# Digital Twins for Ocean Robots

OceanPredict24, Paris, France

Gaël FORGET, 2024/11/21

# Definition : Digital Twins

U.S. Committee on Foundational Research Gaps and Future Directions for Digital Twins

A digital twin is a set of virtual information constructs that mimics the structure, context, and behavior of a natural, engineered, or social system (or system-of-systems), is dynamically updated with data from its physical twin, has a predictive capability, and informs decisions that realize value. The bidirectional interaction between the virtual and the physical is central to the digital twin.

The National Academies Press, 2024. https://doi.org/10.17226/26894

# Definition : Ocean Robots

- Wikipedia : A robot is a machine—especially one programmable by a computer—capable of carrying out a complex series of actions automatically. A robot can be guided by an external control device, or the control may be embedded within
- GF : ocean robot can be anything from AUV, ROV, Argo float, Drifter, ... to CTD rosette ... to lab-on-chip device

# The Genesis of DTOR

- earlier work on Argo, MITgcm, ECCO, Julia
- 2020-2021 : covid 😭 ... divest from zoom 😩 ... invest in Julia dev 😀
- 2022 : Symposium on Advances in Ocean Observation
- 2023 : US-CLIVAR presentation
- 2023 : JuliaCon presentation
- 2024 : JuliaCon proceedings paper (under review)

# Physical Twins

## List of supported observing networks + those planned

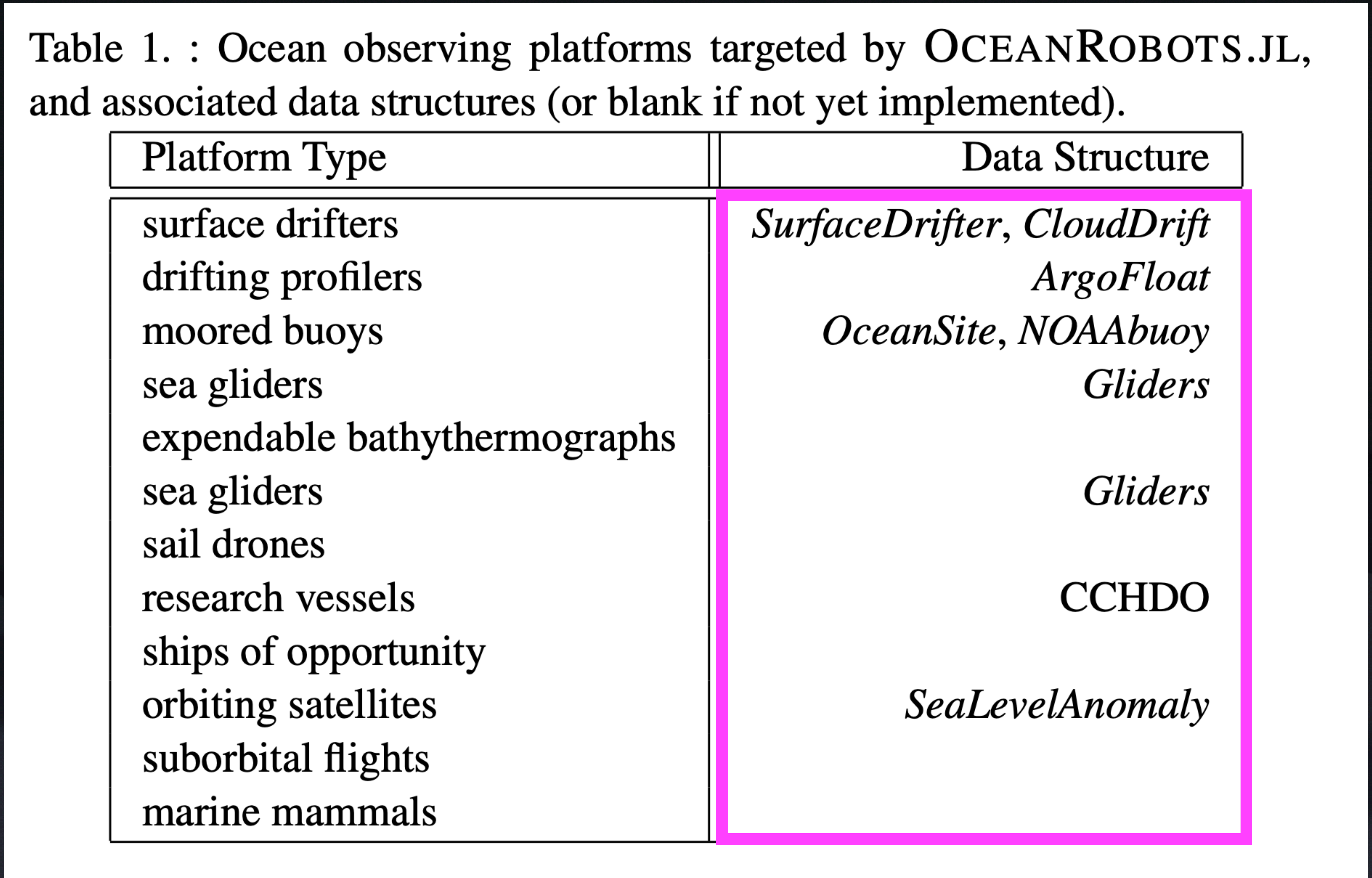
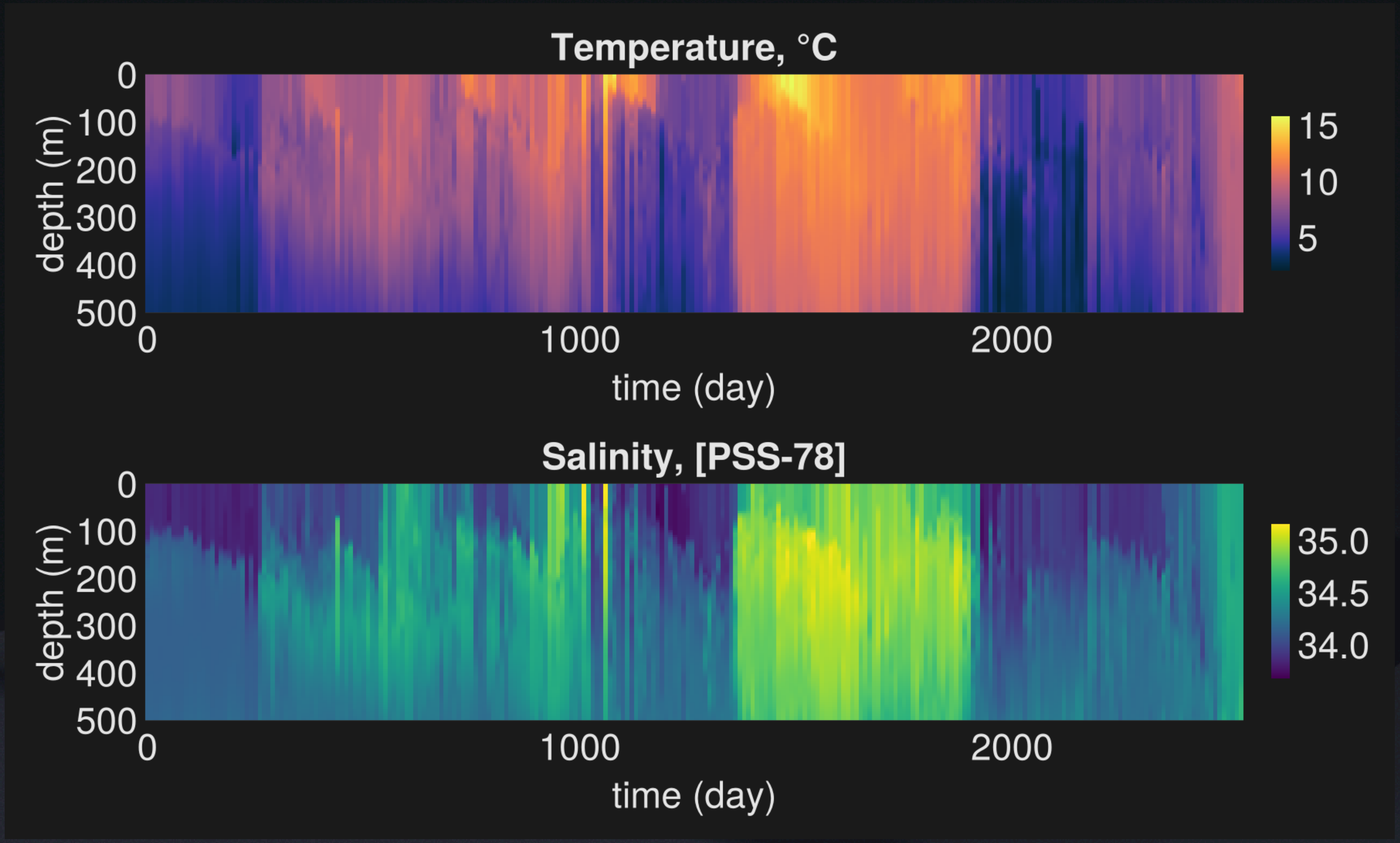


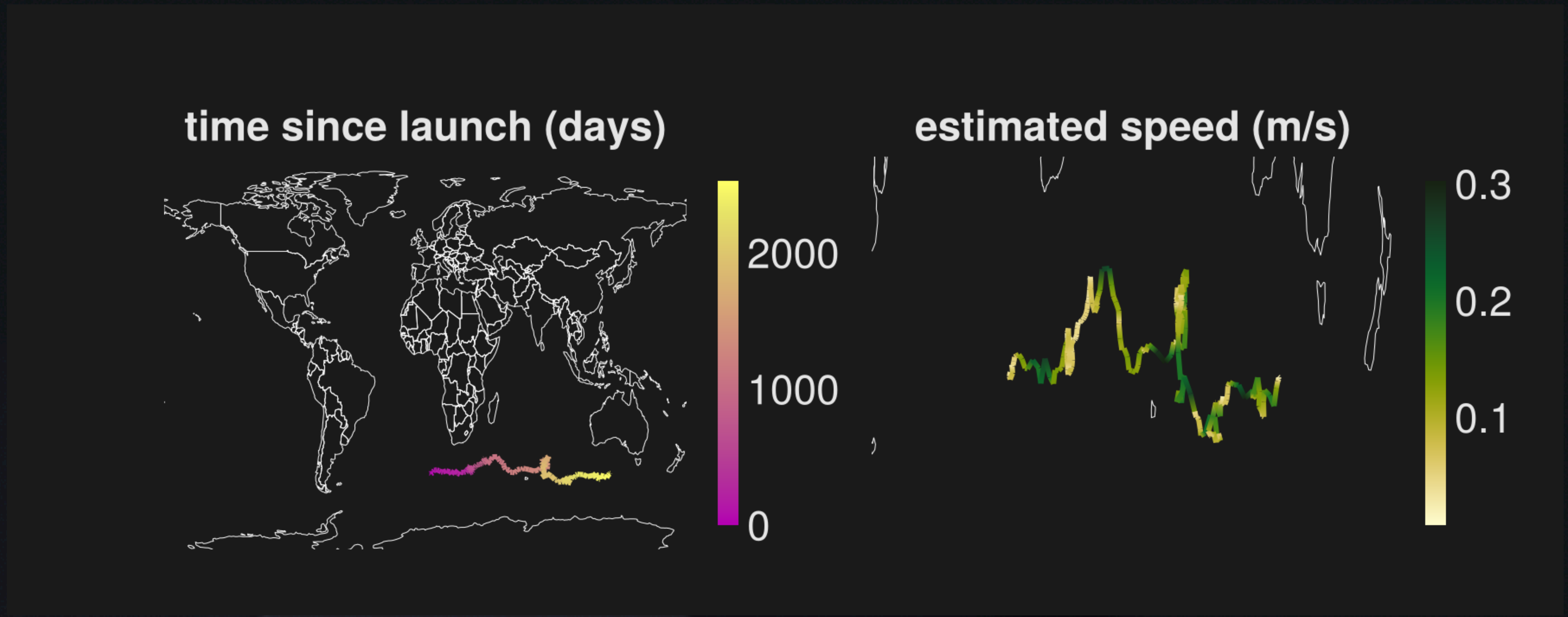
Table 1. : Ocean observing platforms targeted by OceanRobots.jl, and associated data structures (or blank if not yet implemented).

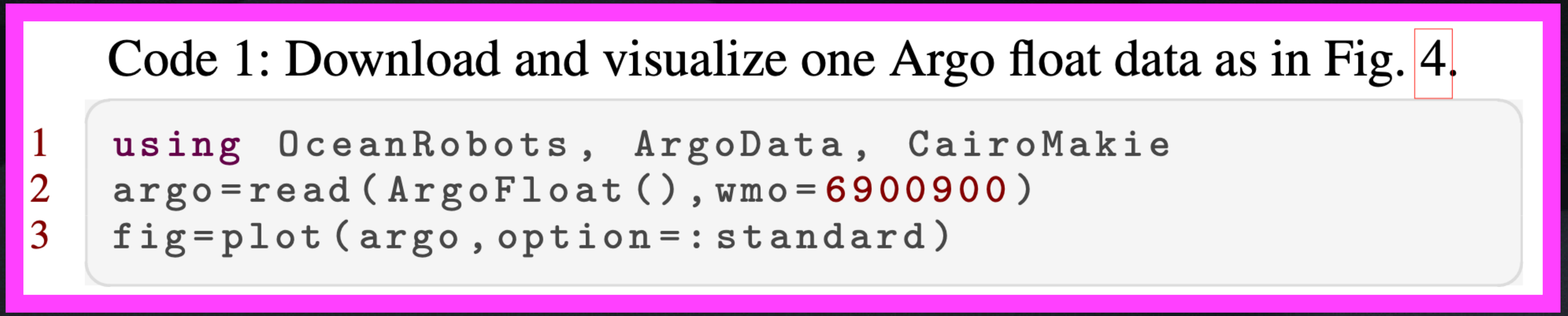
| Platform Type | Data Structure |
|---|---|
| surface drifters | *SurfaceDrifter*, *CloudDrift* |
| drifting profilers | *ArgoFloat* |
| moored buoys | *OceanSite*, *NOAAbuoy* |
| sea gliders | *Gliders* |
| expendable bathythermographs | |
| sea gliders | *Gliders* |
| sail drones | |
| research vessels | CCHDO |
| ships of opportunity | |
| orbiting satellites | *SeaLevelAnomaly* |
| suborbital flights | |
| marine mammals | |

# Example 1 : Argo Floats

## Building a simple Julia interface to observations in DTOR







Code 1: Download and visualize one Argo float data as in Fig. 4.

```
using OceanRobots, ArgoData, CairoMakie
argo=read(ArgoFloat(),wmo=6900900)
fig=plot(argo,option=:standard)
```

# Example 1 : Argo Floats

## Pairing Physical and Virtual Twins

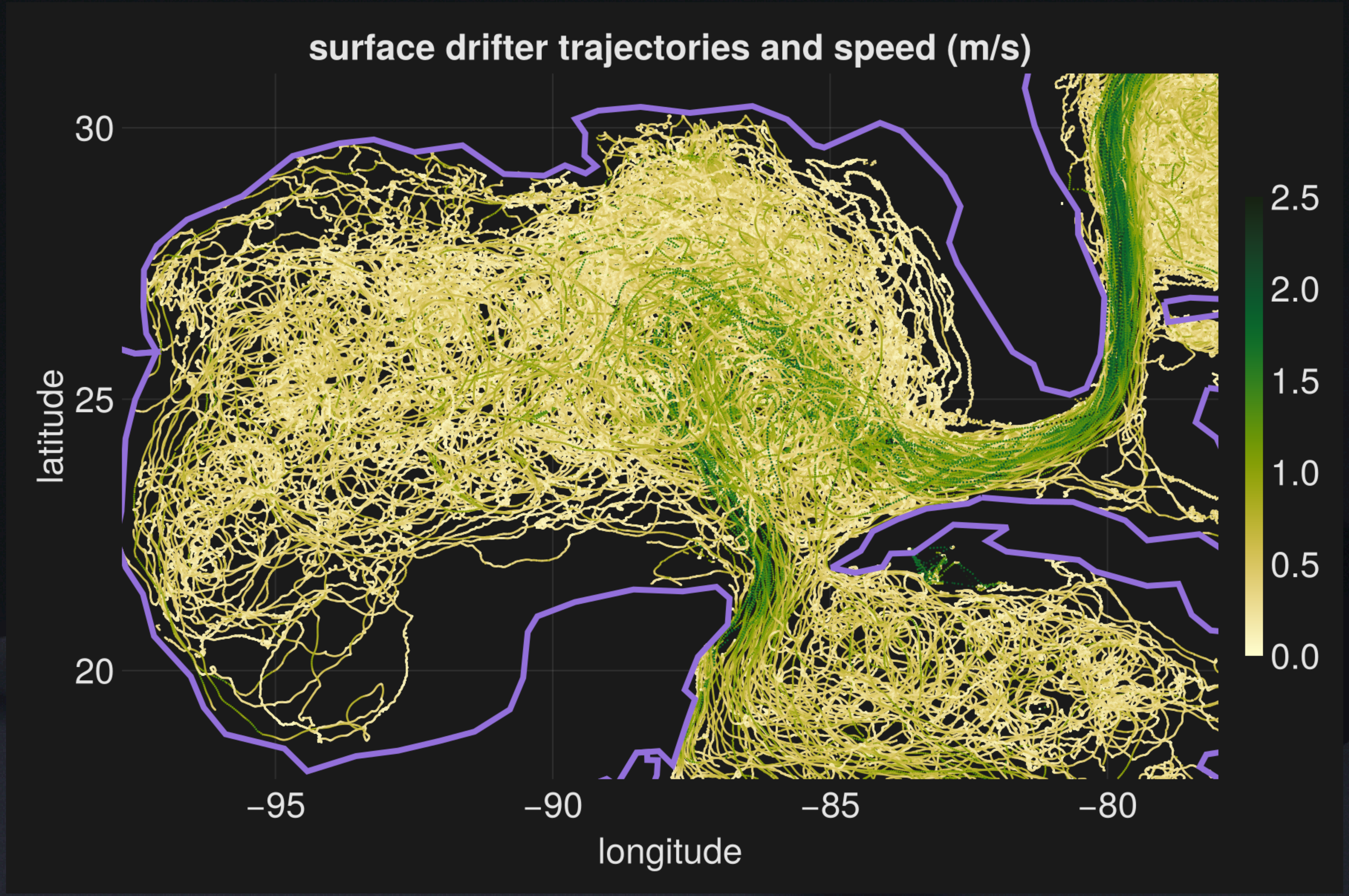
Temperature, °C
depth (m)
0
100
200
300
400
500
0
1000
2000
time (day)
Temperature, °C
depth (m)
15
10
5
Salinity, [PSS-78]
depth (m)
time (day)
Salinity, [PSS-78]
depth (m)
35.0
34.5
34.0

Argo (real world profiles)

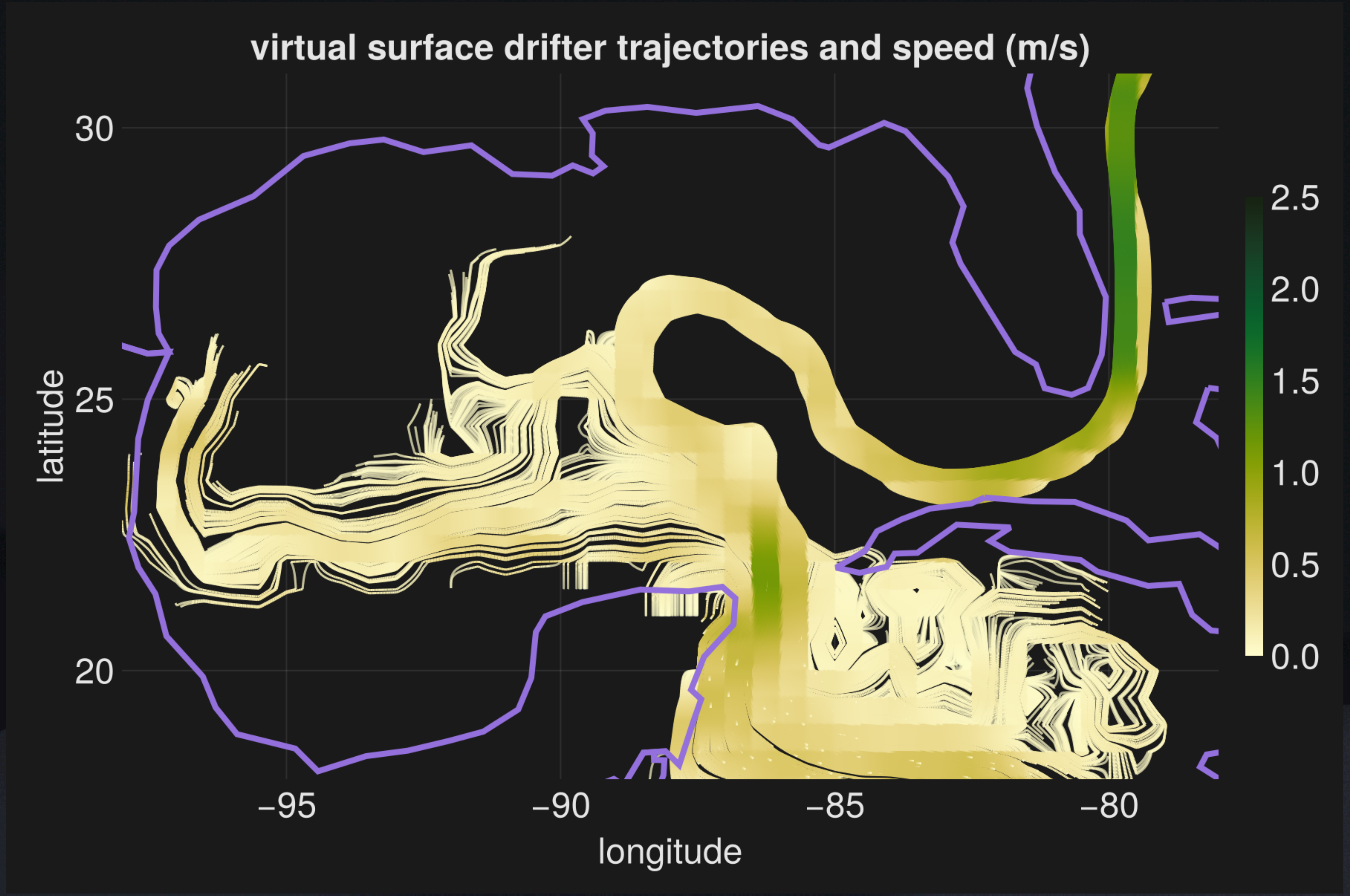
OCCA (emulated profiles)

# Example 2 : Drifters

## Pairing Physical and Virtual Twins



Drifters (real world trajectories)



Drifters.jl (basic emulator)

# Example 3 : BGC-Argo

## Building Emulators & Assessing Performance

- Blue : model truth
- Black : neural emulators
- Software : Flux.jl, MITgcm.jl , Makie.jl

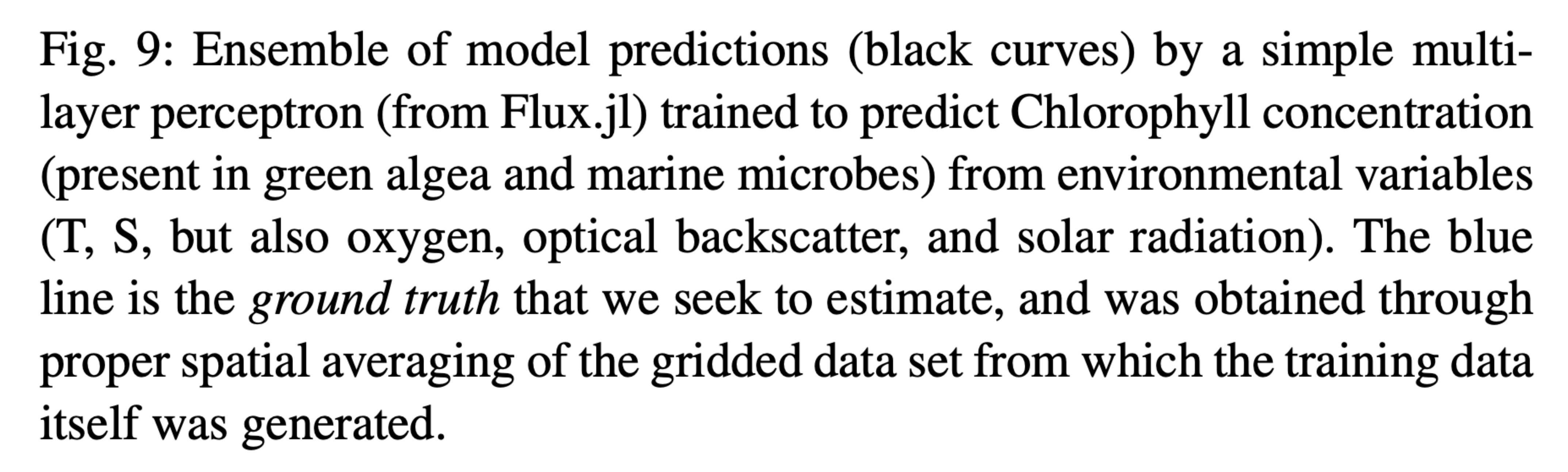
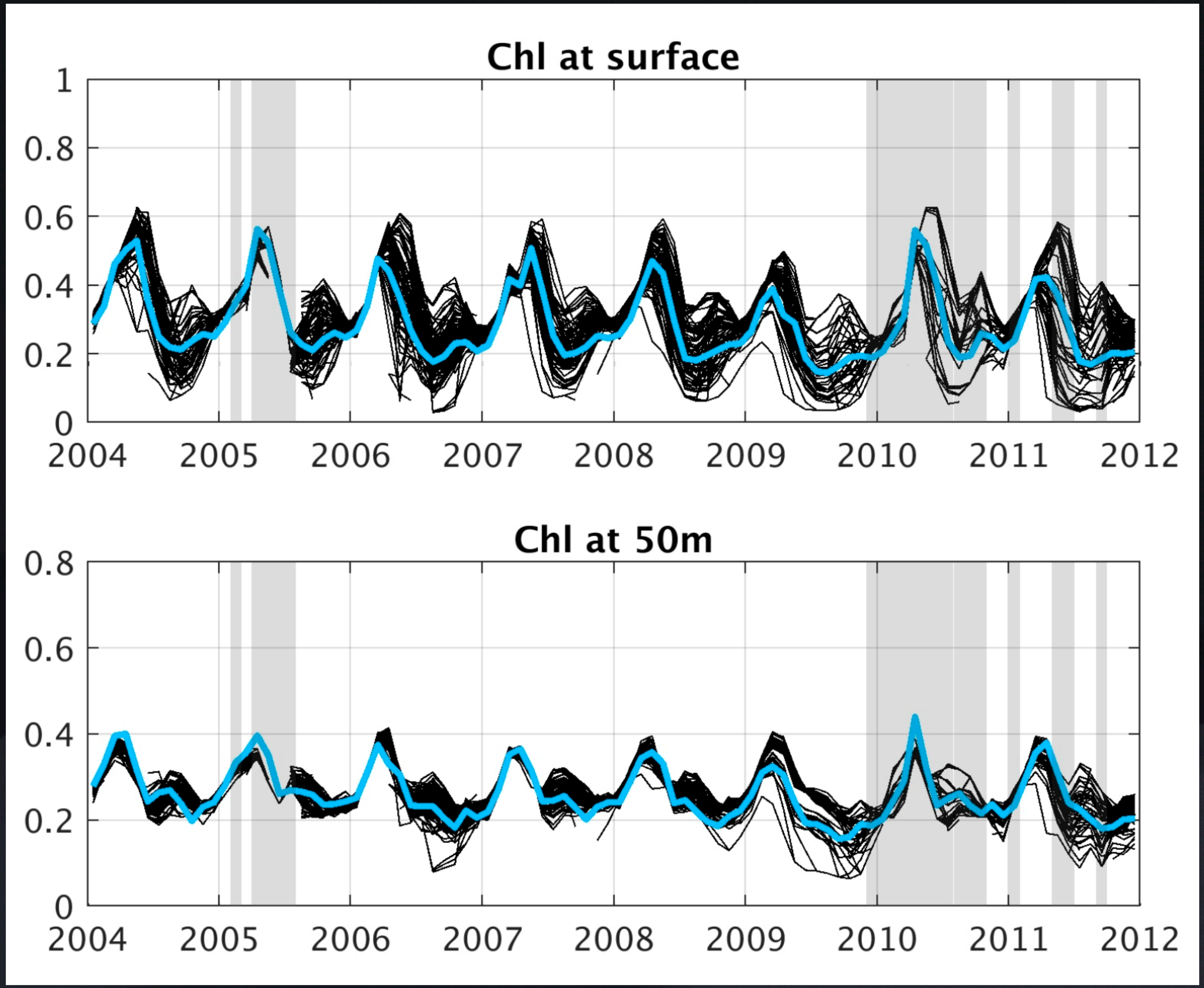
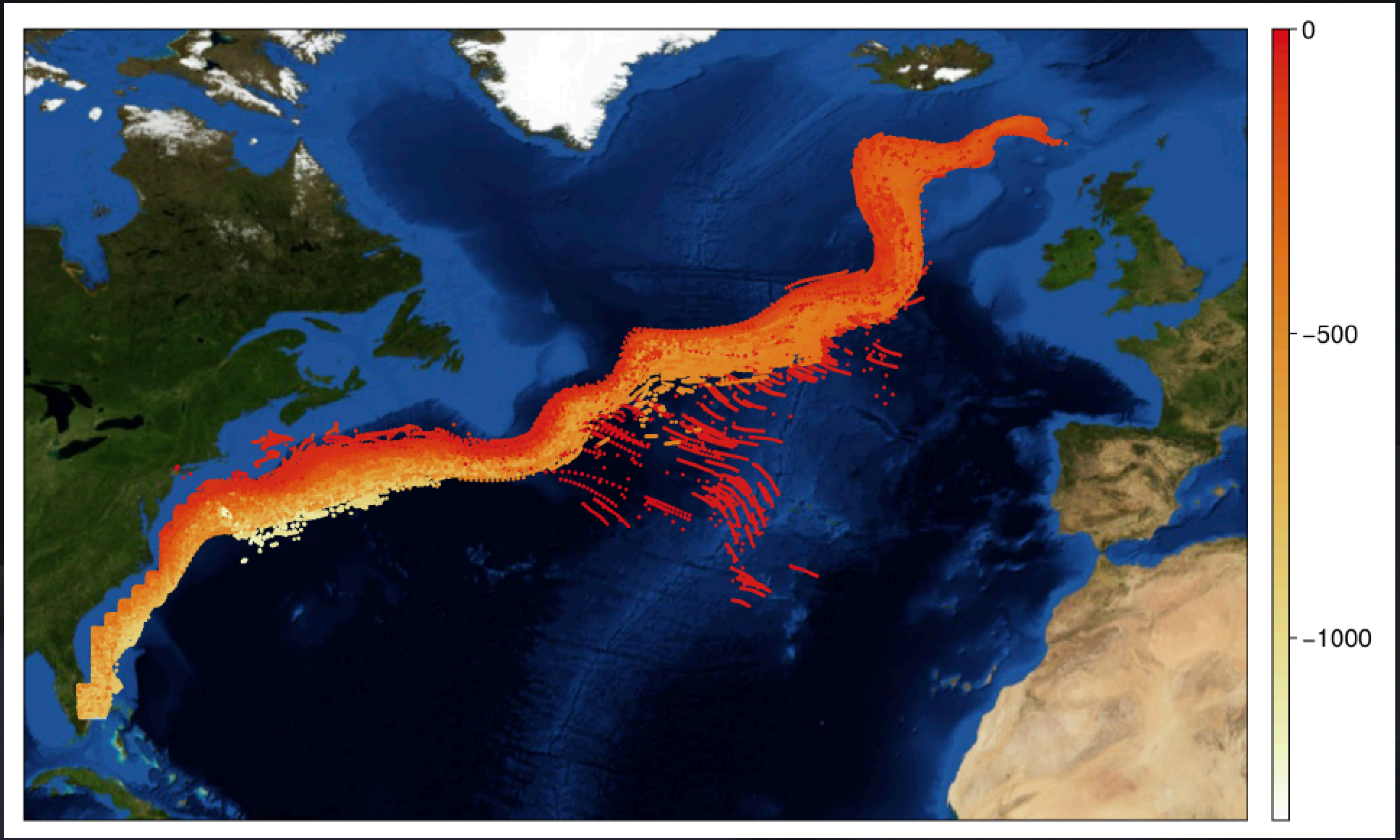


Fig. 9: Ensemble of model predictions (black curves) by a simple multi-layer perceptron (from Flux.jl) trained to predict Chlorophyll concentration (present in green algea and marine microbes) from environmental variables (T, S, but also oxygen, optical backscatter, and solar radiation). The blue line is the *ground truth* that we seek to estimate, and was obtained through proper spatial averaging of the gridded data set from which the training data itself was generated.



# Example 4 : Emulating C-Streams

## Targeting the SPG with drifting buoys



- Climatology.jl
  - three-dimensional monthly flow fields from ECCO4
- Drifters.jl
  - three-dimensional trajectory calculations
- Makie.jl
  - Visualization and Animation

# Example 5 : Emulating Fine-Scales

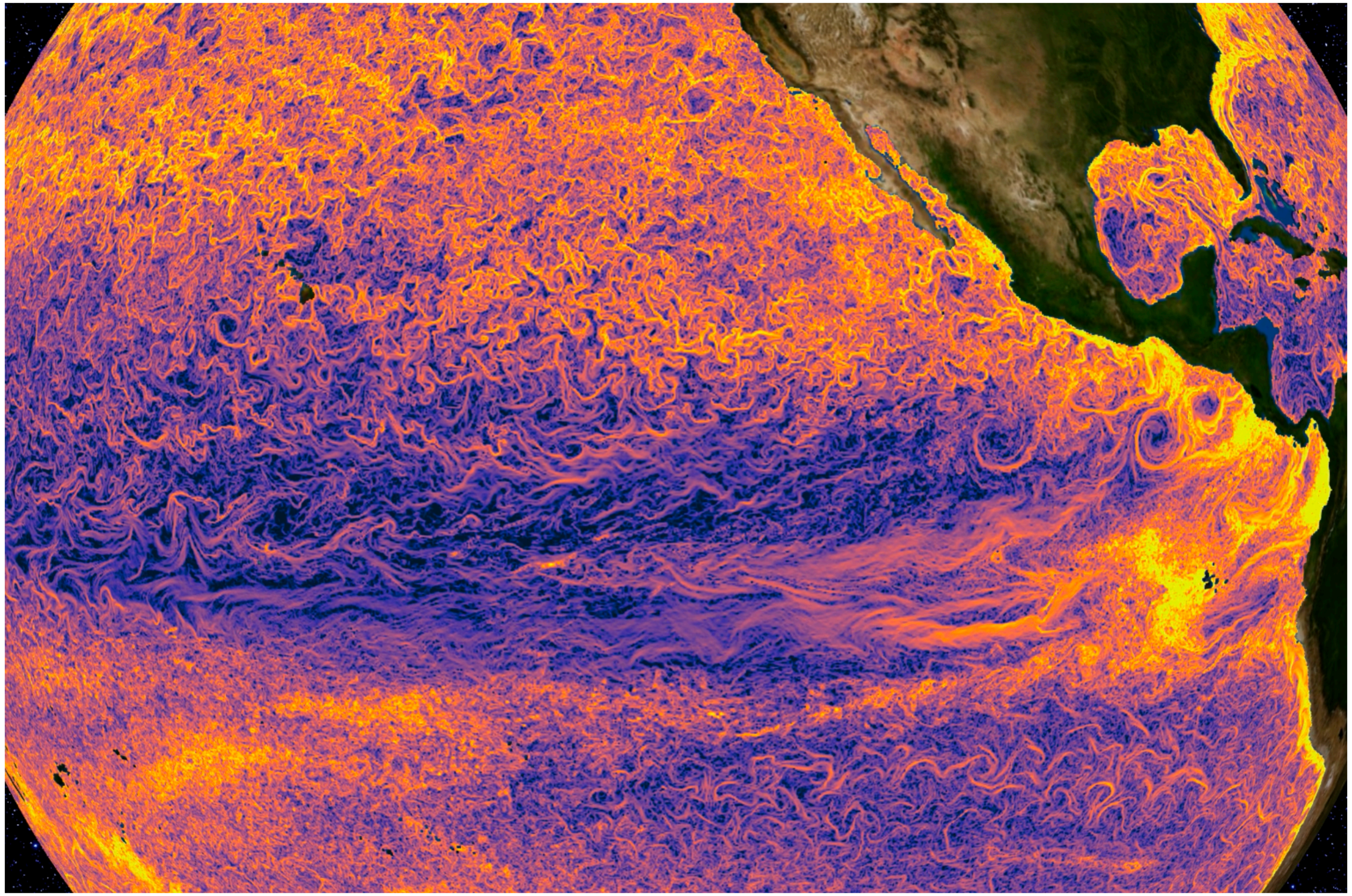


Fig. 8: Temperature fronts in a global km-scale MITgcm simulation. Plotted is the logarithm of the spatial gradient of a temperature snapshot.

- MITgcm
  - Km-scale model run
- MITgcm.jl
  - Handling MITgcm model output
- MeshArrays.jl
  - Interpolation from native grid to regular grid
- Makie.jl
  - Visualization and Animation

# Example 6 : Emulating Future Environments

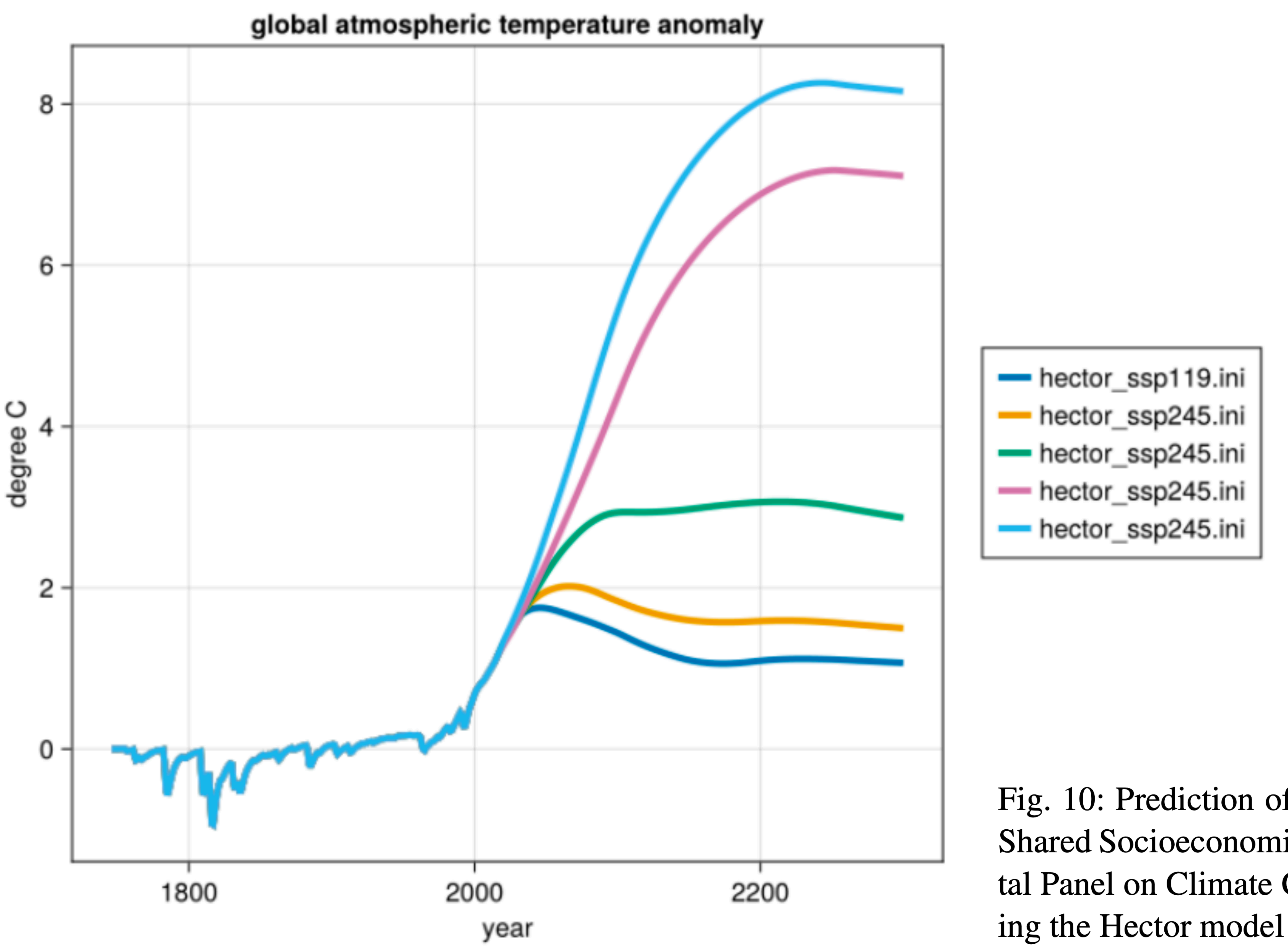
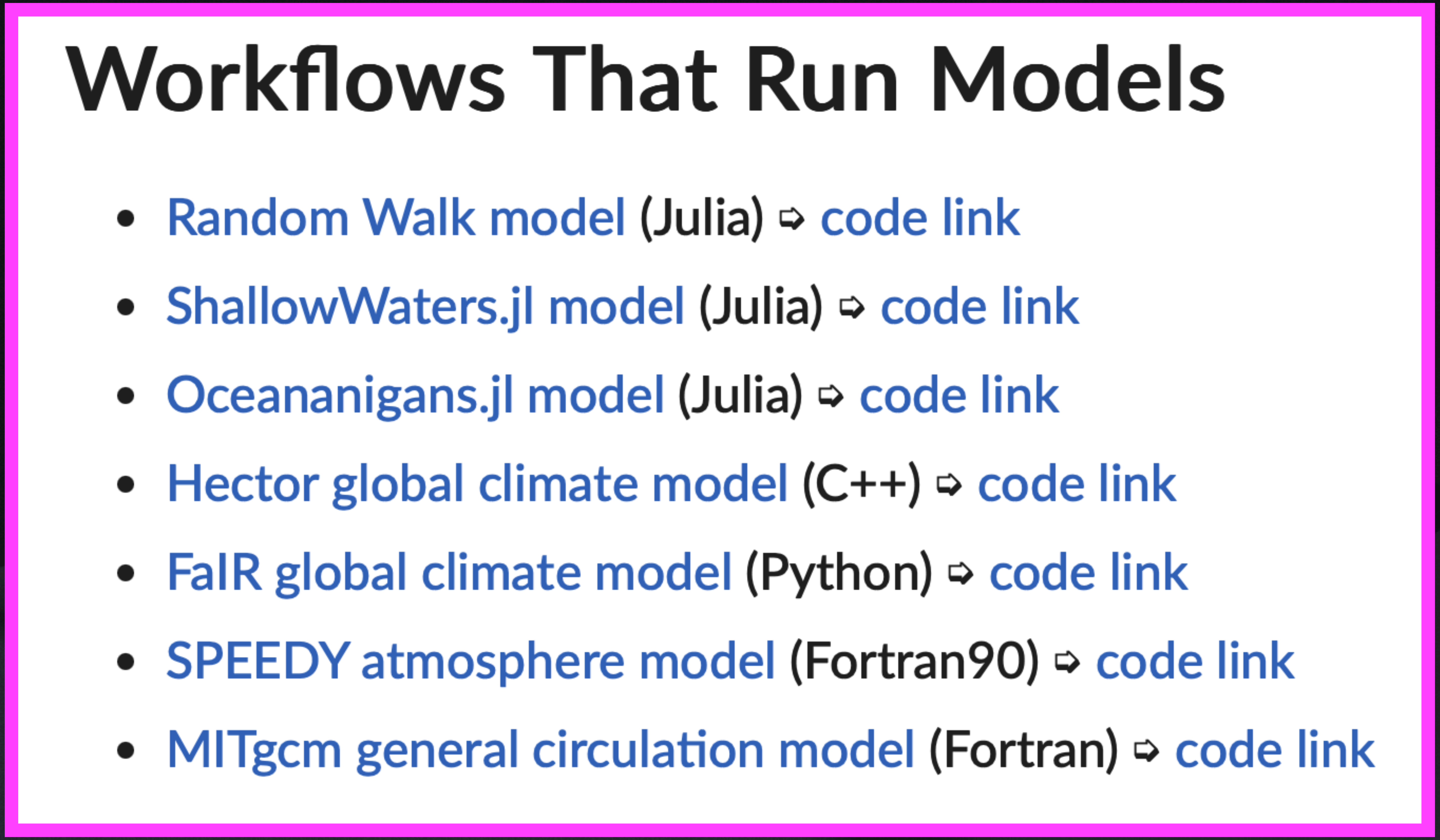


Fig. 10: Prediction of global warming based on different scenarios, called Shared Socioeconomic Pathways (SSPs), as defined by the Intergovernmental Panel on Climate Change (IPCC). These predictions were generated using the Hector model [13] via CLIMATEMODELS.JL.

- ClimateModels.jl
    - Interface to build and run Hector emulator
- Hector
    - Climate scenario emulation
- Makie.jl
    - Visualization and Animation

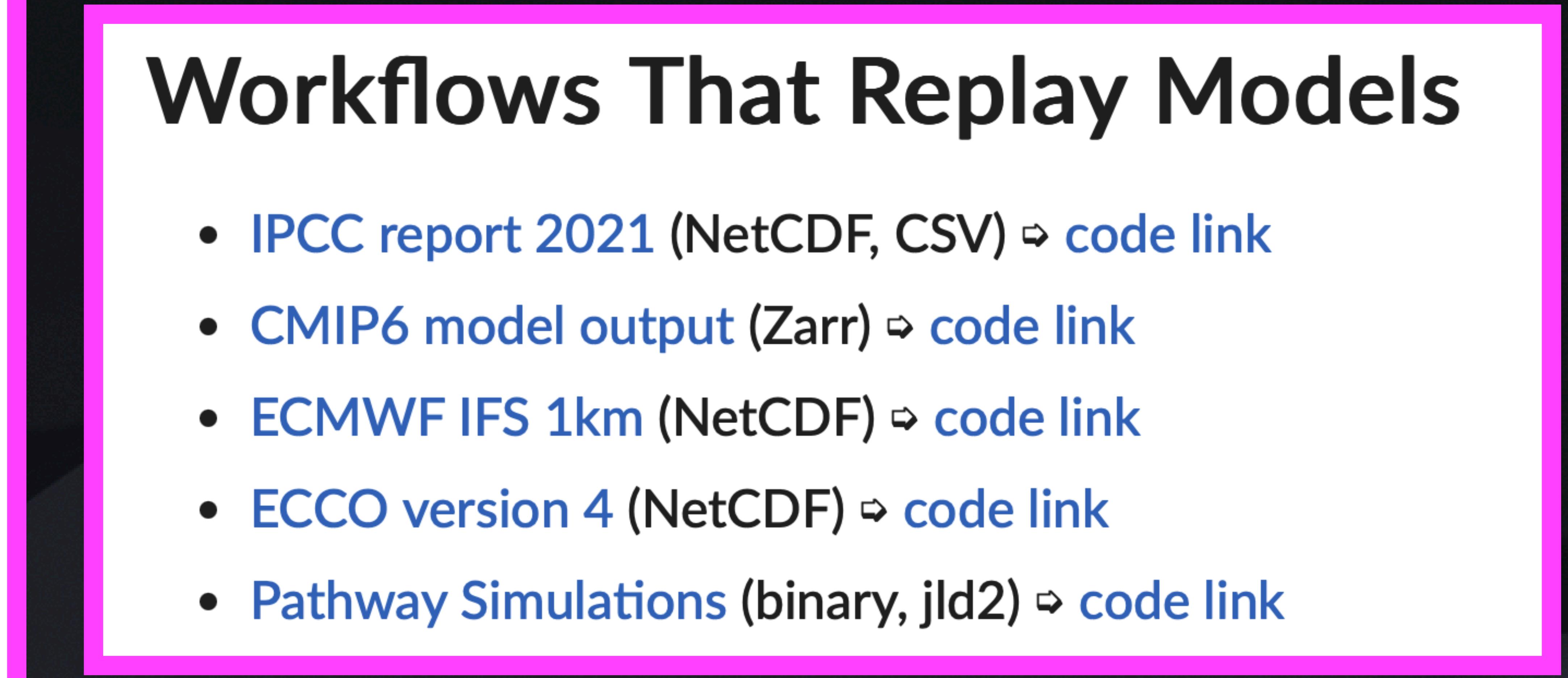
# Mechanistic Models

## A Versatile and Easily Extensible Lineup



### Workflows That Run Models

- Random Walk model (Julia) ➪ code link
- ShallowWaters.jl model (Julia) ➪ code link
- Oceananigans.jl model (Julia) ➪ code link
- Hector global climate model (C++) ➪ code link
- FaIR global climate model (Python) ➪ code link
- SPEEDY atmosphere model (Fortran90) ➪ code link
- MITgcm general circulation model (Fortran) ➪ code link

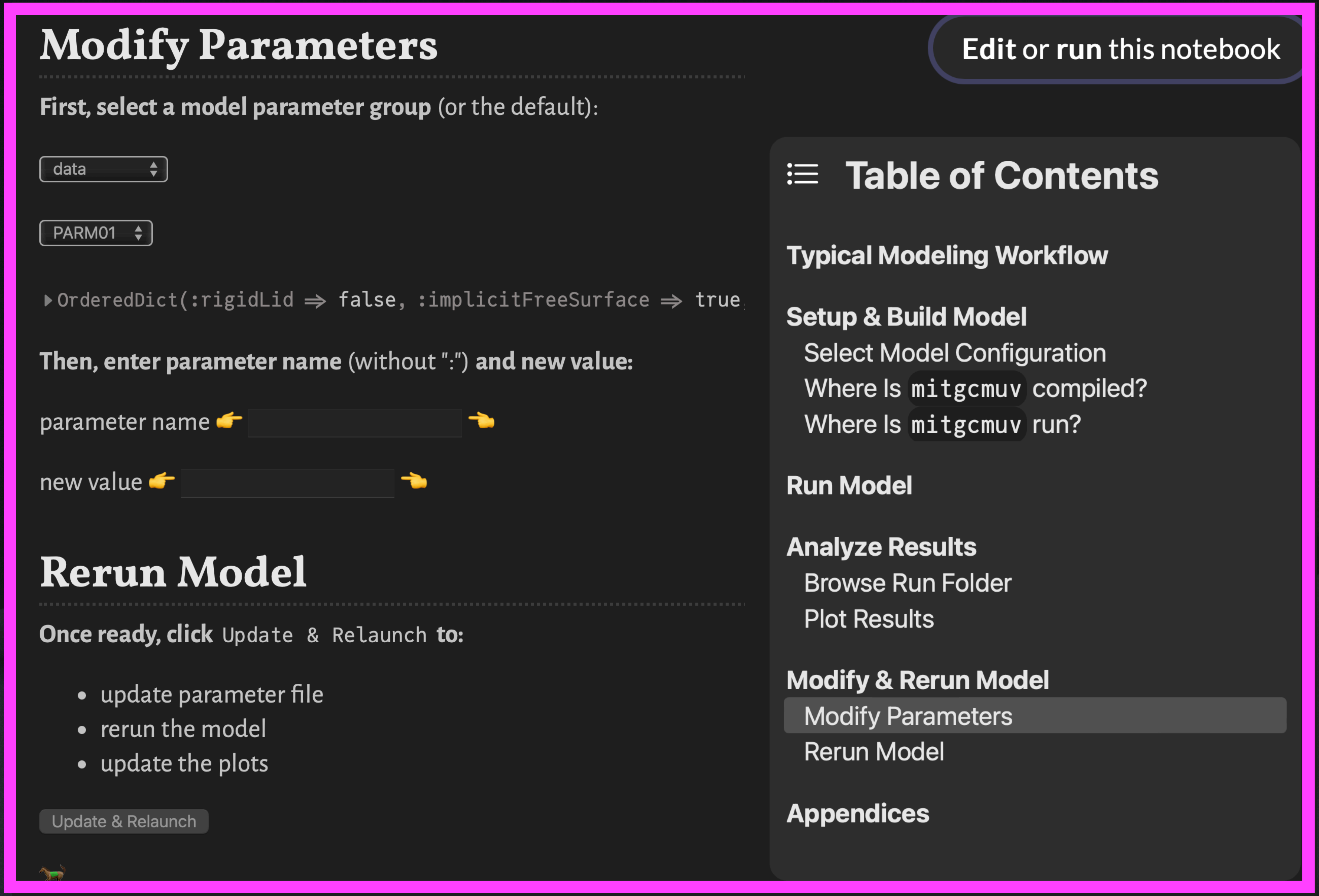


### Workflows That Replay Models

- IPCC report 2021 (NetCDF, CSV) ➪ code link
- CMIP6 model output (Zarr) ➪ code link
- ECMWF IFS 1km (NetCDF) ➪ code link
- ECCO version 4 (NetCDF) ➪ code link
- Pathway Simulations (binary, jld2) ➪ code link

# Interactivity

## Pluto reactive notebook examples



### Modify Parameters

**First, select a model parameter group** (or the default):

data

PARM01

```
▸OrderedDict(:rigidLid ⇒ false, :implicitFreeSurface ⇒ true,
```

**Then, enter parameter name** (without ":") **and new value:**

parameter name 👉 👈

new value 👉 👈

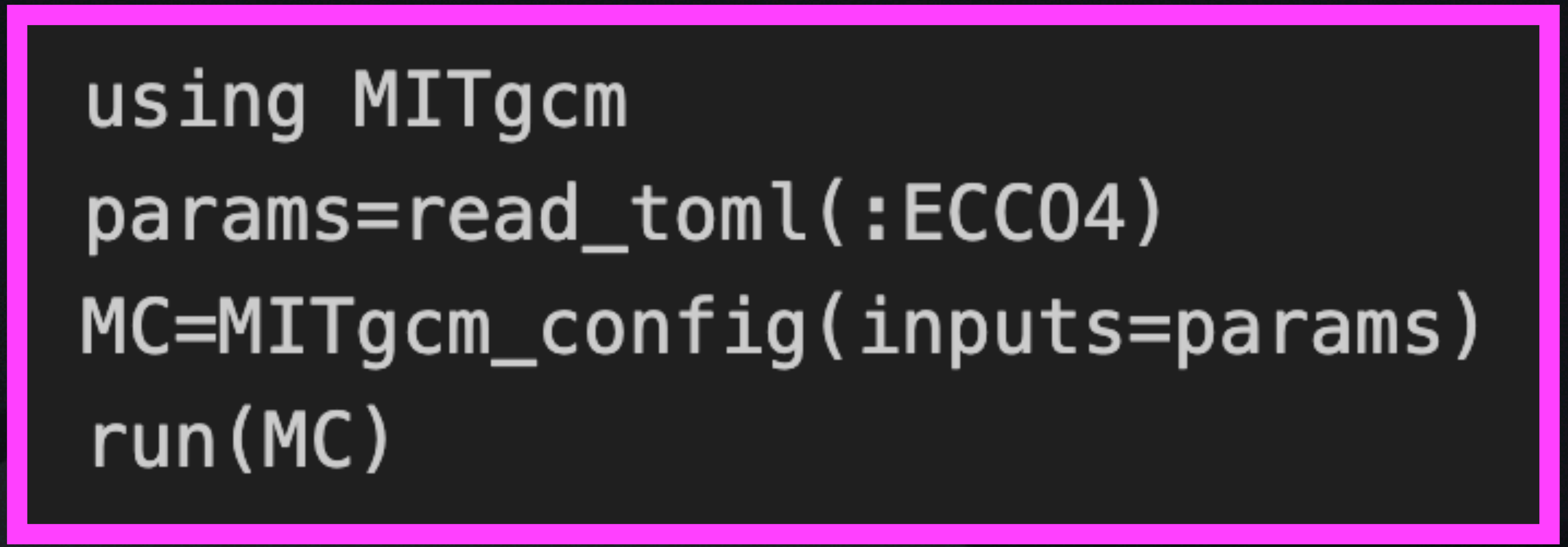
### Rerun Model

**Once ready, click** `Update & Relaunch` **to:**

- update parameter file
- rerun the model
- update the plots

Update & Relaunch

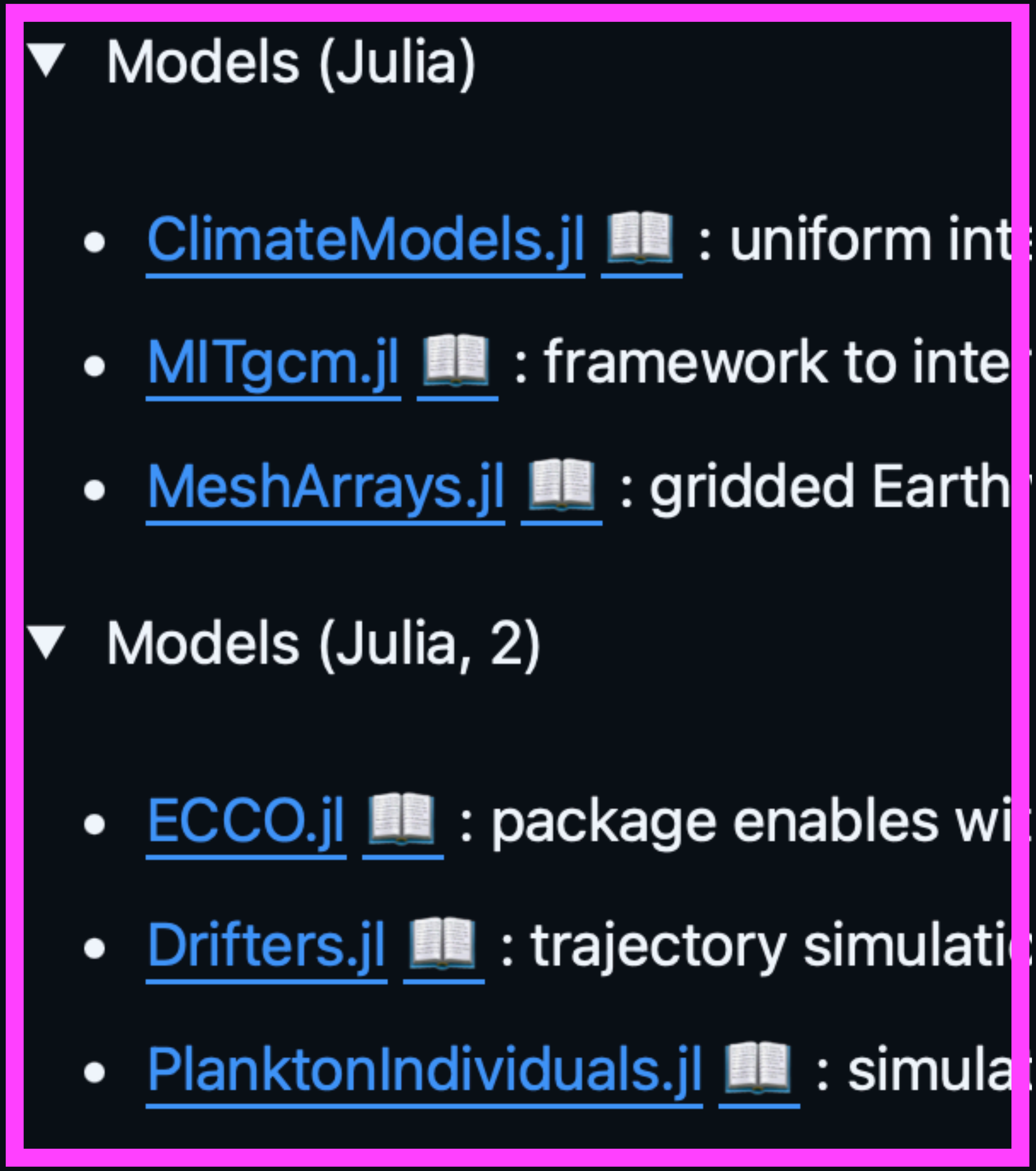
Table of Contents



```
using MITgcm
params=read_toml(:ECCO4)
MC=MITgcm_config(inputs=params)
run(MC)
```

# Open Source Software

https://github.com/gaelforget

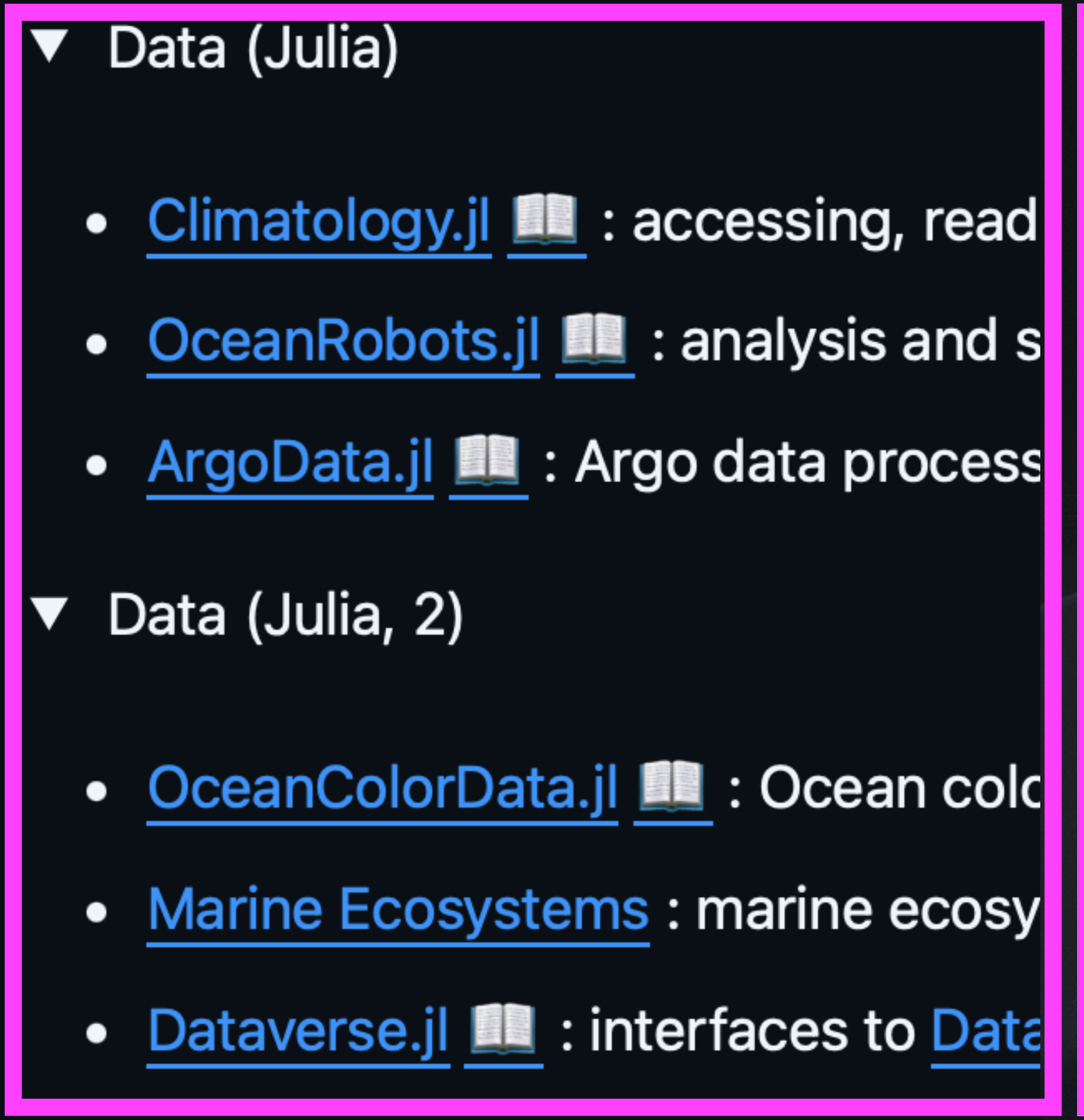


▼ Models (Julia)

- ClimateModels.jl 📖 : uniform int
- MITgcm.jl 📖 : framework to inte
- MeshArrays.jl 📖 : gridded Earth

▼ Models (Julia, 2)

- ECCO.jl 📖 : package enables wi
- Drifters.jl 📖 : trajectory simulati
- PlanktonIndividuals.jl 📖 : simula



▼ Data (Julia)

- Climatology.jl 📖 : accessing, read
- OceanRobots.jl 📖 : analysis and s
- ArgoData.jl 📖 : Argo data process

▼ Data (Julia, 2)

- OceanColorData.jl 📖 : Ocean colo
- Marine Ecosystems : marine ecosy
- Dataverse.jl 📖 : interfaces to Data

# Outlook

- All systems go!
- Next for me ?
  - science, art, education, and commercial applications
  - maintain and continue building up DTOR *(BGC-Argo, NEMO, ODIS, cloud, ...)*
  - build up on the AI/DA/ML/UQ side of DTOR *(Enzyme.jl, Flux.jl, Turing.jl, SciML, ...)*
- Next for you ?
  - Ignore 😎, Benefit 😎, Contribute 😎
  - https://github.com/gaelforget
  - JuliaEO25 *(6-10 January 2025)*