Deep Learning Emulator for Greenland Ice Sheet Runoff

Elke Schlager[1, 2], Peter L. Langen[1], Ruth H. Mottram[2] and Andreas P. Ahlstrøm[3]

[1]Dept. Environmental Science, iClimate, Aarhus University, [2]National Centre for Climate Research (NCKF), Danish Meteorological Institute, [3]Geological Survey of Denmark and Greenland

1. Introduction & Motivation

Mass loss of the Greenland ice sheet not only contributes to global mean sea-level rise, but also has various consequences on local and regional scales. Development of adequate mitigation and adaptation efforts regarding climate change requires knowledge of the full range of possible future runoff scenarios. Traditional runoff models rely on Regional Climate Models (RCM) to refine General Circulation Models (GCM) data for a smaller geographic area. This downscaling is computationally too expensive to obtain comprehensive projections. Therefore, we develop a runoff emulator that is efficient enough to allow for production of large runoff ensemble projections. However, RCM data does not directly constitute a spatially higher-resolution rendition of the GCM, as it is solely influenced by the GCM through boundary conditions but exhibits some variability within its domain. Furthermore, we want to predict runoff, which is a variable not present in our low-resolution data but is computed by the RCM and depends on the ice sheet properties that evolve over time. We address these issues by including various additional information in our model.

2. Datasets

The proposed approach relies on climate model data of different spatial resolutions:
Climate data from RCM with a coupled snow/ice subsurface scheme (~5-12km resolution, referred to as HR data):

- main target: runoff
- additional targets (optional): temperature, pressure, rainfall, snowfall, snowmelt, etc.
- Input: HR elevation maps (and down-sampled elevation to medium resolution (MR))

Climate data from the driving GCM as model input (~30-100km resolution, referred to as LR data),

- Variables from Greenland area (local features)
- Variables from whole North Atlantic region (non-local/remote features)
- Monthly aggregated variables of previous months that influence the characteristics of the snowpack and thus runoff

3. Model Design: Modules

The proposed model consists of several components. The baseline network consists of an Encoder-Decoder structure with additional layers for super-resolution and final computation of the runoff prediction map $\hat{R}_t^{(HR)}$. All the other components in the network are optional and are each designed to serve a distinct purpose:

A. A convolutional architecture is used for local feature extraction from LR data of Greenland area. Super-resolution techniques are applied to reconstruct the HR data from those extracted features.

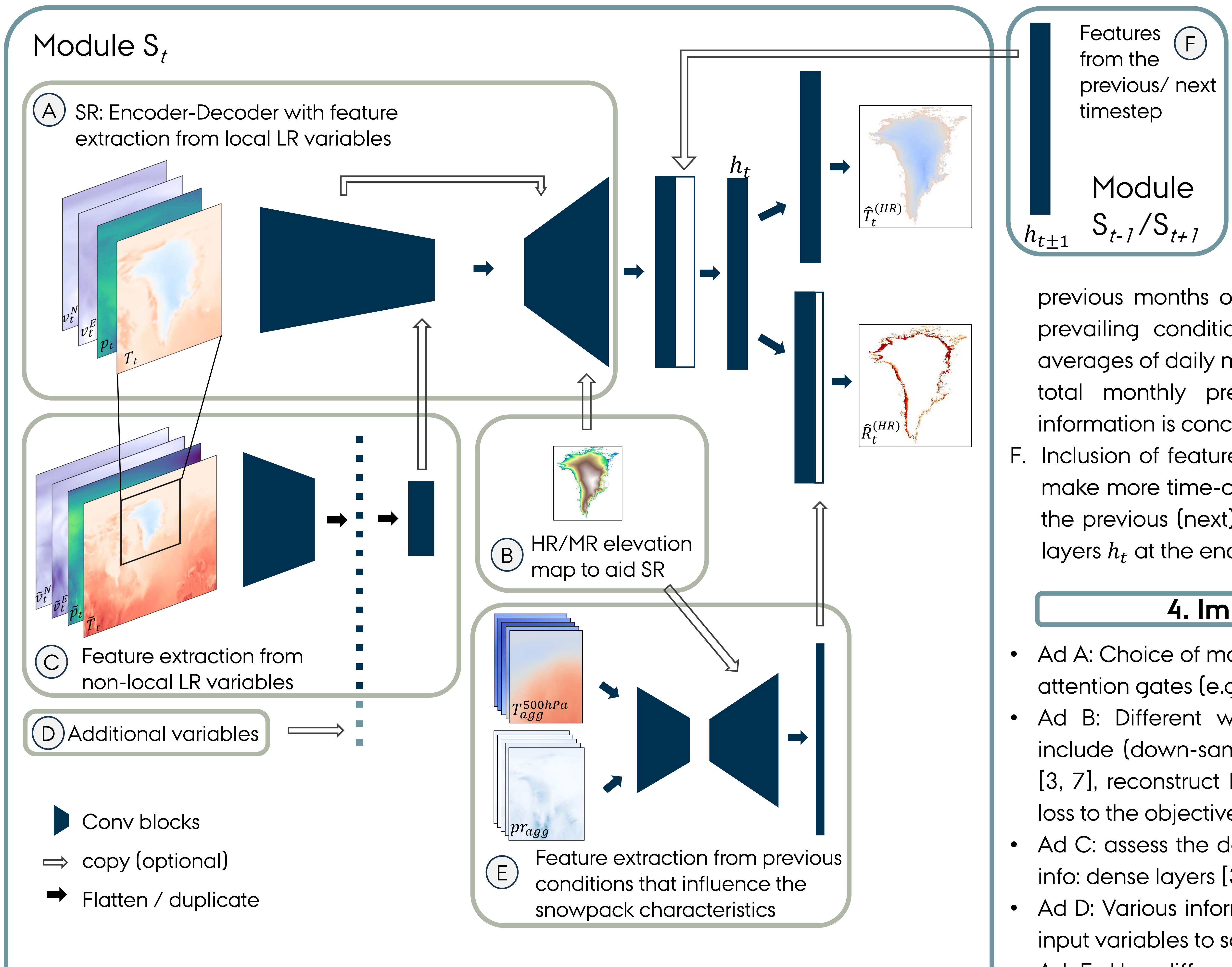
B. HR and MR (i.e., down-sampled HR) elevation maps are incorporated to assist reconstructing details at the HR scale. These details can not be derived from the LR maps, leading to too smooth predictions.

C. Remote information from within the North-Atlantic area is captured via a dense network. The results of this dense network are replicated over the spatial dimensions such that they can be concatenated to the layers of block A and are in the receptive fields of the upcoming convolutions for each pixel.

D. Information such as seasonal indicators are added to the dense layer.

E. Runoff is affected by snowpack characteristics which is defined by the prevailing conditions of the previous months or even years. Thus, we extract features from previous prevailing conditions at the Greenland ice sheet area, e.g., monthly averages of daily maximum temperature at the 500hPa pressure level and total monthly precipitation, to infer snowpack characteristics. This information is concatenated to the layers predicting runoff.

F. Inclusion of features from the previous and/or next timestep can help to make more time-consistent predictions. To do so, the layers h_{t-1} (h_{t+1}) of the previous (next) timestep can be concatenated to the current hidden layers h_t at the end of the network.


Module S_t
A SR: Encoder-Decoder with feature extraction from local LR variables
v_t^N
v_t^E
p_t
T_t
h_t
$\hat{T}_t^{(HR)}$
$\hat{R}_t^{(HR)}$
B HR/MR elevation map to aid SR
$\bar{v}_t^N$
$\bar{v}_t^E$
$\bar{p}_t$
$\bar{T}_t$
C Feature extraction from non-local LR variables
D Additional variables
T_{agg}^{500hPa}
pr_{agg}
E Feature extraction from previous conditions that influence the snowpack characteristics
Conv blocks
copy (optional)
Flatten / duplicate
Features from the previous/ next timestep F
Module S_{t-1}/S_{t+1}
$h_{t\pm1}$

Pixel-wise loss on targets + basin-wise loss for runoff:

$$\mathcal{L}_t^{px} = w_1 \mathcal{L}(R_{t,i}^{(HR)}, \hat{R}_{t,i}^{(HR)}) + w_2 \mathcal{L}(T_{t,i}^{(HR)}, \hat{T}_{t,i}^{(HR)})$$

$$\mathcal{L}_t^{\mathcal{B}} = w_3 \ \sum_{B\in\mathcal{B}} \mathcal{L}(\sum_{i\in B} R_{t,i}^{(HR)}, \sum_{i\in \mathrm{B}} \hat{R}_{t,i}^{(HR)})$$

4. Implementation & Open Questions

- Ad A: Choice of model framework and design for SR [9], residual learning, attention gates (e.g., CBAM) to foster learning channel dependency [10]
- Ad B: Different ways of including elevation and terrain information: include (down-sampled) elevation map at various levels in the network [3, 7], reconstruct HR elevation during training [6], add a terrain-guided loss to the objective [12]
- Ad C: assess the domain of influence [5]; different ways to catch remote info: dense layers [3], non-local NN [8], or axial attention [11]
- Ad D: Various information, also such as daily spatial mean and std of the input variables to scale them for block A (as done by [2]) are possible
- Ad E: Use different variables and aggregation methods; most simple version: make spatial means of the monthly data and pass over the values in D instead of using block E
- Ad F: include the previous and the following timestep; use a recurrent structure [1, 4, 13]
- Use various output variables – do they help each other? Are the results physically more consistent? (e.g., [6] learns the topography instead of using it as input only)

References:

[1] Chou, C., Park, J., & Chou, E. (2021). Generating high-resolution climate change projections using super-resolution convolutional LSTM neural networks. *2021 13th International Conference on Advanced Computational Intelligence (ICACI)*, 293–298.
[2] Doury, A., Somot, S., Gadat, S., Ribes, A., & Corre, L. (2022). Regional climate model emulator based on deep learning: Concept and first evaluation of a novel hybrid downscaling approach. *Climate Dynamics*.
[3] Dujardin, J., & Lehning, M. (2022). Wind-Topo: Downscaling near-surface wind fields to high-resolution topography in highly complex terrain with deep learning. *Quarterly Journal of the Royal Meteorological Society*, *148*(744), 1368–1388.
[4] Harilal, N., Singh, M., & Bhatia, U. (2021). Augmented convolutional LSTMs for generation of high-resolution climate change projections. *IEEE Access*, *9*, 25208–25218.
[5] Rampal, N., Gibson, P. B., Sood, A., Stuart, S., Fauchereau, N. C., Brandolino, C., Noll, B., & Meyers, T. (2022). High-resolution downscaling with interpretable deep learning: Rainfall extremes over New Zealand. *Weather and Climate Extremes*, *38*, 100525.
[6] Sha, Y., Gagne II, D. J., West, G., & Stull, R. (2020). Deep-learning-based gridded downscaling of surface meteorological variables in complex terrain. Part I: Daily maximum and minimum 2-m temperature. *Journal of Applied Meteorology and Climatology*, *59*(12), 2057–2073.
[7] Vandal, T., Kodra, E., Ganguly, S., Michaelis, A., Nemani, R., & Ganguly, A. R. (2017). DeepSD: Generating High Resolution Climate Change Projections through Single Image Super-Resolution. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1663–1672). Association for Computing Machinery.
[8] Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7794–7803.
[9] Wang, Z., Chen, J., & Hoi, S. C. H. (2021). Deep Learning for Image Super-Resolution: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(10), 3365–3387.
[10] Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). CBAM: Convolutional Block Attention Module. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018* (pp. 3–19). Springer International Publishing.
[11] Yu, T., Kuang, Q., & Yang, R. (2021). ATMConvGRU for weather forecasting. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1–5.
[12] Yu, T., Yang, R., Huang, Y., Gao, J., & Kuang, Q. (2022). Terrain-guided flatten memory network for deep spatial wind downscaling. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 9468–9481.
[13] Zhang, S., & Li, X. (2021). Future projections of offshore wind energy resources in China using CMIP6 simulations and a deep learning-based downscaling method. *Energy*, *217*, 119321.

Contact:
eschlager@envs.au.dk