Review of "Predicting ocean-induced ice-shelf melt rates using a machine learning image segmentation approach" by Rosier et al.

The Cryosphere

Round 2

1 General comments

In this second review I address the new version submitted by the authors, as well as their previous response to the reviewers.

I am happy to see some of the changes suggested in the new version, but I must admit there are still some aspects that remain unclear to me. I will just focus on two main ones, since as I said in my previous review, I think the paper is overall of high quality, with only some methodological shortcomings.

1.1 GC1: Choice of a classification modelling framework

I appreciate the extended explanations on this modelling choice, but according to the authors' reply to the reviewers, I am still not sure that we are on the same page in terms of our previous questions.

The authors justify their choice saying that if they wanted to use a regression network, they would have to apply a CNN like the one in Jouvet et al. (2021). I don't understand this comparison between different architectures, since any neural network architecture can be used for both classification and regression. Even U-Net can be used for regression, it would be as simple as changing the very last layer, with a single neuron and a linear activation function, and replacing the loss function for something like RMSE or MAE. What defines a network as a classification or regression one is not its architecture, but its output layer and loss function. It is unclear if the authors have attempted this extremely simple change, or if they have tried other different architectures. Directly adjusting U-Net for a regression problem would the most straightforward and fair way to test this.

I understand that if this current architecture works, it makes sense to keep it. But I'm also worried that such an architecture might end up being used as a reference for this sort of tasks, which is clearly and overcomplication of the problem. If the U-Net and the autoencoder where optimized at the same time (i.e. with a single loss at the output of the autoencoder), this would already make more sense, since backpropagation would take place as a regression

problem. However, destroying information to train a first classification network and then asking another network to undo the damage looks problematic to me.

In order to avoid a sterile debate over this, I ask the authors the following:

- Can you please clearly explain what exact architecture and changes you applied to the U-Net in order to use it as a regression problem? This should be clearly explained in the Appendix. I would strongly encourage the authors to test the very simple changes proposed above to use the U-Net for regression. Simply change the output layer and the loss function. This should be less than 5 lines of code.

- You should clearly state in the methods section and the abstract that this is a regression problem. You can explain that this more complex combined method of classification + regression works well here, but the reader should know that this is an exception, not the norm.

1.2 GC2: Absence of test dataset

One aspect that I admit escaped my first review is the absence of a test dataset. Currently the authors are splitting their total dataset into 90% training and 10% validation. Then, they tune the hyperparameters of the networks based on the results of the validation dataset, but no independent assessment of model performance is made. This is highly problematic, since the model hyperparameters are overfitting the validation dataset, which is also used to assess the final model performance.

The usual and recommended practice in machine learning is to split the dataset into train, validation and test datasets. The reasons to use an independent test dataset are widely explained in the literature, so I leave a reference here for more details (see points 2.2 and 3.1): Lones (2022), How to avoid machine learning pitfalls: a guide for academic researchers.

Before publication, the authors should repeat the training process, putting aside a test dataset. A possible division could be 70% training, 20% validation and 10% test. The authors are free to choose these ratios, but I would encourage them to use at least 10% on test. The split between these 3 datasets should be done before training, and the test dataset should be kept aside and not used until the very last moment, which will serve to have the "real" model performance. The final performance on the test dataset should be at least slightly lower than the one on the validation dataset (if the model is generalizing correctly). All figures displaying model performance should be updated with the results of the test dataset.

In order to allow a fair comparison against PICO and PLUME, these dataset divisions should also be applied throughout all models (particularly for Fig 4).