Australian
National University

A data-driven approach for developing and calibrating a parameterization of the ocean mesoscale eddy fluxes

Australian Government
Australian Research Council

Navid Constantinou

Remark: Not to be confused with Van Gogh's "Starry Night"

KITP, ML for Climate
November 1st, 2021

Visualization using output from the MIT/JPO project
Estimating the Circulation and Climate of the Ocean, Phase II (ECCO2)

Credit: NASA/Goddard Space Flight Center Scientific Visualization Studio

special thanks go to my pals at MIT

Gregory L. Wagner
glwagner

Oceans + Julia

122 followers · 76 following · 109

Massachusetts Institute of Technology
Salt Lake City, Utah
gregory.leclaire.wagner@gmail.com
glwagner.github.io

Adeline Hillier
adelinehillier

6 followers · 10 following · 7

also Raffaele Ferrari, Andre Souza, Xiaozhou Ruan (MIT)
and Stephen Griffies (GFDL)

ocean currents modelled at different horizontal resolutions

(why ocean eddies give headaches to climate scientists?)


typically used
for climate predictions
IPCC, etc…

state-of-the-art
ocean—sea-ice model

[ACCESS-OM2 ocean—sea-ice models, Kiss et al., Geosci. Model Dev. 2020]

ocean currents modelled at different horizontal resolutions

(why ocean eddies give headaches to climate scientists?)

typically used
for climate predictions
IPCC, etc…

state-of-the-art
ocean—sea-ice model

[ACCESS-OM2 ocean—sea-ice models,
Kiss et al., Geosci. Model Dev. 2020]

ocean currents modelled at different horizontal resolutions

typically used
for climate predictions
IPCC, etc…

state-of-the-art
ocean—sea-ice model

[Gogh, V., *MoMA*, 1889]

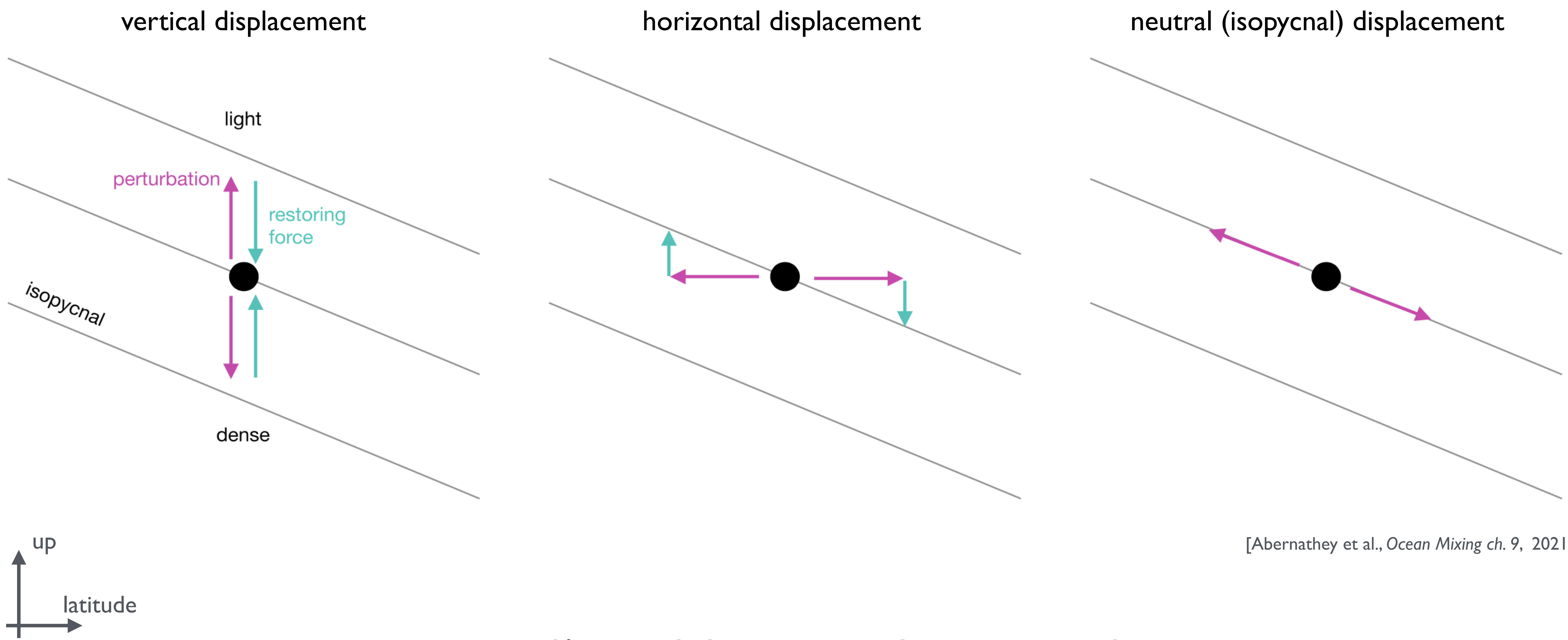
can we make the coarse model feel the effect of the flow details that it does not resolve?

[in technical terms: "eddy parameterisation"]

we don't need to know what each eddy is doing!
we care for the low-order, long-time statistics of the system
(climate *vs* weather)

a small primer on how eddies
affect the large-scales

eddies move tracers along neutral directions



[Abernathey et al., *Ocean Mixing ch. 9*, 2021]

isopycnal/neutral direction = along isopycnal
diapycnal = across isopycnal (costs potential energy)

how eddies affect tracers?

😳 (few equations hardly ever hurt)

tracer (e.g. heat) dynamics

$$\frac{\partial c}{\partial t} + \boldsymbol{u} \cdot \nabla c = \kappa \nabla^2 c$$

$$c = \bar{c} + c'$$

resolved unresolved

how eddies affect tracers?

😳 (few equations hardly ever hurt)

tracer (e.g. heat) dynamics

$$\frac{\partial c}{\partial t} + \boldsymbol{u} \cdot \nabla c = \kappa \nabla^2 c$$

$$c = \underbrace{\overline{c}}_{\text{resolved}} + \underbrace{c'}_{\text{unresolved}}$$

$$\underbrace{\frac{\partial \overline{c}}{\partial t} + \overline{\boldsymbol{u}} \cdot \nabla \overline{c} = \kappa \nabla^2 \overline{c}}_{\text{the dynamics the model solves for}} \underbrace{- \nabla \cdot \left(\overline{\boldsymbol{u}' c'} \right)}_{\text{subgrid eddy fluxes}}$$

$\overline{\boldsymbol{u}' c'}$ eddy tracer flux

parametrization

express **eddy tracer flux** in terms of the **resolved fields**

$$\overline{\boldsymbol{u}'c'} \quad = \quad \mathscr{F}(\overline{\boldsymbol{u}}, \overline{c}, \ldots)$$

eddy tracer flux | eddy tracer flux parametrization

isoneutral diffusion

eddies mix tracers, therefore $\overline{\boldsymbol{u}'c'} \approx -\kappa_{\text{eddy}} \nabla \overline{c}$ $\implies -\nabla \cdot \left(\overline{\boldsymbol{u}'c'}\right) = \kappa_{\text{eddy}} \nabla^2 \overline{c}$

downgradient flux

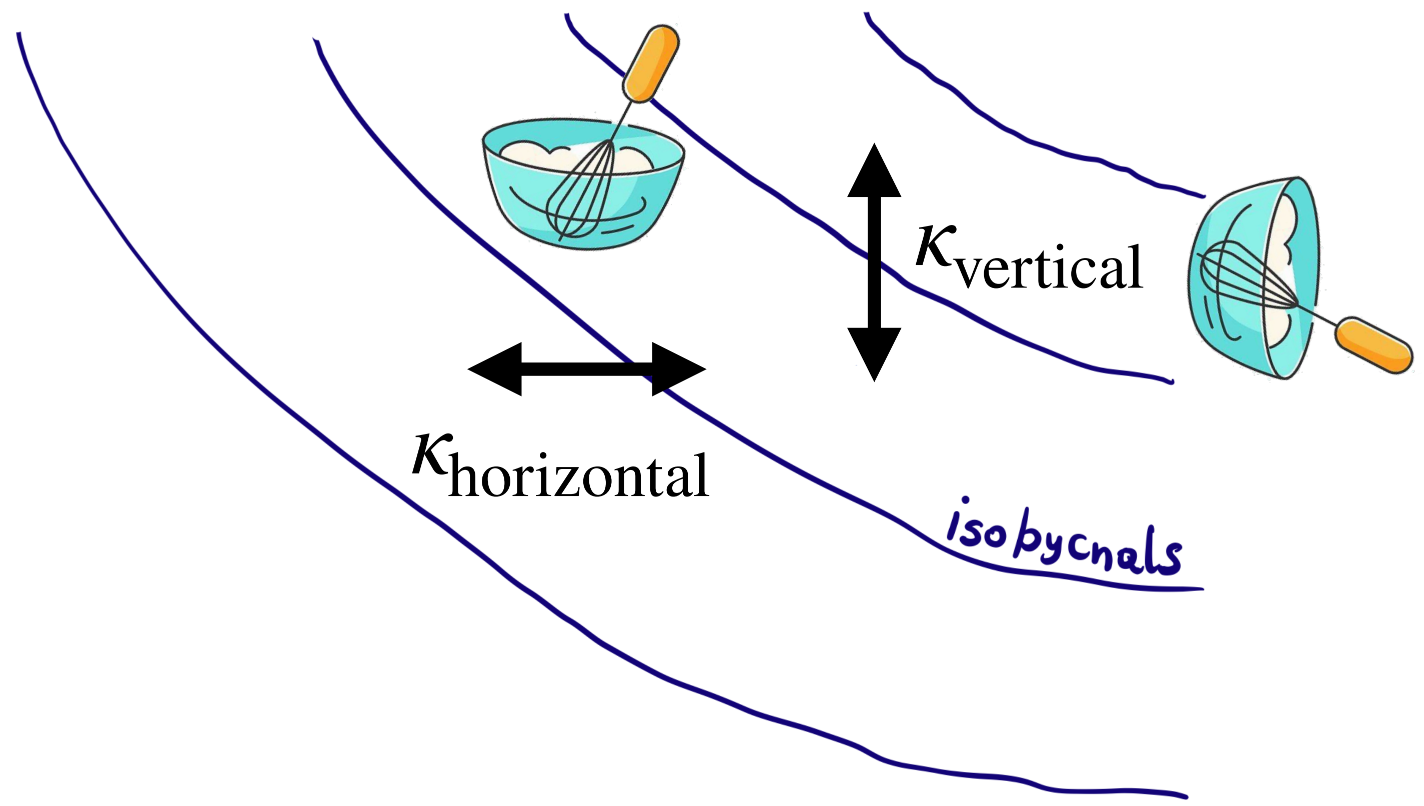
"eddy diffusivity"

isoneutral diffusion

eddies mix tracers, therefore $$\overline{\boldsymbol{u}'c'} \approx -\kappa_{\text{eddy}} \nabla \overline{c}$$ $$\implies -\nabla \cdot \left(\overline{\boldsymbol{u}'c'}\right) = \kappa_{\text{eddy}} \nabla^2 \overline{c}$$

downgradient flux

"eddy diffusivity"



$$\overline{\boldsymbol{u}'c'} \approx -\begin{pmatrix} \kappa_{\text{h}} & 0 & 0 \\ 0 & \kappa_{\text{h}} & 0 \\ 0 & 0 & \kappa_{\text{v}} \end{pmatrix} \cdot \nabla \overline{c}$$

anisotropic downgradient flux

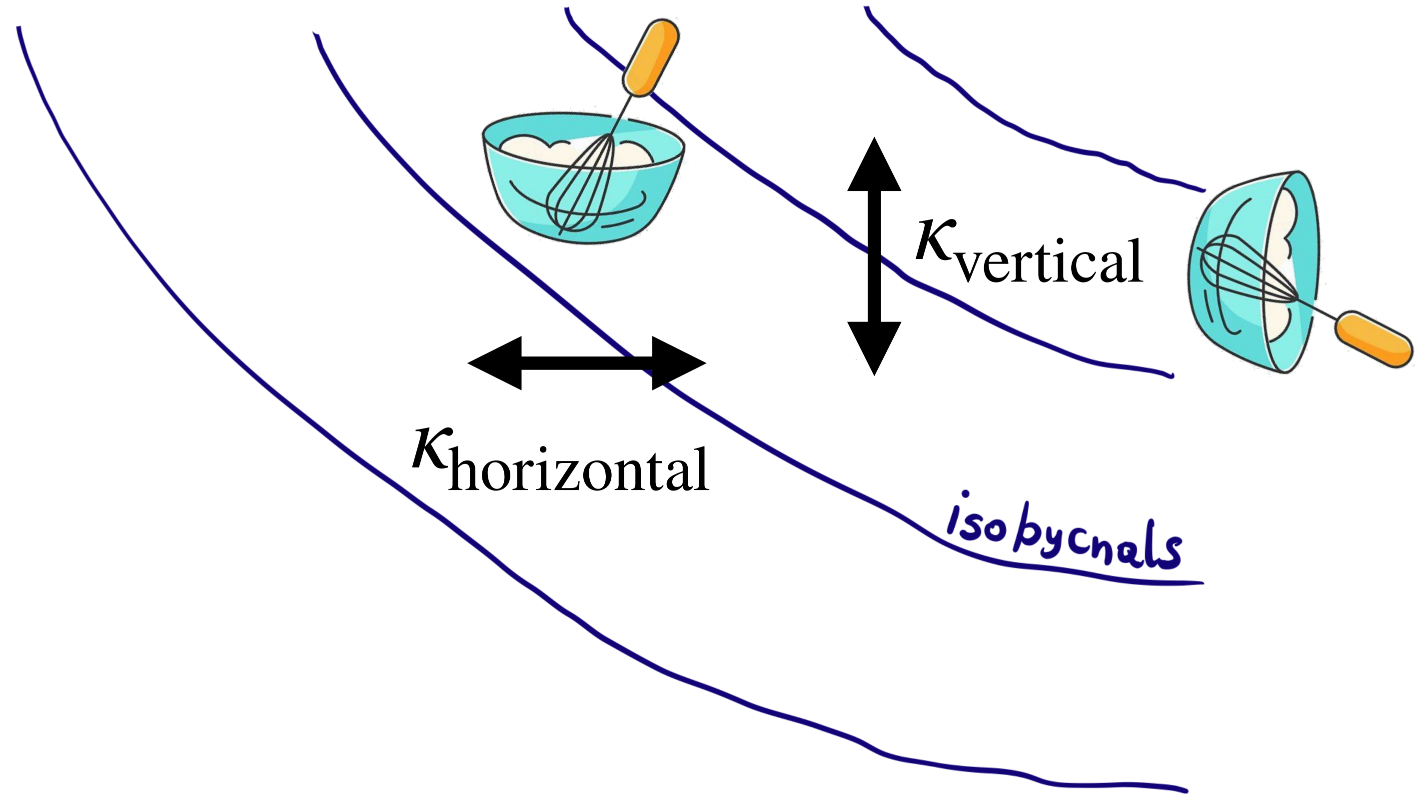
isoneutral diffusion

eddies mix tracers, therefore

$$\overline{\boldsymbol{u}'c'} \approx -\kappa_{\text{eddy}}\nabla\bar{c} \quad\Longrightarrow\quad -\nabla\cdot\left(\overline{\boldsymbol{u}'c'}\right) = \kappa_{\text{eddy}}\nabla^2\bar{c}$$

downgradient flux

"eddy diffusivity"



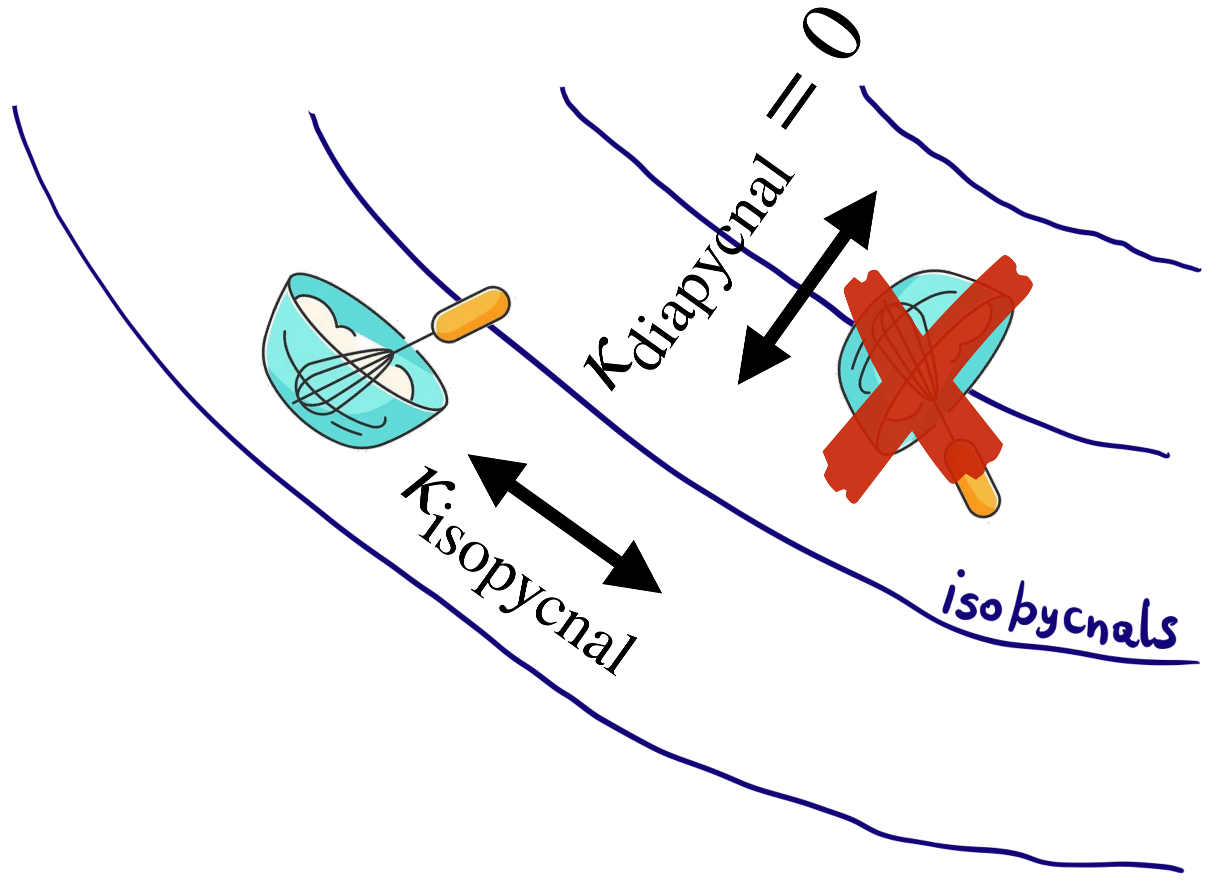
$$\overline{\boldsymbol{u}'c'} \approx -\begin{pmatrix}\kappa_{\text{h}} & 0 & 0\\ 0 & \kappa_{\text{h}} & 0\\ 0 & 0 & \kappa_{\text{v}}\end{pmatrix}\cdot\nabla\bar{c}$$

anisotropic
downgradient flux

$$\overline{\boldsymbol{u}'c'} \approx -\mathbb{K}_{\text{eddy}}\cdot\nabla\bar{c}$$

$\mathbb{K}_{\text{eddy}}$ a 3x3 tensor
that rotates coords
to neutral-cross neutral
directions

downgradient flux
locally aligned with
neutral direction



isoneutral diffusion

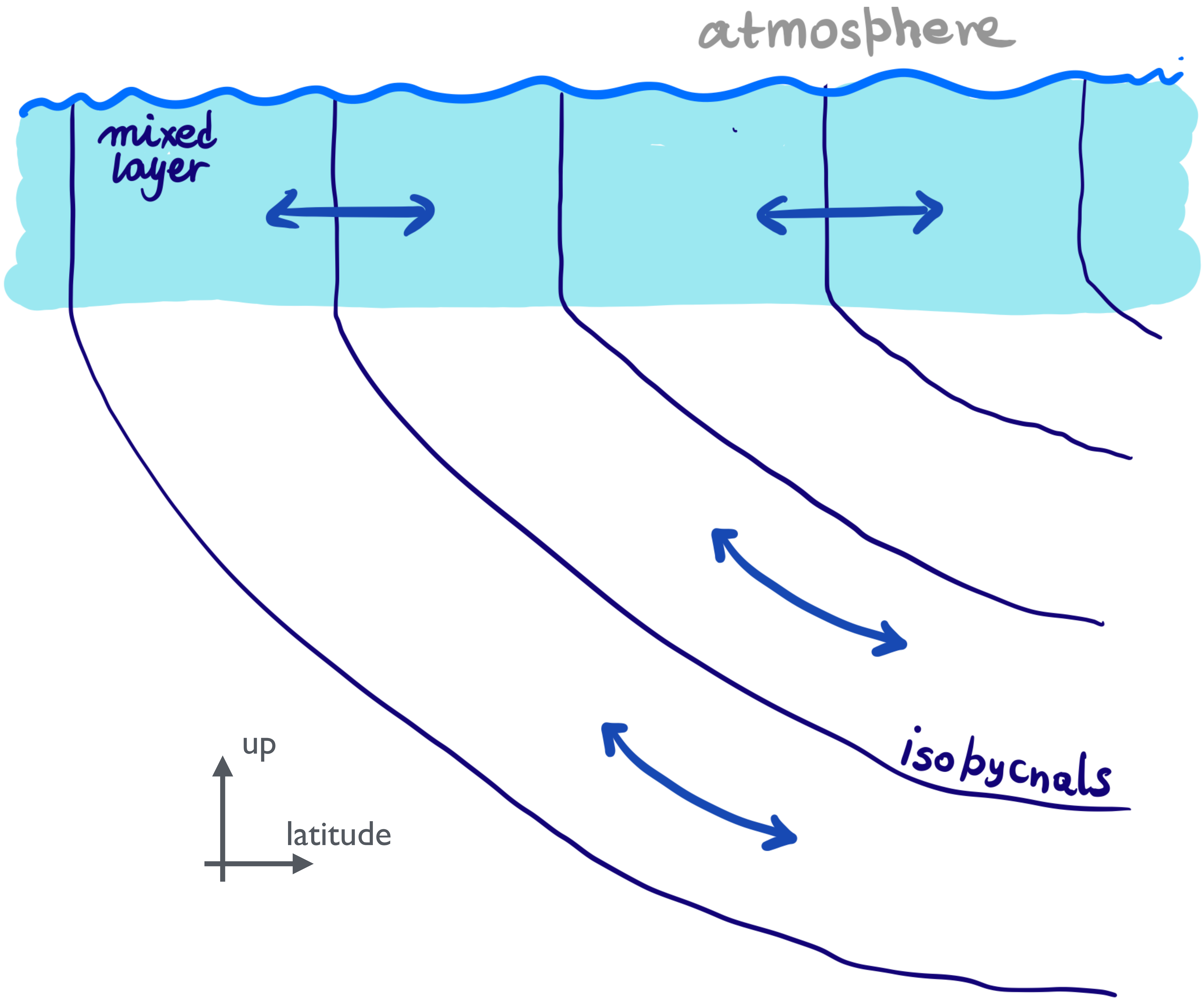
$$\overline{\boldsymbol{u}'c'} \approx -\left(\mathbb{K}_{\mathrm{GM}} + \mathbb{K}_{\mathrm{Redi}}\right) \cdot \nabla \overline{c}$$

skew flux modeling stirring along isopycnals

tracer diffusion along isopycnals

[Reddi 1982, Gent and McWilliams 1990, Griffies 1998, Griffies et al 1998]

it was all fun and games until…



towards the surface dominant dynamics change

mesoscale parametrization should "convert" to boundary layer turbulence parametrization

[eg Ferrari et al. 2008]

GM-Redi should "turn off" when isopycnal become too steep

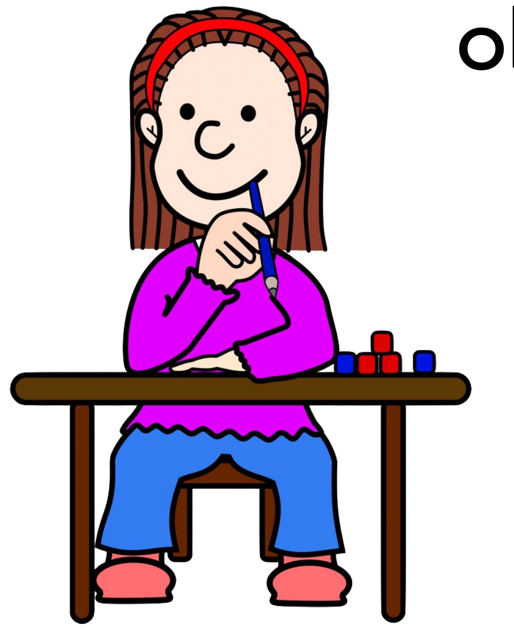
[slope-clipping, slope tapering Gerdes et al. 1991, Dabanasoglou and McWilliams 1995, Large et al. 1997]

parametrization should "turn off" in places where model is able to resolve eddies (double-counting)

[scale-aware eg Zanna et al. 2017]

eddy activity varies (laterally, vertically, seasonally?) GM/Redi diffusivities may depend on space/time

how do we come up with new parametrizations?



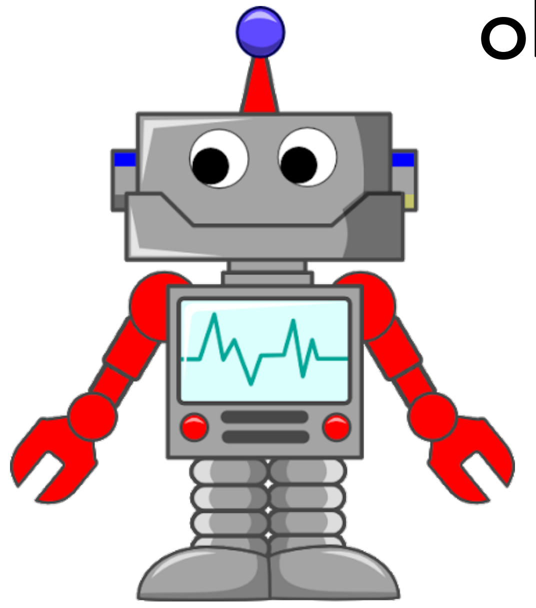
get inspired by data
(model output,
observations, night sky,…)

derive a model
from physical intuition
(usually involves some
free parameters)

calibrate free parameters
to match data

implement in climate model
and produce IPCC reports, etc

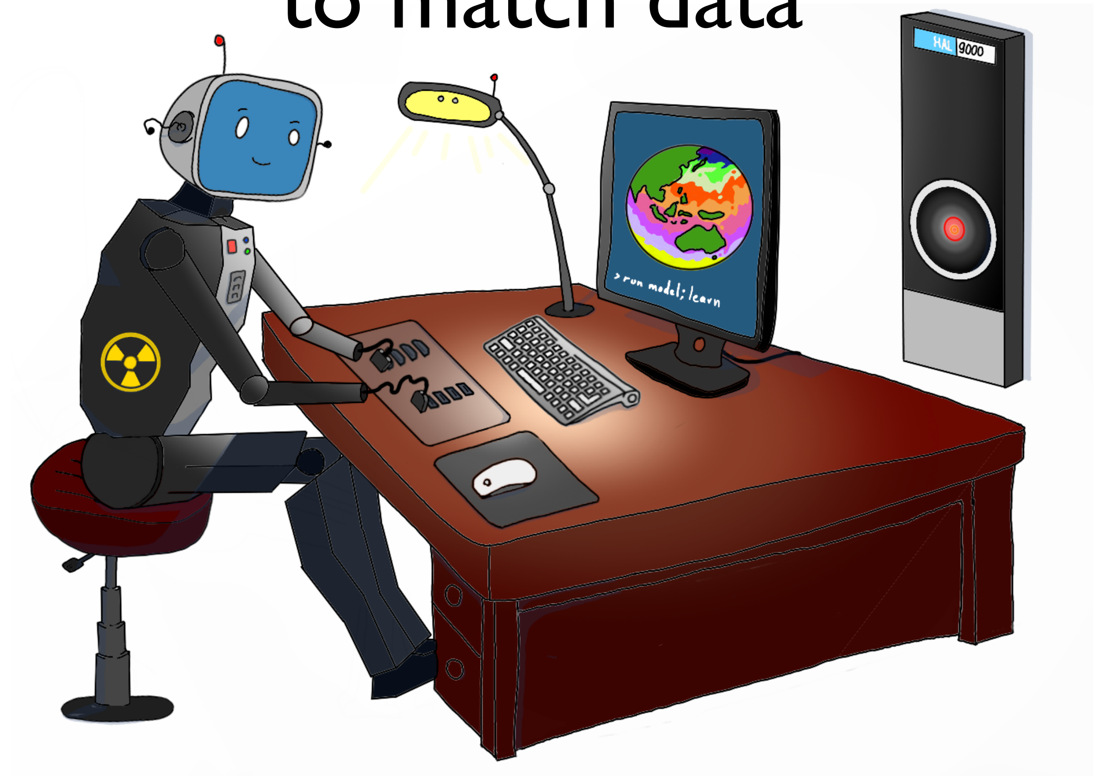
how do we come up with new parametrizations? and how machines can help?



get inspired by data
(model output,
observations, night sky,…)

derive a model
from physical intuition
(usually involves some
free parameters)

calibrate free parameters
to match data



implement in climate model
and produce IPCC reports, etc

how about data-driven?
doesn't that involve a neural network?

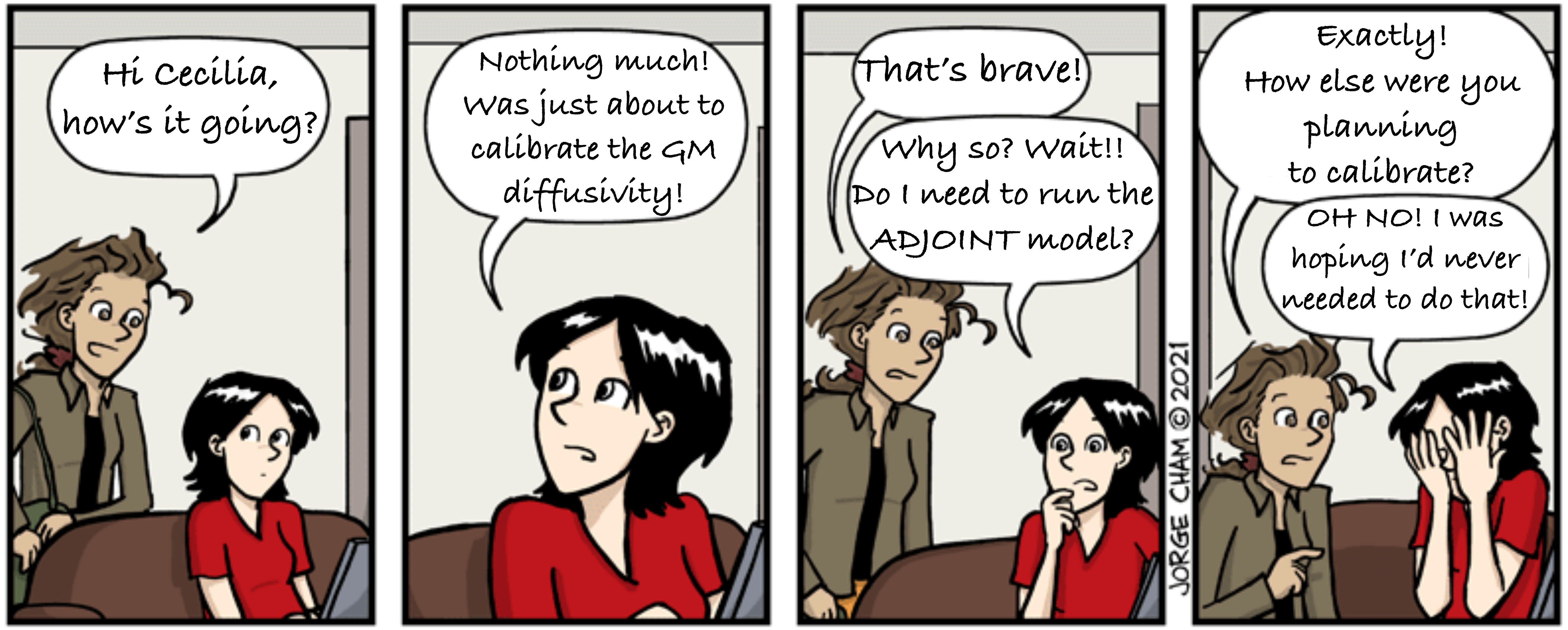
Newton's laws were, actually, "data-driven"

Instead of starting from a neural network
with O(1e6) free parameters
we start from what we currently have
and enhance our physical models adding few more free parameters

calibration is *data-driven*

calibration

*"All agree that calibration is great!
But most don't do it in a systematic manner
because it is so cumbersome!"*
— *adage*



adopted from WWW.PHDCOMICS.COM
and slightly modified

derivative-free Bayesian optimization
using ensemble Kalman filters

[Iglesias et al., *Inverse Problems*, 2013]

Ensemble Kalman Inverse process

Derivative-free ensemble optimization method
that seeks to find the optimal parameters θ for inverse problem

$$\underbrace{y}_{\text{observations (data)}} = \underbrace{G}_{\text{ocean model}}(\underbrace{\theta}_{\text{free parameters}}) + \underbrace{\eta}_{\text{noise}}$$

Calibration is done *online* by running ensembles of forward model runs

[Iglesias et al., *Inverse Problems*, 2013]

software enables research

CliMA / **Oceananigans.jl** Public

An oceanic library for fast, friendly, data-driven fluid dynamics on CPUs and GPUs

clima.github.io/oceananigansdocumentation/stable

MIT License

527 stars 95 forks

Unstar Unwatch

Code Issues 218 Pull requests 40 ...

main ...

navidcy ... 2 days ago

View code

README.md

Oceananigans.jl

Fast and friendly ocean-flavored Julia software for simulating incompressible fluid dynamics in Cartesian and spherical shell domains on CPUs and GPUs.

CliMA / **EnsembleKalmanProcesses.jl** Public

Implements Optimization and approximate uncertainty quantification algorithms, Ensemble Kalman Inversion, and Ensemble Kalman Processes.

clima.github.io/ensemblekalmanprocesses.jl/dev

Apache-2.0 License

12 stars 7 forks

Unstar Watch

Code Issues 12 Pull requests 2 Actions ...

main ...

bors and **trontrytel** ... 19 days ago

View code

README.md

EnsembleKalmanProcesses.jl

Implements Optimization and approximate uncertainty quantification algorithms, Ensemble Kalman Inversion, and Ensemble Kalman Processes.

adelinehillier / **OceanTurbulenceParameterEstimation.jl** Public

Parameter estimation for column models of the ocean surface boundary layer.

MIT License

1 star 0 forks

Unstar Unwatch

Code Issues 6 Pull requests 2 Actions Projects ...

main ...

navidcy Update README.md ... now 157

View code

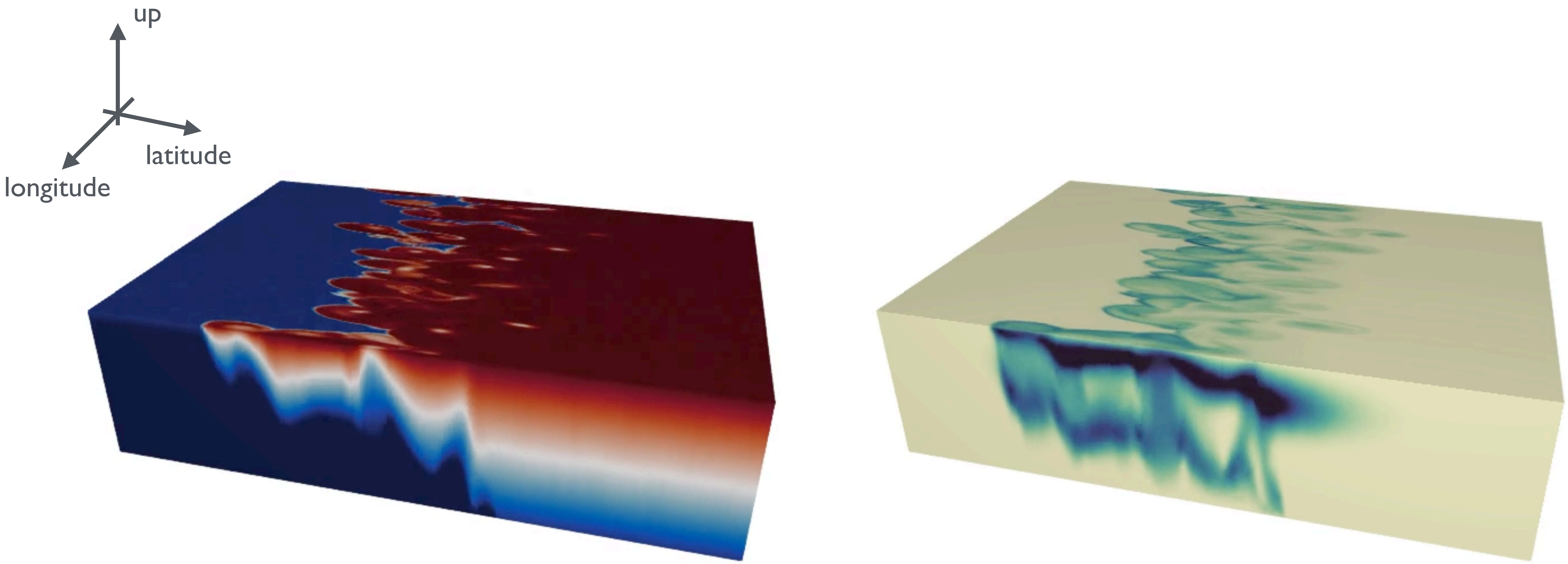
README.md

OceanTurbulenceParameterEstimation.jl

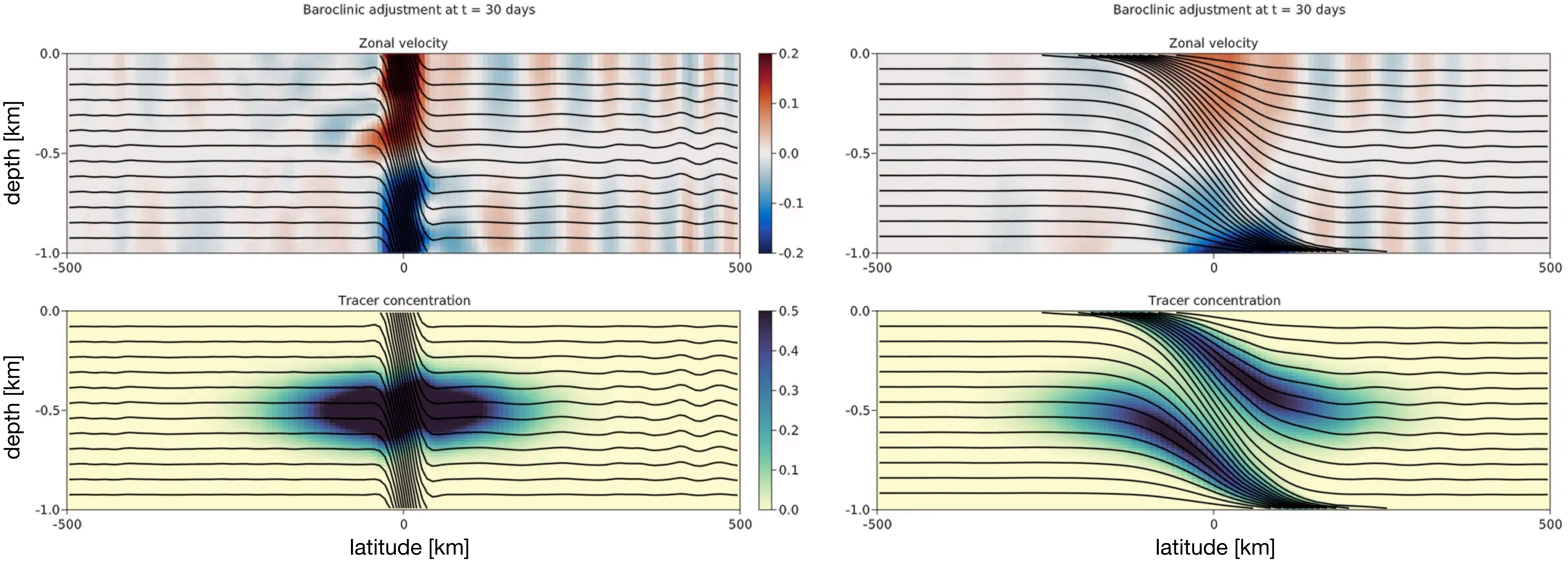
A Julia package designed to leverage Oceananigans.jl and EnsembleKalmanProcesses.jl to allow for calibration of ocean turbulence parametrizations.

baroclinic adjustment of a front

Buoyancy and tracer concentration at t = 30 days



zonally-averaged baroclinic adjustment of a front



with GM + Redi diffusion

$$\kappa_{\mathrm{GM}} = 1000\,\mathrm{m}^2\,\mathrm{s}^{-1} \quad \kappa_{\mathrm{Redi}} = 900\,\mathrm{m}^2\,\mathrm{s}^{-1}$$

perfect model calibration (proof-of-concept) using Ensemble Kalman Inverse process

$$y = G(\theta) + \eta$$

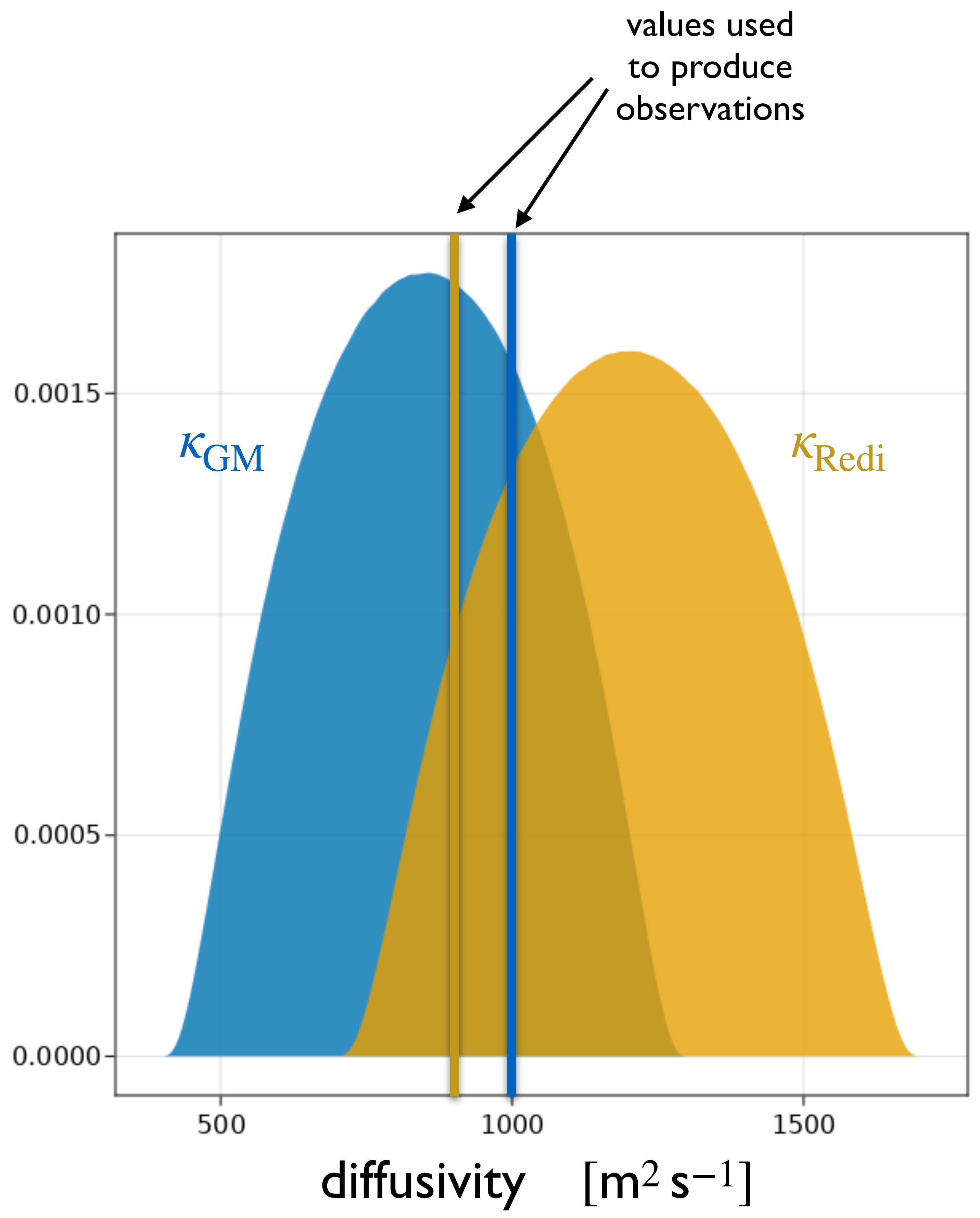
observations (data)

ocean model

free parameters

noise

prior distributions

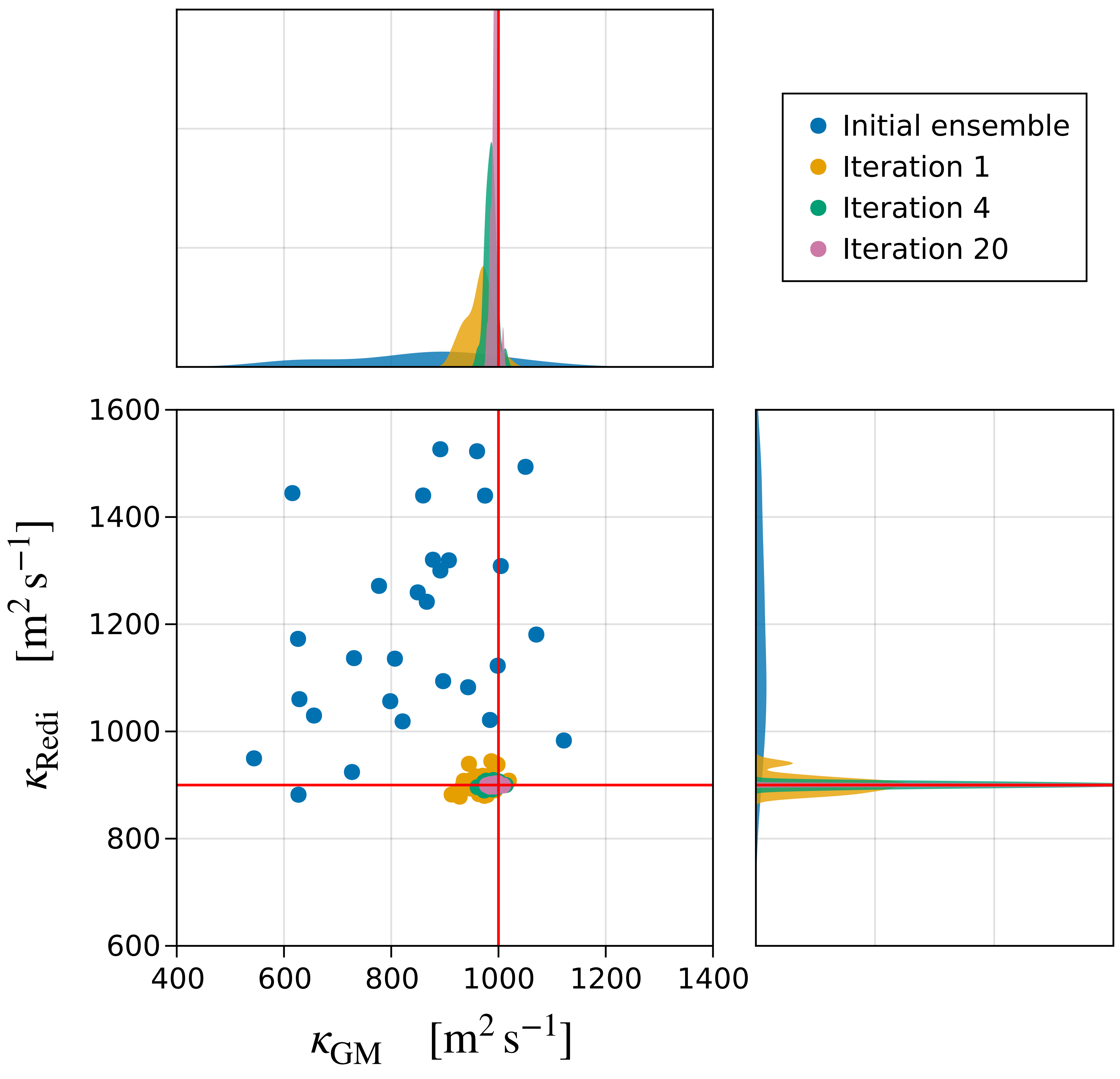


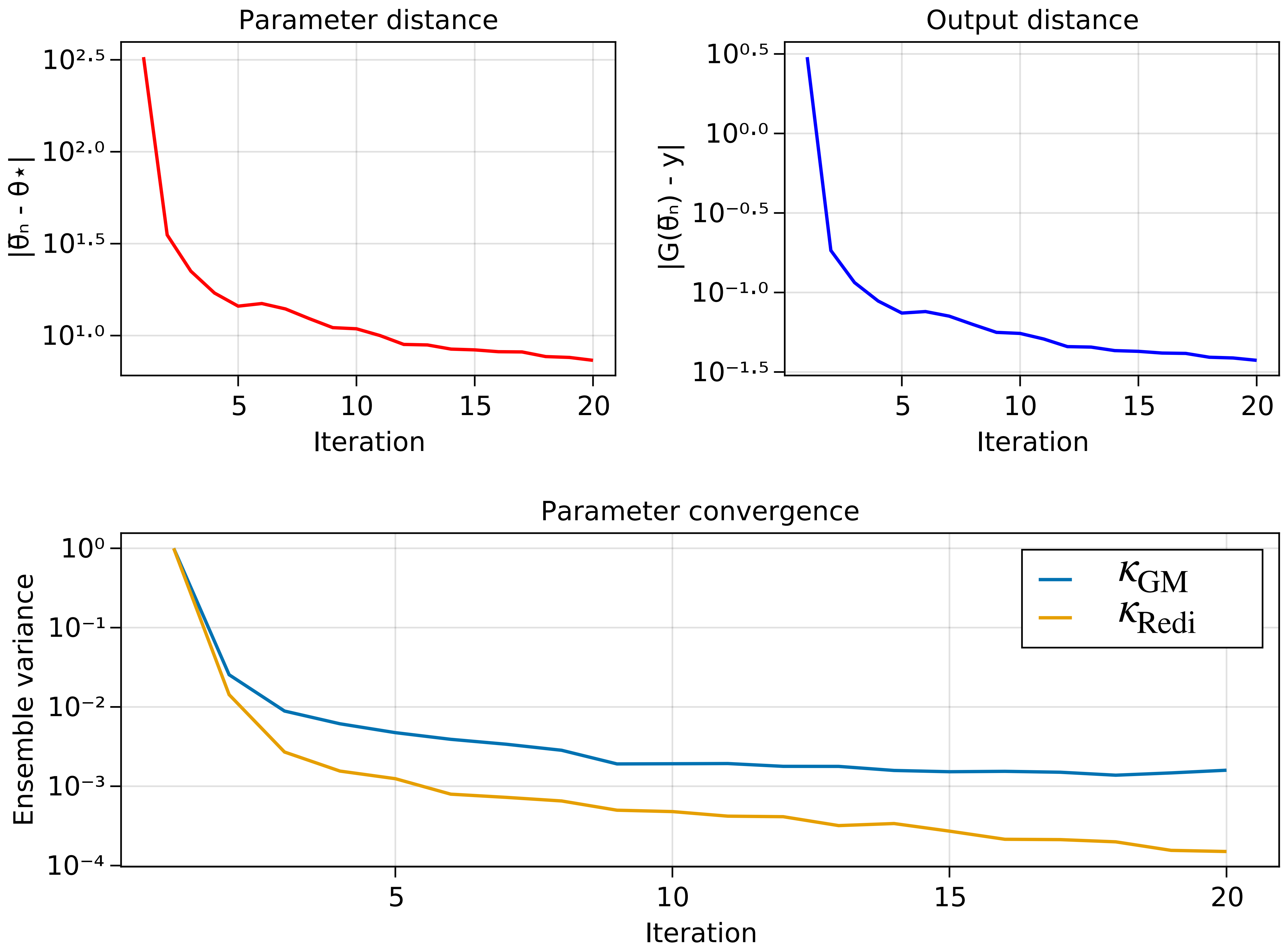
1. produce observations with given GM and Redi diffusivities

2. "close eyes" and see if EKI can converge figure out the values

("perfect model calibration" = obs y were generated by model G)

perfect model calibration (proof-of-concept) using Ensemble Kalman Inverse process

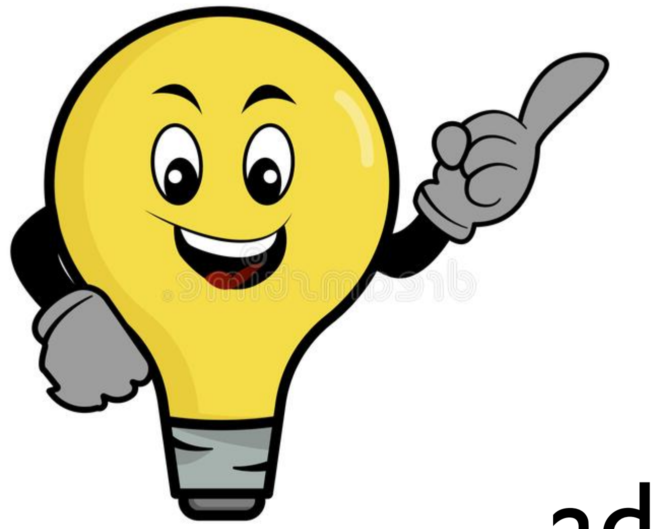




OK, so what?

we can easily calibrate free parameters of a turbulence closure

we can even calibrate *simultaneously* across various scenarios
and find optimal parameters that are robust

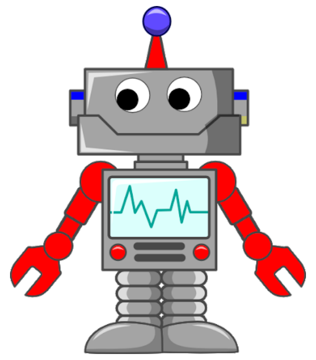


add depth/time/anything dependence in diffusivities is trivial

any parametrization obtain this ways
is, *by construction*, numerically stable
when added back to the model

but that's only the beginning

Oceananigans.jl



produce data
(high-resolution models,
LES, DNS)
or gather observations
and use as "ground truth"

use physical intuition
enhance parametrizations
(add physics, not `if`-statements)

possibly this adds few
more free parameters

calibrate free parameters
to *robustly* match data
across various scenarios

Ensemble Kalman
Processes

when cycle
"converges"

implement in
climate model