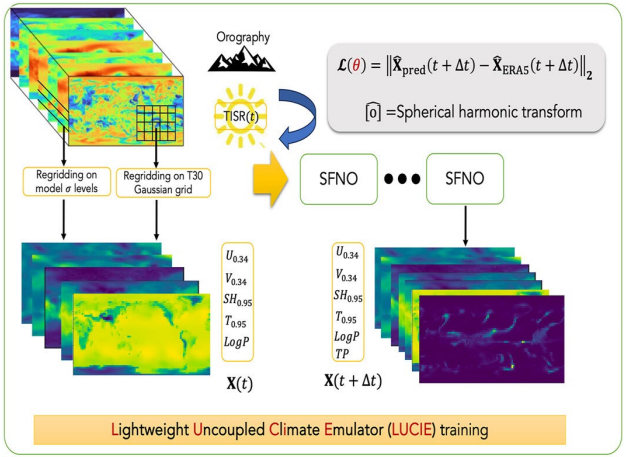
Watson-Parris *(GRL 2026)*

Finding the sweet spot



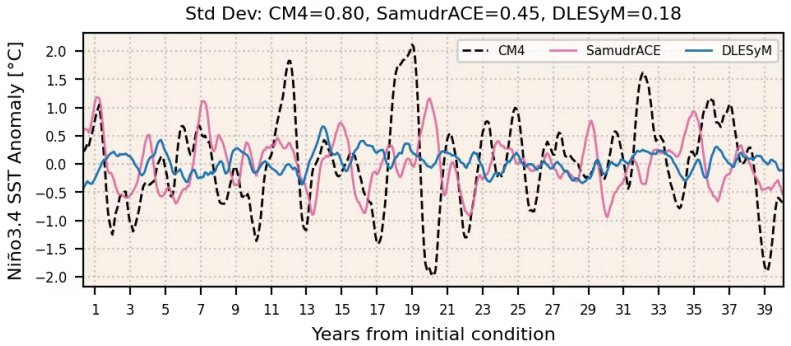
Nascent Climate Emulators

LUCIE



Guan, H., *et al. JAMES 2025*

SamudrACE



Duncan, J., *et al. arXiv 2025*

How can we trust an ML climate model?

How should we test a climate model?



How should we test a climate model?

- Without getting too philosophical, we don't know the truth
- That doesn't mean we don't know anything though
- There is information in our available observations that we can use to assess the likelihood of different climate outcomes
- Importantly, there is also more information in the observations (signal) than there was 10-15 years ago

The Task:

Given a plausible future emissions pathway, what will regional temperature and precipitation changes be in 2050 relative to 1990-2020?

Transient, not equilibrium

More policy-relevant and easier to discern from observations

Minimizes baseline uncertainty

Measures relative to a well-observed 20th century state

Verifiable within my career

Can be refined and updated as the climate evolves

A 3-Tier Protocol for Scoring Climate Models

TIER I

THE PHYSICAL ENTRY TICKET

A "sanity check" verifying models respect fundamental laws: Energy Conservation, Global Water Budgets, and Arctic Amplification.



TIER II

BENCHMARKING AGAINST OBSERVATIONS

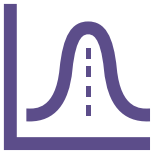
Models scored on ability to reproduce real-world Temperature and Precipitation Trends using reserved post-2015 data.



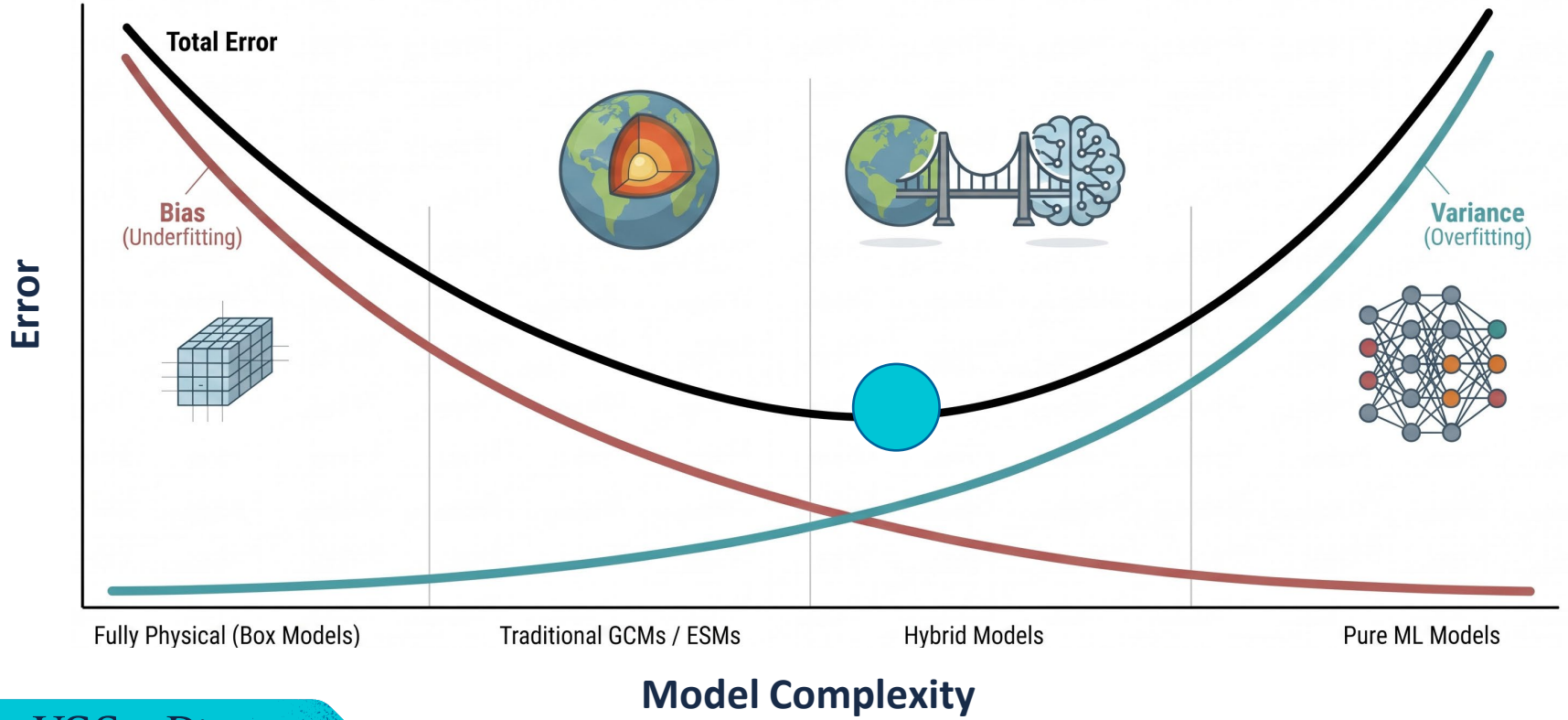
TIER III

OUT-OF-DISTRIBUTION GENERALIZATION

The ultimate test of predictive skill to forecast unseen climate states. Paleoclimate records + perfect-model experiments.



Finding the sweet spot



Neural parameterizations

Radiation emulation

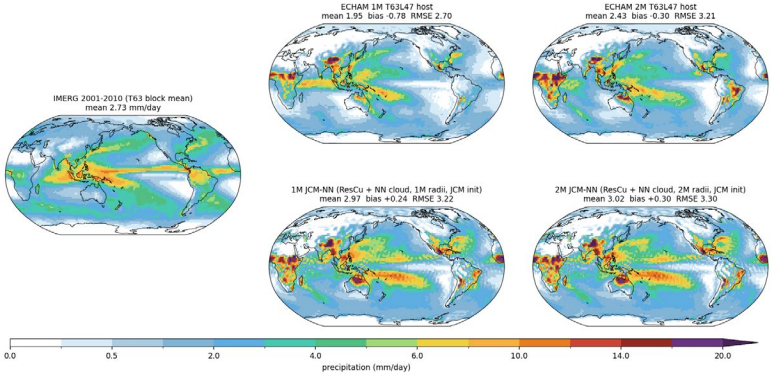
4× **model throughput in JCM with an NN emulator of RRTMGP**

radiation becomes essentially free

- Contrast with Ukkonen et al. (2020): NN gas optics in a Fortran code gave only a modest speed-up of the radiation scheme
- Emulates the full radiation call (not just gas optics) on GPU, alongside the rest of the model
- Emulator is differentiable, so it can sit inside gradient-based calibration

cf. Ukkonen et al. (2020), JAMES

NN convection (ResCu) running online in JCM



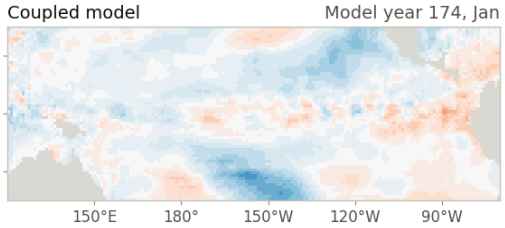
- Stable 5-year online runs with no drift in T63 JCM-ECHAM hosts
- With realistic surface fluxes the precipitation pattern moves towards IMERG: a target for online fine-tuning through the differentiable host

Han & Zhang (prelim.)

Neural components

Physics atmosphere + ML ocean

JCM (SPEEDY) + SamudraX, all in JAX

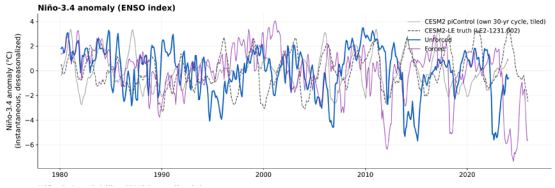


- End-to-end differentiable coupled model
- Enables joint fine-tuning of the ML ocean with the physical atmosphere

Wilke et al. (in prep.)

ML atmosphere + physics ocean

Non-autoregressive ML fluxes driving POP2

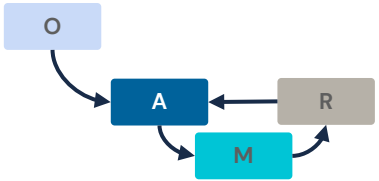


- Stable multi-decade coupled runs with ENSO-like variability
- ~35 SYPD on 2 CPU nodes + 1 A100 vs ~7 SYPD for full CESM2 on 8 nodes

Aggarwal et al. (in prep.)

Learning each component

Reflective: each process, incl. a dry dycore



- Emulate each component separately, then reassemble
- Errors can be attributed to a component, improving interpretability

Reflective (in prep.)

Summary

- There is no single right level of complexity: the sweet spot depends on the question and the goal, so we work across the whole model hierarchy
- Gradients: JCM shows a fully differentiable atmospheric model in JAX is feasible, efficient and useful, unlocking calibration and sensitivity analysis that legacy Fortran cannot support
- Benchmarks: ClimateBench v2 provides a rigorous framework for scoring any climate model, physical or ML, on a concrete, policy-relevant prediction task
- Hybrids: neural parameterizations and components slot into the same framework, e.g. 4× throughput from NN radiation and stable ML-atmosphere/ocean couplings

dwatsonparris@ucsd.edu · github.com/climate-analytics-lab

SPARE SLIDES



Climate Model Emulation

Tier I: Establishing Physical Credibility

Top-of-Atmosphere (TOA) Energy Balance

Models must close the global energy budget to within 0.1 W m^{-2} per decade across atmosphere, ocean, and land.

Arctic Amplification

Arctic (>66.5°N) must warm ≥1.5× global mean under CO_2 forcing. Scored against abrupt-4xCO2.

Land–Ocean Warming Contrast

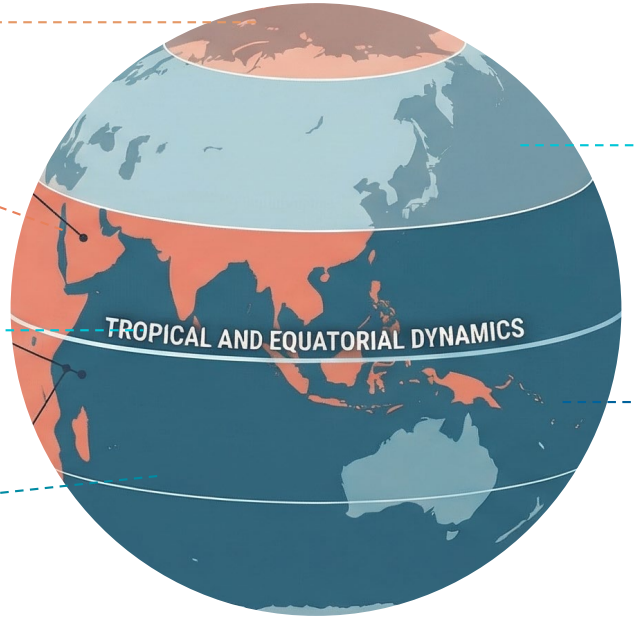
Land warms 1.2–1.6× faster than ocean due to lower heat capacity & reduced evaporation

Coupled Variability (ENSO Check)

ENSO spectral peak 2–7 yr; Niño-3.4 SD within 0.4–1.8 K. Correct sign of teleconnections.

ITCZ – Energy Flux Equator

ITCZ shifts ~3° toward warmer hemisphere per PW of cross-equatorial atmospheric energy flux.



LARGE-SCALE COUPLED DIAGNOSTICS

Meridional Heat Transport Partitioning

Energy crossover at ~30°. Ocean dominates tropics (1.5–2.0 PW); atmosphere dominates mid-latitudes (4–5 PW). Crossover latitude scored.

Bjerknes Compensation

Anomalous poleward OHT at 40–70°N is anti-correlated with atmospheric heat transport on decadal timescales. Primarily boreal winter response.

Tier II: Benchmarking Against Observations

Core Evaluation Variables

2m Temperature
mean, trends (seasonal/annual) — GISS, Berkeley, HadCRUT

Temperature Extremes
daily PDFs — Berkeley Earth (daily), HadGHCND

Precipitation
mean, trends, PDFs, extremes — GPCP, IMERG, MSWEP

TOA Radiative Fluxes
LW/SW, clear & all-sky — CERES EBAF

Sea Ice
extent & trends — NSIDC, OSI SAF

Sensible & Latent Heat
Surface energy fluxes — CERES SYN

Scoring Metrics

CRPS — primary probabilistic score across all variables
Spread–skill ratio — requires ≥3 ensemble members (ideally 10)
RMSE + correlation — diagnostic metrics for mean state & trends
Wasserstein distance — for precipitation PDFs and extremes

Test / Train Split

All observational data after 2015 is reserved for testing — not used for training or calibration.



Tier III: Out-of-Distribution Generalization

Paleoclimate Out-of-Distribution Tests

Last Interglacial (LIG, ~127 ka) +0.5 K

Strong high-latitude warming driven by orbital forcing. Reduced Arctic sea ice, smaller Greenland ice sheet.

Proxies: Osman (2026); Otto-Bliesner et al. (2021)

Last Glacial Maximum (LGM, ~21 ka) −5 to −7 K

Expanded ice sheets, CO_2 ~190 ppm. Strongest constraint on upper bound of climate sensitivity.

Proxies: Tierney et al. (2020); Osman et al. (2021)

Mid-Holocene (Mid-H, ~6 ka) Small ΔT

Orbital forcing, strengthened boreal monsoon, 'Green Sahara'. Tests regional precipitation response.

Proxies: Osman et al. (2021); Kaufman et al. (2020)

Scoring: CRPS-based, adapted for large proxy uncertainties. Spatial patterns evaluated, not just global mean.

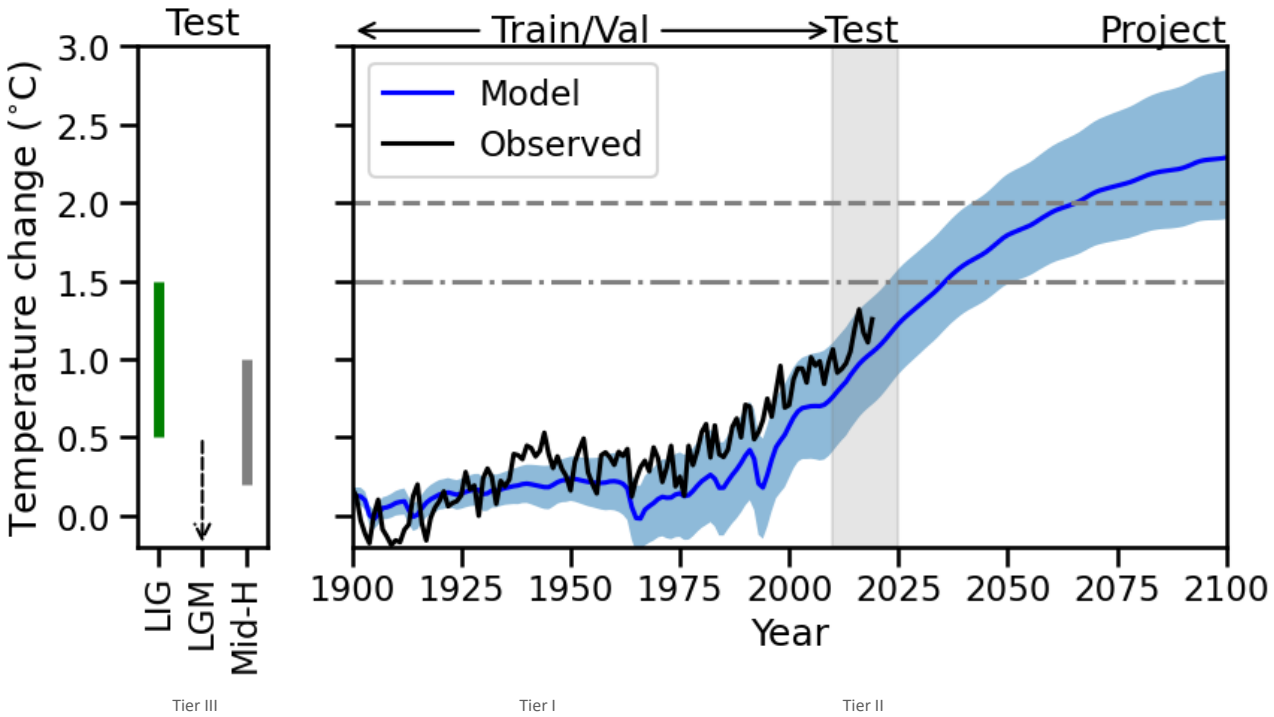
Perfect-Model Experiments (ML models)

Train on a target ESM's historical output; evaluate against its held-out future simulations. Tests whether a model can predict future climate from present-day data alone.

CESM2 ECS ≈ 5.2 K	REQUIRED
MPI-ESM ECS ≈ 3.63 K	Optional
GISS ModelE2 ECS ≈ 2.31 K	Optional

Large ensemble test (>20 members vs CESM-LE):
Checks spread across initial-condition perturbations under a common forced signal - probes the interaction of forcing with internal variability.

Test-train split



Down the hierarchy: JAX-LES

- GPU-first atmospheric large-eddy simulation in JAX, extracted from Google's Swirl-LM (Chammas et al. 2023) and rebuilt for GPUs
- Dry and moist anelastic cores ported module by module against golden data, reproducing the TensorFlow original to float32
- Sharded multi-GPU meshes with checkpointed cycles; differentiability a design constraint throughout, not an afterthought
- Shares the jax-gcm PhysicsTerm interface, so a parameterization is literally the same object in the LES and in the GCM

github.com/climate-analytics-lab/jax-les · pre-alpha, Phase 3 (moist physics)



DYCOMS-II RF01 nocturnal stratocumulus

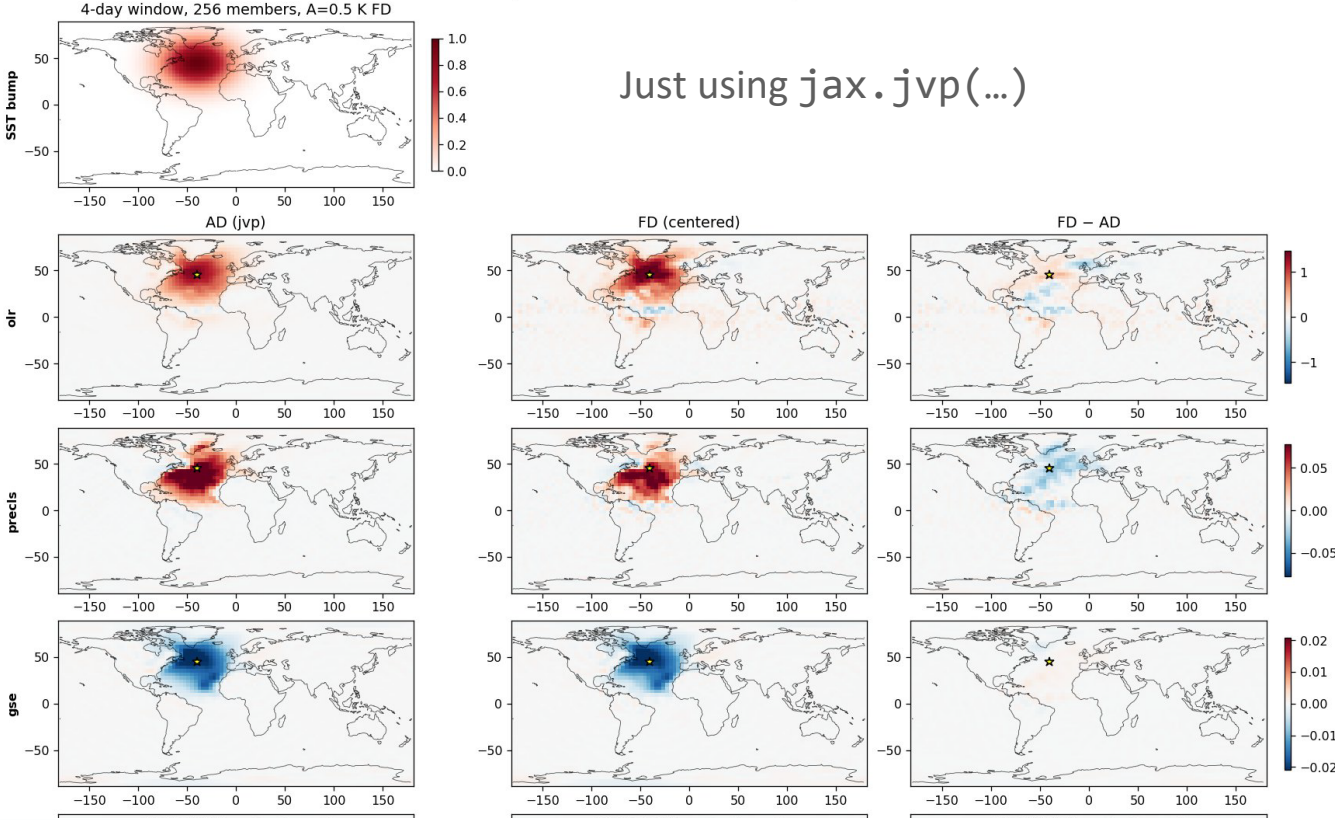
4 simulated hours, 32 m × 6 m grid



BOMEX trade-wind cumulus

6 simulated hours, 100 m × 40 m grid

Boundary Condition Sensitivity

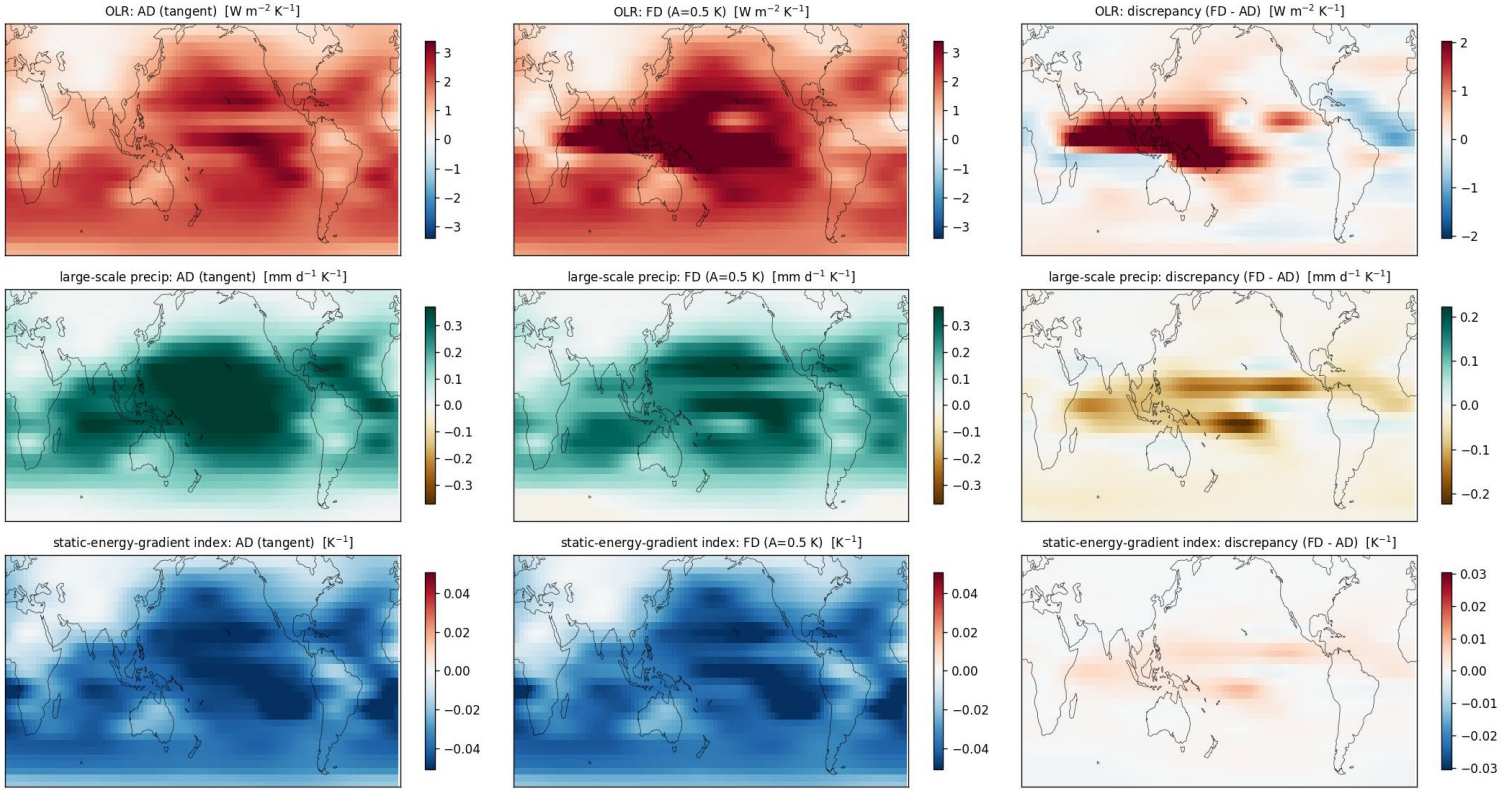


Just using `jax.jvp(…)`

Maddan et al. *(in prep.)*

UC San Diego

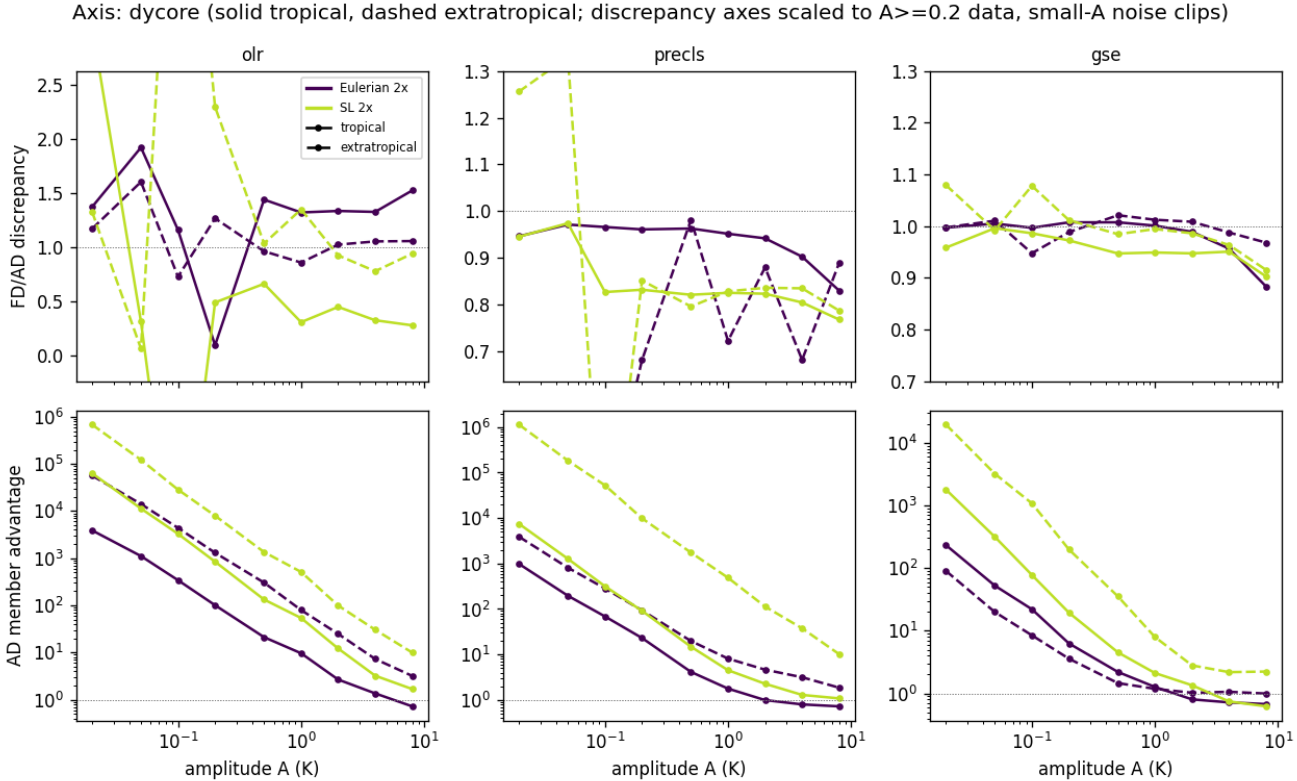
Boundary Condition Sensitivity



Maddan et al. *(in prep.)*

UC San Diego

Accelerated convergence



Maddan et al. *(in prep.)*

UC San Diego

Towards a coupled model: JAX-ESM

- In parallel we have developed a flexible, JAX based coupler as the basis for a full ESM
- Our current prototype can couple JCM with a simple slab ocean an run stably for years
- Current work is ongoing to develop interpolation routines and a coupler to the Veros ocean model and DifferLand

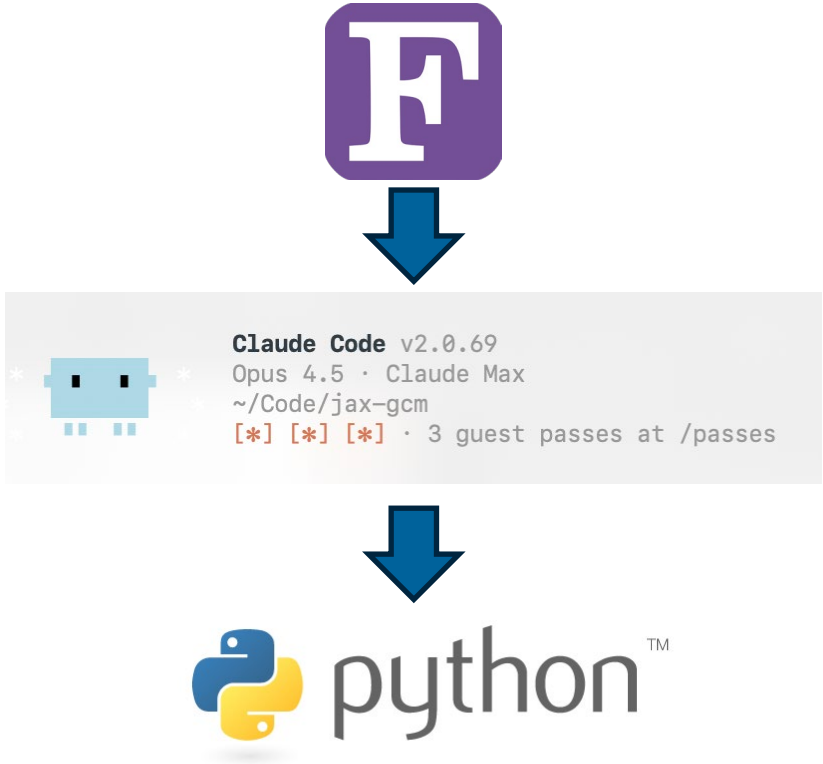
```
# Creating components
model = couple(dict(
   atm = JCM(
      model =jcm.model.Model(),
      coupling_timestep = timestep,
   ),
   ocn = SlabOceanModel(
                  timestep = timestep,
                  start_dt = start_datetime,
   ),
))

initial_state = model.initialize()

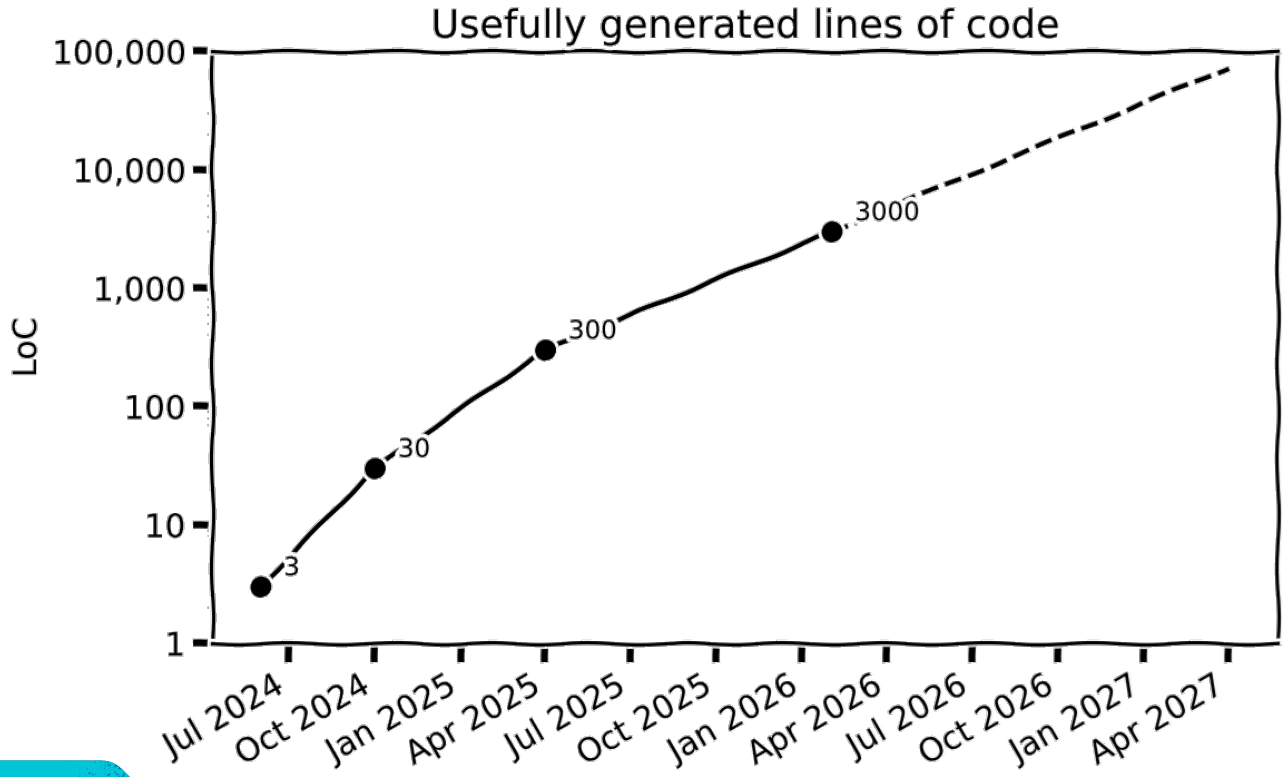
# Run coupled model
model.run(initial_state, end_time = 10)
```

AI Assisted Conversion of Legacy Code

- Original effort (~18 months ago) used ChatGPT to support conversions of specific routines / loops from Fortran
- We then hosted a hackathon to workout high-level APIs and develop unit tests
- Since then we have used agentic tools to convert whole parameterizations at a time: ECHAM6 physics, and now the GISS ModelE convection scheme



The future of model development?



Thank you!