Responses to the comments from Anyonymous Referee 2
Preprint `https://doi.org/10.5194/amt-2022-184`

Adrià Amell[*] Patrick Eriksson Simon Pfreundschuh

24 August 2022

The original text from the Anonymous Referee is presented in colour grey and our responses in black. At the end there is a section specifying the changes to the manuscript after reading and answering each comment.

[*]amell@chalmers.se

Response to comments from Anonymous Referee #2

The paper proposed by Adrià Amell and colleagues presents an inversion technique based on machine learning for the estimation of ice wather path (IWP) form Meteosat-9 observations with a focus on low latitudes. In their work, the authors both introduce and describe the topic with good details and discuss the potential and advantages of using artificial intelligence quantile-based regression methodologies over physics-based methods present in the literature.

In this context, the authors test various neural network architectures and compare the use of observations in the thermal infrared (IR) and/or visible bands as inputs. Finally, authors conclude that the architecture based on convolutional neural networks (CNNs) in which spatial information is integrated is the architecture that performs better, using, moreover, only observations in the infrared band as input. The presented approach offers several advantages over traditional methods, such as the ability to calculate diurnal cycles, a problem that for example CloudSat cannot solve due to its limited temporal and spatial sampling. Then, since the methodology is quantile based, it allows the developed methodology to obtain directly and in an integrated way an estimate of the uncertainty of the regressions.

The authors validated their work using CLASS that is thoroughly validated dataset based on traditional approaches. The obtained retrievals compare favourably with IWP retrievals in CLAAS. In my opinion, this last result arguably demonstrates the potential of this methodology highlight the possibilities to overcome limitations from physics-based approaches as demonstrated in other works recently published in literature Holl et al. (2014), Islam and Srivastava (2015) and Mastro et al. (2022).

We thank the referee for the nice summary of the paper, as well as for the comments and suggestions below.

However, in my opinion, some shortcomings are present in the paper framework that require a major review.

1. In section 3.2 authors describe the Network architecture and specifically they discuss the multilayer percepton (MLP) and the CNN configurations indicating their structural hyperparameters. I would argue that it is essential to describe in more detail this information and how the choice of these configurations was made. For example, for the MLP configuration, the authors indicate an architecture consisting of 16 hidden layers each composed of 128 hidden units assuming that it is the setup that achieves the best performance. How did they reach this finding? Has a tuning framework been used? If so, how was the hyperparameter space configured from which to begin the search for the best configuration? Also, were configurations with fewer hidden layers explored?

   We understand that it can be concerning to not provide more information about the design choices, particularly for those readers who are more knowledgeable in machine learning. We want the reader to be focused on the retrieval performance with machine learning, and not distracted by details of the choices in the machine learning models. In the three works cited that use neural networks to retrieve IWP, no description is given for the network choice (Holl et al., 2014), a rule of thumb is used without any hyperparameter search employed (Islam and Srivastava, 2015) or, the most recent work, only states what framework was used but not how it was configured (Mastro et al., 2022). Despite the level of detail clearly differs, we are updating the paper with some information at a very high-level. Nevertheless we answer the questions in detail here, as the hyperparameter tuning represented a substantial amount of work.

A subset $\mathcal{S}_t$ of the training data set $\mathcal{S}$ was used to train different models, which are described in the next two paragraphs. Another subset $\mathcal{S}_v \subset \mathcal{S}$, $\mathcal{S}_v \cap \mathcal{S}_t = \emptyset$ was used to evaluate the performance of each model. The performance on $\mathcal{S}_v$ determined our model choice. All models were trained and evaluated several times to make solid choices. No software tuning framework was used for this step. Instead, the (human) evaluation explained in the manuscript (line 175) was used for this step.

The CNN configuration and choices comes from experience and expertise acquired through previous or parallel projects from the authors. All these projects consists of retrievals with QRNNs. Furthermore, choices for the CNN are imposed by data and hardware constraints. In using a similar network as in those other projects we were able to adapt existing code for this work. This reduced the chances of running into silent bugs or inapproriately using the ML libraries. We find that presenting the choices for the blocks in the CNN model itself would require a paper of its own, since we also evaluated small changes in the CNN choices, which did not present remarkable benefits in this retrieval problem. The number of Xceptions blocks n was searched thoroughly over the grid $n = \{0, 1, 2, 4\}$, and the number of filters in $k = \{64, 128\}$. We observed that with $n = 2$ and $k = 128$ the performance tended to be better, yet only marginally better than the other choices in several repetitions of this hyperparameter exploration. Therefore, we did not explore deeper for a better final configuration.

Concerning the MLP, the goal was to have a simple MLP without any elaborated design choice as a benchmark base for any CNN network. This also made it computationaly cheaper to train and define a hyperparameter space. In this case, we explored configurations with $l = \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ hidden layers, $n = \{8, 16, 32, 64, 128, 256\}$ hidden neurons, and the three input settings used. Through the manual evaluation, we observed that $l = 16$ and $n = 128$ tended to perform equal or better than the other choices in several repetitions of the hyperparameter exploration. More details could be given on the MLP performance for this problem, but as in the CNN case, we also think this would deserve a paper of its own: only determining the MLP network candidate to then train with $\mathcal{S}$ represents $|l| \times |n| \times (3 \text{ input settings}) = 162$ options to compare, but since more than one training execution was performed for each hyperparameter configuration, there would be many more results to discuss.

2. The authors indicate that Table 2 shows the input characteristics used by the analyzed architectures. I believe that as presented, the table does not make it easy to understand which of the inputs shown are used of the architectures presented. I understand that various configurations of inputs were used for each architecture. Anyway, I sugges the authors reformulate more clearly the information in Table 2 and contextualize it better.

   We think that the information in Table 2 is clear, but we are updating the Table caption sentence “Input features used.” to “Input features used for each input settings.”. Throughout the text, the input settings for each network are indicated whenever they cannot be determined from the context, whether they are contained in the Figure (as in Figure 5), in the Figure caption (Figure 7, Figure 9), or in the text itself, for example, lines 247 or 259. This also implies that various configurations of inputs were used only for the MLP, as described in Sect. 4.1, and the rest of results, including the features used for the CNN, build only on using the features referred as IR input settings, as the first line (line 247) of Sect. 4.2 states and the following (sub)sections remind.

3. In section 3.3 the authors discuss the training of the proposed configurations. Here they also introduce information regarding the inputs used. In general as presented the section is very confusing and a possible reader might find it difficult to read. I propose to move the choice of inputs to section 3.2 following the corrections of Table 2 indicated previously and to focus section 3.3 in providing details concerning only the training phase. In addition, a useful piece of information would be to show the learning curves (for each epoch of training and validation) of the two configurations in order to demonstrate the absence of overfitting and underfitting problems.

   We appreciate this observation and are moving the text accordingly. We would also like to remark that we did observe overfitting, which is why early stopping on the validation loss was used (line 178). Therefore, we do not see any added value in adding the learning curves of the seven networks presented in the paper, but we will provide Figure RC2.1 as supplementary material. For comparison with the three works cited that use neural networks to retrieve IWP, only Mastro et al. (2022) present a learning curve, but only for one of the networks therein (cf. comment #1). As a side note, we identified that the bursts in the loss function are related to the behaviour of the chosen optimizer (and its hyperparameters) in the landscape of the loss function for this problem, but we consider any further discussion and analysis on this regard out of the scope of the paper.

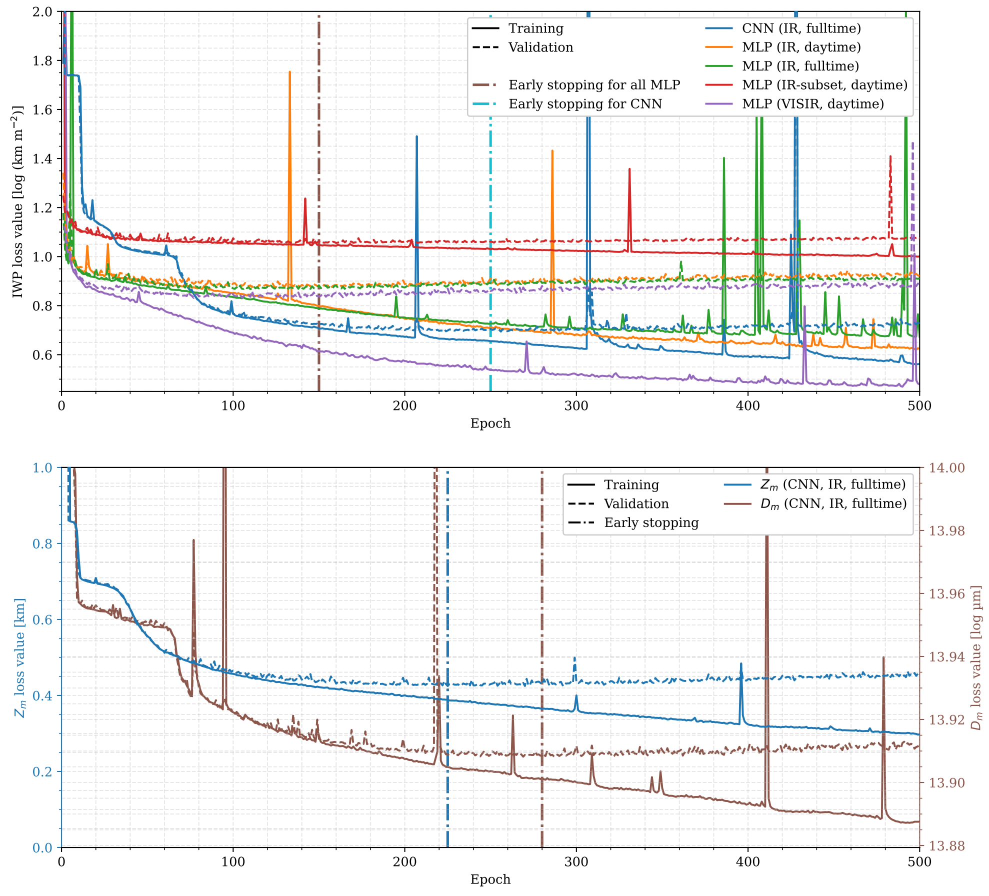

Figure RC2.1: Learning curves, with the early stopping epochs indicated.

4. Figure 4 shows the CNN architecture and in my opinion it is a bit misleading. I would like to propose to the authors to change the position of the DXception and Xception blocks next to the blocks themselves, because as they look they appear to be part of the input and output blocks.

   Figure 4 is designed such that it fits nicely in a two-column paper, and the suggestion does not fit our intention. We understand that it can be misleading. Figure RC2.2 will replace the current Figure in the paper (the colour choice is arbitrary).

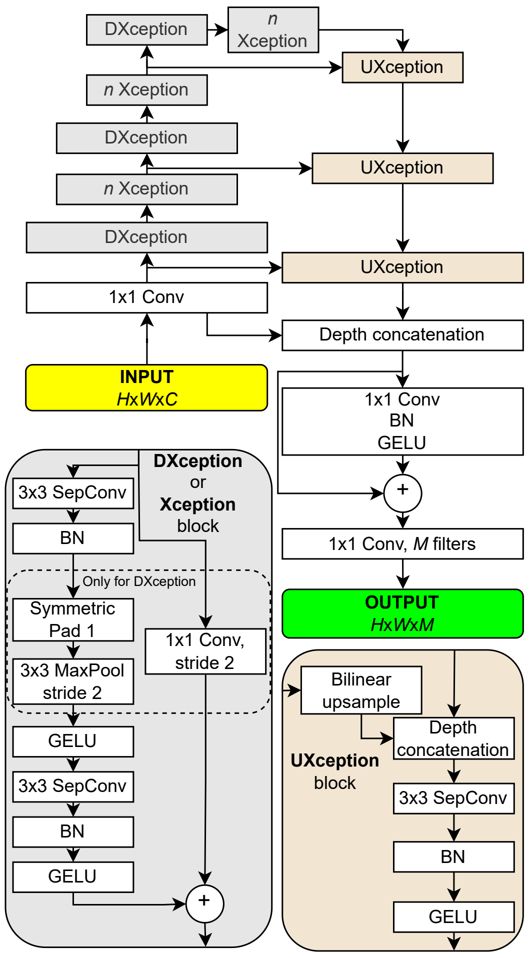

Figure RC2.2: Diagram that will replace Figure 4.

Manuscript changes after the comments from Anonymous Referee #2

- Added a high-level info on how the hyperparemeters for the networks were selected in Sect. 3.2.
- Table 2 caption from "Input features used." to "Input features used for each input settings.".
- Move text according to referee comment #3.
- All learning curves from all networks presented in the paper (Fig. RC2.1) added as supplementary material.
- Figure 4 replaced by Fig. RC2.2.