Supporting Information for "Accelerating subglacial hydrology for ice sheet models with deep learning methods"

Vincent Verjans[1], Alexander Robel[1]

[1]School of Earth and Atmospheric Sciences, Georgia Institute of Technology, Atlanta, GA, USA

Contents of this file

1. Text S1 to S7
2. Tables S1 to S3
3. Figures S1 to S2

Introduction This Supporting Information provides all the methodological details of the study. Text S1 details the simulations performed with the Glacier Drainage System model. Text S2 details the architecture of the artificial neural network (ANN) developed in this study. Text S3 details our selection and processing of inputs for the ANN. Text S4 details how training, validation, and test data have been separated. Text S5 details the training procedure of the ANN. Text S6 details the configuration of the ice sheet model simulations, which are presented in section *Ice sheet model forcing* of the main text. Text S7 presents an additional sensitivity analysis to quantify the importance of each input in

Corresponding author: V. Verjans, School of Earth and Atmospheric Sciences, Georgia Institute of Technology, Atlanta, GA, USA (vverjans3@gatech.edu)

the performance of the ANN.

Table S1 provides the parameters used for the Glacier Drainage System model. Table S2 shows the architecture of the ANN. Table S3 shows the separation between training, validation, and test data. Figure S1 shows the configuration of the glaciers used for the subglacial hydrology model simulations. Figure S2 shows the results of the input importance sensitivity analysis.

All references are provided here, as well as in the reference list of the main text.

Text S1: Hydrology model simulations

We use the Glacier Drainage System model (GlaDS, Werder et al., 2013) implemented into the Ice-sheet and Sea-level System Model (ISSM, Larour et al., 2012) to generate data for this study. GlaDS is run separately over the seven calibration glaciers (Petermann, Jakobshavn, Helheim, Kangerlussuaq, Humboldt, Koge Bugt, and Russell, see Fig. S1) for 40 years with a two hour time step, saving outputs every three days. Outputs from these simulations are used to train, validate, and test our Artificial Neural Network (ANN) in mapping a set of inputs to a spatial field of hydraulic potential (ϕ). All domains have dimensions 100×100 km^2, except Petermann and Jakobshavn which have dimensions of 100×200 and 200×100 km^2, respectively (Fig. S1). We use a mesh resolution varying between 800 m in areas of fast ice flow and 2 500 m in areas of slow ice flow. We prescribe ice velocities from Joughin et al. (2017), and bedrock topography and ice thickness fields from Morlighem et al. (2017). We note that limited areas need bedrock smoothing to help with numerical stability of GlaDS for the Helheim, Petermann, and Jakobshavn domains (14%, 3%, and 4% of the domains, respectively). For our simulations, we integrate surface runoff over the glacier domains from the diurnal Energy Balance Model over the period 1970-2009 (Krebs-Kanzow et al., 2020). The surface runoff is directed to the bedrock at 30 locations representing moulins. The moulin locations are randomly distributed, under the conditions that ice thickness is greater than 500 m, ice velocity is greater than 25 m/yr, and that there is at least 10 km distance to the ocean, to the domain borders, and to any other moulin location (see Fig. S1). We refine the mesh resolution to 800 m around moulin locations to help with numerical stability. At any time step, runoff is equally partitioned between the 30 moulins.

We use an ice viscosity parameter corresponding to an ice temperature of 271.15 K in the parameterization of (Cuffey & Paterson, 2010). The spatial fields of the basal friction coefficient are obtained through an inversion method, based on the present-day geometry and ice velocities. GlaDS requires several parameters for the subglacial hydrology system. We take all the parameter values following the original implementation (Werder et al., 2013) and the default values of a recent intercomparison of subglacial hydrology models (de Fleurian et al., 2018). The parameter values of the subglacial hydrology system are listed in Table S1. Emulating GlaDS with other parameter values would require re-training the ANN. For the subglacial water system, we use zero-flux boundary conditions on the domain borders, and fixed hydrostatic ocean pressure at the grounding line. To preserve numerical stability, GlaDS is run with a 2 hour time step. Numerical instabilities still appeared in the simulations, in the form of infinite growth of the subglacial hydrological sheet thickness at some mesh elements on domain boundaries, close to the grounding line, or close to peripheral ice zones. This was caused by a negative cavity closing term, due to negative effective pressure values. At such mesh elements, we enforce a zero effective pressure, i.e, floatation.

In addition to these seven simulations, we perform a simulation at a test glacier (Upernavik, Fig. S1h) to generate additional test data. This simulation uses the same strategy for runoff generation and meshing as described above. The domain of Upernavik is of size 100×100 km^2, and all the GlaDS parameter values remain the same (Table S1).

We note here that GlaDS calculations can lead to unphysical negative water pressure values. Water pressure, p_w is defined as the hydraulic potential, ϕ, minus the elevation potential: $p_w = \phi - \phi_m$, where $\phi_m = \rho_w g B$ with ρ_w, g, B being water density, gravitational

acceleration, and bedrock elevation, respectively. In the GlaDS routine for computing ϕ, there is no constraint on enforcing that $\phi \geq \phi_m$, and negative p_w values can arise in areas with high bed elevation and thin ice thickness (Siu, 2022). This only affects zones of peripheral ice.

Text S2: Architecture of the Artificial Neural Network

Our ANN is implemented with the Pytorch library (Paszke et al., 2019), and is a modified version of the U-Net architecture developed in Ronneberger et al. (2015). Our ANN architecture, detailed in Table S2, consists of an encoding and a decoding pathway. In the encoding stage, features are extracted from the two-dimensional input fields and spatial resolution is progressively reduced. Encoding is performed through a series of down-convolution blocks. Each down-convolution block consists of three operations: two convolution operations, each with a nonlinear activation function, and one pooling operation. The convolutions use a 3×3 kernel size, a stride of 1, zero-padding, and a bias term. We use the ReLU activation function after each convolution:

$$\mathrm{ReLU}(x) = \max(0, x). \tag{1}$$

The pooling operation is a 2×2 max-pooling, and thus reduces the horizontal resolution at each down-convolution block by a factor of 2 along each spatial dimension. The number of output features from the first down-convolution block is 24, and is then doubled for each subsequent down-convolution block. We experimented with different numbers of output features from the first down-convolution block, and found that 24 gives optimal model performance. The last down-convolution does not use pooling and has 94 output features instead of 96 to allow concatenation of two additional inputs at the end of the encoding stage. Specifically, we concatenate the time inputs to the 94 features. The time

inputs are the cosine and sine of the time step (in units of years) multiplied by 2π. The decoding stage is symmetric to the encoding stage. It consists of up-convolution blocks. Each up-convolution block consists of four operations: one transpose convolution, one concatenation, and two convolutions. Every transpose convolution uses a 2×2 kernel size, halves the number of features, and enhances the horizontal resolution by a factor of 2 along each spatial dimension. The concatenation process allows to concatenate the features from the corresponding encoding level, allowing propagation of information from higher-resolution features. The convolution operations are similar to those from the encoding stage. We exclude the concatenation operation from the last up-convolution block, as we found that excluding it improves the spatial smoothness of the results from the ANN, in better agreement with the validation data. This is explained by not passing information at the high resolution of the initial data directly to the last up-convolution block, but rather forcing all features to undergo at least one pooling operation. The final layer of the ANN is a 1×1 convolution operation. No activation function is applied to this last convolution, of which the output is the standardized predicted ϕ field.

Text S3: Inputs to the Artificial Neural Network

Because our ANN is a convolutional neural network, its input consists of two-dimensional images, referred to as input features. We use as input features the bed topography, the ice thickness, and the ice velocity fields. These input features are fixed in time, and are therefore the same at any time step. However, they differ between the seven calibration glaciers, and are thus different for training samples corresponding to the different glaciers. In addition to these three input features, we use the spatial distribution of surface meltwater inflow. Surface meltwater inflow is non-zero only at moulin locations,

over which the meltwater is distributed uniformly at any time step. We integrate the spatial meltwater inflow over different past time periods to be provided as input features to the ANN. To select these past time periods, we use a feature selection method. This consists of adding an increasing number of input features until no performance gain is achieved. In our feature selection method, our baseline case is considering only the instantaneous meltwater inflow, and meltwater inflow integrated over the previous 10 days. The first step is to add also meltwater inflow integrated over the previous month. The second step is to add meltwater inflow integrated over month-minus-1 to month-minus-2. We proceed iteratively, adding one month of meltwater inflow information at a time. We find that the optimal combination of features includes meltwater inflow (i) at the current time step, integrated over (ii) the previous 10 days, (iii) the previous month, (iv) month-minus-1 to month-minus-2, (v) month-minus-2 to month-minus-3, and (vi) month-minus-3 to month-minus-4. Including months beyond this time period does not improve model performance, when evaluated on the validation data. However, through our feature selection process, we find that adding (vii) the meltwater inflow integrated over the entire previous year further improves the ANN performance. This feature selection process results in a total of 12 inputs: 10 two-dimensional input features, and the cosine and sine of the time step. One input sample thus consists of the 12 inputs for a given glacier at a given time step. While GlaDS is run at a 2-hourly time step to ensure numerical stability, model outputs are saved every three days for storage reasons.

The two-dimensional input features have an inherent spatial scale, which our ANN is sensitive to. For this reason, we consistently train and evaluate our ANN over 100×100 km^2 windows. GlaDS runs on an irregular finite-element mesh, but results are bilinearly

interpolated on a regular 128×128 mesh. It is important to preserve consistency in the spatial scaling, and using the ANN for predictions over a domain size different than the domain size used for training would be inappropriate. However, for larger domains of interest, it is straightforward to use the ANN multiple times over separate 100×100 km^2 parts of the domain, and concatenate the results. This is what has been done for the Jakobshavn and Petermann glaciers in this study, each being separated in two subdomains. As such, our data set of 8 glaciers corresponds to 10 domains. The ANN could also be trained, and thus used for predictions, on any other domain size.

As explained in Section *Architecture of the Artificial Neural Network*, we concatenate the time input at the end of the encoding stage as cosine and sine of the current time multiplied by 2π (Table S2). This improves the ANN performance due to the assymetry between early- and late-melt season behavior of the subglacial hydrology system. The time inputs are not passed in the first input layer because they are not spatial fields. Still, passing them at the end of the encoding stage allows the ANN to capture interactions between time of year and the other inputs through the decoding stage. Each time step is treated independently by the ANN, but our method of integrating past meltwater inflow provides, de facto, some temporal dependence. Future work can focus on associating the convolutional structure of our ANN with recurrent neural networks, which explicitly simulate temporal dependencies.

Text S4: Separation of training, validation, and test data

The first 5 years (1970-1974) of the GlaDS simulations are discarded, as GlaDS evolves transiently from an arbitrary initial state. For the seven calibration glaciers, years 1975-2004 are used as calibration years, and years 2005-2009 are preserved for test data. The

data of the seven calibration glaciers consist of nine domains because two glaciers (Jakobshavn and Petermann) span the size of two domains, and are therefore provided separately to the ANN. The calibration years are further split between training and validation data, with 90% (years 1975-2001) and 10% (years 2002-2004) of the data, respectively. For the test glacier (Upernavik), all the data (years 1975-2009) are preserved for test data. Furthermore, we proceed to data augmentation by applying three transformations of each glacier domain and its input fields to use as additional training and validation data. Data augmentation improves performance of artificial neural networks by increasing the amount of data for calibration (Lemley et al., 2017). The transformations are a vertical axial symmetry, a horizontal axial symmetry, and a diagonal axial symmetry. For each training glacier, one of these transformations is used exclusively as validation data , and the two others are used as training (years 1975-2001) and validation (years 2002-2004) data. The splitting of the data between training, validation, and test data is detailed in Table S3.

Text S5: Training of the Artificial Neural Network

For training efficiency, we standardize every two-dimensional input feature and the ϕ output feature, such that our variables have zero mean and unit standard deviation. For predictions, our ANN thus requires inputs standardized accordingly, and predicted ϕ must be rescaled accordingly. Cosine and sine of time are not scaled, because they range between -1 and 1. We initialize the parameters of our ANN using the He Normal initialization method (He et al., 2015). We train our ANN with the training data such that parameter values are updated through backpropagation by minimizing a loss function measuring the misfit between ϕ fields predicted by the ANN and the GlaDS output. We

use the L2 loss function, as it showed better results than when using alternative loss functions such as mean absolute error or the Huber loss. The L2 loss is defined as:

$$L(\boldsymbol{\phi}_{ANN}) = \sqrt{\frac{1}{N}\sum_{i}\left(\phi_{i,ANN} - \phi_{i,GlaDS}\right)^2}, \tag{2}$$

where $\phi_{i,GlaDS}$ denotes a ϕ value calculated by GlaDS, $\phi_{i,ANN}$ denotes the corresponding value of ϕ calculated by the ANN, and $\boldsymbol{\phi}_{ANN}$ denotes the full sample of ϕ values calculated by the ANN, with dimensions determined by the number of samples, and by the number of pixels in the two-dimensional spatial domain. In the loss calculation, we exclude all pixels with ice thickness less than 20 m or ice velocity less than 5 m/yr. Simulating ϕ in such regions is not necessary, as ice flow variability has minimal impact on ice sheet dynamics. And, because these outlier regions lead to different behaviors of subglacial hydrology models, we prefer to make the ANN calibration insensitive to these regions. During training, an epoch consists of passing the entire training data in sequences of randomly selected batches to the ANN. We use a batch size of 32 samples, as we found that it results in optimal model performance and training speed. The loss function is evaluated on the training batch, used to update parameter values via backpropagation, and on the validation data. Our backpropagation algorithm uses the Adam optimizer (Kingma & Ba, 2014) with an initial learning rate of 0.001. We use an adaptive learning rate, decreasing it by a factor of 2 after 5 consecutive epochs without improving the validation loss. This allows more localized search in the parameter space as the training procedure approaches a local minimum of the loss function. We stop the training after 10 consecutive epochs without improving the validation loss to avoid overfitting the training data. The final parameter values saved from the training procedure are those having led to the best validation loss score. As an additional tool to avoid overfitting, we use

dropout. Similarly to (Ronneberger et al., 2015), we implement dropout only after the last convolution operation of the encoding stage (layer 8 in Table S2), and we use a dropout probability of 0.2. We train separately an ensemble of 20 networks. The training procedure is identical for these networks, and they only differ due to the random initialization of the parameters and the randomness of the optimization algorithm. Our final ANN is the ensemble mean output of these 20 members, as this averaging approach has been shown to improve deep neural network performance (Lakshminarayanan et al., 2016).

Text S6: Details on ice sheet model runs

The ice sheet model runs at Upernavik glacier, shown in section *Ice sheet model runs*, are performed using the Ice sheet and Sea-level System model (Larour et al., 2012). The ice rheology and basal friction coefficient parameters are kept identical as in the GlaDS simulations (see Supporting Information). As initial conditions, we use the ice geometry and ice velocity fields applied in the GlaDS runs. We prescribe a surface mass balance field that is constant in space and time, which is taken as the mean 1970-2009 surface mass balance averaged over the domain from the diurnal Energy Balance Model (Krebs-Kanzow et al., 2020). We start the simulations from 1975, to avoid impacts from the first 5 years of GlaDS run, during which GlaDS evolves from an arbitrary initial state. We perform a GlaDS-forced run, which applies the p_w field as predicted by GlaDS. Similarly, we perform an ANN-forced run, which applies the p_w field as predicted by the ANN. Ice flow dynamics are coupled to p_w through the basal sliding law, for which we use the Budd sliding law (Budd et al., 1984):

$$\boldsymbol{\tau_b} = -C^2 \boldsymbol{u_b} N, \tag{3}$$

where $\boldsymbol{\tau_b}$ is the basal stress [Pa], $\boldsymbol{u_b}$ is the basal ice velocity [m yr^{-1}], and C^2 is the basal friction coefficient, varying in space [m^{-1} yr]. We also perform a control run, in which the p_w assumes a simple hydrostatic connection to the ocean: $p_w = -g\rho_w B$. The control run captures the transient changes caused by the initial ice geometry not being in equilibrium. We subtract these transient changes from the GlaDS-forced and ANN-forced runs when analyzing the results in terms of ice thickness and ice velocities. As existing ice sheet sliding laws are not applicable at very low effective pressures, we follow (Ehrenfeucht et al., 2022) in applying a lower limit on N equal to 6% of the ice overburden pressure.

Text S7: Input importance

We evaluate the importance of each input feature to the quality of the ANN predictions. To this end, we perturb randomly each input feature individually. After each perturbation, we use the ANN to predict a $\widetilde{\boldsymbol{\phi}}_{ANN}$ field, which is of lower accuracy than the $\boldsymbol{\phi}_{ANN}$ field predicted without input perturbation. To perturb a given input feature, we add white Gaussian noise of standard deviation 1 to the input feature. We add the noise to the standardized features. In this way, each input feature is perturbed in a similar fashion, because all the standardized input features have mean 0 and standard deviation 1 by construction. The cosine and sine of time are not standardized. As such, we perturb these variables by their standard deviation, which is approximately 0.67. We evaluate $\widetilde{\boldsymbol{\phi}}_{ANN}$ on the validation data of the non-transformed glacier domains (see Table S3). Figure S2 shows the ratio in coefficient of determination (R^2), in Root Mean Square Error (RMSE), and in absolute bias of $\widetilde{\boldsymbol{\phi}}_{ANN}$ with respect to these metrics evaluated with $\boldsymbol{\phi}_{ANN}$. These ratios thus show the reduction in model accuracy caused by each perturbation. Results are shown for each perturbed input individually. Figure S2 shows

that perturbing the ice thickness input has the most consequential impact on the ANN accuracy, as both the R^2 coefficient and the RMSE are more strongly impacted than when perturbing any other input. The different time periods over which we integrate the meltwater inflow show a similar importance on the ANN accuracy, except for the period over the entire previous year, which has a stronger impact on accuracy. This could be due to the strong correlation between each meltwater inflow input with its neighboring meltwater inflow input, for example month-minus-0 to month-minus-1 and month-minus-1 to month-minus-2. In contrast, the meltwater inflow integrated over the entire previous year has no neighboring time period with which it is strongly correlated. Finally, we find that perturbing any input causes a decrease in the ANN accuracy, which implies that all the inputs are to some extent useful for prediction.

Table S1. Parameters of the GlaDS simulations

Parameter	Value	Units
Englacial void ratio	10^{-3}	/
Pressure melt coefficient	7.5×10^{-8}	K Pa^{-1}
Latent heat of fusion	334×10^{3}	J kg^{-1}
Bedrock bump height	0.1	m
Cavity spacing	2.0	m
Sheet conductivity	0.01	$m^{7/4}$ $kg^{-1/2}$
Channel conductivity	0.1	$m^{3/2}$ $kg^{-1/2}$

Table S2. Architecture of the ANN

Layer number	Layer used as input	Layer type	Activation	Output shape
0	-	Input	-	128×128×10
1	0	Conv 3×3	ReLU	128×128×24
2	1	Conv 3×3	ReLU	128×128×24
3	2	MaxPool 2×2	-	64×64×24
4	3	Conv 3×3	ReLU	64×64×48
5	4	Conv 3×3	ReLU	64×64×48
6	5	MaxPool 2×2	-	32×32×48
7	6	Conv 3×3	ReLU	32×32×94
8	7	Conv 3×3	ReLU (dropout p=0.2)	32×32×94
9	8,cos(2πt),sin(2πt)	Concat	-	32×32×96
10	9	Trans-Conv 2×2	-	64×64×48
11	5,10	Concat	-	64×64×96
12	11	Conv 3×3	ReLU	64×64×48
13	12	Conv 3×3	ReLU	64×64×48
14	13	Trans-Conv 2×2	-	128×128×24
15	14	Conv 3×3	ReLU	128×128×24
16	15	Conv 3×3	ReLU	128×128×24
17	16	Conv 1×1	-	128×128×1

Table S3. Training, validation, and test data split

Glacier (subdomain)	Transformation	Years 0-5	Years 5-32	Years 32-35	Years 35-40
Jakobshavn (0)	None	Discarded	Train	Validation	Test
Jakobshavn (0)	Diagonal symmetry	Discarded	Validation	Validation	Discarded
Jakobshavn (0)	Vertical symmetry	Discarded	Train	Validation	Discarded
Jakobshavn (0)	Horizontal symmetry	Discarded	Train	Validation	Discarded
Jakobshavn (1)	None	Discarded	Train	Validation	Test
Jakobshavn (1)	Diagonal symmetry	Discarded	Train	Validation	Discarded
Jakobshavn (1)	Vertical symmetry	Discarded	Validation	Validation	Discarded
Jakobshavn (1)	Horizontal symmetry	Discarded	Train	Validation	Discarded
Helheim (0)	None	Discarded	Train	Validation	Test
Helheim (0)	Diagonal symmetry	Discarded	Train	Validation	Discarded
Helheim (0)	Vertical symmetry	Discarded	Train	Validation	Discarded
Helheim (0)	Horizontal symmetry	Discarded	Validation	Validation	Discarded
Petermann (0)	None	Discarded	Train	Validation	Test
Petermann (0)	Diagonal symmetry	Discarded	Validation	Validation	Discarded
Petermann (0)	Vertical symmetry	Discarded	Train	Validation	Discarded
Petermann (0)	Horizontal symmetry	Discarded	Train	Validation	Discarded
Petermann (1)	None	Discarded	Train	Validation	Test
Petermann (1)	Diagonal symmetry	Discarded	Train	Validation	Discarded
Petermann (1)	Vertical symmetry	Discarded	Validation	Validation	Discarded
Petermann (1)	Horizontal symmetry	Discarded	Train	Validation	Discarded
Kangerlussuaq (0)	None	Discarded	Train	Validation	Test
Kangerlussuaq (0)	Diagonal symmetry	Discarded	Train	Validation	Discarded
Kangerlussuaq (0)	Vertical symmetry	Discarded	Train	Validation	Discarded
Kangerlussuaq (0)	Horizontal symmetry	Discarded	Validation	Validation	Discarded
Humboldt (0)	None	Discarded	Train	Validation	Test
Humboldt (0)	Diagonal symmetry	Discarded	Validation	Validation	Discarded
Humboldt (0)	Vertical symmetry	Discarded	Train	Validation	Discarded
Humboldt (0)	Horizontal symmetry	Discarded	Train	Validation	Discarded
Koge Bugt (0)	None	Discarded	Train	Validation	Test
Koge Bugt (0)	Diagonal symmetry	Discarded	Train	Validation	Discarded
Koge Bugt (0)	Vertical symmetry	Discarded	Validation	Validation	Discarded
Koge Bugt (0)	Horizontal symmetry	Discarded	Train	Validation	Discarded
Russell (0)	None	Discarded	Train	Validation	Test
Russell (0)	Diagonal symmetry	Discarded	Train	Validation	Discarded
Russell (0)	Vertical symmetry	Discarded	Train	Validation	Discarded
Russell (0)	Horizontal symmetry	Discarded	Validation	Validation	Discarded
Upernavik (0)	None	Discarded	Test	Test	Test

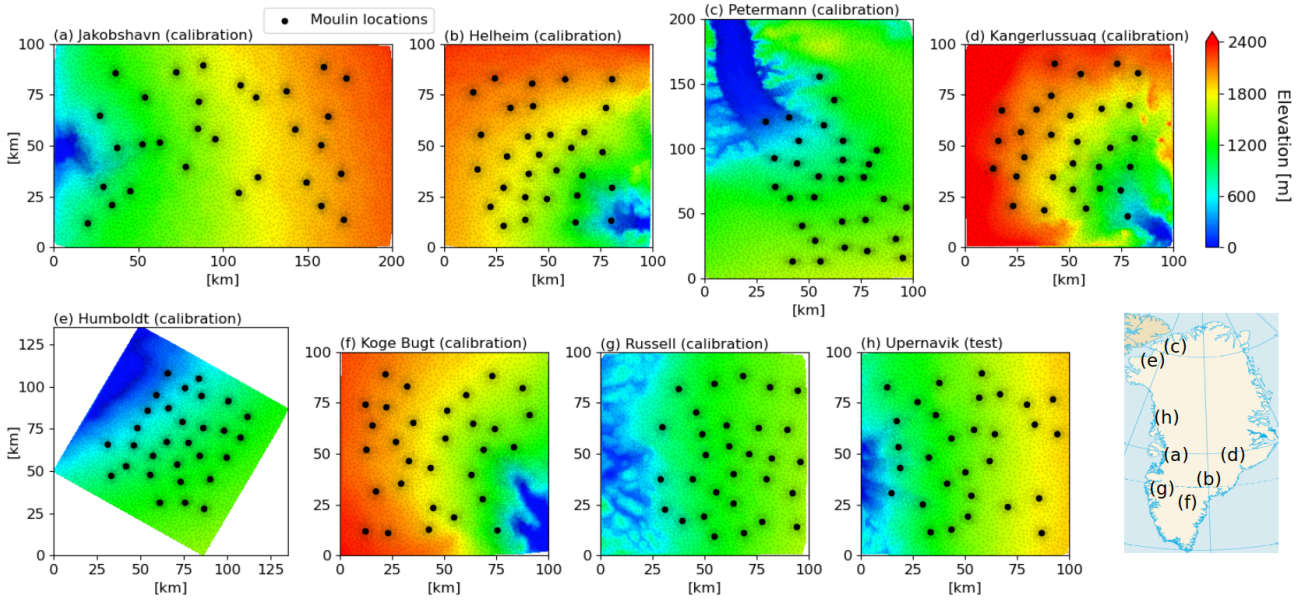

Figure S1. Model domains of the seven calibration glaciers ((a) Jakobshavn, (b) Helheim, (c) Petermann, (d) Kangerlussuaq, (e) Humboldt, (f) Koge Bugt, (g) Russell), and of the test glacier ((h) Upernavik). Map in the inset shows glacier locations. Light-grey points show mesh vertices.

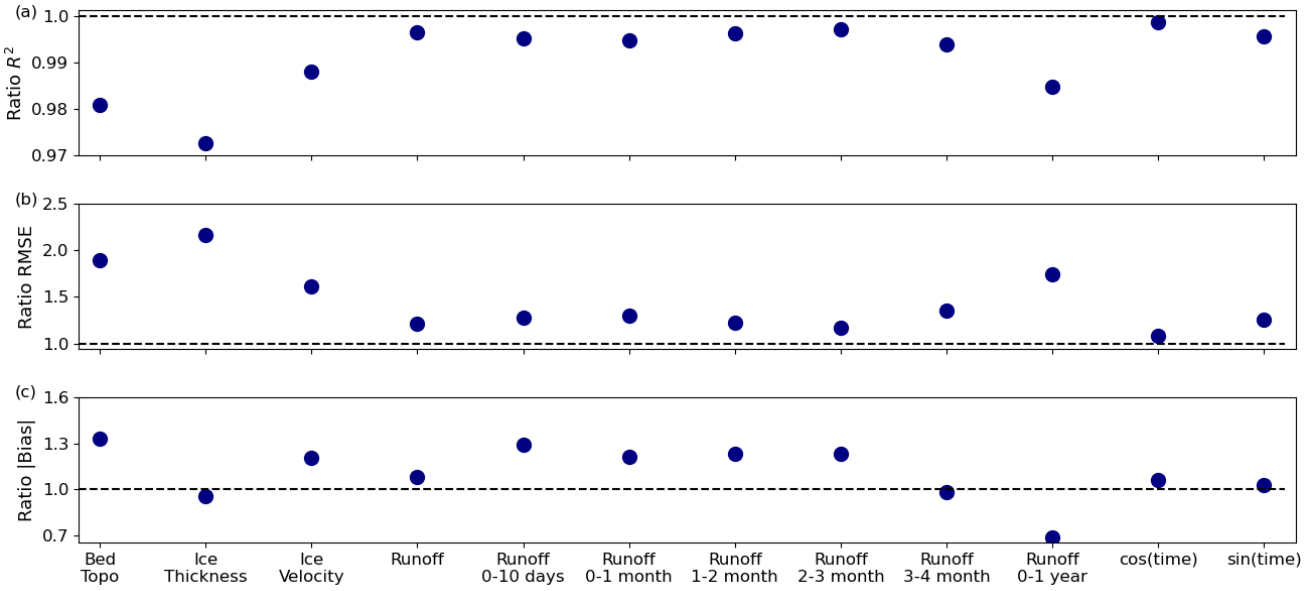

Figure S2. Ratio of performance metrics on the validation data after random white noise perturbation of input fields. The ratio is computed as the performance metric of the ANN with input perturbation with respect to the performance metric of the ANN without input perturbation. Metrics are (a) the coefficient of determination, (b) the Root Mean Square Error, and (c) the absolute bias. The black horizontal dashed line shows the value of 1, corresponding to no performance deterioration due to random input perturbation. Bed Topo is bed topography, Runoff denotes the meltwater inflow at the instantaneous time step, Runoff x-y period denotes the meltwater inflow integrated over the time interval between the past period x and period y.

References

Budd, W. F., Jenssen, D., & Smith, I. N. (1984). A three-dimensional time-dependent model of the west antarctic ice sheet. *Annals of Glaciology*, *5*, 29 - 36.

Cuffey, K. M., & Paterson, W. S. B. (2010). The physics of glaciers ed. 4..

de Fleurian, B., Werder, M. A., Beyer, S., Brinkerhoff, D. J., Delaney, I., Dow, C. F., ... Sommers, A. N. (2018). Shmip the subglacial hydrology model intercomparison project. *Journal of Glaciology*, *64*, 897 - 916.

Ehrenfeucht, S., Morlighem, M., Rignot, E., Dow, C. F., & Mouginot, J. (2022). Seasonal acceleration of petermann glacier, greenland, from changes in subglacial hydrology. *Geophysical Research Letters*, *50*.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, 1026-1034.

Joughin, I. R., Smith, B. E., & Howat, I. M. (2017). A complete map of greenland ice velocity derived from satellite data collected over 20 years. *Journal of Glaciology*, *64*, 1 - 11.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, *abs/1412.6980*.

Krebs-Kanzow, U., Gierz, P., Rodehacke, C. B., Xu, S., Yang, H., & Lohmann, G. (2020). The diurnal energy balance model (debm): a convenient surface mass balance solution for ice sheets in earth system modeling. *The Cryosphere*, *15*, 2295-2313.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2016). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Nips.*

Larour, E. Y., Seroussi, H., Morlighem, M., & Rignot, E. (2012). Continental scale, high order,

high spatial resolution, ice sheet modeling using the ice sheet system model (issm). *Journal of Geophysical Research*, *117*.

Lemley, J., Bazrafkan, S., & Corcoran, P. M. (2017). Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, *5*, 5858-5869.

Morlighem, M., Williams, C. N., Rignot, E., An, L., Arndt, J. E., Bamber, J. L., ... Zinglersen, K. B. (2017). Bedmachine v3: Complete bed topography and ocean bathymetry mapping of greenland from multibeam echo sounding combined with mass conservation. *Geophysical Research Letters*, *44*, 11051 - 11061.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Neural information processing systems.*

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, *abs/1505.04597*.

Siu, K. (2022). Modelling subglacial hydrology under future climate scenarios in wilkes subglacial basin, antarctica..

Werder, M. A., Hewitt, I. J., Schoof, C., & Flowers, G. E. (2013). Modeling channelized and distributed subglacial drainage in two dimensions. *Journal of Geophysical Research: Earth Surface*, *118*, 2140 - 2158.