Universal Differential Equations for glacier ice flow modelling using ODINN.jl

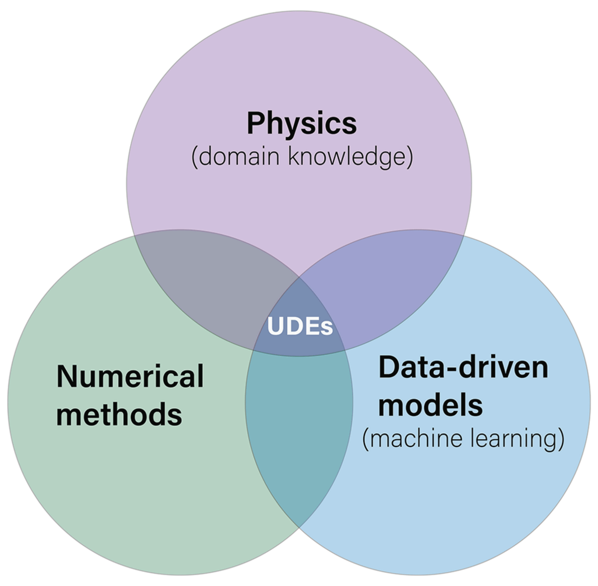
Jordi Bolibar[1,2], Facundo Sapienza[3], Fabien Maussion[4], Redouane Lguensat[5], Bert Wouters[1], Fernando Pérez[3]

jordi.bolibar@univ-grenoble-alpes.fr

ØDINN

@ODINN_SciML

- Open source modelling framework based on **Universal Differential Equations** (UDEs, Rackauckas et al. 2020)



Modelling philosophy:

1. Make use of as much existing physical knowledge as possible.

2. Only use regressors (i.e. data-driven models) for the subparts of the equation that need to be learnt or expanded.

Bolibar et al. (2023)

- Focused on global glacier modelling
- Multi-language

julia python

The Datasets

- GlaThiDa ice thickness database
- Millan et al. (2022); Ice velocity and thickness of the world's glaciers
- Hugonnet et al. (2021); Accelerated global glacier mass loss in the early twenty-first century

We can use this framework to invert/calibrate glacier rheology in positions where we know both the ice thickness and ice surface

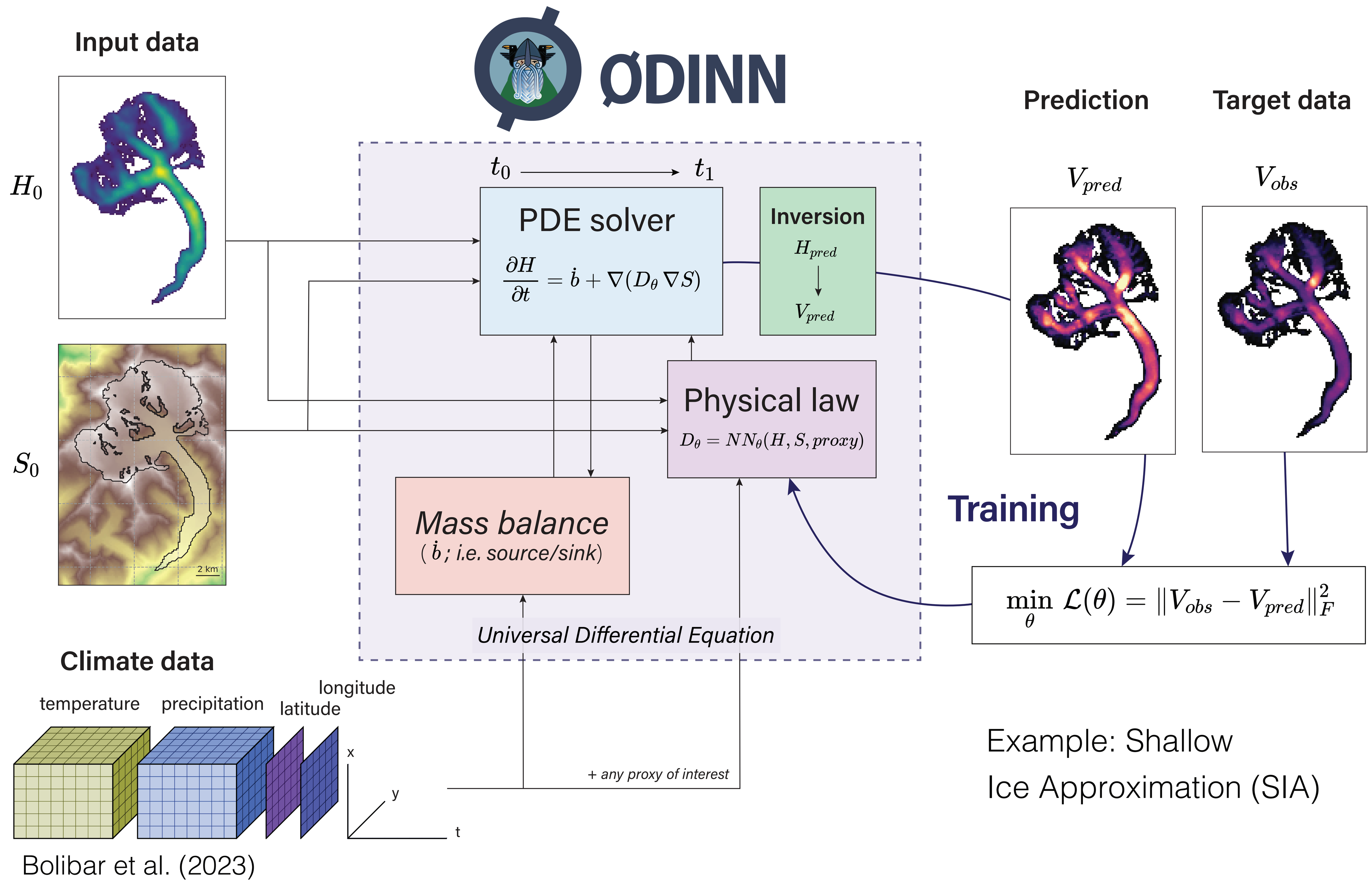
$$V_{\mathrm{surf}} = V_{\mathrm{basal}} + \frac{2A}{n+1}(\rho g)^n H^{n+1} \|\nabla S\|^{n-1} \nabla S$$

$$\downarrow$$

$$V_{\mathrm{surf}} = V_{\mathrm{surf}}(H_{\mathrm{obs}}, A_\theta)$$

$$\min_\theta \sum_{i \text{ glacier}} \mathrm{Loss}(V_{\mathrm{surf}}(H^i_{\mathrm{obs}}, A_\theta), V^i_{\mathrm{obs}})$$

Overview



Bolibar et al. (2023)

Example: Shallow Ice Approximation (SIA)

Functional inversions

The differential equation / inversion is inside the loss funcion that we try to minimize

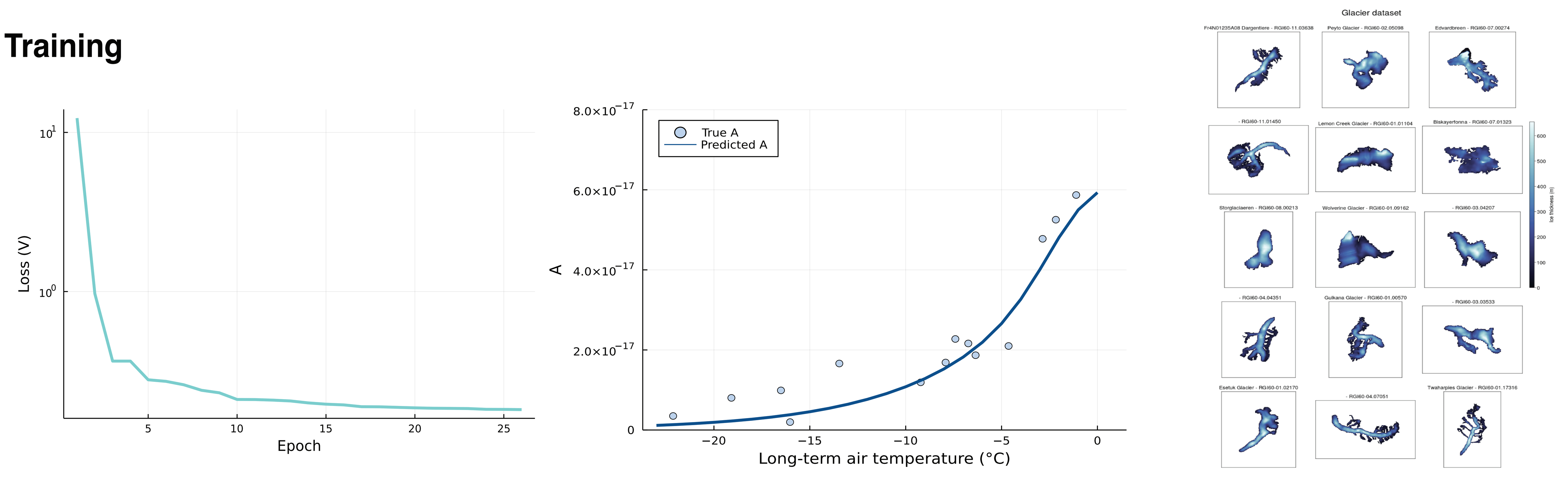
$$D = \left(C + \frac{2A}{n+2} H\right)(\rho g)^n H^{n+1} \|\nabla S\|^{n-1} \longrightarrow \min_A \mathrm{Loss}(V_0^{\mathrm{obs}}, \mathrm{Solver}(H_0^{\mathrm{obs}}, t_0, t_1, A))$$

Instead of adjusting single parameters based on observations, we find functional relations of these parameters with respect to other variables (e.g. long-term air temperature, bed properties).

$$D_\theta = \left(C + \frac{2A(T;\theta)}{n+2} H\right)(\rho g)^n H^{n+1} \|\nabla S\|^{n-1} \longrightarrow \min_\theta \sum_{i=\text{glaciers}} \mathrm{Loss}(V_i^{\mathrm{obs}}, \mathrm{Solver}(H_i^{\mathrm{obs}}, t_0, t_1, D_\theta))$$

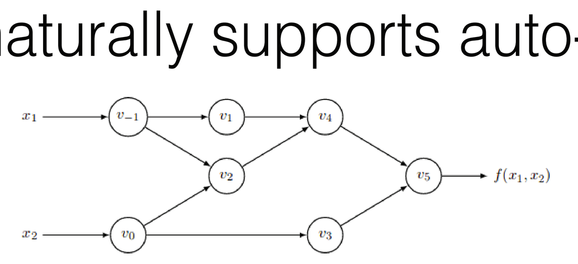
The optimized solution is a function that captures global patterns based on observed data.
This allow us to calibrate parameters based on global trends and as a function of some other observables.

Training



What's next for this research? What are your remaining questions?

- Efficient solvers for differential equations SciML
- Automatic differentiation: In order to minimize the loss function, we need to be able to compute gradients. Julia naturally supports automatic differentiation



- Discretized PDE into thousands of ODEs: memory and performance challenge when computing the sensitivities.
- Application with large Earth datasets to discover parametrizations of poorly represented physical processes of glaciers (e.g. basal sliding, ice viscosity, calving)

1 TU Delft
2 UGA Université Grenoble Alpes
3 Berkeley University of California
4 universität innsbruck
5 IRD Institut de Recherche pour le Développement France; IPSL