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

The Cryosphere

## 1 General comments

Rosier and colleagues present a novel method to predict ocean-induced basal melt of ice shelves using machine learning. The manuscript is well written, it is pleasant to read and the figures are of high quality. Many aspects of the study are well thought, such as the generation of random synthetic ice shelves, or the use of a GAN to generate temperature and salinity profiles. Overall, this is a good quality scientific study, which tackles an important modelling issue with the right tool (deep learning surrogate models). Despite all this strong aspects, I must say I am very surprised by the modelling choices regarding the neural network(s). Right from the title it is clear that the authors have decided to use an image classification architecture for this problem. This is quite bizarre, since this problem is clearly a regression problem, not a classification/segmentation one. As I was reading the manuscript, I was expecting the authors to explain such a strange choice, but I could not find any justification for that choice.

I have structured my comments into global comments (GC) that cover the main issues I found with the manuscript, and particular comments, which address line-by-line small comments or issues.

### 1.1 GC1: Choice of a classification modelling framework

This has been by far the most striking feature of this manuscript. The authors have chosen to use an image classification network for a regression problem. From the results, this method seems to work, but in order to do so the authors have had to ”force” the architecture into this problem, resulting in some awkward modelling strategies. Since there is no justification for this choice, there might be two potential explanations to this: (1) The authors have a clear strategy behind this, but did not explain it in the main manuscript. (2) The authors have re-used an already existing architecture (a very common and totally correct ML practice) from image classification, and tried to apply it to this problem without knowing that it was designed for a completely different task. I would like to know the exact reasons behind this choice, but on the meantime I will try to argue why I think such an architecture is not the best choice for this task.

Deep learning models can be applied to two main different types of problems: classification and regression. Classification is the most popular one, involving a nonlinear transformation of input data into a new space, in which a segmentation is performed based on a specific number of classes or labels in a supervised or unsupervised manner. On the other hand, regression models are in general less well known, and they are more challenging to train, validate and apply to physical systems. While the validation of a classification model is quite straightforward, since it is very easy to verify if the labels are accurate or not, this is not true for a regression problem. Regression problems for physical systems are trickier to validate, as one needs to make sure that the model is learning the physical relationships for the right reasons.

The fact that the authors chose a classification model for a regression problem has a series of consequences which add unnecessary complexity to the modelling framework:

- The discussion on the choice and impact of the number of classes for the first network could have been completely avoided just by choosing a regression model. Since the modelled variable (melt) is a continuous variable, it does not make sense to model it in a discrete way with a classification framework.
- The authors compensate this strange choice by adding a second neural network, an autoencoder, in order to interpolate the discrete classes obtained by the first network. As for the previous point, this second network could have been directly discarded if a regression model had been chosen.
- The model(s) presented in this study do seem to work, but I cannot help wondering how simpler and potentially faster might have been a solution with a regression network.

Since everything else in the study is well conceived, the model seems to work and the authors have even verified the physical plausibility of the learnt model, I do not think these reasons above are enough to deny publication. However, I would ask the authors at the very least to clearly explain in the discussion the reasoning behind this strange choice and comment on what the use of a regression model could imply for such a modelling framework.

Moreover, if the authors think it is something relatively simple to achieve, I would encourage them to re-train a regression CNN to see if the results are improved. In lines 357-359, the authors mention that they trained a single CNN that performed the same tasks as both networks. If that is really the case, that should be a regression network, otherwise it would not be possible to go from a continuous input to a continuous output. They also mentioned that such a network proved harder to train. It would be interesting to know if that is because they simply re-used the same architecture with some minor changes (e.g. just changing some activation functions), or if they chose a specific architecture suited to regression problems. As I said, training and validating regression networks is often trickier, but it is very likely that this might result in a better model. I will not enforce these changes, due to the above mentioned reasons. If they decide that it is too much work and they would rather keep the current model, then this should be clearly added in the discussion as a future perspective, including the current shortcomings of the model. The current model is overly complicated for this problem. A regression model would largely simplify the modelling pipeline, and could potentially result in a more accurate and expressive model.

### 1.2 GC2: Model validation

Another aspect of the modelling framework that I believe should be improved is its validation. According to the manuscript, only 5% of the dataset is used for validation, which seems extremely low. The authors justify this low fraction of data for test arguing that this maximizes the training dataset, thus improving the overall model performance. This is even more surprising knowing that this is in fact a surrogate model, whose training and validation data can be generated at will. Expanding the validation dataset would be as easy as generating more synthetic ice-shelf geometries and running NEMO on them. From Figure B4 we can see that the train performance plateaus at around 2500 synthetic cases. However, there is no information on how the test set impacts the performance. In machine learning it is essential to monitor the simultaneous evolution of the train and test performance, since they give important clues regarding overfitting or underfitting.

Some extra analyses should be performed in order to improve our confidence in the surrogate model(s):

- I believe the test dataset should be expanded. 5% might (or will likely) not be enough to correctly evaluate the out-of-sample model performance in a large variety of ice-shelf and ocean configurations.

- The test performance should be added to Figure B4, in order to track its evolution with different dataset sizes. If computational costs are behind the use of just 5%, I would still encourage the authors to expand it as much as possible, and then add these reasons explicitly in the manuscript.

### 1.3 GC3: Code availability and model details

Another downside of the manuscript is the lack of transparency regarding the model details. The main issue in my opinion is the fact that the model source code is not open-source. There is only a statement saying that the synthetic geometries are available upon request, without any mention of the model code itself. This makes it even harder to review the model, and goes against the open science values from journals such as The Cryosphere. Many of my doubts or questions could have been directly resolved by checking a properly documented repository on GitHub (or elsewhere). Therefore, I strongly encourage the authors to share their source code in a public repository. By making it citeable (e.g. using Zenodo), there are virtually no downsides to sharing it.

This has also been commented by the other reviewer. I think overall there is a lack of details regarding the model configuration in the manuscript. I understand that the authors do not want to flood the text with technicalities, but it would still be interesting to know a little bit more about the model in an Appendix or Supplementary material. Details regarding the optimizer for the gradient descent, regularization techniques used to avoid overfitting, learning rates, etc...

## 2 Specific comments

- **L120** Please add more details about the optimizer and gradient descent in either the text or an additional section in the Appendix or Supplementary material.
- **L126-127** By simply evaluating the loss at the pixels covering the ice shelf this could be easily solved. A matrix mask could be used to filter out those values. This yet another consequence of using a classification framework.
- **L133** A simple leaky ReLu could have sufficed, which is also less computationally expensive.
- **Figure 2** Great figure!
- **L246-247** Nice, this is indeed a very good idea, which allows for an infinite number of training samples.
- **L277-280** This should be explained in the legend, otherwise it is impossible to understand.
- **L281** By remaining panels do you mean the panels shown in Fig. 4?
- **L283-284** This should also be mentioned in the figure. It is important to mention that you are showing an out-of-sample performance.

  It is also unclear why the performances of the two parametrizations are not included. One would expect to see the comparison here, otherwise there is no baseline performance to compare with.
- **Figure 4** "Note the colour map gradient is not linear, but is greatest around zero, to make it easier to distinguish the magnitude of melting/refreezing over the bulk of the ice shelves." What do you mean? To me the colourmap from the plot appears to be linear.
- **L303-304** This is another strong aspect of this study. When working with surrogate models this risk is highly reduced, but it is still very nice that the learnt model physics were verified.
- **L303-304** Couldn't you change the loss of these two models? This could be easily solved by tuning all models with a combined loss: e.g. the $(NRMSE\_local + NRMSE\_average)/2$.
- **L315** Indeed, this study has focused on modelling the spatial information of ice shelf melt. Modelling of the temporal dimension remains untackled, and it might prove more challenging to do (see e.g. Bolibar et al. (2020) The Cryosphere). A validation in the spatial dimension doesn't ensure a good performance in the temporal dimension, which would be mandatory for any real world application as a surrogate for NEMO.

- **L331** Do you mean to the surrogate model? How would you add new physical processes to a surrogate model? I am not sure this is that straightforward to achieve. This model acts as a black box here, it just can be trusted because it is emulating a physical model that can be well understood.

- **L334** The authors use the term ”image” throughout the manuscript to refer to the input matrices used to train the networks. They seem to have re-used all the jargon from the computer vision field, despite not working in a computer vision problem. I would strongly suggest to refer to this as either a matrix or simply training features. What the authors are using are not really images, they are just gridded values, which for the case of a CNN they need to be presented in a 2D matrix. If the architecture changed to a simple feedforward NN these would be flattened, so the input shape is completely arbitrary.

- **L338-339** Please see GC1 .This should be explained in more detail. Do you mean a CNN used for regression? What type of architecture? If the authors have re-used architectures used for computer vision it is rather obvious that they will obtain better results by *forcing* them to a regression problem. However, this still seems really awkward, and a regression architecture tailored to this problem could likely perform better.

- **L345-346** Indeed, possibly NEMO is difficult to apply at such large geographical scales, but MELTNET should in theory be easily applicable. It would only provide a qualitative impression though.

- **L349-351** This could easily be constrained directly within the NN architecture. Just by applying a custom activation function at the ouput of MELTNET, one could already limit the simulated values within a physically plausible range.

- **L376** It would be nice to explain the ”mode collapse” concept for the non experts.

- **Figure B4** Please see GC2. Here it would be important to also see the validation set. I imagine that its fraction is reduced when the training fraction increases? With such an experiment it is not possible to know if performance increases due to additional training data and/or due to a reduced (and therefore) easier test set. Has any regularization been used for the training?

  Moreover, a plot showing the evolution of the train/test performance would help identify if the current network is overfitting or underfitting. As I previously said, I think that 5% for the test set is very low, so I’m a little bit worried that the network might be overfitting. The fact that the authors do not mention any regularization techniques at all also aggravates this.