

Emulating a Subglacial Hydrology Model with a Neural Operator

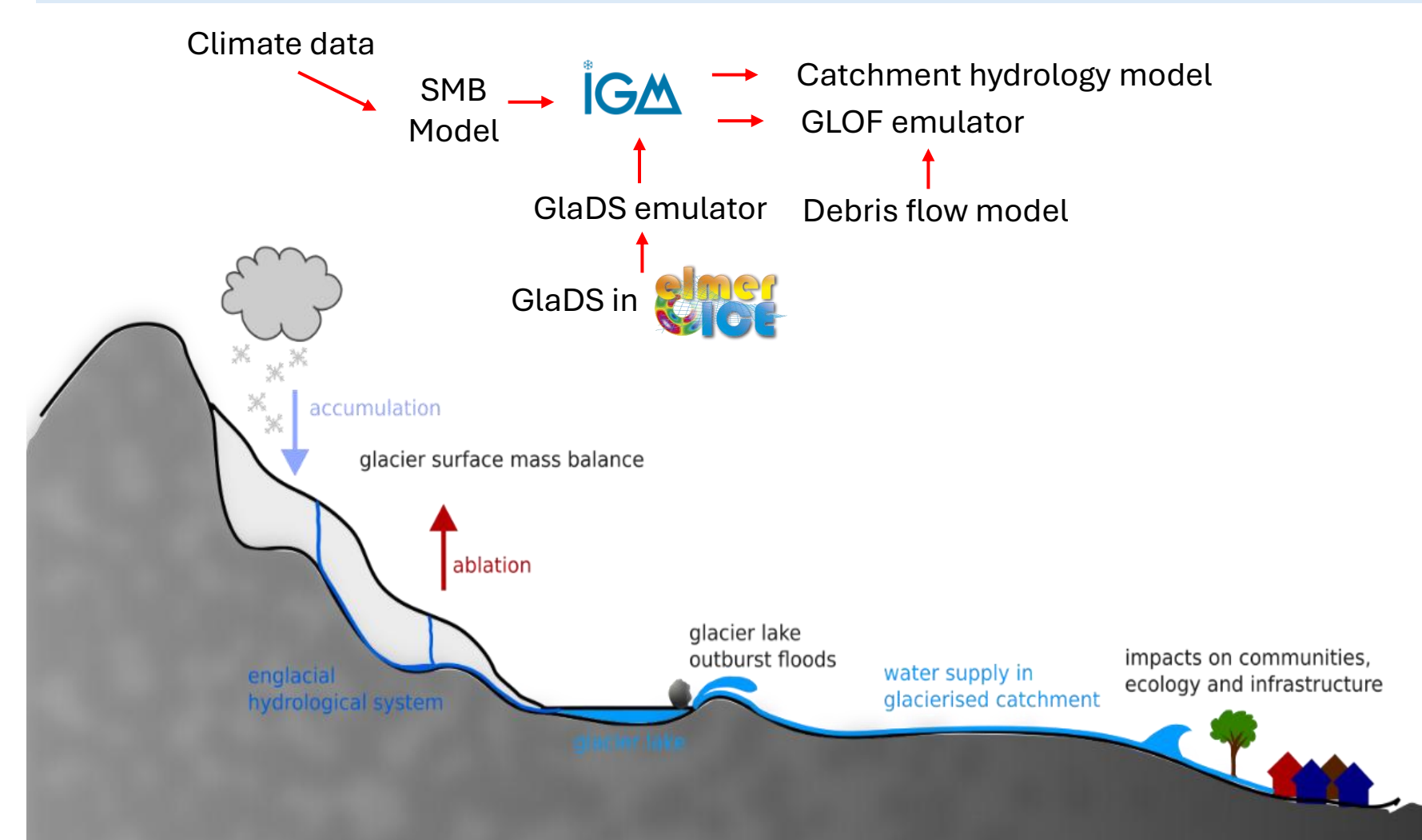
Nikola Jovanović¹, Samuel Cook¹, Thomas Zwinger², Johannes Fürst¹, Céline Walker¹

¹Friedrich-Alexander-Universität Erlangen-Nürnberg, Institute of Geography, Germany (nikola.jovanovic@fau.de)

²CSC - IT Center for Science Ltd., High Performance Computing, Espoo, Finland

1. Introduction

Subglacial hydrology exerts an important control on ice flow and influences the evolution of downstream hydrology, as well as the occurrence of glacial lake outburst floods. However, large-scale modelling of subglacial hydrology remains computationally expensive due to the presence of nonlinear processes, as well as processes acting on different spatial and temporal scales. Within the DeLIGHT (Deep-Learning-Informed Glacio-Hydrological Threat) framework, we aim at enabling coupled simulations of ice flow, subglacial hydrology, and downstream hydrology, with the goal of improving predictions of ice flow evolution and the timing of peak runoff. This research focuses on the development of a subglacial hydrology emulator trained using output from the Glacier Drainage System¹ (GlaDS) model, implemented within Elmer/Ice.



2. GlaDS in Elmer/Ice

1. GlaDS

- 2D subglacial hydrology model¹ implemented in Elmer/Ice
- Couples inefficient, distributed subglacial flow, with efficient, channelised flow
- Unstructured mesh
- Allows two-way coupling with an ice flow model

2. Model Setup

- Glacier outlines (RGI7)
- Mesh extrusion using Millan ice thickness²
- Moulin generation (the lowest elevation node in a Voronoi region, lower density of moulines with higher elevation³)
- Hydraulic potential, ϕ , equal to elevation potential ϕ_m at the glacier terminus
- Glacier discharge = channel flux + sheet discharge at the lowest elevation node



Figure 1: Workflow for generating a 3D mesh for GlaDS simulations.

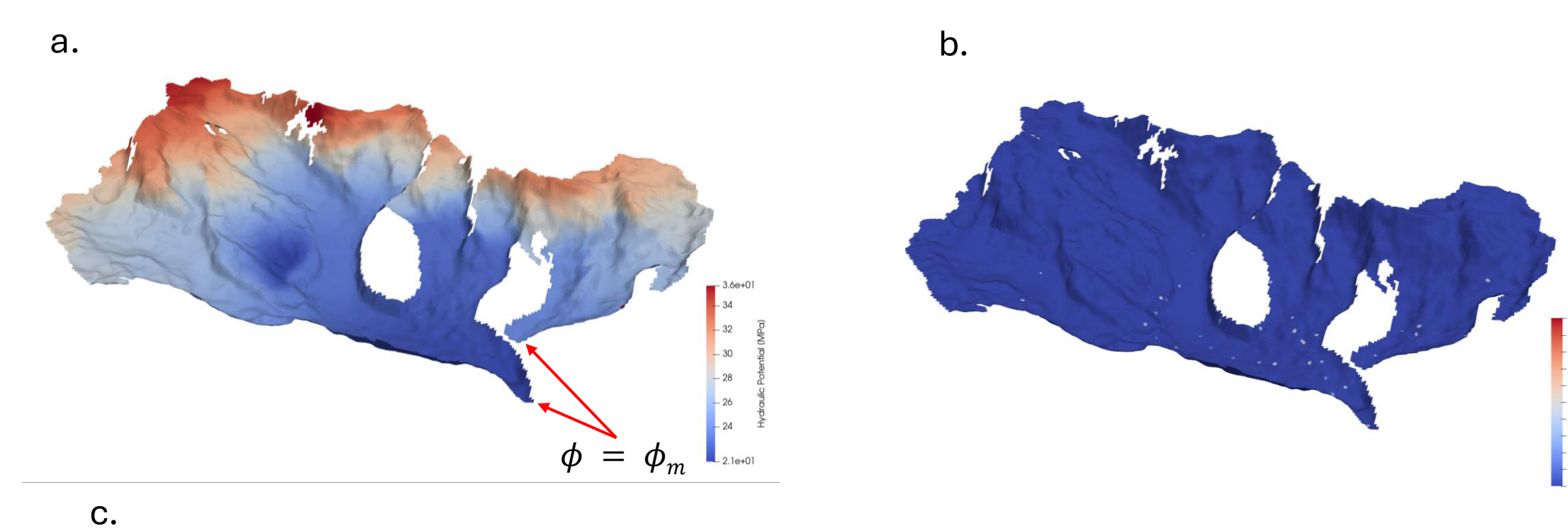


Figure 2: GlaDS configuration, Gornegletscher example. a) Hydraulic potential field, b) Moulines as lowest points in a Voronoi cell, c) Moulin flux using the reference (2000–2010) climate forcing.

3. Example Simulation

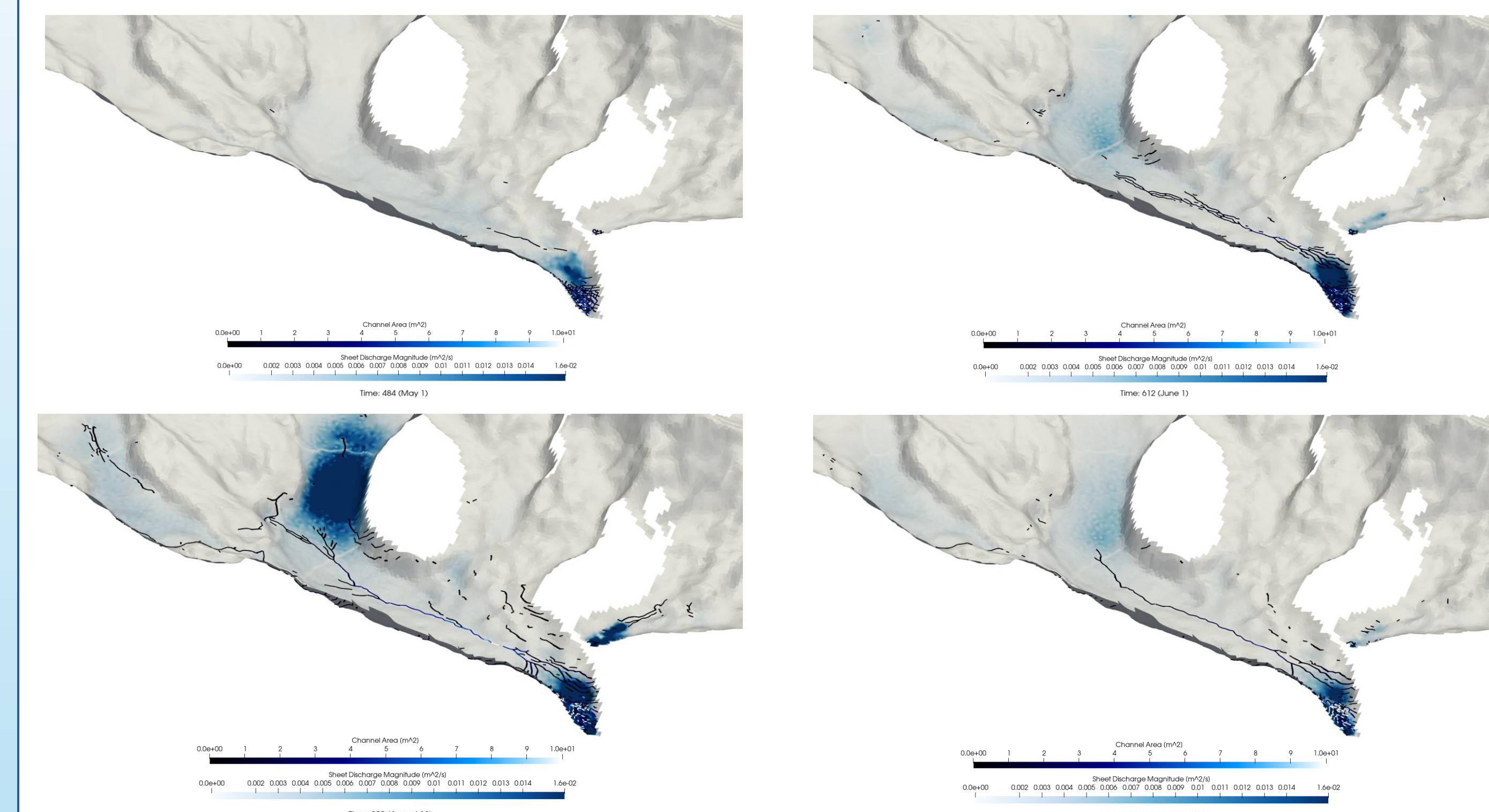


Figure 5: Gornegletscher subglacial system throughout a melt season

- GlaDS can simulate the growth of the channelised network at the outset of the melt season, as well as its closure towards the end of the melt season
- An efficient channelised system dominates at lower elevations during peak melt, whereas a linked cavity system dominates at higher elevations where the melt input is lower

4. Glacier Selection

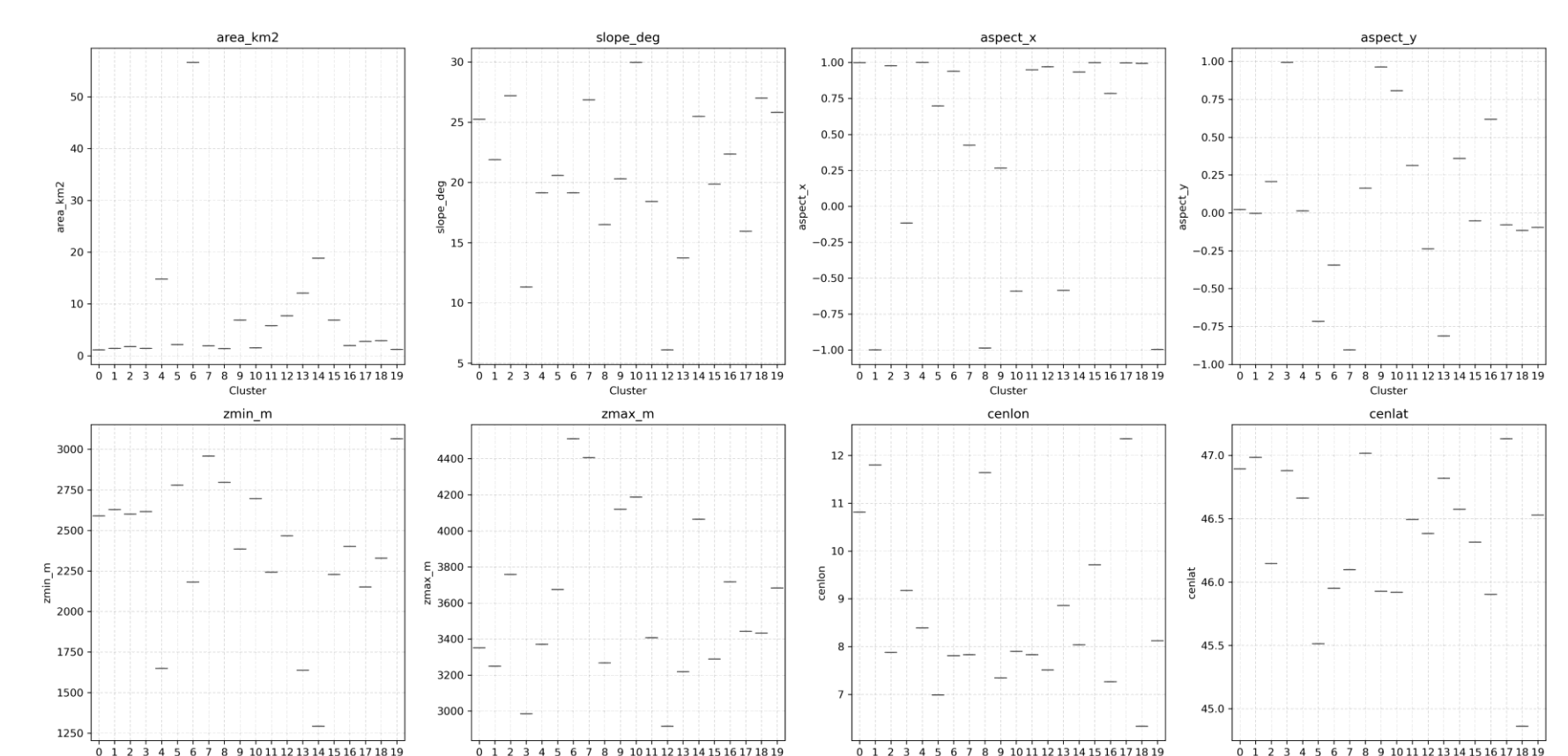


Figure 3: Selection of glaciers in the Alps (RGI 11) using K-means clustering.

- 70 glaciers selected from different regions: 10 in Svalbard (RGI7), 10 Scandinavia (RGI8), 20 in the Alps (RGI11), and 30 in Central (RGI13) and Southeast Asia (RGI14 and 15)
- RGI7 data – glaciers grouped in different clusters based on different characteristics: area, slope, aspect, elevation, latitude and longitude (K-means clustering)
- Minimum area $\sim 1 \text{ km}^2$ - bias towards smaller glaciers

5. Climate Forcing

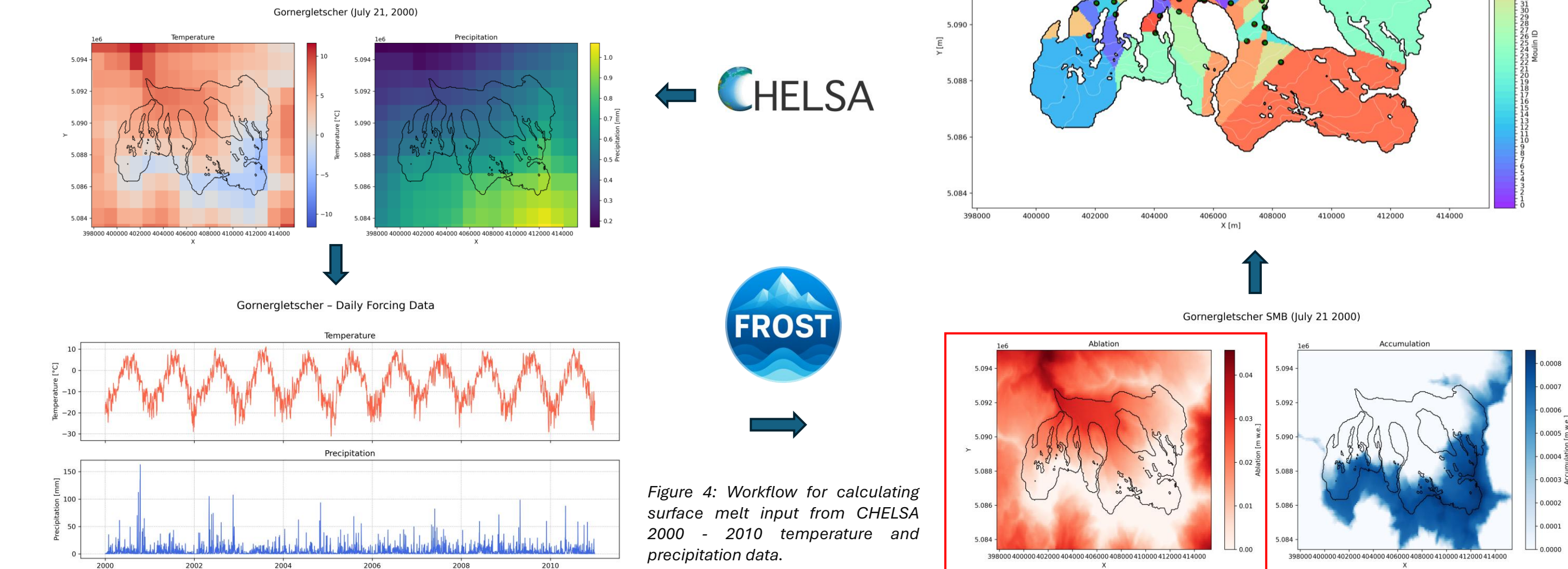


Figure 4: Workflow for calculating surface melt input from CHELSA 2000–2010 temperature and precipitation data.

6. Training Dataset Generation

Training strategies:

- 70 glaciers, 10 simulations per glacier
- 6h temporal resolution with the idea of capturing diurnal variation and 50 m spatial resolution for all glaciers
- Keeping the GlaDS parameter set the same across all simulations, neglecting basal melt – generalization to different climates and topographies
- Scaling surface melt input with different melt factors s_i , mimicking colder and warmer climate: $M_i = s_i \times M_{ref}$ where $s_i = \{0.5, 1.5\}$ (ex. below)
- Introducing noise to surface melt input to remove spatial bias across different scenarios

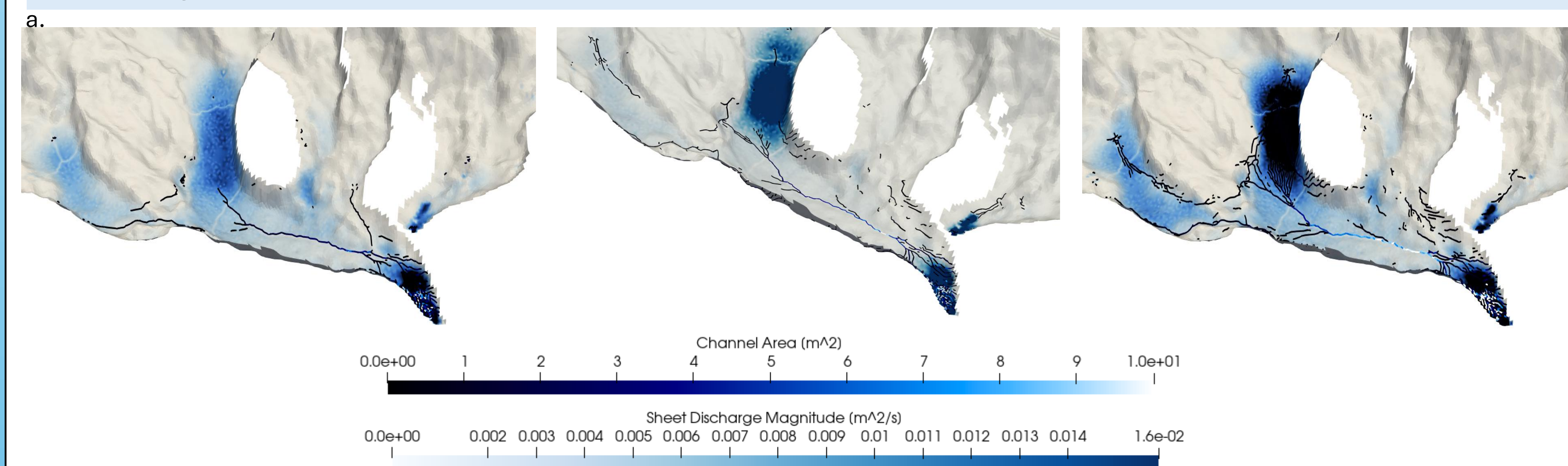


Figure 6: Gornegletscher subglacial hydrology simulations with a varying melt factor. a) Channel area and sheet discharge, b) Subglacial discharge.

7. Developing the Emulator

One of the biggest challenges in deep learning is generalisation to unseen data. Neural operators (NOs) can circumvent this obstacle, since they learn mappings between functions (i.e., infinite-dimensional spaces). We currently plan to use the so-called Fourier neural operator⁵ (FNO), that turns a convolution in real space to a point-wise multiplication in Fourier space and is therefore relatively fast. We intend the emulator to generalise across different topographies and climates, i.e., be applicable to any mountain glacier worldwide. Further, we plan to implement a physical constraint in the loss function, making the emulator physically-informed.

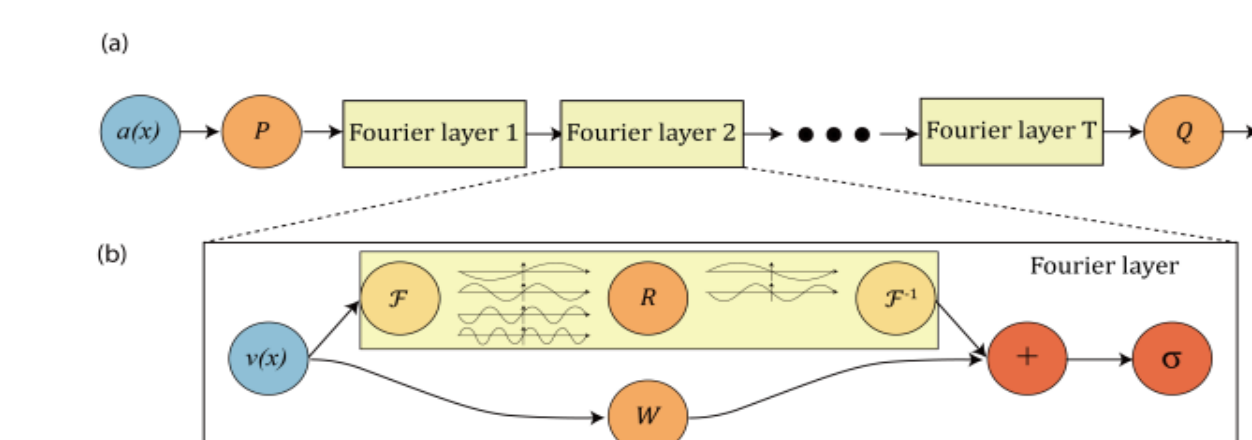


Figure 7: Fourier neural operator. a) Full architecture, b) Fourier layers. Illustration borrowed from the original publication⁵.

- Inputs for the emulator:
 - Ice thickness, bed topography, surface melt input, moulin locations, ice mask, basal velocity⁷
- Expected output from the emulator:
 - hydraulic potential, subglacial discharge
- Current state of the research: preprocessing the training data - current dataset is too large, selecting a smaller subset, deciding on glaciers for training and testing etc.
- Any ideas or directions on different architectures, training the NO, incorporating physics, and splitting the dataset are welcome!