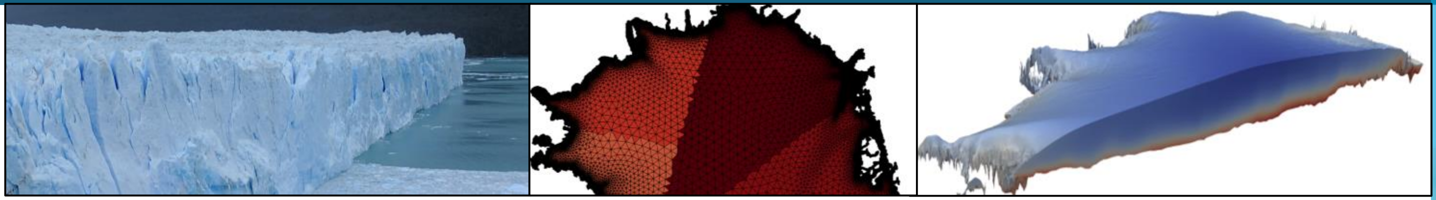
SAND2022-7804C

Sandia National Laboratories

Advances in Computing Probabilistic Projections of Sea Level Rise Due to Ice-sheet Mass Loss

CCR Center for Computing Research



Mauro Perego

Main collaborators:

K. Liegeois, J. Jakeman, T. Seidl (Sandia National Labs)
Q. He (University of Minnesota)
T. Hillebrand, M. Hoffman, S. Price (Los Alamos National Lab)

U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Talk Outline

- Brief motivation and introduction to ice sheet models
- Ice sheet initialization *(ProSPect/FASTMath)*
- Approaches to speed up the uncertainty quantification
  - Multifidelity approach for uncertainty quantification *(ProSPect/FASTMath)*
  - Neural Network surrogate to accelerate simulations *(PhILMs/ProSPect)*

Supported by US DOE Office of Science projects:

- ***ProSPect**: Probabilistic Sea-Level Projections from ice sheets and Earth System Models*
- ***FASTMath**: Frameworks, Algorithms and Scalable Technologies for Mathematics*
- ***PhILMs**: Physics Informed Learning Machines*

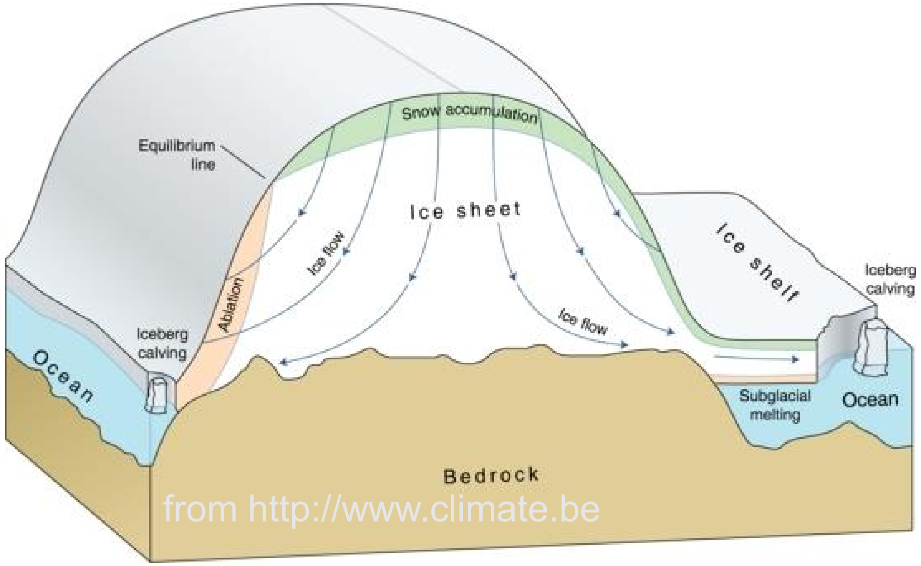
Brief Motivation an basic physics

- Modeling ice sheets (Greenland and Antarctica) dynamics is essential to provide estimates for sea-level rise in next decades to centuries.

- Ice behaves like a very viscous shear-thinning fluid (similar to lava flow) driven by gravity.

- Several unknown or poorly known parameters (e.g. basal friction, bed topography) and processes (calving laws, basal hydrology)

Perito Moreno glacier front



from http://www.climate.be

Model: Ice velocity equations

Stokes equations:

$$\begin{cases} -\nabla \cdot \sigma = \rho \mathbf{g} \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

gravit. acceleration

ice velocity

Stress tensor:

$$\sigma = 2\mu \mathbf{D} - pI, \qquad \mathbf{D}_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

Ice viscosity (dependent on temperature):

$$\mu = \frac{1}{2} A(T)\, |\mathbf{D}(\mathbf{u})|^{\frac{1}{n}-1}, \quad n \geq 1, \quad (\text{tipically } n \simeq 3)$$

In this work we use a simplification of Stokes equations, called **First Order** equations, obtained by scaling arguments given the shallow nature of the ice sheets and using hydrostatic pressure.

Model: Ice velocity equations

Stokes equations:

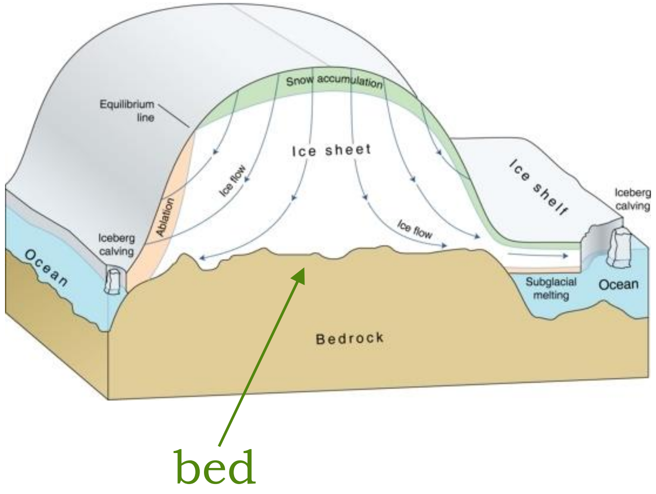
$$\begin{cases} -\nabla \cdot \sigma = \rho \mathbf{g} \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

Sliding boundary condition at ice bed:

$$\begin{cases} \mathbf{u} \cdot \mathbf{n} = 0, \quad \text{(impenetrablity)} \\ (\sigma \mathbf{n})_{||} = \beta \mathbf{u} \end{cases}$$

Free slip: $\beta = 0$

No slip: $\beta = \infty$



Hierarchy of approximations of Stokes equations

Stokes equations are typically simplified exploiting the shallow nature of the ice sheets and using hydrostatic pressure.

Increasing fidelity and cost

First Order (FO) model
(3D elliptic PDE)

membrane stress tensor

$$-\nabla \cdot \left(2\mu\tilde{\mathbf{D}}\right) - \partial_z(\mu\,\partial_z \mathbf{u}) = -\rho g \nabla s$$

$$2\mu\tilde{\mathbf{D}}\mathbf{n} = \beta\mathbf{u}, \quad \text{on bed}$$

upper surface

MALI
(production)

Mono-Layer Higher-order (MOLHO) model
(two 2d PDEs)

Solve FO with trial function

$$\mathbf{u} = \bar{\mathbf{u}}(x,y) + \mathbf{u}_{\mathrm{def}}(x,y)\,\varphi(z)$$

Shallow Shelf Approx. (SSA)
(2d PDE, for floating fast-flowing ice)

$$-\nabla \cdot \left(2\mu H \tilde{\mathbf{D}}(\bar{\mathbf{u}})\right) + \beta\bar{\mathbf{u}} = -\rho g H \nabla s$$

Shallow Ice Approx. (SIA)
(for grounded slow-flowing ice)

$$\bar{\mathbf{u}} = -\left(\frac{2A\rho^3 g^3}{5} H^4 |\nabla s|^2 + \frac{\rho g}{\beta} H\right) \nabla s$$

FEniCS
(prototyping)

Model: Temperature equation

Heat equation (for cold ice):

$$\rho c\, \partial_t T + \nabla \cdot (k \nabla T) + \rho c\, \mathbf{u} \cdot \nabla T = 4\mu |D(\mathbf{u})|^2$$

conductivity

heat capacity

dissipation heating

Boundary condition at the ice bed (includes melting and refreezing):

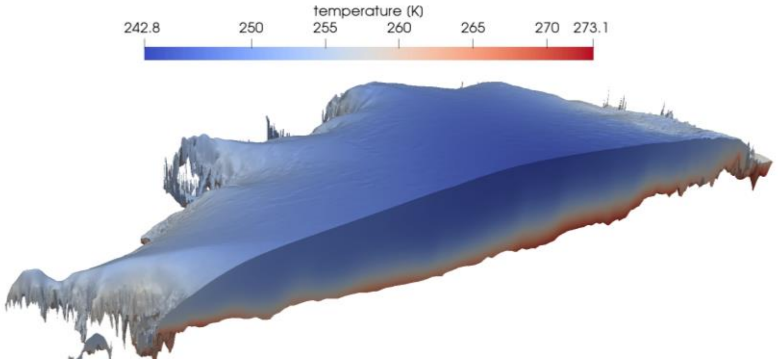
$$\mathrm{m} = \mathrm{G} + \beta |\mathbf{u}|^2 - k \nabla T \cdot \mathbf{n}$$

frictional heating

melting rate

geothermal heat flux

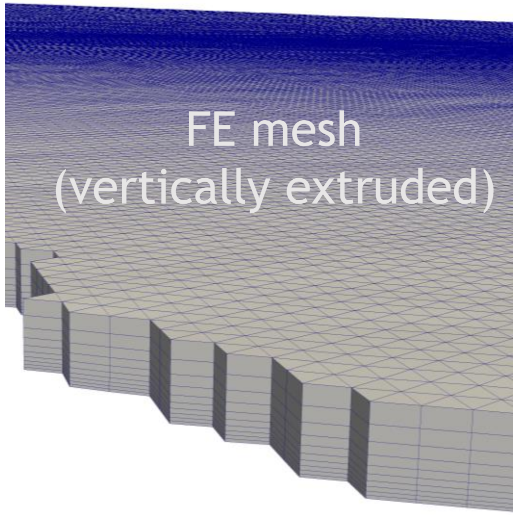
temperature flux



In this work we use a enthalpy formulation that accounts for temperate ice as well.

Software: MPAS-Albany Land Ice model (MALI)

ALGORITHM	SOFTWARE TOOLS
Linear Finite Elements on tets/prisms	Albany Land Ice
Optimization	ROL
Nonlinear solver (Newton method)	NOX
Krylov linear solvers/Prec	Belos/MueLu, Belos/FROSch
Automatic differentiation	Sacado



MPAS (Model for Prediction Across Scales): *Fortran*, **finite volumes** library, conservative Lagrangian schemes for advecting tracers (evolution of ice thickness)
Albany Land Ice: *C++* finite element library built on top of **Trilinos** achieving performance portability through ***Kokkos*** programming model. Provides large scale PDE constrained optimization capabilities

References:
Hoffman, et al. GMD, 2018
Tuminaro, Perego, Tezaur, Salinger, Price, SISC, 2016.
Tezaur, Perego, Salinger, Tuminaro, Price, Hoffman, GMD, 2015
Perego, Price, Stadler, JGR, 2014

Ice sheet initialization

(w/ K. Liegeois, T. Hillebrand, M. Hoffman and S. Price)

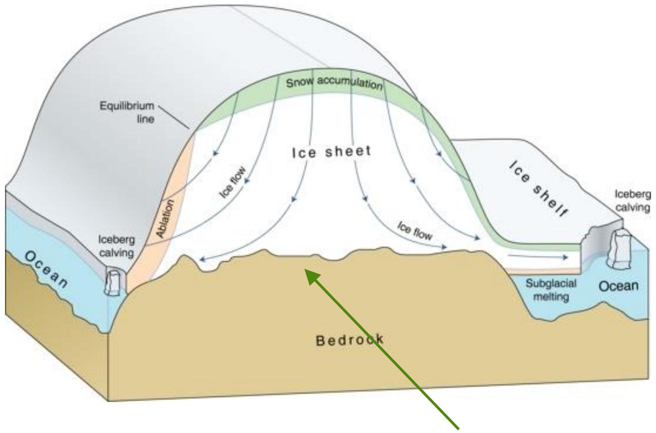
Goal: Find the initial/present-day thermo-mechanical state of the ice sheet and estimate the unknown/poorly known model parameters, by matching observations

Approach: **PDE-constrained optimization**

Find basal friction coefficient β that minimizes the mismatch with surface velocity:

$$\min_{\beta} \mathcal{J}(\beta) = \int_{\Omega} \frac{|u - u_{obs}|^2}{\sigma^2} + R(\beta)$$

Subject to the coupled velocity/temperature problem



unknown sliding parameter β

Software Requirements

- Large Scale optimization library (ROL), featuring gradient-based methods (ROL)
- Computation of gradients of the PDE residual and the loss functional w.r.t. the solution and the parameters. **Automatic Differentiation** is crucial for complex physics
- Faster, more robust methods available using **Hessian** (second derivatives)

Ice sheet initialization

Hessian computation using automatic differentiation (using Sacado package)

Newton-Krylov optimization methods require Hessian mat-vec products:

Hessian of residual $\boldsymbol{f}$ dotted with the Lagrange multiplier $\boldsymbol{\lambda}$ in the direction $\boldsymbol{v}$:

$$\partial_{\boldsymbol{uu}}(\boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{u},\boldsymbol{p}))\,\boldsymbol{v},\quad \partial_{\boldsymbol{up}}(\boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{u},\boldsymbol{p}))\,\boldsymbol{v},$$
$$\partial_{\boldsymbol{pu}}(\boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{u},\boldsymbol{p}))\,\boldsymbol{v},\quad \partial_{\boldsymbol{pp}}(\boldsymbol{\lambda}^T\boldsymbol{f}(\boldsymbol{u},\boldsymbol{p}))\,\boldsymbol{v}$$

work by
Kim Liegeois

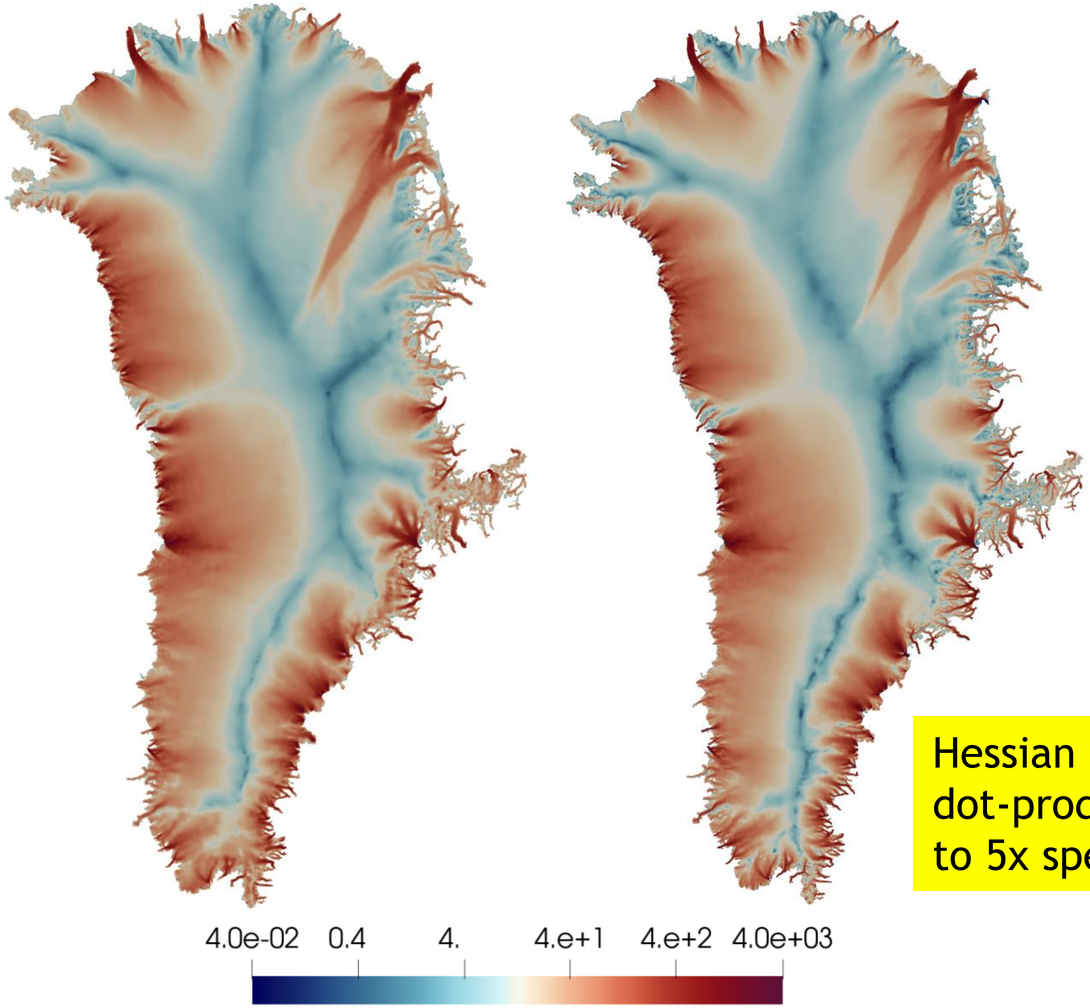
Computed w/ **automatic differentiation**, differentiating twice, based on the formula:

$$\partial_{\boldsymbol{pp}}\,\mathcal{J}(\boldsymbol{p})\,\boldsymbol{v} = \partial_r\left(\partial_{\boldsymbol{p}}\,\mathcal{J}(\boldsymbol{p}+r\,\boldsymbol{v})\right)\Big|_{r=0}$$

We also build the sparse matrix $H_{\boldsymbol{pp}} = \partial_{\boldsymbol{pp}}\,\mathcal{J}$, **efficiently** computed using coloring, seeding and performing mat-vec products.

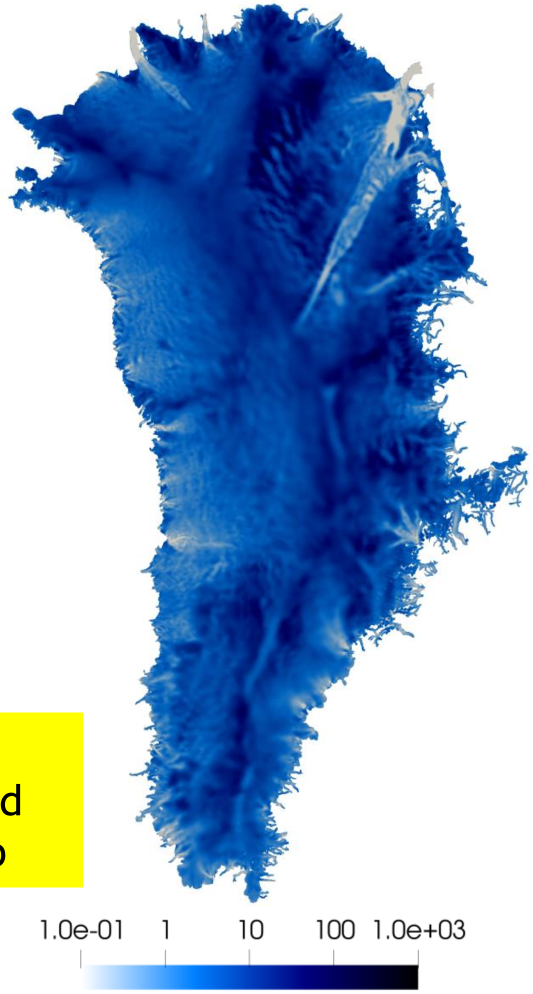
$H_{\boldsymbol{pp}}$ is used to define a Hessian-based vector-product for the Optimization package ROL instead of the Euclidean dot-product, leading to improved convergence of the optimization algorithms.

Thermo-mechanical initialization of Greenland ice sheet

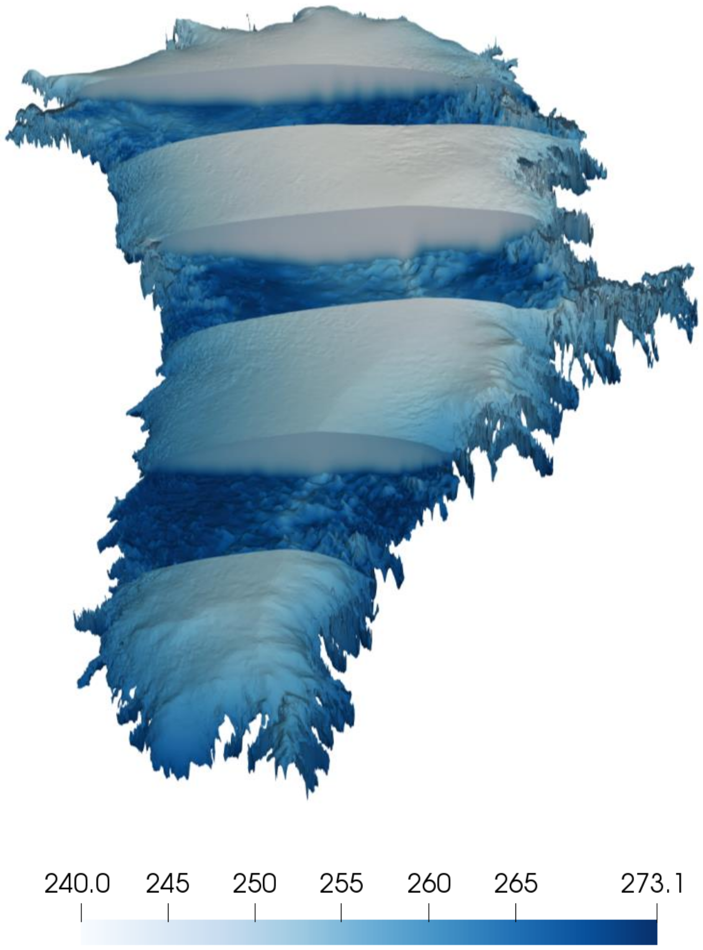


modeled ice speed

observed ice speed



modeled basal friction



modeled temperature

Hessian based dot-product led to 5x speed-up

300K parameters, 14M unknowns. Initialization: ~10 hours on 2k nodes on NERSC Cori (Haswell),
The optimization is constrained by the **coupled velocity-temperature** solvers. Most large scale-ice sheets codes constrain the optimization only with the velocity solver, which results in a temperature field that is not consistent with the velocity

Approaches to accelerate uncertainty quantification

We are interested in computing uncertainty in the *total ice mass loss*, our Quantity of Interest (QoI), due to the uncertainty in the basal friction.
We assume that the basal friction distribution is lognormal, centered on the value obtained during optimization:

$$\beta = \exp(\gamma)\,,\ \text{where}\ \gamma \sim \mathcal{N}\left(\log(\beta_{\text{opt}}), k\right),\ \text{and}\ k(\boldsymbol{x_1},\boldsymbol{x_2}) = \sigma \exp\left(-\frac{|\boldsymbol{x_1}-\boldsymbol{x_2}|^2}{2\,l^2}\right)$$

variance

correlation length

Computing the uncertainty requires a **huge number** of solution of the ice flow problem, for different samples of the parameter β.

We present two approaches for reducing the cost:

- **multi-fidelity**
- **neural network surrogates**

Multi-fidelity Models

Ice thickness equation:

$$\partial_t H + \nabla \cdot (\bar{\mathbf{u}} H) = f_H$$

ice thickness

accumulation/ablation

Increasing fidelity and cost

Mono-Layer Higher-order (MOLHO) model
(two 2d PDEs)

Solve FO with trial function

$$\mathbf{u} = \bar{\mathbf{u}}(x, y) + \mathbf{u}_{\text{def}}(x, y)\, \varphi(z)$$

Shallow Shelf Approx. (SSA)
(2d PDE, for floating fast-flowing ice)

$$-\nabla \cdot \left(2\mu H \tilde{\mathbf{D}}(\bar{\mathbf{u}})\right) + \beta \bar{\mathbf{u}} = -\rho g H \nabla s$$

Shallow Ice Approx. (SIA)
(for grounded slow-flowing ice)

$$\bar{\mathbf{u}} = -\left(\frac{2A\rho^3 g^3}{5} H^4 |\nabla s|^2 + \frac{\rho g}{\beta} H\right) \nabla s$$

FEniCS
(prototyping)

FEniCS code, developed by C. Sockwell, M. Perego from an original implementation by D. Brinkerhoff

Multi-fidelity Approach (w/ John Jakeman and Tom Seidl)

We use approximate control variate (ACV) Monte Carlo which is a generalization of Multi-level Monte Carlo (MLMC)

$$\begin{aligned} Q^{\mathrm{ACV}} &= Q_{0,\mathcal{Z}_{0,1}} + \sum_{\alpha=1}^{M} \eta_\alpha \left(Q_{\alpha,\mathcal{Z}_{\alpha,1}} - \mu_{\alpha,\mathcal{Z}_{\alpha,2}} \right) = Q_{0,\mathcal{Z}_{0,1}} + \sum_{\alpha=1}^{M} \eta_\alpha \Delta_{\alpha,\mathcal{Z}_{\alpha,1},\mathcal{Z}_{\alpha,2}} \\ &= Q_{0,N} + \boldsymbol{\eta}\boldsymbol{\Delta} \end{aligned}$$

$$\boldsymbol{\eta} = -\mathbb{C}\mathrm{ov}[\boldsymbol{\Delta}, \boldsymbol{\Delta}]^{-1} \mathbb{C}\mathrm{ov}\left[\boldsymbol{\Delta}, Q_0\right]$$

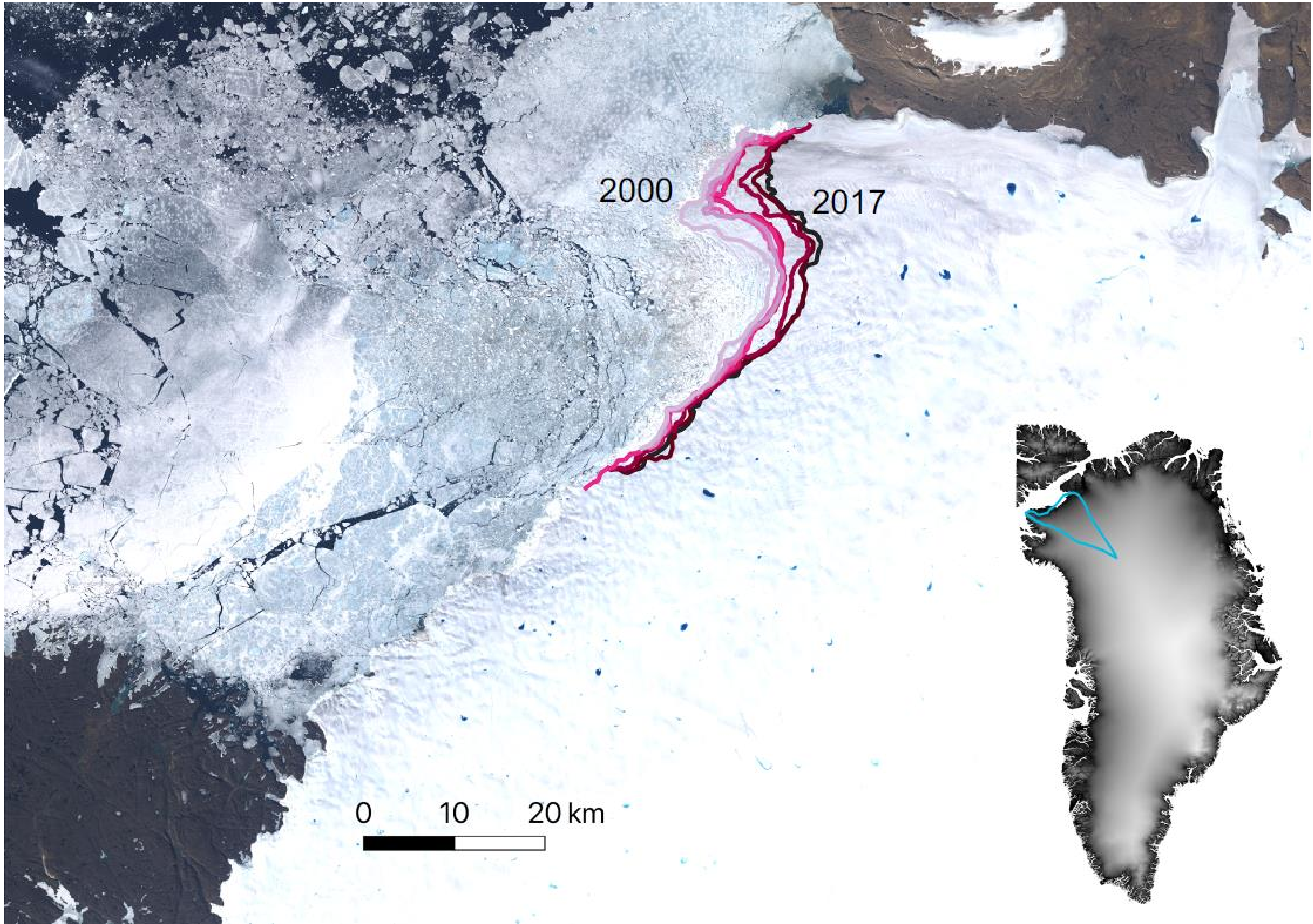
$$\gamma = 1 - \mathbb{C}\mathrm{ov}[\boldsymbol{\Delta}, Q_0]^T \frac{\mathbb{C}\mathrm{ov}[\boldsymbol{\Delta}, \boldsymbol{\Delta}]^{-1}}{\mathbb{V}\left[Q_0\right]} \mathbb{C}\mathrm{ov}\left[\boldsymbol{\Delta}, Q_0\right]$$

$$V\left[Q^{\mathrm{ACV}}\right] = \gamma \mathbb{V}[Q_0]$$

We consider three different *mesh resolutions* and *three different models*: MOLHO, SSA, SIA.

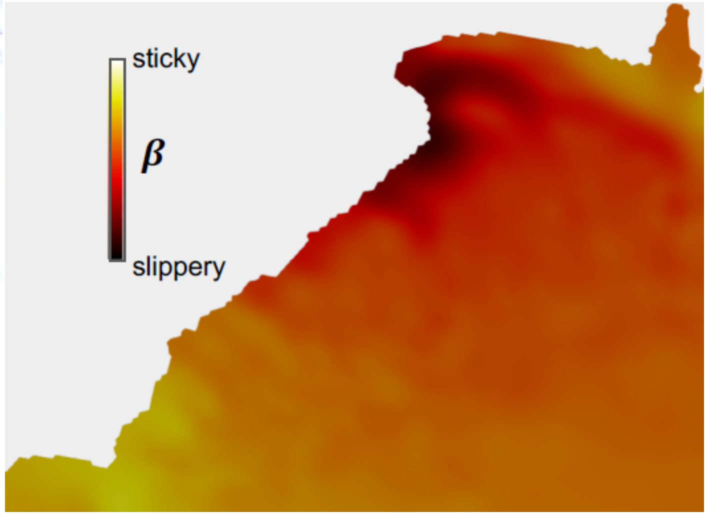
Focus on Humboldt glacier

(Humboldt is one of the largest glaciers in Greenland)



Observed grounding line retreat from year 2000

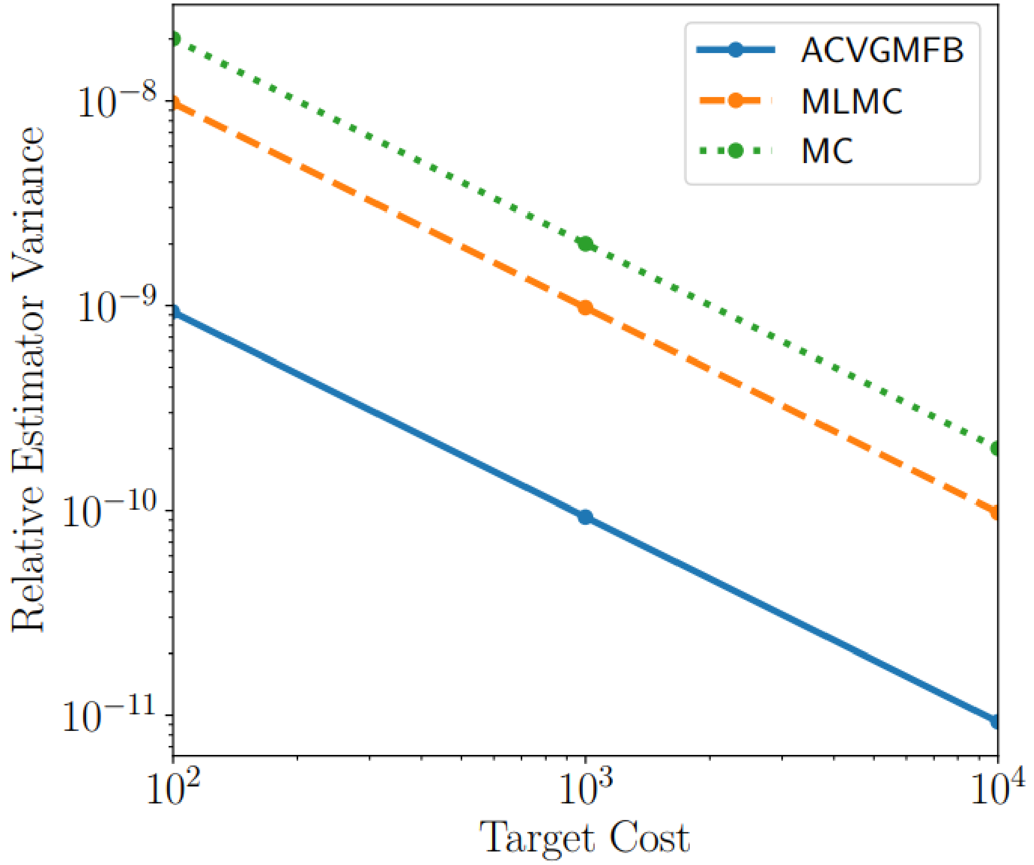
Estimated basal friction

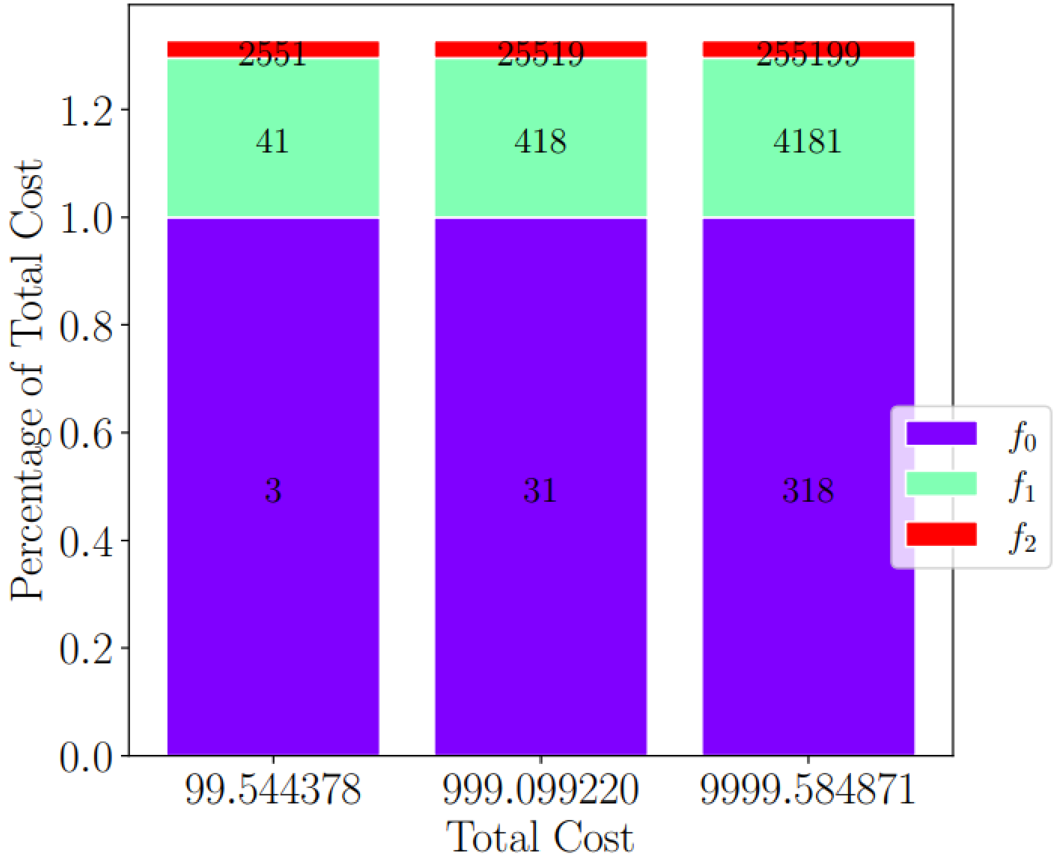


Courtesy of **T. Hillebrand**

Multi-fidelity Results (w/ J. Jakeman and T. Seidl)

We compare vanilla Monte Carlo approach, using the MOLHO model and the finest mesh (f_0) with
Multi-Level Monte Carlo approach, using MOLHO model and three mesh resolutions
Approximate-Variate Monte-Carlo, which automatically select the MOLHO model with finest mesh, SSA model on medium mesh (f_1) and SSA using coarse mesh (f_2). Note that no SIA model is chosen (as it should be!)





Neural Network surrogates
(w/ Qizhi He, A. Howard, S. Panos, G. Karniadakis)

Thickness equation:

$$\boxed{\partial_t \underset{\nearrow \atop \text{ice thickness}}{H} + \nabla\cdot(\underset{\nearrow \atop \text{vertically avg. velocity}}{\bar{\mathbf{u}}} H) = \underset{\nwarrow \atop \text{accumulation/ablation}}{f_H}}$$

Time discretization:

$$\frac{{\color{red}H_\beta^{(n+1)}} - H_\beta^n}{\Delta t} + \nabla\cdot\left({\color{blue}\bar{\mathbf{u}}_\beta^{(n+1)}}\,{\color{red}H_\beta^{(n+1)}}\right) = F_H^{(n+1)}$$

$$\boxed{\begin{cases} -\nabla\cdot\sigma({\color{blue}\mathbf{u}_\beta^{(n+1)}}) = \rho\mathbf{g} & \text{in } \Omega_{\color{red}H^{n+1}} \\ \nabla\cdot{\color{blue}\mathbf{u}_\beta^{(n+1)}} = 0 & \text{in}\Omega_{\color{red}H^{n+1}} \end{cases}}$$

$$\bar{\mathbf{u}}_\beta^{n+1} = \boxed{\mathcal{G}(\beta, H^{n+1})}$$

Stokes equation maps the thickness and the basal friction into the velocity

Neural Network surrogates
(w/ Qizhi He, A. Howard, S. Panos, G. Karniadakis)

The velocity solver is the most expensive part of the model.
Idea: replace the velocity solve with a Deep Operator Network*

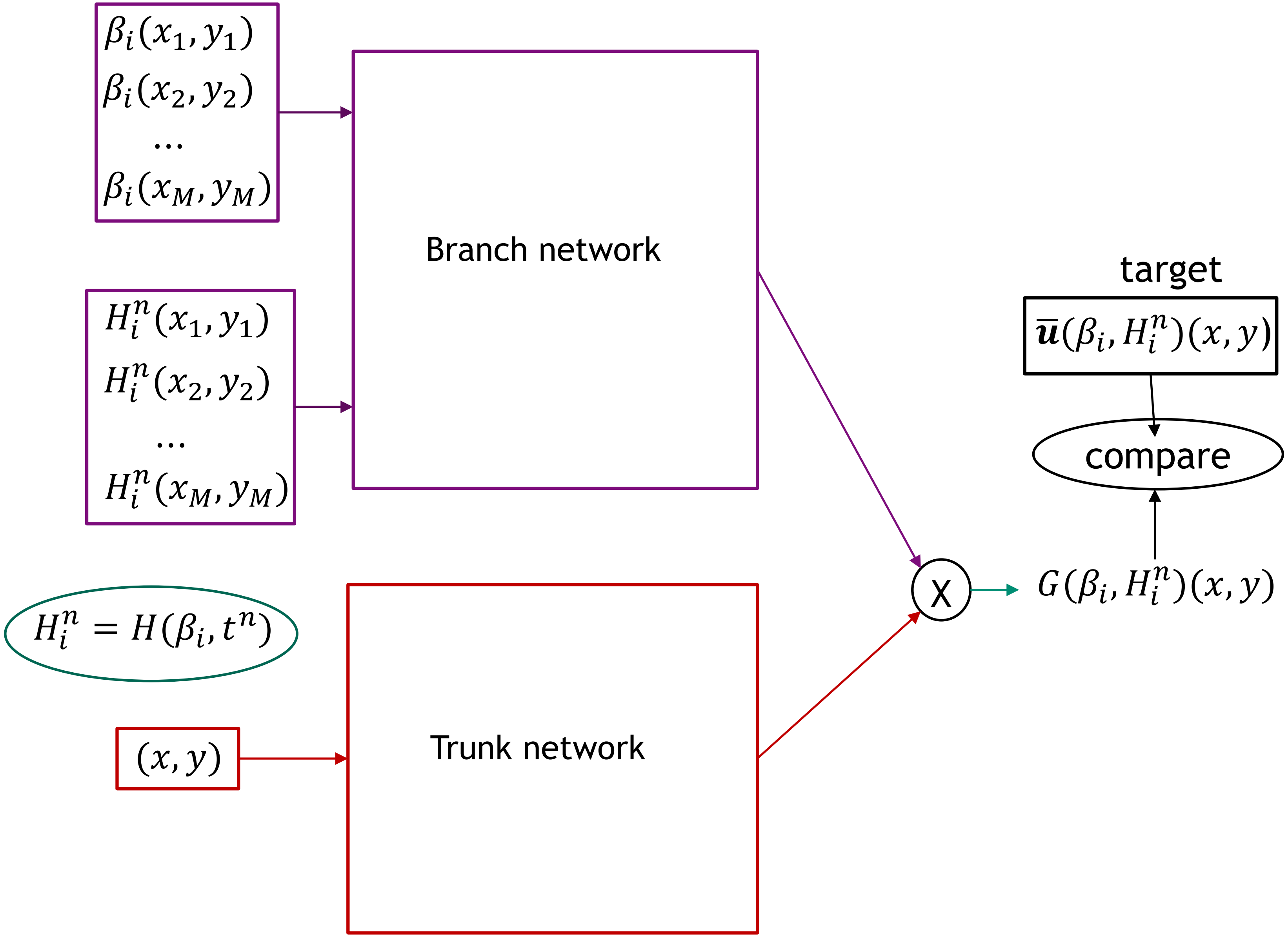
$$\bar{\mathbf{u}}_\beta^{n+1} = \boxed{\mathcal{G}(\beta, H^{n+1})} \quad \longleftrightarrow \quad \boxed{\text{DeepONet}}$$

Instead of approximating functions, DeepONet approximate *nonlinear* continuous operators.

The universal approximation theorem provides a strong mathematical foundation of DeepONets

*Lu, L., Jin, P., Pang, G. *et al.* Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat Mach Intell* **3,** 218–229 (2021).

DeepONet architecture



Velocity and thickness data are generated by the FEM code, implemented in FEniCS

Input/Output:

Branch input size: $(N_\beta N_T, 1, \mathbf{2M})$

Trunk input size: $(N_\beta N_T, M, \mathbf{2})$

Target size: $(N_\beta N_T, M, \mathbf{2})$

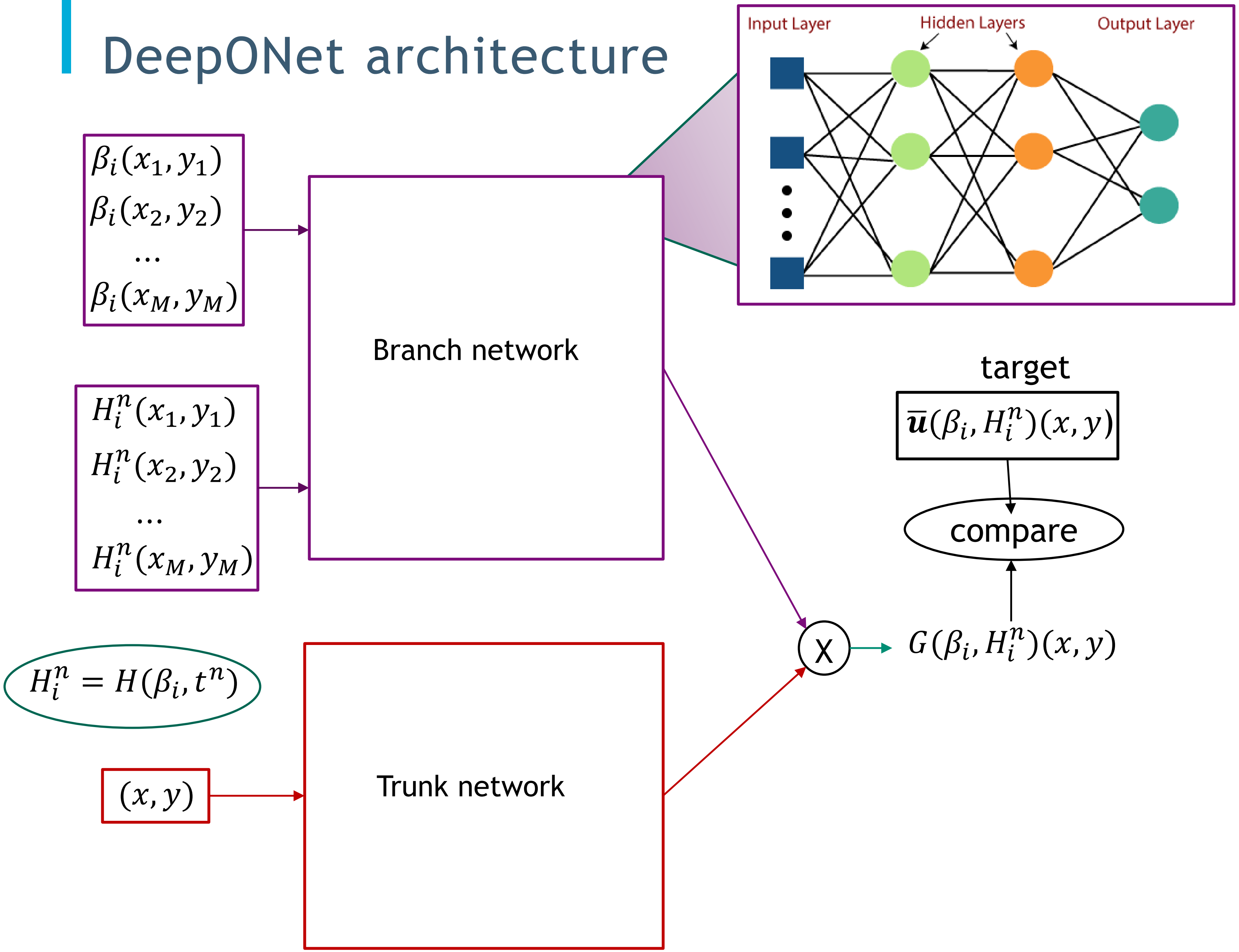
Legend:

M: size of spatial grid

N_β: number beta samples

N_T: number of time snapshots

DeepONet architecture



Velocity and thickness data are generated by the FEM code, implemented in FEniCS

Input/Output:

Branch input size: $(N_\beta N_T, 1, \boldsymbol{2M})$

Trunk input size: $(N_\beta N_T, M, \mathbf{2})$

Target size: $(N_\beta N_T, M, \mathbf{2})$

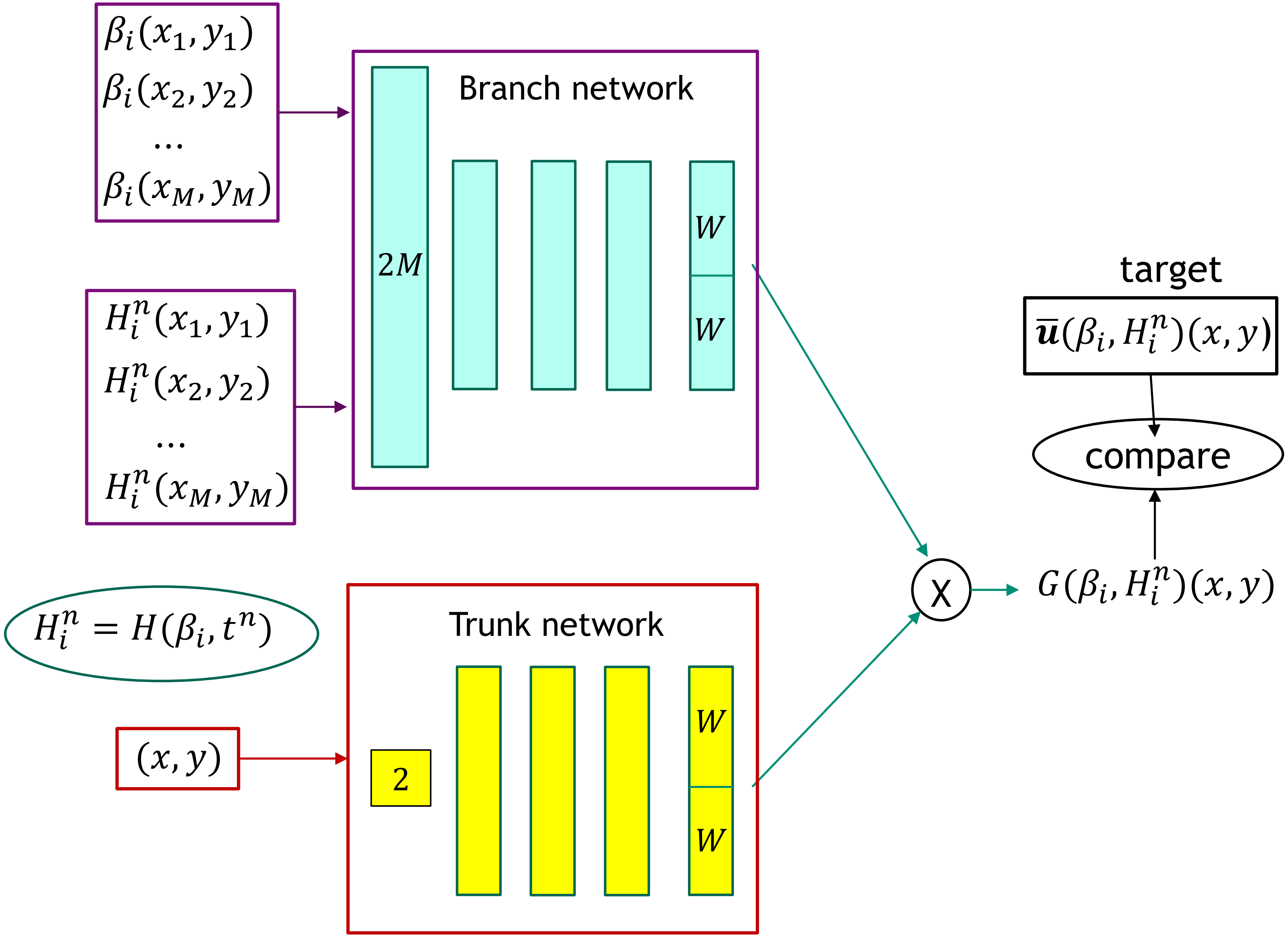
Legend:

M: size of spatial grid

N_β: number beta samples

N_T: number of time snapshots

DeepONet architecture



Velocity and thickness data are generated by the FEM code, implemented in FEniCS

Input/Output:

Branch input size: $(N_\beta N_T, 1, \mathbf{2M})$

Trunk input size: $(N_\beta N_T, M, \mathbf{2})$

Target size: $(N_\beta N_T, M, \mathbf{2})$

Legend:

M: size of spatial grid

N_β: number beta samples

N_T: number of time snapshots

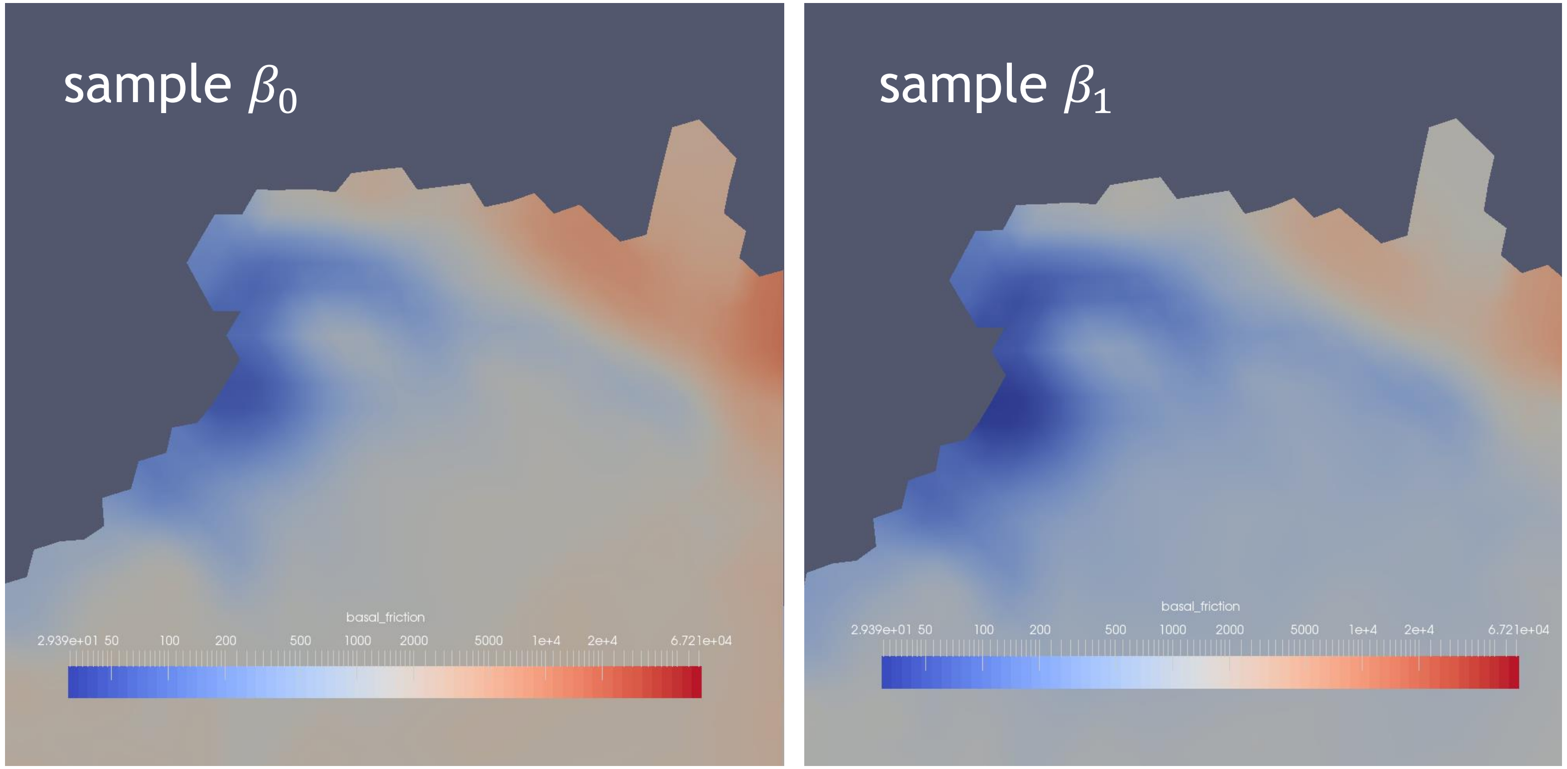
Humboldt (basal friction samples)

Basal friction sampled from a log-normal distribution:

$$\beta = \exp(\gamma)\,,\ \text{where}\ \gamma \sim \mathcal{N}\left(\log(\beta_{\text{opt}}), k\right),\ \text{and}\ \ k(\boldsymbol{x_1}, \boldsymbol{x_2}) = \sigma \exp\left(-\frac{|\boldsymbol{x_1} - \boldsymbol{x_2}|^2}{2\, l^2}\right)$$

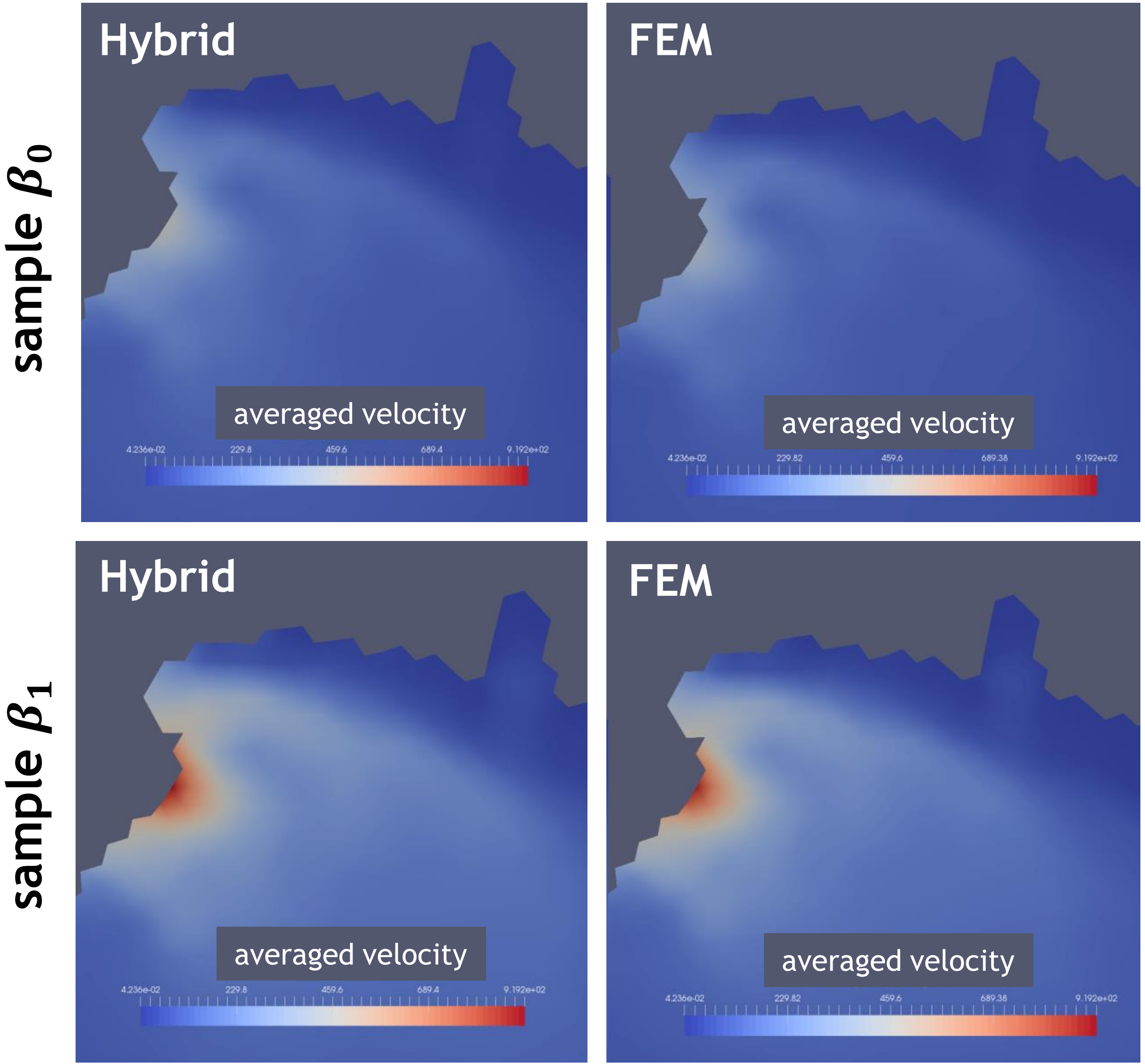
Workflow:

- Generated beta samples
- Generate thickness and velocity data for different beta samples using Finite Elements (FEM) code
- Train the DeepONet w/ velocity data



Basal friction samples

Humboldt – SSA model (computing averaged velocity w/ DeepONets)

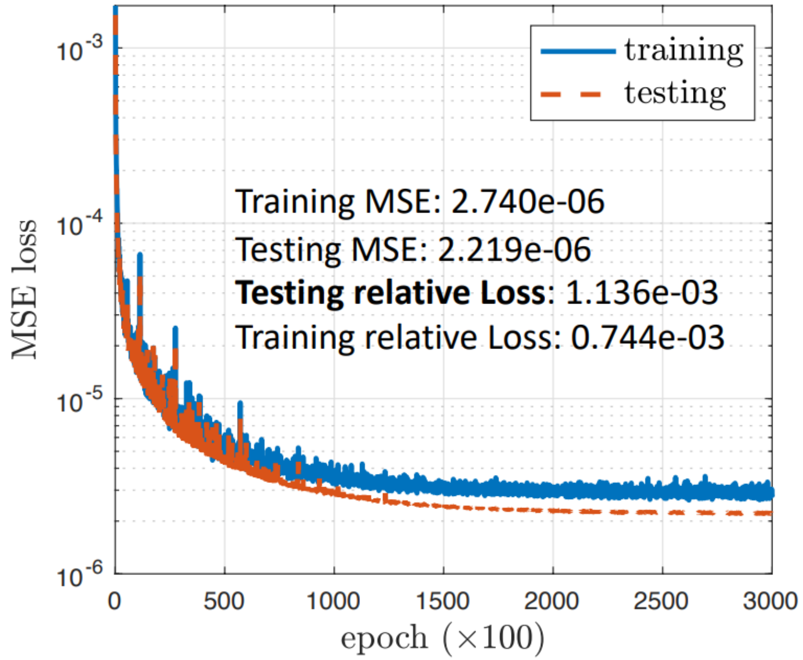


Hybrid: thickness solved w/ ***FEM*** calling the ***DeepONet*** at each time step to compute velocity

FEM: thickness and velocity models solved with FEM

Left: Averaged velocity at T=100 yr for *test* beta samples (***NOT*** used for training)

SSA Training data: $\{\beta_i\}_{i=20}^{300}$



Training MSE: 2.740e-06

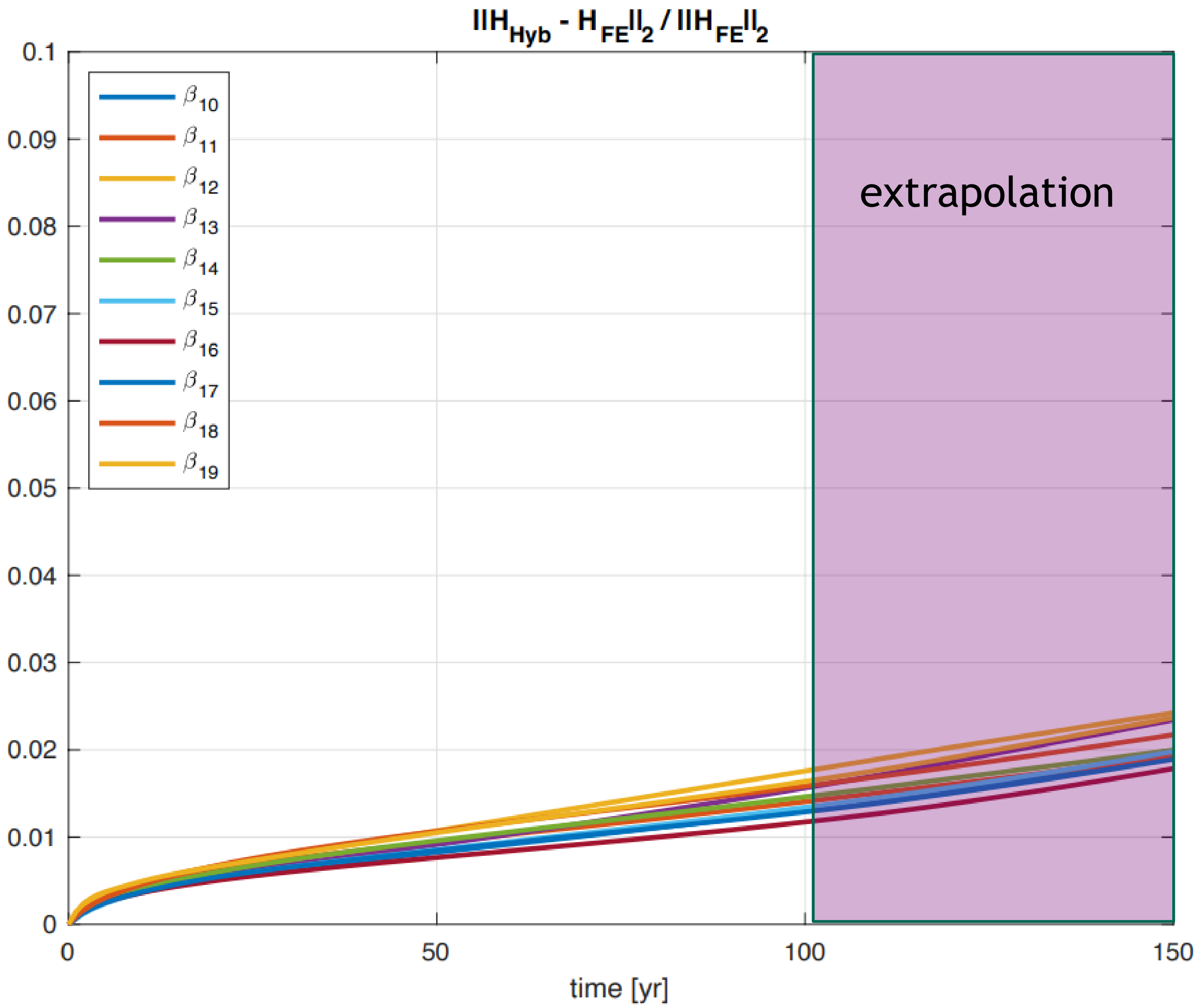
Testing MSE: 2.219e-06

Testing relative Loss: 1.136e-03

Training relative Loss: 0.744e-03

Humboldt – SSA model

(relative error as a function of time for different data)

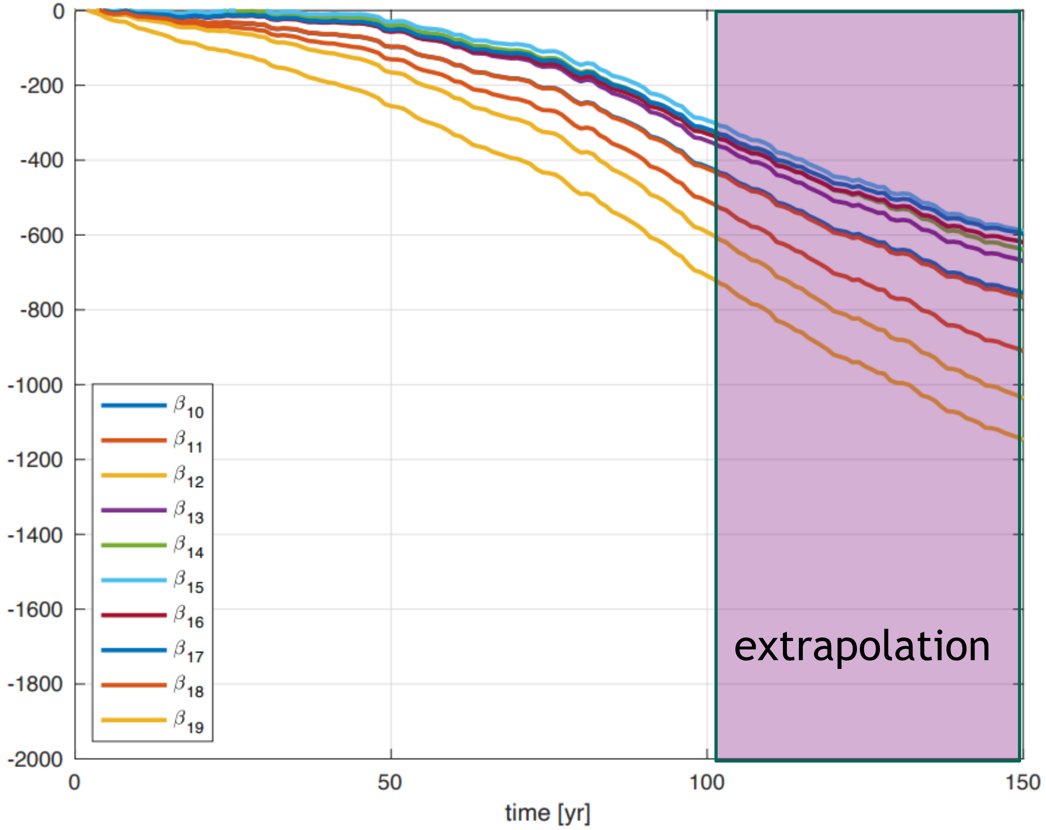


Setting:

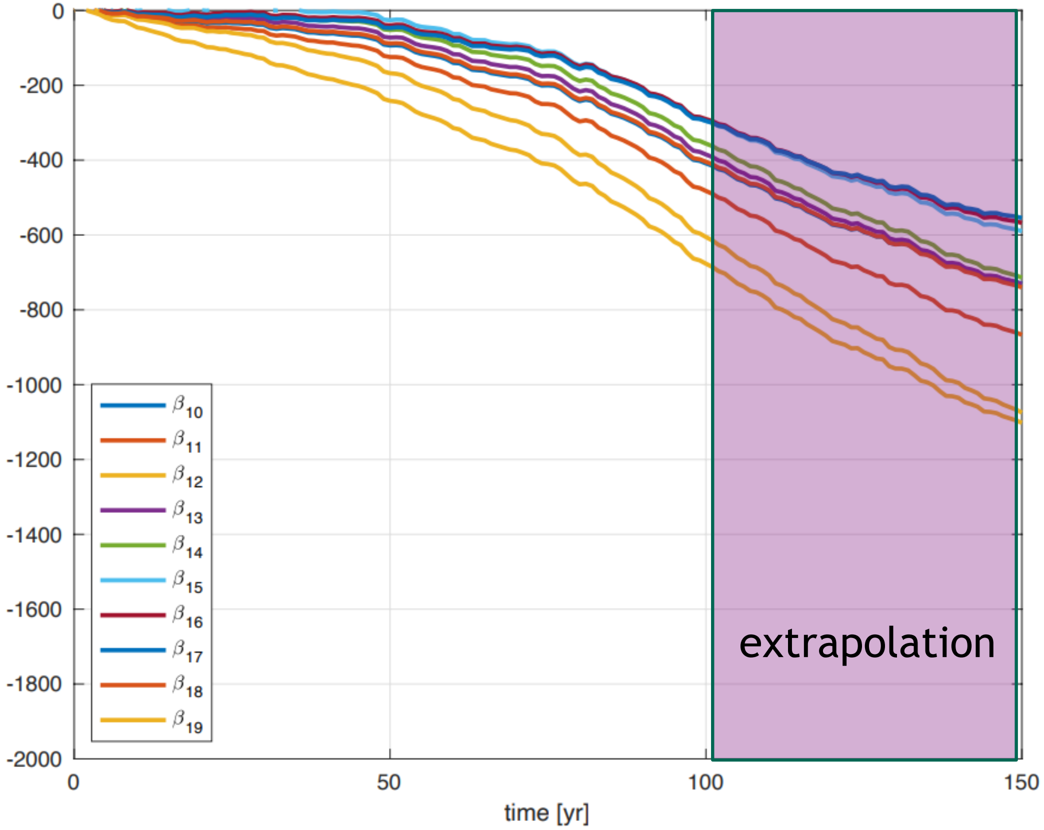
# training beta samples: 280
#epochs: 300,000
4 layers of width W: 700
time snapshots: 1, 2, …, 100

ice thickness (FEM)

Humboldt – SSA model (glacier mass loss)



Glacier mass change in gigatons, FEM model



Glacier mass change in gigatons, Hybrid model

Fast evaluation of forward model will enable the quantification of uncertainty on of sea level rise