We'd like to thank the editor for handling our manuscript, as well as reviewer #3 for reading our manuscript and providing numerous helpful suggestions for improvement.

We have carefully read through all the comments and questions and revised the manuscript accordingly. Please find our point-by-point response to reviewer #3 below. Here, the reviewer's general and specific questions/comments are formatted to be left-aligned text in bold font. Our responses are indented and formatted in regular font.

Here is a summary of the major changes in the revised manuscript:

1) Table 2 reports ANN performance metrics for both the validation and independent test data set.
2) We added additional information on the ranges for each hyperparameter and computational costs to section 3.
3) We added explanations on why the temperature ANN model appears to be more complex than other models.
4) 2) Tables 2 and 3 report the respective ANN performance metrics (RMSD, bias, and percentile differences) for each species in both their natural units (K, ppmv, ppbv), as well as percentages.
5) We added a subsection on data quality assessment to section 3.
6) We discuss areas in the global maps, where the ANN-NRT algorithm exhibits clear underestimations.

GENERAL COMMENTS

==========

The paper describes the application of an artifical neural network (ANN) to the retrieval of trace gas profiles from the MLS instrument. ANN have been applied recently to different problems, partially with large success.

Here, the intent is to replace a primarily fast but comparatively inaccurate near-real-time retrieval with something both faster and more accurate. The presented results indicate that the approach has succeeded on both ends.

The study is on the point, well described, and executed. The topic fits the journal. I recommend publication.

SPECIFIC COMMENTS

==========

lines 80ff: The underlying software seems to be readily available. Could the training model employed here be made available as well? This might be applicable for similar tasks and/or other limb sounders.

> The referenced "Keras" and "Tensorflow" software packages are open source tools to set up machine learning platforms. Internally, we use specifically developed Python routines to access those open source tools and to streamline the training process. We are currently in the process of preparing a Python package that could be hosted on Github and made available to the public. However, note that these are simply wrappers to simplify access to "Keras" and "Tensorflow.
>
> Following the steps outlined in the Keras user manual (https://keras.io/guides/sequential_model/) to set up a feedforward neural network and using the settings summarized in Section 3.2 and Table 1 of the manuscript is all it takes to set up the exact models described in the manuscript. However, it is highly unlikely that these exact models produce reliable results for different tasks or instruments. Instead, the correct settings need to be determined individually for each application and data set; these settings are probably very different from the ones used here.
>
> Please reach out if you want help with setting up similar models for different tasks, we are happy to help. (…so long as we aren't forbidden to do so by US export controls.)

lines 130ff: A general problem with trained models is how the model copes with unexpected situations. Here, you describe how you adapted the training data set to cope with volcanic activity. How important was this for the performance and how likely is it that, e.g. the Ozone hole would have been missed?

> This problem (performance for situations not seen before) is indeed inherent in all supervised machine-learning applications, not just for the SO_2 model described in the manuscript. However, the MLS SO_2 profile retrievals are somewhat special (compared to

the other species) as they are basically noise at all levels in the absence of volcanic activity. An example of that is presented in Fig. 1 of this reply, which shows joint histograms of the operational L2 SO_2 concentrations and those provided by OE-NRT, ANN-NRT trained with the reduced data set, and ANN-NRT trained with all data over 01/01/2005–04/30/2022. Joint histograms are shown for two pressure levels; data is from MLS observations in May 2022.

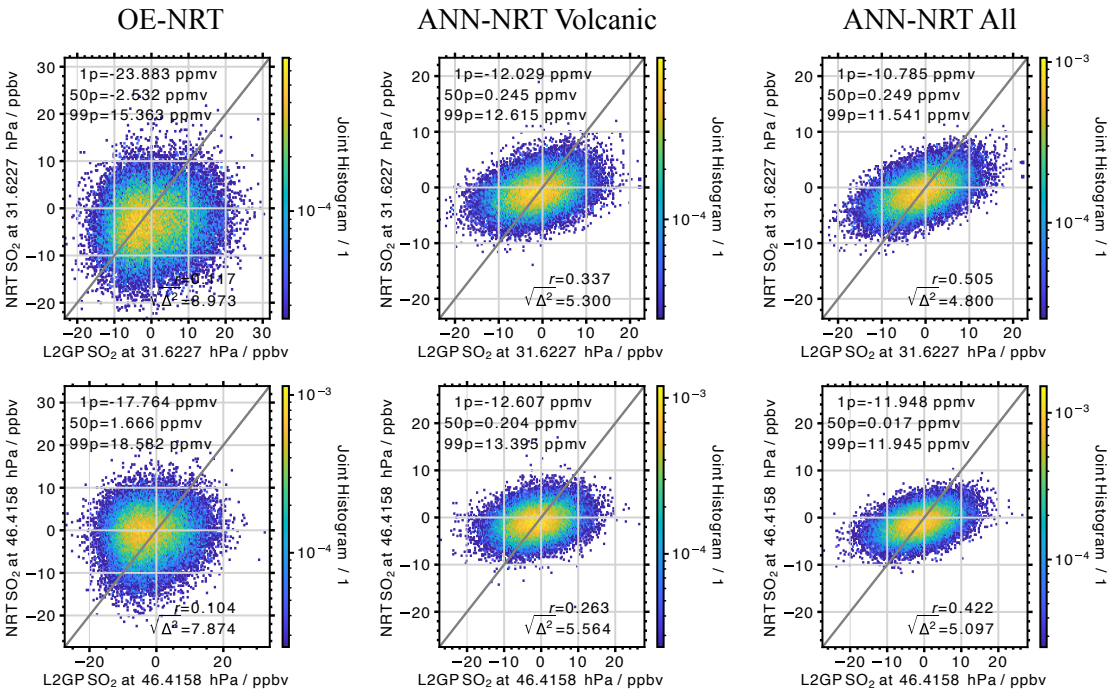

Fig. 1: Joint histograms of L2 SO_2 and three NRT models for two pressure levels each.

The ANN-NRT that was trained with all profiles performs better in predicting the noise at each pressure level than the model trained with the reduced data set. This is simply due to the fact that the former was trained on more noisy data. Both ANN-based models perform better than OE-NRT. Note the slight tilt of the ANN-based distributions in relation to the 1:1 line, which illustrates the tendency of the ANNs to predict 0 ppmv.

Both the correlation coefficient and bias for an independent test data set improve when using the ANN model that was trained on all data over 01/01/2005–04/30/2022. Average R=0.37 and average bias =5.73%, compared to 0.26 and 7.26%, respectively. However, this only illustrates that the model can predict noise better. We compared model performance for profiles with elevated values (from the independent test data set) and found that the model trained on the limited data set performs better (correlation coefficients of 0.57 vs 0.52; RMSDs of 5.41 ppbv vs 5.72 ppbv). In other words, the model trained on the full data set focuses slightly too much on the noise.

We added additional information to the revised manuscript. Table 2 now lists the performance metrics for the two different models. In the text we then motivate the use of the model trained on the limited data set:

"If the training data set is increased to include all MLS retrievals between 01/01/2005 and 04/30/2022 (named 2nd model in Table 2) rather than being restricted to periods of volcanic activity, the associated correlation coefficients and biases slightly improve to 0.37 and <7%, indicating a better ability to predict noise. However, further analysis indicates that this model performs slightly worse for profiles containing elevated SO_2 concentrations; correlation coefficients for such profiles in the test data set are decreased by about 0.05 (R=0.52 compared to R=0.57), while the RMSD increases by about 0.31 ppbv (5.72 ppbv compared to 5.41 ppbv). Since the main objective of the SO_2 NRT is to detect volcanic activity, we decided to employ the model trained on the reduced (volcanic only) data set."

lines 235ff: This result suggests that the training data set contains a lot of redundancy, as is expected for such a large set measuring effectively the same planet all over. Do you have means to identify profiles with high influence on the training performance? And if yes, what were they?

This would be an interesting analysis. There are several ways to determine feature importance, i.e., determine which input variables (MLS bands, channels, MIFs) are most important during the prediction. There also ways to set sample weights, i.e., a way to make sure certain profiles are more important than others.

Unfortunately, identifying individual profiles that most contributed to the training performance is not possible, at least not with our current setup. That's because during training the loss function is not calculated for individual profiles, but for a collection of profiles (called a batch). The batch size is determined by the "mini-batch size" parameter (listed in table 1 of the manuscript), which in our case is almost always 32. That means that during each training iteration, an average loss is calculated for M batches (where M is the total number of profiles, divided by 32). Each of the M batches contains 32 randomly selected profiles. After each iteration, the ANN weights are updated based on the average loss, the input data get randomly shuffled and assigned to a new set of M batches, and a new loss is calculated.

There is the possibility to set the mini-batch size to 1. However, this is not recommended as the calculated losses become very noisy, which almost certainly will prevent the model from converging. It also means that during each iteration we have to loop over every profile in the training data set, which dramatically increases the training time.

Do you foresee a possibility to generate a synthetic set of training data for a new instrument, for which no historic data is available? How would this compare for instruments, which measure more seldomly, such as ACE-FTS. Would a year of data still be sufficient to train the retrieval?

Machine learning approaches are statistical in nature. Using a wide array of synthetic composition profiles and radiance data should indeed provide the means to facilitate near-real-time predictions for a new instrument. Such an approach is not dissimilar to calculating look-up tables of synthetic observations for a wide range of viewing

geometries and cloud variables in MODIS-like cloud property retrievals. As long as the radiances accurately describe the actual (noisy) observations and the set of composition profiles cover a wide array of possible atmospheric states, that approach should yield reliable results. Again, a retrieval approach based on look-up tables is very similar.

We ran a small test to, at the very least, confirm the feasibility of such an approach. Instead of creating a large set of possible atmospheric states and running a forward model on each to create synthetic MLS radiances, we used simulated radiances for day 51 in 1996 as input for our ANN-NRT temperature model. That data set is part of our testing procedure for the MLS retrieval algorithm. Note that the ANN-NRT models were trained on the relationship between a set of noisy MLS radiances and noisy MLS L2 retrievals. Applying these models on noise-free radiances and climatological temperature profiles introduces considerable uncertainties.

The results are shown in Fig. 2 of this reply. Panels a and b show scatter plots of predicted vs modelled temperatures at 100.00 hPa and 21.54 hPa, respectively. While model performance is worse compared to our analysis for actually observed MLS radiances and retrieved temperature profiles, it still performs reasonably well. Correlation coefficients are 0.95 (100.00 hPa) and 0.93 (21.54 hPa). The RMSD>2 K is in the range of the results in table 3 of the manuscript. These metrics are also worse than the ones based upon a single year of MLS observations (see Fig. 5 of the manuscript).

This approach might also be preferrable for instruments with low sample frequency. ACE-FTS, for example, samples about 5,000 composition profiles per year. In our experience this is roughly an order of magnitude too low to train a reliable machine learning model. However, there are a number of data augmentation techniques (like applying Gaussian noise to the input features, as well as to the neuron output in the model) that can make the model predictions more robust even for smaller datasets.

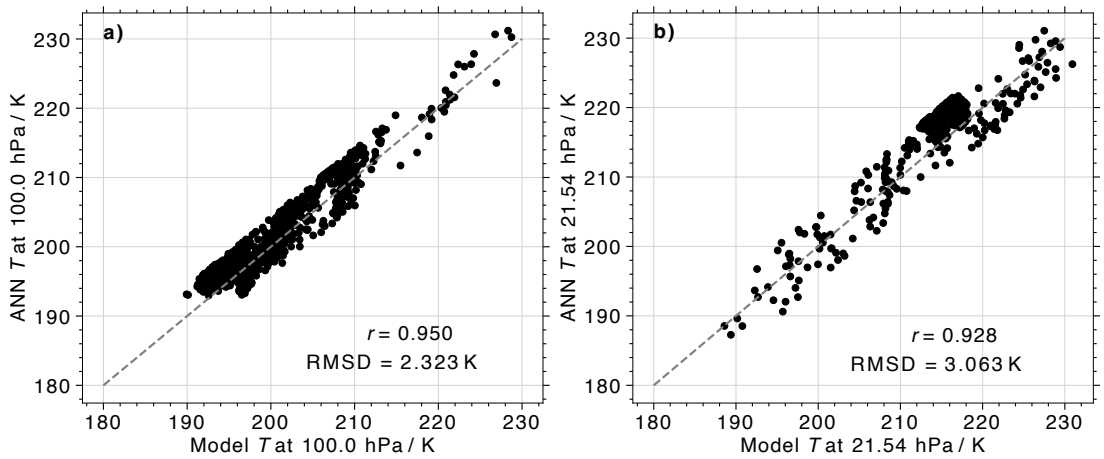

Fig. 2: Model performance for simulated temperature profiles and radiances.

We performed another small test, where we tried to predict ACE-FTS CH_4 based only on ACE-FTS N_2O concentrations, i.e., predicting the relationship shown in Fig. 1 of Minnschwaner and Manney (2014). While not the same thing as relating radiances to

composition profiles, it still gives us an idea about the impact of data set size. We compared model performance for a model that was trained on 5% (~1 year) to the performance of a model that was trained on 25% (~5 years) of data. The size of each validation data set is 2% of all ACE-FTS data up to 2022. We then compared the results for the remaining data points (i.e., test data); the results are shown in Fig. 3 of this response.

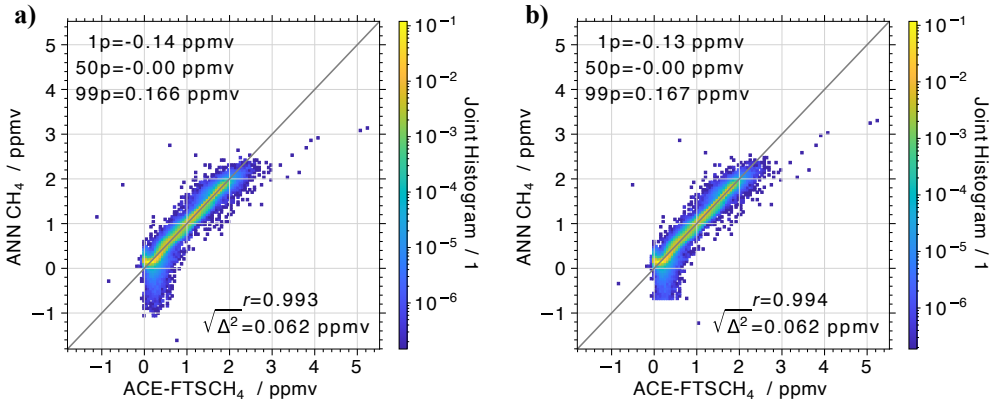

Fig. 3: ACE-FTS predictions of CH_4.

Overall, there is not a lot of difference between the two models. Using 5 years of data increases the correlation coefficient from 0.993 to 0.994. The RMSD is the same between the two models at 0.062 ppmv. Naturally, the relationship between radiances and compound profiles is a lot more complex than the relationship between N_2O and CH_4.

While we don't think it makes sense to add this analysis to the revised manuscript, we changed the last sentence of the conclusions to the following:
"…, which demonstrates the potential of applying machine learning to generate NRT products for other current and future mission concepts with similar sampling frequency. Alternative approaches, like training ANNs on synthetic profiles of atmospheric constituents and simulated brightness temperatures, may be needed for instruments with significantly lower sampling rates."

Reference: Minschwaner, K., Manney, G.L. Derived methane in the stratosphere and lower mesosphere from Aura Microwave Limb Sounder measurements of nitrous oxide, water vapor, and carbon monoxide. J Atmos Chem 71, 253–267 (2014). https://doi.org/10.1007/s10874-015-9299-z

lines 244ff: Typically, level 2 products are associated with a zoo of diagnostic data from precision to resolution etc. How is the data provided by the ANN characterised?

This is one of the disadvantages of neural networks compared to Random Forests (another popular machine learning framework): the usual implementation of neural networks does not supply any uncertainty information. However, we attempt to estimate the precision of the ANN predictions based on statistics. We also perform some basic data quality checks.

We agree that it is important to add this information to the revised manuscript. We therefore added a new subsection on data quality to section 3 of the revised manuscript. Here is a quick summary of the information:
The only data quality flag that is used going forward is the precision, which is derived as the root mean square of (i) the typical MLS L2 precisions for the given pressure level taken from the training data set, and (ii) RMSD between MLS L2 products and the predictions for the independent test data set. Negative precisions are assigned to values outside the valid pressure range, profiles in overlap regions. Data values with negative precisions should not be used. An additional data quality check assures that predictions at each pressure level are within a predefined confidence range; precisions for profiles where predictions are outside that confidence range (at any pressure level) are set to negative 1.

Note that this information is also given in the Version 5 Level-2 Near-Real-Time Data User Guide.

lines 261ff: The speed-up of the NRT retrieval is impressive and very useful for the purpose of providing near-real-time data. How does this relate to the computational effort for training the model? Is this (over the foreseen runtime) still a net positive or does one trade in training effort for faster operational results? Does one need a super-computer/cloud service for training or is this feasible with a well-equipped work station?

The computational costs of training the ANN-NRT models are not too crazy. The exact numbers depend on the specific model setup (number of hidden layers and neurons, mini-batch size) and the size of the input matrix (number of features and samples). Training a model on 1 year of MLS O_3 data, for example, requires about 60 GB of memory and takes a ~10 hours to converge when trained using 16 CPUs. The size of the data set (i.e., how many years are included) does not affect the memory requirements for the training process, as the model calculates average losses for a batch of samples; adding more data of course affects overall memory usage because the data needs to be readily available. Training the O_3 model on 18 years of data requires about 100 GB of memory and takes about 1 week to fully train when using 16 CPUs. This means that including the time it took to determine the best hyperparameters, each ANN-NRT model can be set up and trained in about 1 month.

A well- equipped work station is sufficient to develop and train these ANN models. Note that tree-based machine learning architectures, like Random Forests and Gradient Boosted Decision Trees, can offer similar performance at a fraction of the computational costs. These models also convergence significantly faster than ANNs.

We added the following information to the revised manuscript:
“The computational costs associated with the training procedure of each ANN-NRT model, while dependent on the respective hyperparameters and size of the m x n input matrix, are generally as follows: it takes about one month to develop and train each ANN, using 12 CPUs and requiring ≈100 GB of memory.

lines 263ff: Are NRT retrievals the only application of the ANN model discussed here? Could this data serve as an initial guess to the OE to speed up convergence or are there reasons not to use this?

The new ANN-based NRT predictions, as well as the previous OE-based results, could theoretically be used as an a priori guess for the operational retrieval. There are, however, a number of reasons why we have no plans of doing so:
(1) A well-defined retrieval problem should converge to the correct solution almost independent of the specific a priori profile.
(2) The retrieval uncertainty/precision would be different, as the a priori uncertainty would be different.
(3) At this point, 18 years into the MLS mission, we are avoiding massive changes to the rather complex L2 retrieval algorithm. Using the ANN predictions as a priori profiles would require a significant development and testing effort, with possibly little to no benefits (see point 1).