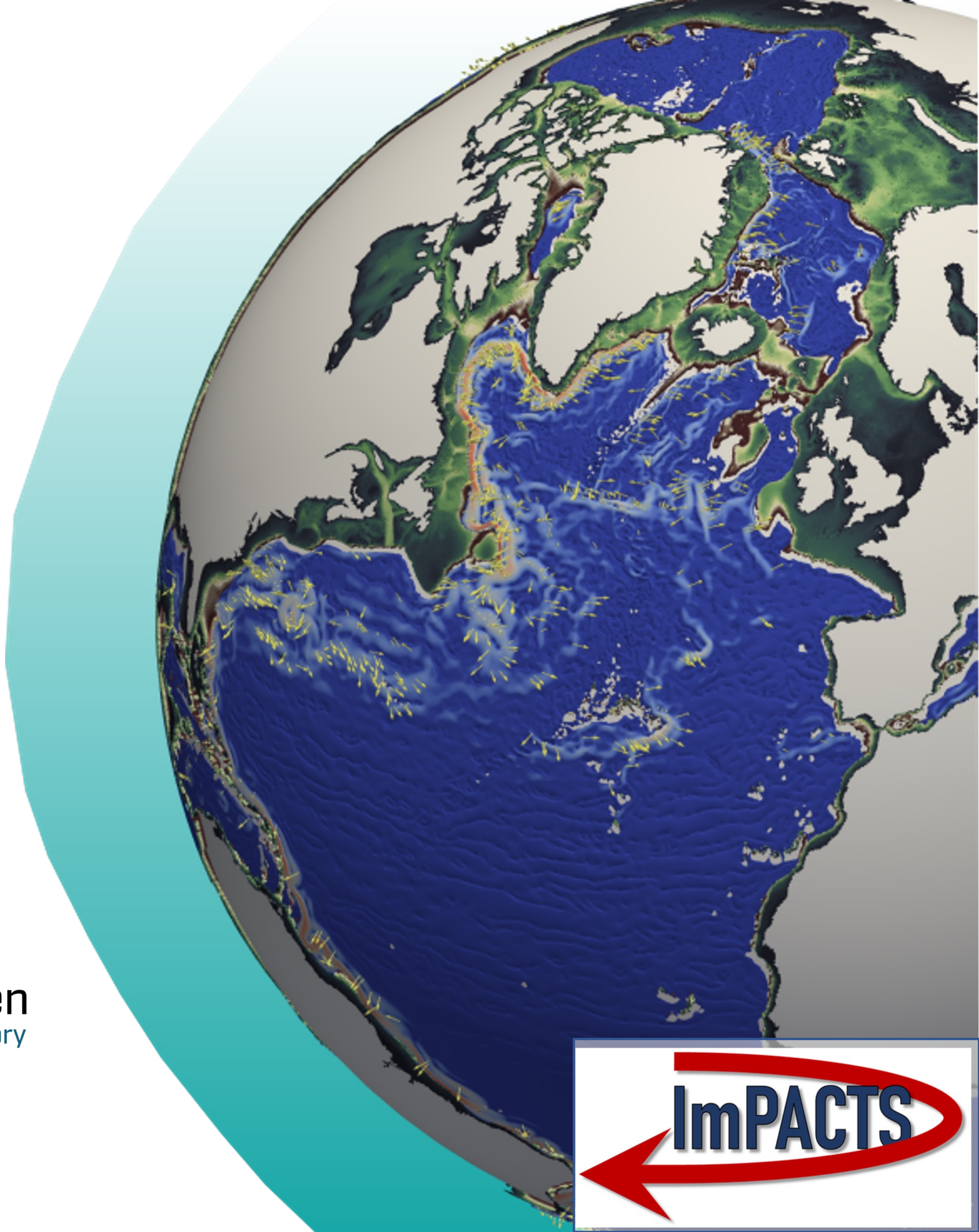
Improving Projections of AMOC and its Collapse Through Advanced Simulations (ImPACTS) project overview

Luke Van Roekel, Alice Barthel, Sri Hari Krishna Narayanan, Wei Xu, Sarat Sreepathi on behalf of the entire ImPACTS team

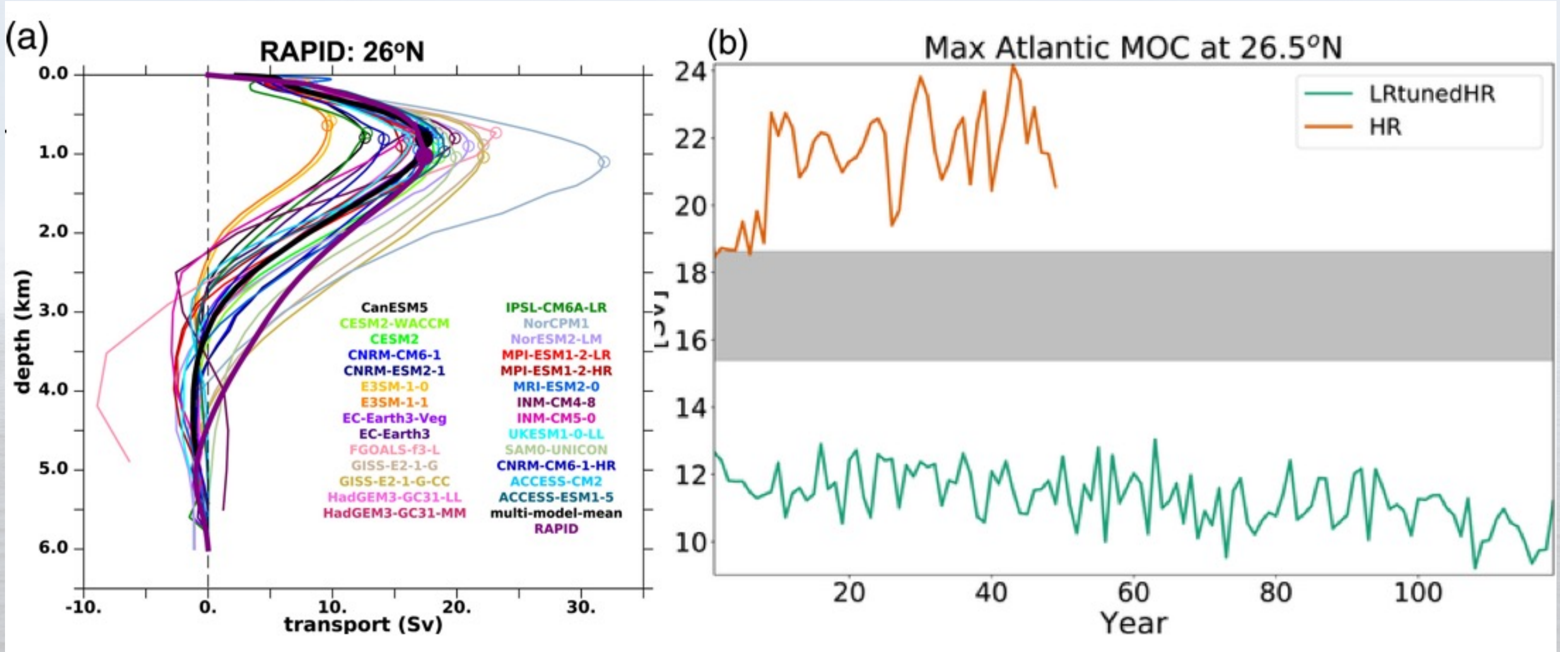
June 22, 2023



Los Alamos NATIONAL LABORATORY
Argonne NATIONAL LABORATORY
OAK RIDGE National Laboratory
Brookhaven National Laboratory
NCAR
SCRIPPS INSTITUTION OF OCEANOGRAPHY
TEXAS The University of Texas at Austin
OHIO STATE
ImPACTS

Project Overview

- The Atlantic Meridional Overturning Circulation (AMOC) is a well known E3SM bias at non-eddy resolving resolutions



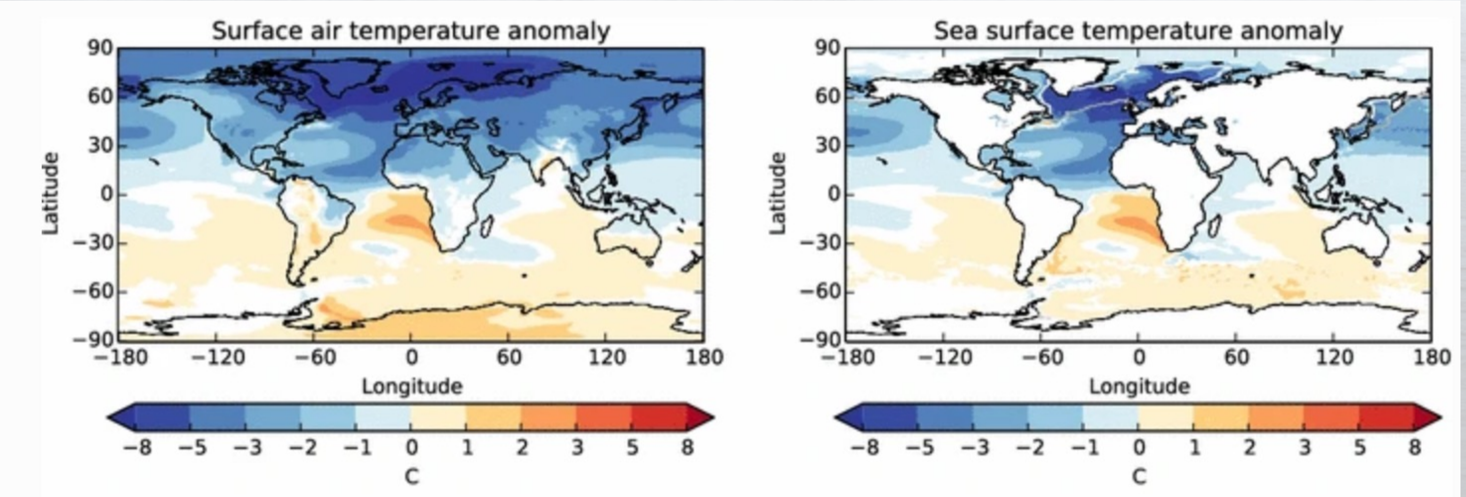
Weijer et al 2020

Caldwell et al 2019

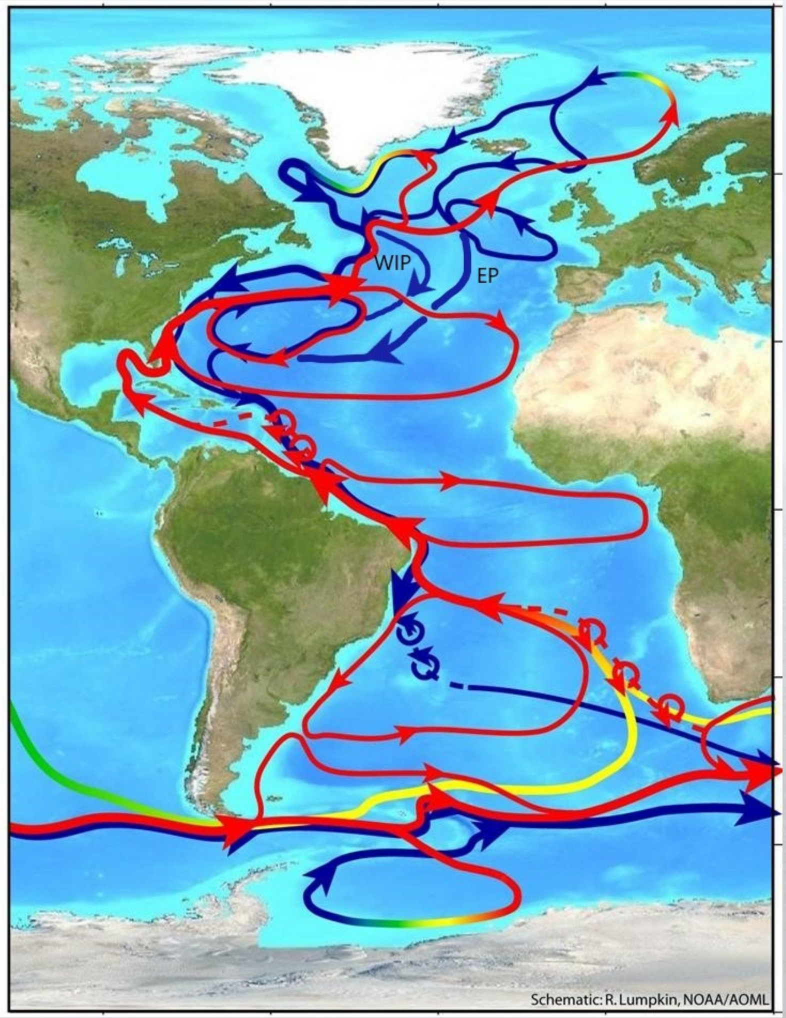
- A major goal of the ImPACTS project is to advance our understanding of AMOC to improve its representation in models like E3SM

AMOC

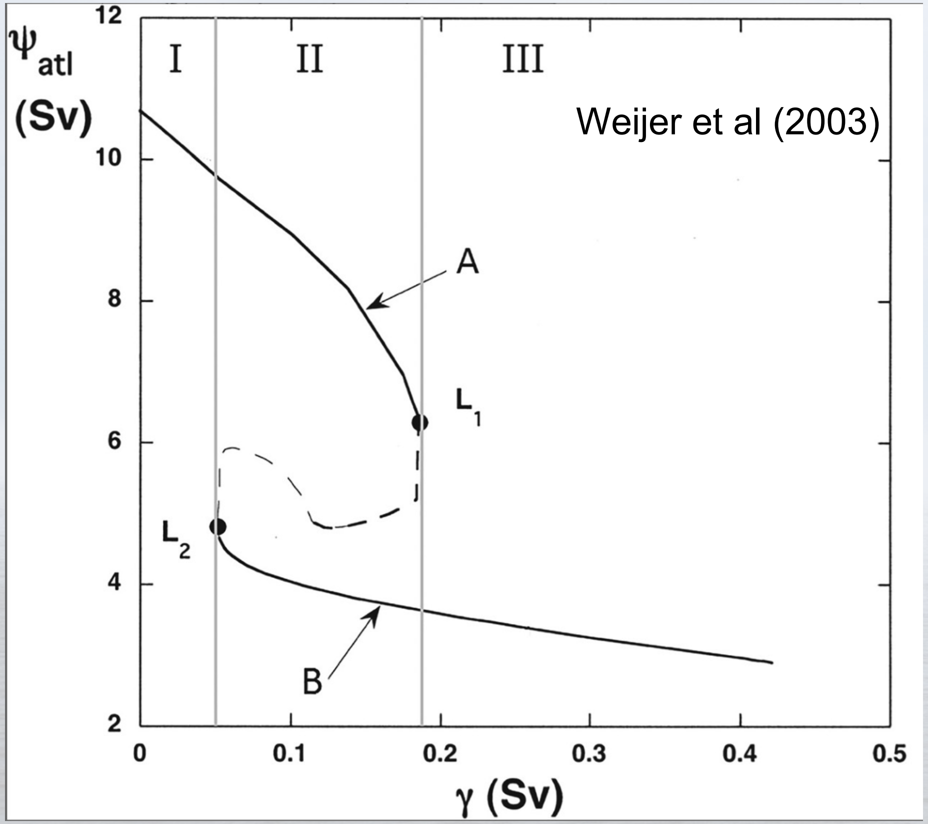
- Redistributes significant amounts of heat and salt
- Linked to numerous climate impacts
    - Regional temperature impacts, Atlantic Hurricanes , Sahel Droughts
- Loss of AMOC has strong climatic impacts.



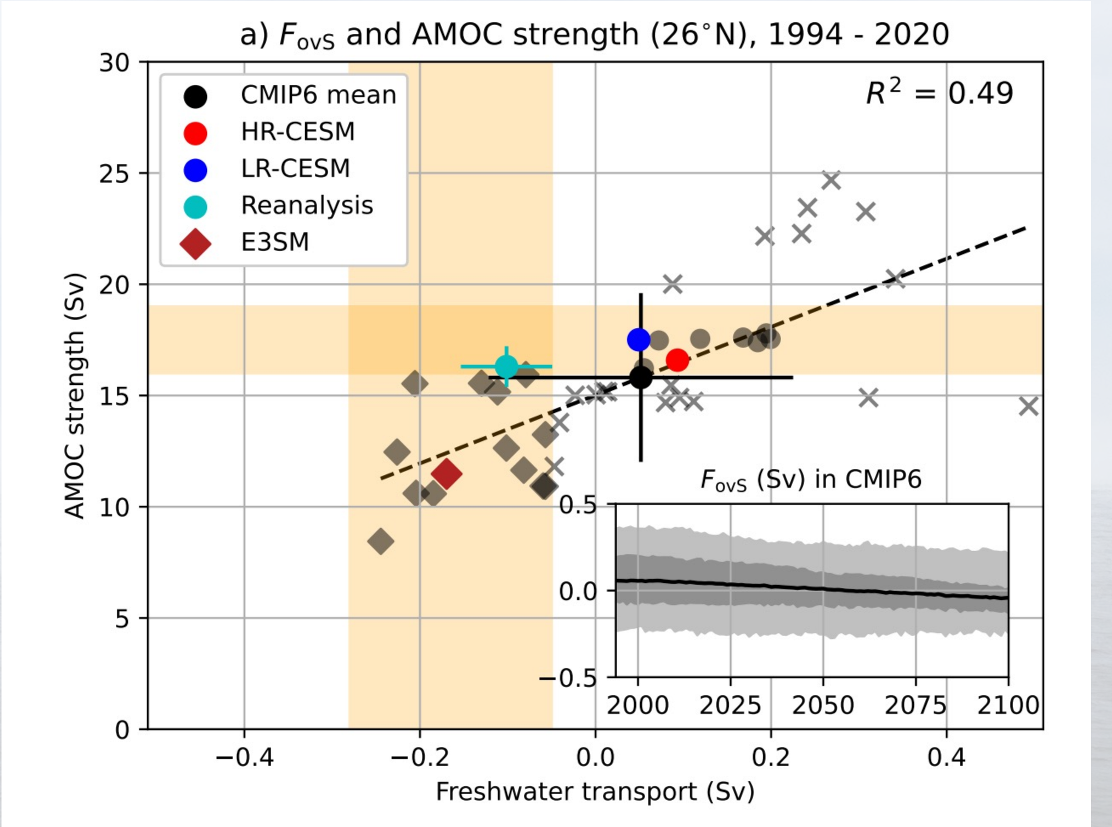
Jackson et al. (2015)



AMOC Stability

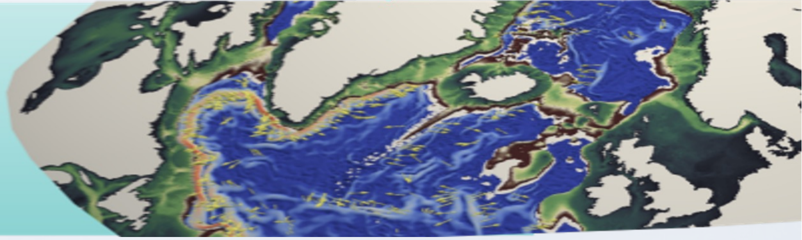


van Westen (2023)



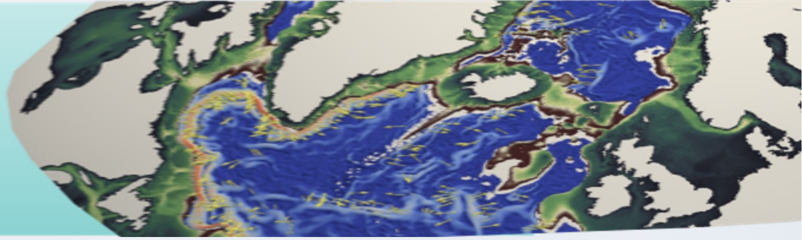
- AMOC bi-stability long hypothesized and shown in simple models (e.g. box models, ICMs)
- Freshwater transport at 34S (FovS) is related to AMOC stability and strength

Project Goals



- Understand the weak AMOC simulated by E3SM and improve the representation.
  - Leverage traditional oceanographic analyses in addition to advanced AI analysis (e.g. NN based model adjoint)
- Improve analysis capability of ocean models
  - Improve in situ diagnostics, contribute to common analyses frameworks (e.g. METRIC), create novel algorithms to make lagrangian particle tracking possible for long term, high resolution configurations
- Assess AMOC stability at coarse and eddy resolving resolution
  - Advance spin up capabilities with AI methods
  - Improve model performance (GPU refactorization, new time stepping)

Project Team and Structure



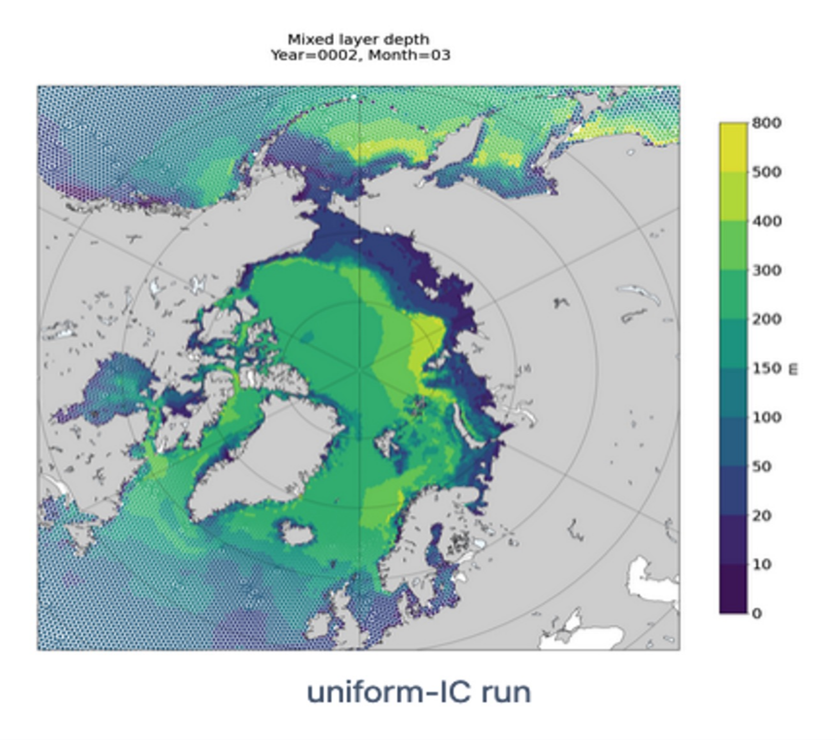
- Project team
  - Eight institutions
  - Eight members of SciDAC RAPIDS2 institute
  - Well balanced between ASCR and BER efforts
- Initial project structure
  - Simulations and Analysis*
  - Adjoint
  - AI/Particles*
  - Performance

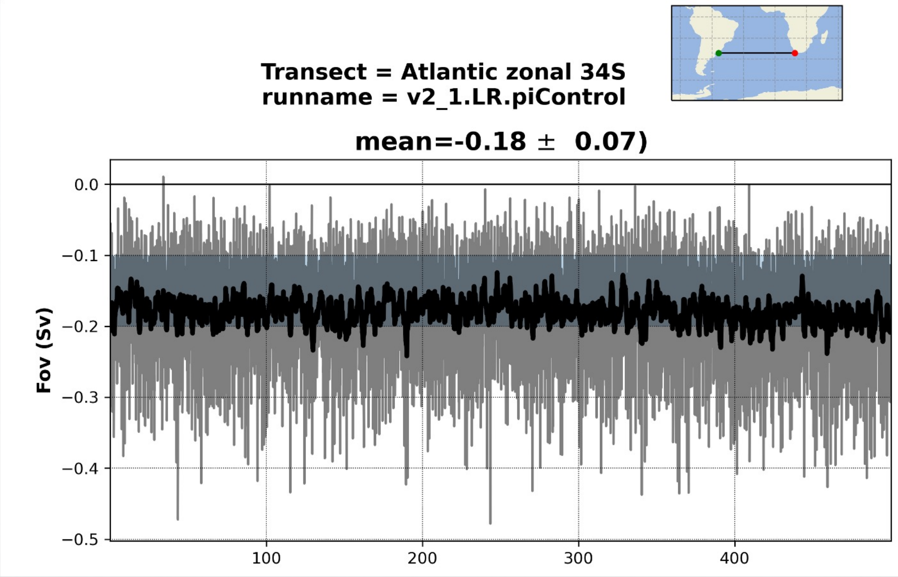
*Early career lead

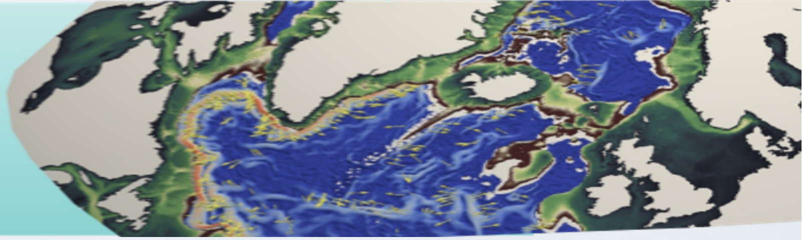
Analysis

Initial progress

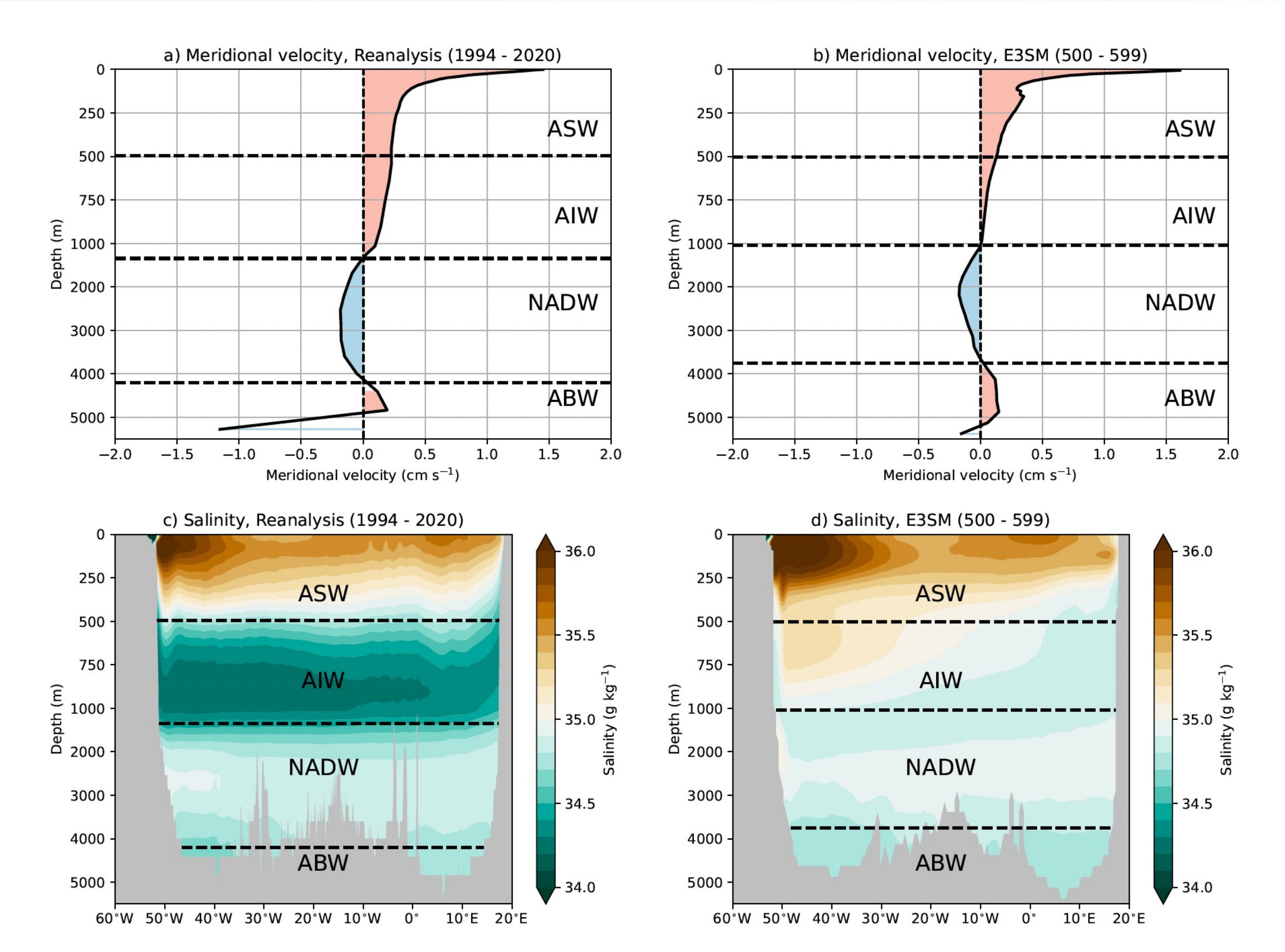
- Designed new simulations (uniform IC)
    - Top right
- Lots of analysis on going
    - Water masses
    - Pathway analysis
    - Fov
    - Streamfunction
- Some analyses being hardened for MPAS-Analysis integration
    - Bottom right



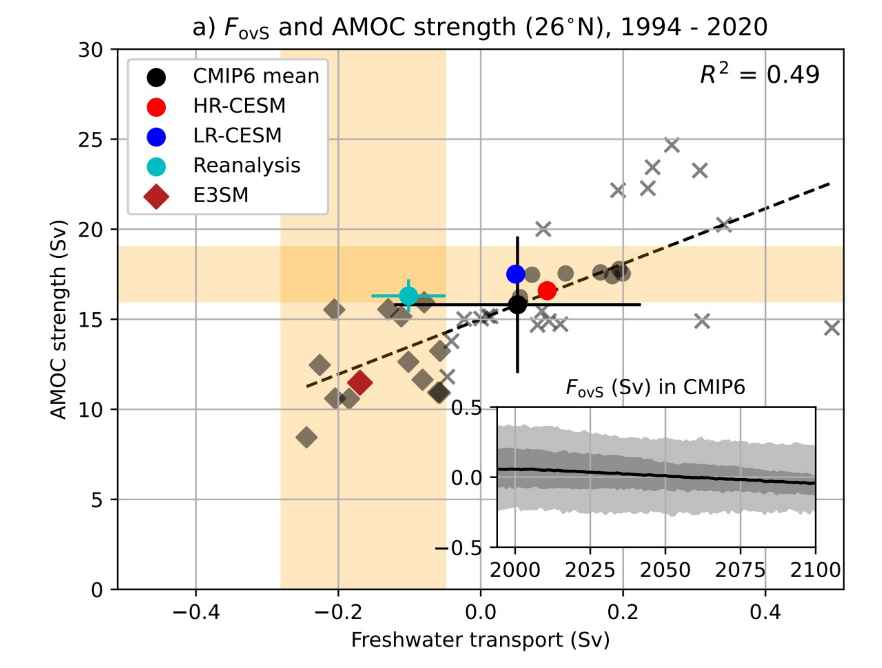




Example: Fov and AIW



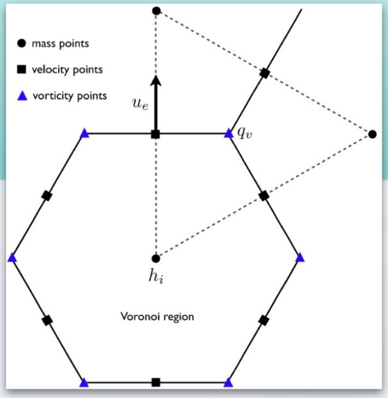
- SORRM simulations missing AIW
- Large impact on Fov
  - Reduces Fov, which may reduce AMOC



Adjoint

- **Goal**: inferring the sensitivity of AMOC with respect to a large number of input model parameters that may or may not have spatial structure, but which are temporally invariant
- **Approach**: obtain neural-network-based adjoints
  - Exploit differentiability of NN emulators of AMOC
  - Use DeepHyper (AutoML software package; RAPIDS2)
  - Training data consisting of
    - CMIP6 data
    - MPAS stacked shallow-water (SSW) simulations
    - E3SM historical and preindustrial simulations
    - ECCO/MITgcm simulations incl. adjoint sensitivity calculations
- **Year 1 tasks**:
  - Pretrain adjoint surrogate on CMIP6 data
  - Create adjoint of MPAS-SSW Julia

MPAS SSW (Julia)



$$\frac{\partial \xi}{\partial t} + \nabla \cdot (h\bar{\mathbf{u}}^z) = 0$$

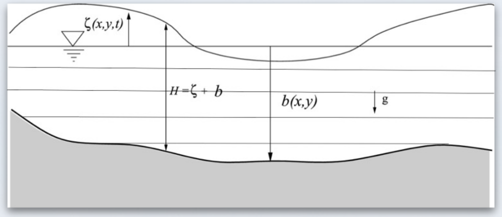
Conservation of Mass $\quad \nabla_3 \cdot \mathbf{v} = 0, \qquad \eta = \mathbf{k} \cancel{\cdot \nabla \times} \mathbf{u} + f$

Momentum Equation $\quad \frac{\partial \mathbf{u}}{\partial t} + \eta \mathbf{k} \times \mathbf{u} + \cancel{w \frac{\partial \mathbf{u}}{\partial z}} = -\frac{1}{\rho_0} \nabla p - \cancel{\nabla K} + \cancel{D_h^u} + \cancel{D_v^u}$

Forward Euler

$g\nabla\xi$

Tracer Equation $\quad \cancel{\frac{\partial \tilde{\rho}\varphi}{\partial t} + \nabla \cdot (\tilde{\rho}\varphi \mathbf{u}) + \frac{\partial}{\partial z}(\tilde{\rho}\varphi w) = D_h^\varphi + D_v^\varphi},$



Pressure Equation $\quad p(x,y,z) = \cancel{p^s(x,y)} + \int_z^{z^s} \rho g dz'$

ρ_0

Equation of state $\quad \cancel{\rho = f_{eos}(\Theta, S, p)}.$

- Simplified MPAS
- Adding Vertical mixing Other possibilities:
  - Temperature and salinity
  - Nonlinear advection
- Differentiating the code using Enzyme.jl
  - Partly done
- Currently working with other models

Physics/Processes

- No momentum advection (linearized)
- No tracer (T, S) advection
- No equation of state (constant density)
- No vertical velocity/momentum advection
- No vertical mixing (KPP)
- No horizontal mixing (del2, del4)
- No bottom friction
- No surface flux/stress/pressure
- No mesoscale parameterizations (GM, Redi)
- Periodic boundary conditions

Algorithmic

- 1st order timestepping
- Domain decomposition limited to rectangular meshes
- Single layer halo exchanges
- No ALE vertical coordinate

Adjoint matching training: Burgers' Equation

- 1D Burgers' Equation was studied.
- Data was generated from a fixed initial condition
- DeepHyper was not used

Physical forward model

$$\mathbf{u_{t+1}} = \mathcal{M}(\mathbf{u_t})$$

Neural network surrogate model

$$\hat{\mathbf{u}}_{t+1} = \mathcal{N}(\mathbf{u}_t) = \mathcal{M}(\mathbf{u}_t) + \epsilon(\mathbf{u}_t)$$

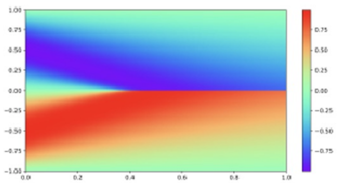
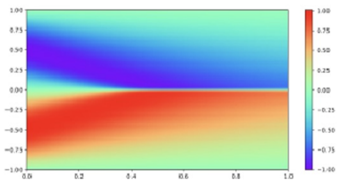
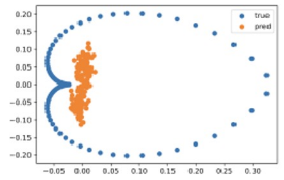
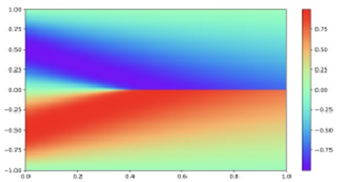
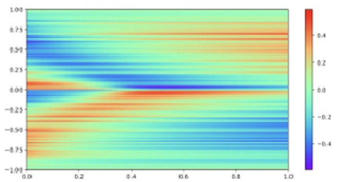
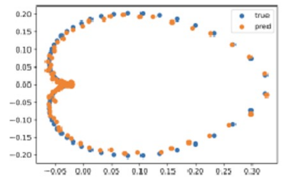
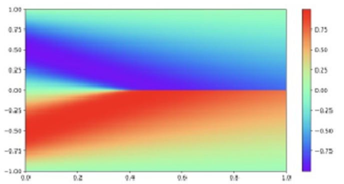
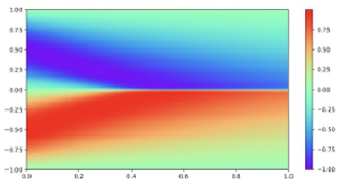
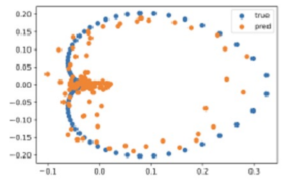
Two-part loss function

$$L = L_{standard} + \alpha L_{adj}$$

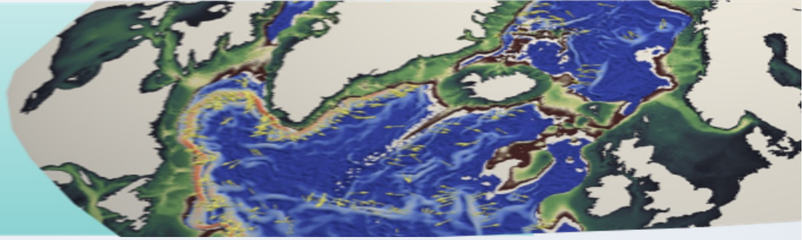
Adjoint matching loss term

$$L_{adj} = \sum_i ||N^T(\mathbf{u}_{t_i}) - M^T(\mathbf{u}_{t_i})||_2^2$$

$$N = \frac{\partial \mathcal{N}}{\partial \mathbf{u}_t}, \quad M = \frac{\partial \mathcal{M}}{\partial \mathbf{u}_t}$$

	True solution	Predicted solution	Adjoints
Forward loss only			
Adjoint loss only			
Forward + adjoint			

Adjoint matching training: Burgers’ Equation



- Varying the initial condition
- Parameter was not varied

Physical forward model

$$\mathbf{u_{t+1}} = \mathcal{M}(\mathbf{u_t})$$

Neural network surrogate model

$$\hat{\mathbf{u}}_{t+1} = \mathcal{N}(\mathbf{u}_t) = \mathcal{M}(\mathbf{u}_t) + \epsilon(\mathbf{u}_t)$$

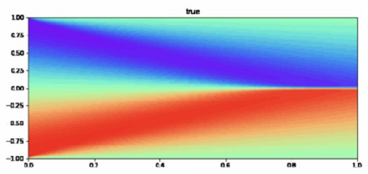
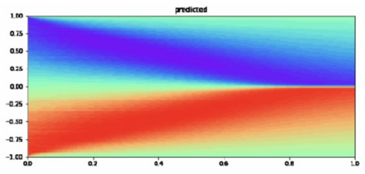
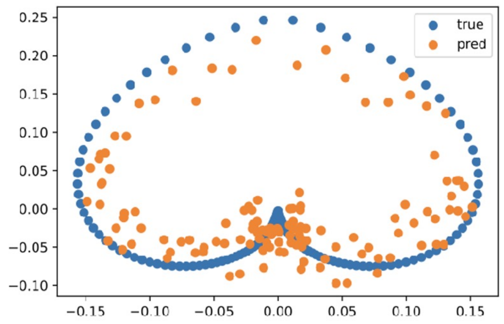
Two-part loss function

$$L = L_{standard} + \alpha L_{adj}$$

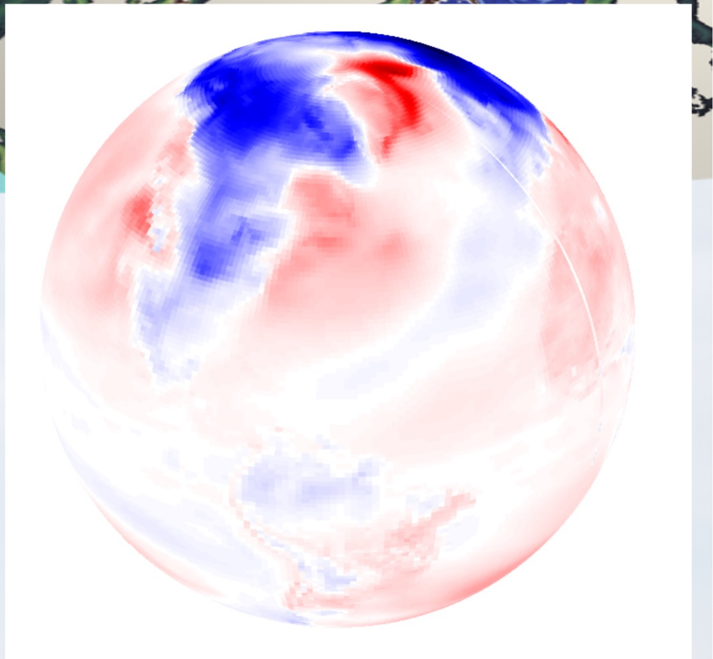
Adjoint matching loss term

$$L_{adj} = \sum_i ||N^T(\mathbf{u}_{t_i}) - M^T(\mathbf{u}_{t_i})||_2^2$$

$$N = \frac{\partial \mathcal{N}}{\partial \mathbf{u}_t}, \quad M = \frac{\partial \mathcal{M}}{\partial \mathbf{u}_t}$$

True solution	Predicted solution	Adjoints
		

ML Analysis

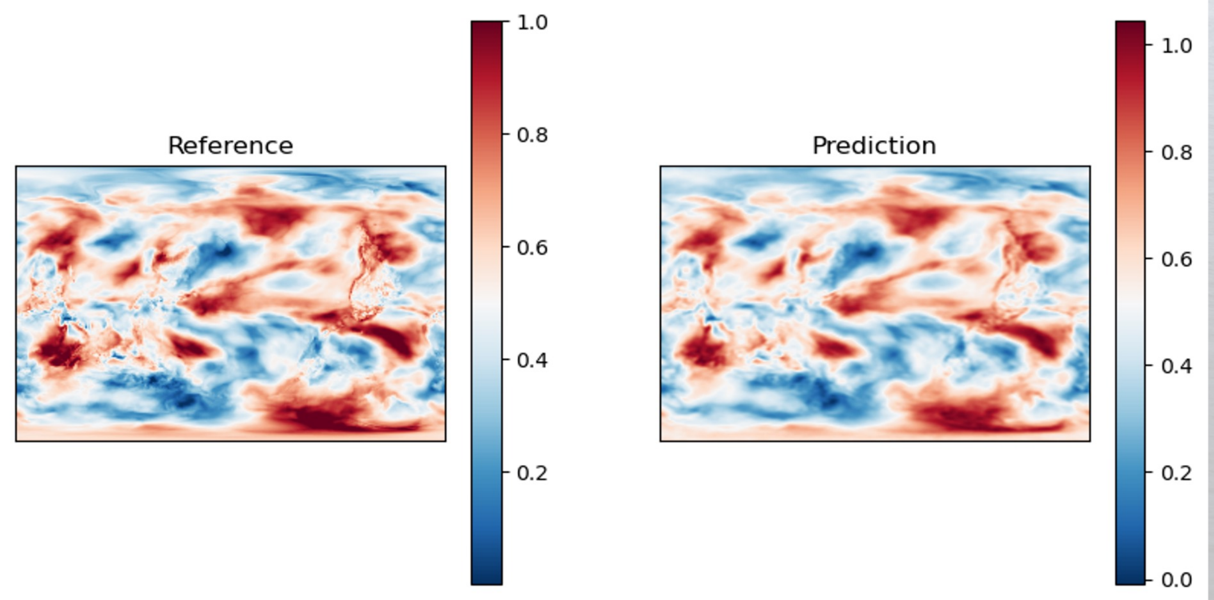


Goal: Develop deep learning-based surrogate models for reconstructing climate maps from limited measurements

Current work: Developing a deep learning-based surrogate model for reconstructing climate maps from limited measurements

- Dataset: Global surface temperature data
- Objective: Reconstructing the whole data from randomly distributed data points
- Methodology: coordinates-based neural networks

Preliminary results: figures on the right



Future plans:

- Fine-tuning the model
- Conducting baseline comparisons to enhance its performance

Tensor Factorizations

Progress:

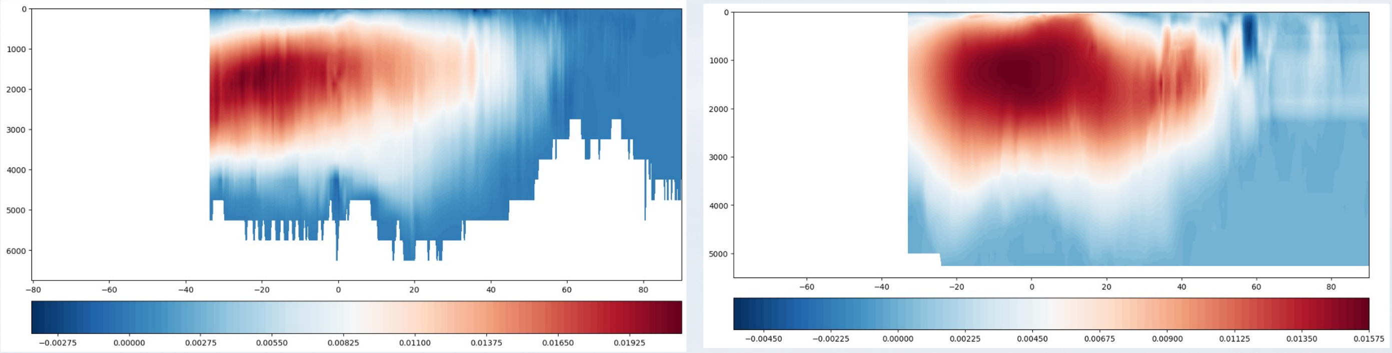
1. How to article/notebook - What are tensor factorizations + code for Earth science
2. Offline TF code for unstructured meshes (graph-based)
    1. First application to ESM output
3. Mode comparison for EOFs between different model representations of AMOC (GFDL and CESM2)

Started:

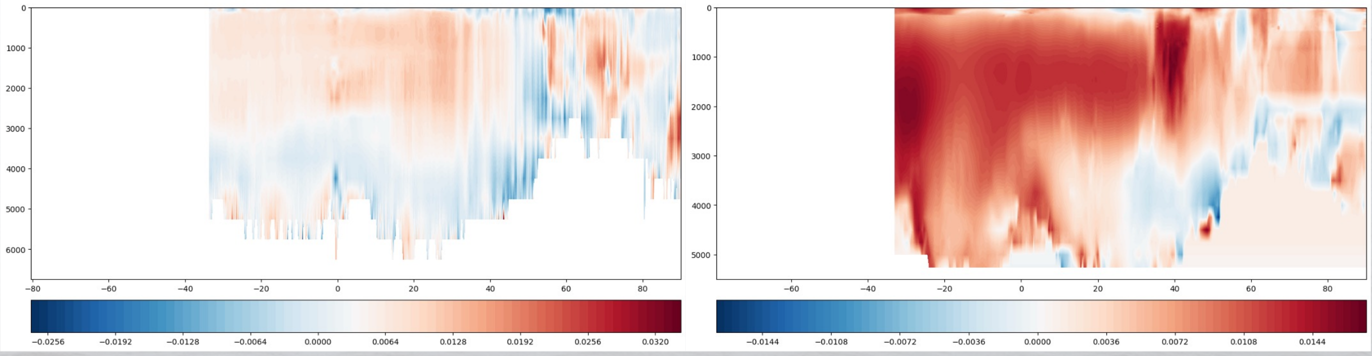
1. Dynamic tensor decompositions
2. Online codes for tensor decompositions

Next - Round out and publish results:

1. Compare modes of CMIP6 AMOC to untangle AMOC strength differences
2. Compare unstructured mesh modes versus traditional EOFs using new mode comparison tool

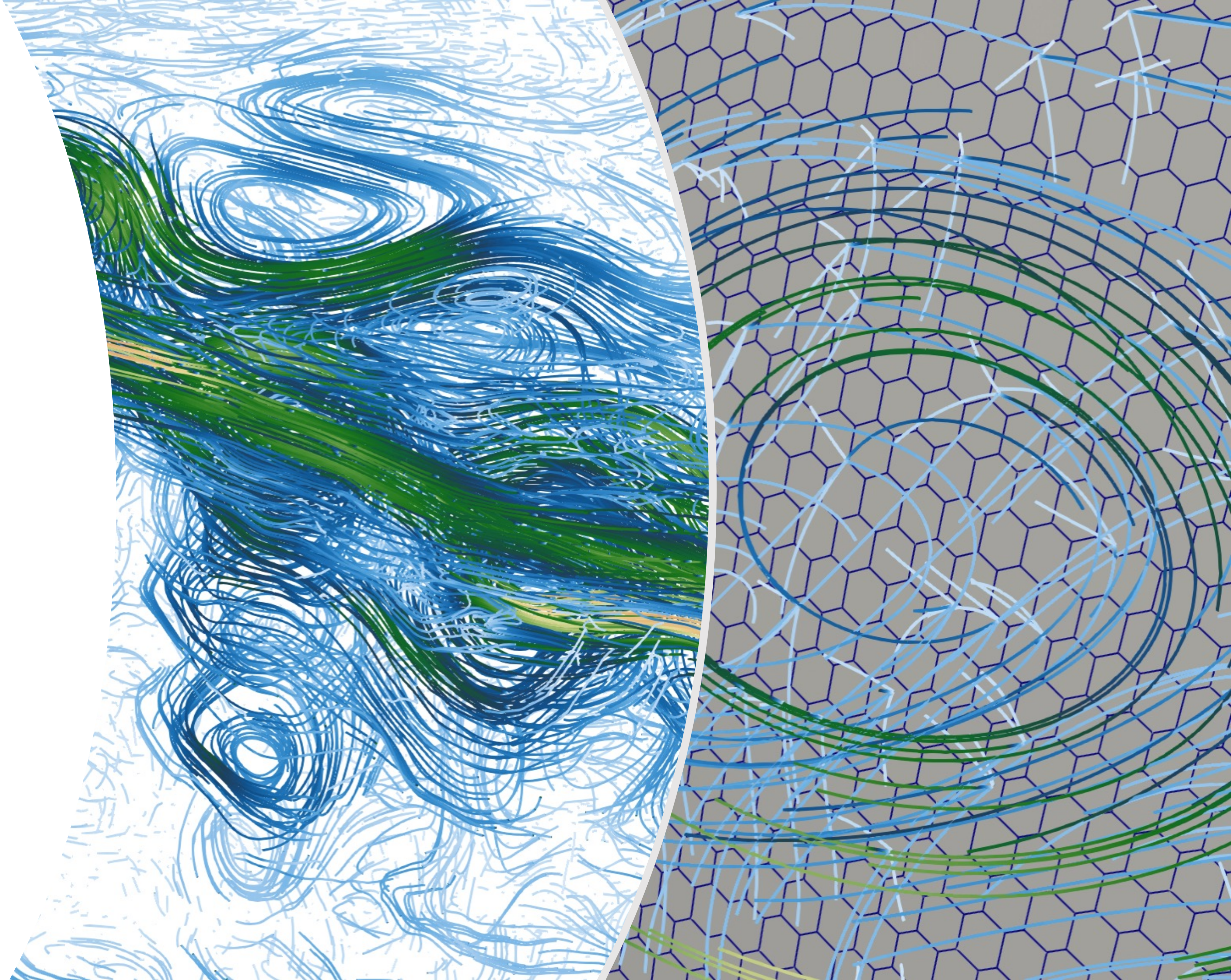


Leading EOF from GFDL and CESM2



Maximal Covariance Analysis of Leading EOF

Parallel Lagrangian Particle Tracking



Particle Tracking

Goal: To enable a long-term particle tracking capability for ocean models that is scalable to exascale resources and runs efficiently on emerging architectures.

Plan: Multiple tracks

- Create a GPU enabled particle tracking code based on a well established ASCR developed particle tracking code.
  - Add ocean specific capabilities (ARGO)
- Exploit a recently developed* reinforcement learning approach to optimize the processor domain decomposition (including on the fly)
  - This new technique has shown to scale well up to more than 16K processors for a domain with $4{,}096^3$.
  - Also improves efficiencies of visualization and analysis

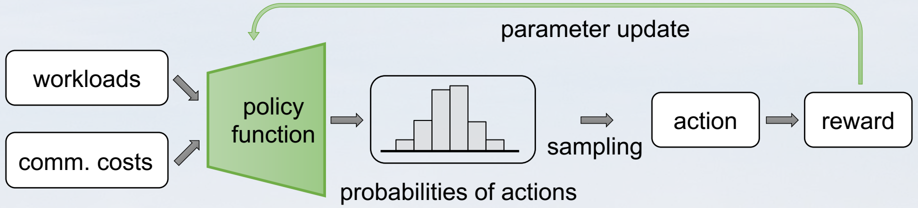


Fig. 1: Decision-making pipeline of a reinforcement learning agent.

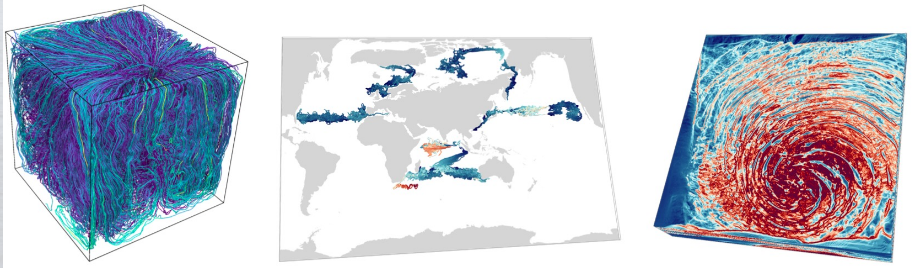
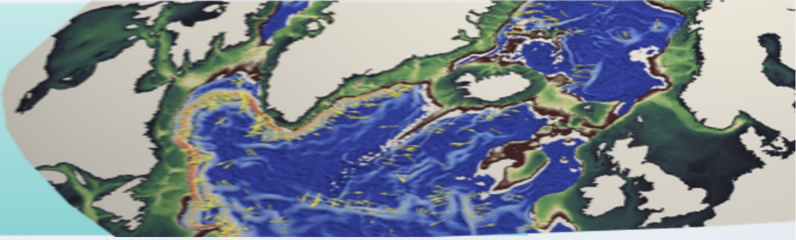


Fig. 2: Flow visualizations.

J. Xu, **H. Guo**, **H.-W. Shen**, M. Raj, S. W. Wurster, **T. Peterka**, *IEEE Transactions on Visualization and Computer Graphics, 2022, Early Access*

*Developed by team members under prior funding

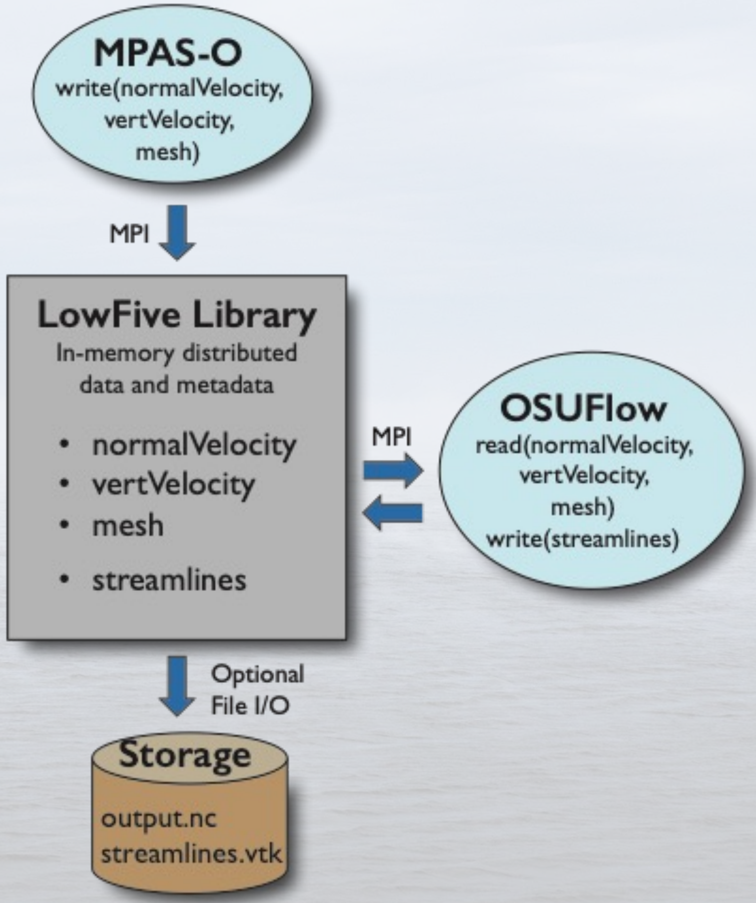
In Situ Workflow Coupling



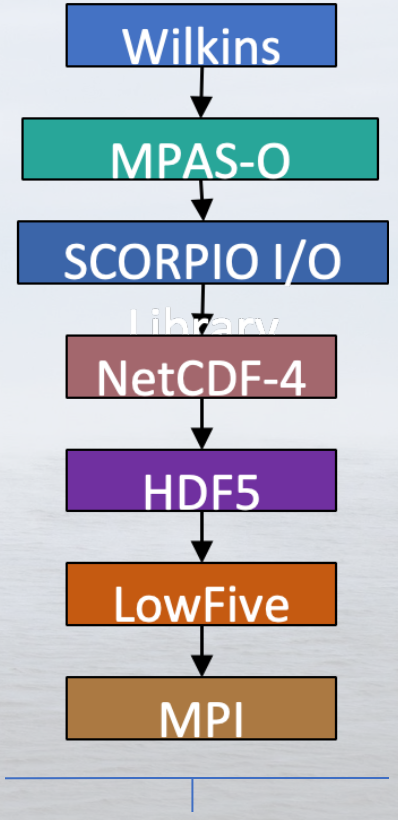
Couple MPAS-O with in situ particle tracing, bypassing storage

Progress to date:

- Custom build of MPAS-O with required dependencies (netCDF, HDF5, parallel netCDF, Scorpio, LowFive)
- Generation of COMPASS test cases (e.g., baroclinic channel) on custom machine and successful execution of custom build of MPAS-O
- In situ coupling of MPAS-O with test code that prints variables (in progress)



In situ coupling of MPAS-O with particle tracing through LowFive



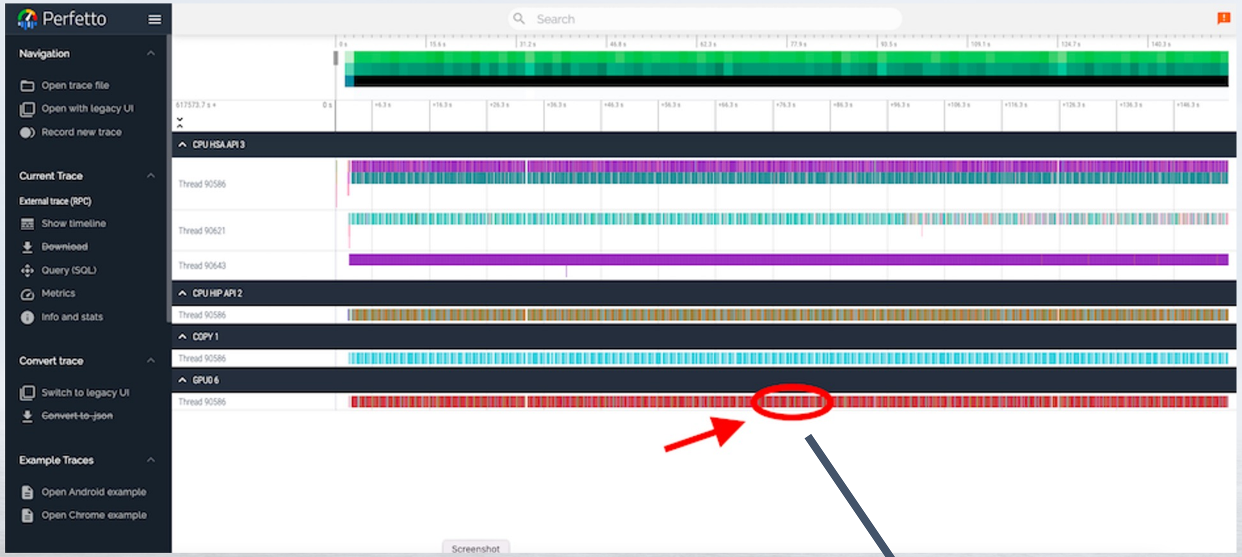
Software stack installed with Spack

Performance

- Working in collaboration with E3SM/ECP and vendors (Cray, Nvidia)
- GPU performance of standalone MPAS Ocean on Crusher is about 7 times slower than CPU performance, while GPU performance is higher on other machines.
- Test configuration (QU240 ocean)
  - Crusher(ORNL), 1 AMD EPYC 7A53 CPU and 1 AMD MI250X GPU
  - Cray compiler cce/15.0.0, AMD rocm/5.1.0
- Initial Performance Measurement

Version	Arch	Total time(sec)	Time integration (sec)	Se tracer tend(sec)	Se implicit vert mix(sec)
Initial	CPU	21.08	16.14	4.22	2.89
	GPU	155.31	141.92	95.30	25.04

GPU Performance Analysis

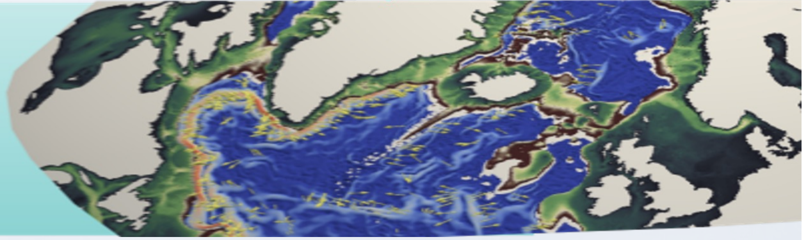


- The three MPAS-Ocean kernels dominates the timeline view.
- Especially "ocn_tracer_advection_mono_tend" kernel takes the longest execution time.

Zoom-in

=> Investigate why the three GPU kernels are expensive.

Performance Optimization



- Further performance analysis revealed…
  - three temporary arrays(wgtTmp, sgnTmp, flxTmp) in “ocn_tracer_advection_mono_tend” kernel create “private” copies of them per every GPU threads(2986) in a OpenACC gang.
- Optimization direction: removing OpenACC private arrays

original

```
!$acc    private(i, k, icell, cell1, cell2, coef1, coef3, &
!$acc            wgtTmp, sgnTmp, flxTmp, tracerWeight)

…

do k = minLevelCell(iCell), maxLevelCell(iCell)
       flxTmp(k) = flxTmp(k) + tracerCur(k,iCell)* &
       wgtTmp(k)*(coef1 + coef3*sgnTmp(k))
end do ! k loop
```

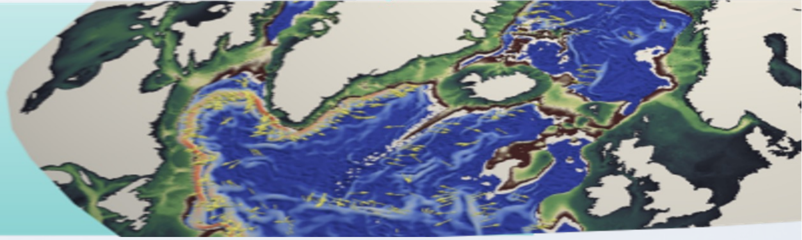
optimized

```
!$acc    private(i, k, icell, cell1, cell2, coef1, coef3, &
!$acc            tracerWeight)

…

do k = minLevelCell(iCell), maxLevelCell(iCell)
   highOrderFlx(k,iEdge) = highOrderFlx(k,iEdge) + tracerCur(k,iCell)* &
    (normalThicknessFlux(k,iEdge)* advMaskHighOrder(k,iEdge))* &
     (coef1 + coef3*sign(1.0_RKIND, normalThicknessFlux(k,iEdge)))
end do ! k loop
```

Speed-up and Future work



Version	Arch	Total time(sec)	Time integration (sec)	Se tracer tend(sec)	Se implicit vert mix(sec)
Initial	CPU	21.08	16.14	4.22	2.89
	GPU	155.31	141.92	95.30	25.04
Optimized	CPU	21.14	16.15	4.17	2.89
	GPU	58.65	52.41	6.29	25.52

- Two versions are verified as bit-for-bit using CPRNC utility
- Further optimizations
  - Found similar issue at the kernels in “ocn_tracer_vmix_tend_implicit”

Summary

- Progress to date
  - Generated new analysis for E3SM simulations
  - Potential new explanations for weak AMOC in E3SM
  - Initial AI based explorations show promise (especially for injecting high res variability)
  - Driving particle code with E3SM data and porting code to GPU
  - Developing new collaborations
- Next steps
  - Perturbed parameter simulations
  - MPAS-Julia and reduced order PPE
  - Streaming ML
  - Time stepping improvements and GPU refactorization