Responses to the comments by Reviewer 2.

We greatly appreciate constructive comments and recommendations by Reviewer 2. A lot of questions raised are very important and would be addressed in future studies with more realistic profiles and even more important, more accurate radiative transfer modeling of aerosol properties and scattering. We will take into consideration the recommendations made by the reviewer in data design and welcome him/her contact us directly regarding potential collaboration.

1. Not enough information about the machine learning (ML) algorithm itself are given. The introduction focuses on aerosols and MAX-DOAS only without introducing ML. In section 5, there is no explanation why the individual ML steps were chosen as they are.

- We have removed some details about the aerosols and have added the following section to the introduction:

"This paper investigates the potential of using advances in machine learning to invert aerosol properties (aerosol extinction coefficient profiles, single scattering albedo and scattering phase function) from a hyperspectral remote sensing technique called multi-axis differential optical absorption.

Machine learning (ML) is a branch of artificial intelligence that derives its roots from pattern recognition and statistics. The goal of ML is to build statistical (or mathematical) models of a real-world phenomenon by relying on training examples. For instance, in supervised ML, a model is first presented with a set of paired examples (termed as the training set), where every training example contains a pair of input variables and output variables, and the goal of ML algorithms is to find the statistical structure of mapping from the input variables to the output variables that match with the training examples and can be generalized to unseen examples (termed as test set). The learned mapping (or the model) can be applied to the inputs of test examples to make predictions on their outputs. There are several advantages of using ML. Firstly, it can sift through vast amounts of training data and discover patterns that are not apparent to humans. Secondly, ML algorithms can have continuous improvement in accuracy and efficiency with increasing amount of training data. Thirdly, ML algorithms are usually very fast to apply on test examples since the time-consuming training process of ML models is offline and one-time. With these advantages as well as the availability of faster hardware, ML has soon become the most popular data analytic technique since the 1990s. In recent years, it has also been applied to the field of remote sensing (Efremenko et al., 2017; Hedelt et al., 2019).
Artificial neural networks (ANN) is one of the many methods studied in the ML field that has found a lot of success in recent years over a number of commercial problems such as image detection, text translation, and speech recognition. It is inspired by the biological neural networks constituting animal brains. As an analogy to a biological brain, an ANN is based on artificial neurons. An artificial neuron is a mathematical function receiving and processing input signals and producing outputs signals or activations. Each neuron comprises of weighted inputs, an activation function, and an output. Weights of the neuron are parameters to be adjusted, while the activation function defines the relationship from the input signals to the output signals. When multiple neurons are composed together in a layered manner (where the output signals of

neurons in a given layer are used as inputs for the neurons in the next layer), we call it an artificial neural network (ANN). A common algorithm for training ANNs is the backpropagation algorithm, that passes the gradients of errors on the training set from the output layer to inner layers to refine the weights at all layers in an incremental way. The backpropagation algorithm converges when there is no change in ANN weights across all layers beyond a certain threshold. There are several optimization methods that are used for performing backpropagation and are behind standalone ANN packages commonly used by the ML community. ANNs have many different types depending on the specifics of the neuron arrangement or architecture. A simple type of ANN is a multilayer perceptron (MLP), where all neurons at a given layer are fully connected with all neurons of the next layer, also termed as dense layers. Other complex types of ANN include convolutional neural network (CNN) and recurrent neural network (RNN). Two important types of artificial neural networks used in this study are the convolutional neural networks (CNN) (Fukushima, 1980; LeCun et al., 1999) and the Long short-term memory (LSTM) neural networks (Hochreiter and Schmidhuber, 1997), which are variants of recurrent neural networks.

Convolutional neural network (CNN) is a class of deep neural networks that uses the convolution operation to define the type of connections from one layer to another. While they have shown impressive results in extracting complex features from images in computer vision applications (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015), they are relevant in many other applications involving structured input data, e.g., 1D-sequences. A CNN is composed of an input layer, multiple hidden layers and an output layer. The hidden layers usually consist of several convolutional layers, followed by pooling layers, fully connected layers (dense layers) and normalization layers. Figure 1 shows a simple example of CNN. The input vector (or sequence) is first passed through a convolutional layer where it is convolved with 3 filters (convolution kernels) of size 3 using the same padding to produce three 6x1 feature maps. Since the ReLU function $(f(\boldsymbol{x}) = max\ (0, \boldsymbol{x}))$ is commonly chosen as the activation function in CNNs, the feature maps only contain positive values. Then the max pooling layer picks the maximum value every 3 elements for each feature map, generating three 2 x 1 vectors. After passing through a flatten layer, the max pooling output is reshaped into a 6 x 1 vector, which is followed by a dense (fully connected) layer with 2 nodes. The dense layer multiplies its input by a weight matrix and add a bias vector for generating the output of the model. The computer adjusts the model's convolutional kernel values or weights through a training process called backpropagation, a class of algorithms utilizing the gradient of loss function to update weights. For the case in Figure 1, there are 26 tunable parameters. $((3 + 1)\times 3 = 12$ from convolution kernels and $(6 + 1)\times 2 = 14$ from the dense layer.)

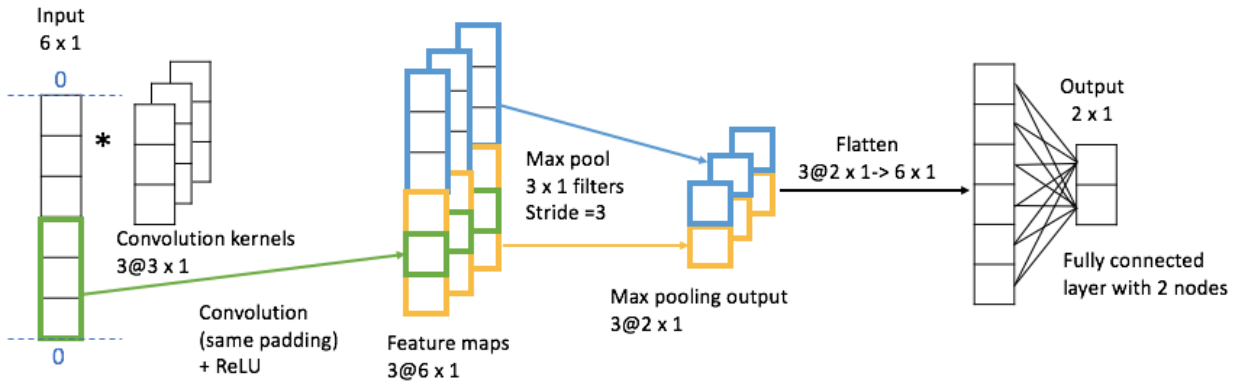

Figure 1: a simple example of CNN

Long short-term memory (LSTM) neural networks have many applications such as speech recognition (Li and Wu, 2015) and handwriting recognition (Graves et al., 2008; Graves and Schmidhuber, 2009). They are a special kind of ANNs termed as recurrent neural networks (RNNs). RNNs are designed for modeling sequence dependent behavior (e.g., in time). They are called "recurrent" because they perform the same operation for every element of a sequence, with the output at a given element dependent on previous computations at earlier elements (Britz, 2015). This is different from traditional neural networks wherein all the input-output examples are assumed to be independent of each other.

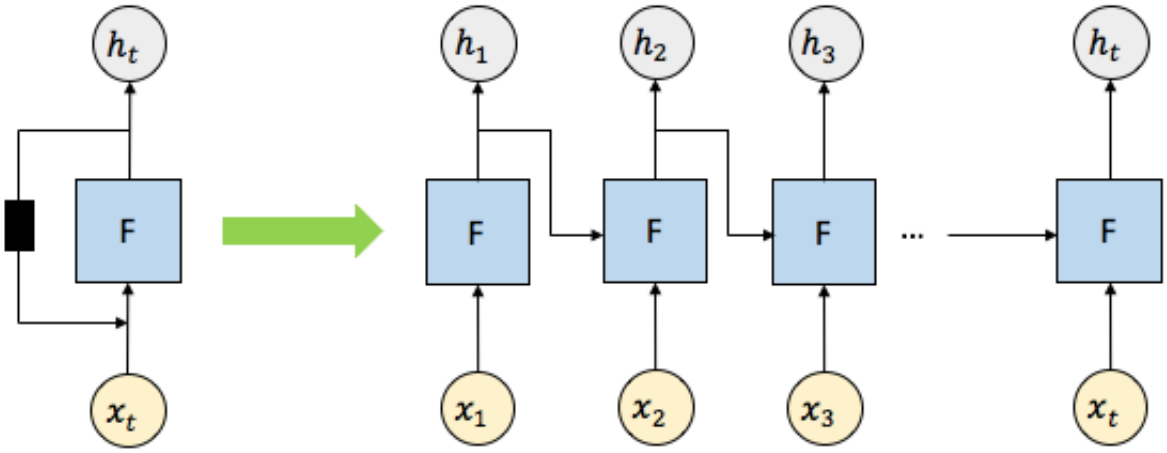

Figure 2: Unrolled recurrent neural network

Figure 2 is a diagram of an unrolled RNN with t input nodes, where "unrolled" means showing the network for the full sequence of inputs and outputs. The RNNs work as follows. At the first element of the sequence, the set of input signals x_1 (which can be multi-dimensioinal) is fed into the neural network F to produce an output h_1. At the next element of the sequence, the same neural network F takes both the next input x_2 and previous output h_1, generating the next output h_2. This recurrent computation continues for t times to produce the output at the tth element of the sequence, h_t. While RNNs are powerful architectures for modeling sequence behavior, classical RNNs are inadequate to capture long-term memory effects where the inputs-outputs at a given element of the sequence can affect the outputs at another element of the sequence separated by a long interval. Long-short-term memory (LSTM) models are variants of RNNs that are able to overcome this challenge and are efficient at capturing long-term dependencies as well as short-term dependencies. It does so by introducing an internal memory state that is operated by neural network layers termed as gates, such as the "input gate," that adds new information from the input signals to the memory state, the "forgot gate," that erases content from the memory state depending on the input signals, and the "output gate," that transforms information contained in the input signals and the memory state to produce output signals.

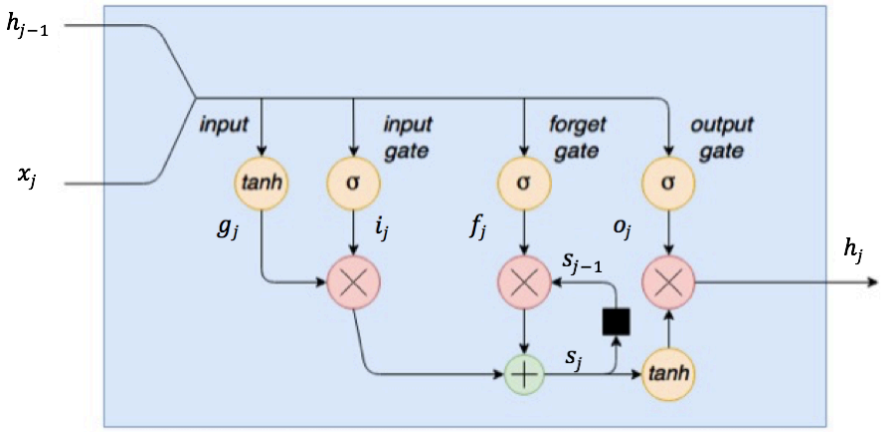

Figure 3: LSTM cell diagram (modified from Thomas, 2018).

The LSTM cell used in this study is illustrated in Figure 3, of which the update rules are:

$$g_j = \tanh\left(b^g + x_j U^g + h_{j-1} V^g\right)$$
$$i_j = \sigma(b^i + x_j U^i + h_{j-1} V^i)$$
$$f_j = \sigma(b^f + x_j U^f + h_{j-1} V^f)$$
$$s_j = s_{j-1} \circ f_j + g_j \circ i_j$$
$$o_j = \sigma(b^o + x_j U^o + h_{j-1} V^o)$$
$$h_j = \tanh(s_j) \circ o_j$$

where j is the element index, σ(x) represents the sigmoid function, and $\tanh(x)$ represents the hyperbolic tangent function. $x \circ y$ denotes the element-wise product of x and y. U^g, U^i, U^f, U^o are the weights for the input x_j, while V^g, V^i, V^f, V^o are the weights for the other input h_{j-1}, and b^g, b^i, b^f, b^o are the scalar terms (termed as bias). The term g_j is the input modulation gate, which modulates the input $b^g + x_j U^g + h_{j-1} V^g$ by a hyperbolic tangent function, squashing the input between -1 to 1. The term i_j is the input gate, which applies a sigmoid function to its input, limiting the output values between 0 and 1. The input gate i_j determines which inputs are switched on or off when multiplying the modulated inputs ($g_j \circ i_j$). The term s_j is the internal cell state that provides an internal recurrence loop to learn the sequence dependence. The terms f_j and o_j are the forgot gate and output gate, respectively. They have similar function to the input gate i_j, regulating the information into and out of the LSTM cell. The term h_j is the output at step j. "

2. The validation section appears to be insufficient to assess the performance of the algorithm: (a) Why not changing the testing dataset to realistic profiles which are not included in the training data? How can you be sure that you do not over-fit your results?

-The mappings contained in the test set are different from those in the training set. And different combination of SSA/ASY/profile produces different values of radiance. The model hasn't seen these input values and output combinations of SSA/ASY/profile before. We split the data into two sets, the training set and the test set, no automatic model selection process using validation set. The training loss converges. We use 75% of the entire dataset for training and then directly apply the model on the remaining 25% for

testing. We use ReLU unit (through sparsity) and Max Pooling layer (through reducing parameters) to control overfitting.

(b) Why not using larger aerosol loads?

- Lower to medium AOD loadings are more common. We will use LIVAS data base in future studies with more realistic profiles and global AOD loadings

(c) Why did you use 16 different elevation angles for the testing dataset even though this number is much too high for most measurement locations? What happens if you just use 8 or 10 elevation angles? Does the algorithm still perform well?

- The main goal of the current work is to evaluate feasibility of ML as a retrieval method. We have included more angles than typical for MAX-DOAS to explore the maximum information content of the measurements and the inversion method. We have also tried a scan with10 angles and the ML methods performed well.

(d) The training dataset was created by using an US standard atmosphere. This is mostly a poor representation of the true atmosphere. What happens if the conditions change?

We have not evaluated the effect of temperature and pressure profiles on the retrieval at this stage. This will be done in the future studies

(e) Why not testing the algorithm on real data?

We do not have real data at 360 nm with accurate partial AOD profiles, ASY and SSA retrieved in the same direction as MAX-DOAS pointing.

3. Are there plans to extend the training dataset by more wavelengths, SSA/asymmetry factors, albedo, profiles, trace gases (as suggested in Sec. 7)?

- Yes

4. Note that many articles are missing in the manuscript.

- we have gone through the references

Specific comments
P2, L36-42: There is no need to show the equation of SSA and its detailed description. I suggest to change those 6 lines into one sentence only.

- we have removed details about SSA from the manuscript

P2, L43-62: This part about the aerosol phase functions is much too long, especially since there is no further discussion about this topic in your paper. The lines about the Legendre expansion could be completely removed without loosing important information for the understanding of the manuscript.

- we have removed details about ASY from the manuscript

P2, L58: When kept, please change the index L of PL(cosθ) P3, L65: "The" MAX-DOAS

- we have removed details about ASY from the manuscript

Figure 1: A single sky scan...

- corrected

P3, L84: "The" DOAS technique...

- corrected

P4, L91: "The" offset term...

- corrected

P4, L101: "Forward model parameters that are considered approximately". I guess you are referring, among other parameters, to a priori knowledge when using "approximately" here? Please change the wording or reformulate this sentence.

- this is reference to such parameters as temperature and pressure profiles. Clarified in the text now: (e.g., temperature and pressure profiles from atmospheric soundings or models).

P4, L114-115: "A priori information about...". I find this sentence to be rather confusing. What do you want to say here?

- we added two commas to improve readability: "…distribution, before the measurements are made…"

P5, L123-125: "None of the algorithms perform perfectly". That depends on what you understand as "perfect". I don't think that it is possible at all to retrieve the true atmosphere in a extremely high vertical resolution. However, as far as I know, the second part of this sentence is correct. I would suggest to reformulate the first part.

- We removed this sentence.

Note that you also used external information about the atmosphere for creating your training dataset. You directly applied a priori knowledge by using exponentially decreasing profiles including Gaussian's for your dataset. And I would not say that a priori knowledge does not exist. You look out of the window and know that it is a hazy day so you adapt the a priori. Sometimes you know about local sources or have ancillary measurements available. I would strongly suggest to change this paragraph.

- a priori information, as applied in MAP, is typically a climatological distribution of the parameters of interest so adjusting the a priori according to the observation at the moment is not an appropriate use of the technique. The only technique that is available to measure aerosol extinction coefficient vertical profiles at 355 nm is LIDAR but even in this case the aerosol property information is very limited, so we do not agree that the true a priori for AOD, ASY and SSA at 360 nm exists for many locations. However, since there might

be some locations where some of this information is available we have replaced the sentence with:

"They also require external information about the atmosphere (e.g. temperature and pressure profiles) that might not be readily available at the measurement time scales, and a priori information might not exist'

P7, L176: AMF represents → An AMF represents

- changed

P7, L178: observations → observation

- changed

P7, L181: Where "the" vertical column density

- changed

P7, L183-191: This part could be shortened as you have already introduced aerosols before.

- we removed some of the introductory aerosol information.

P7, L205: "The" VLIDORT code

- changed

Table 1: Why do your Gaussian profiles don't have center heights higher than 2km? Since the vertical sensitivity for higher altitudes is an issue for common algorithms, I am wondering if your algorithm performs better here?

- in our opinion the limiting factor for the retrieval of elevated layers is the MAX-DOAS approach not the specific algorithm itself. By subtracting the zenith AMF we are removing some information. This information is also reduced by typically considering only a "single species" (O_4) at a "single wavelength". Because of this it made sense to limit the tests to 2 km. Further studies will include actual measurements compiled in LIVAS database.

What is the scaling height of your exponential functions?

- they recalculated depending on the total loading and partitioning between the Gaussian and exponential AOD.

P9, L235: What was the reason for changing the grid step width to a coarser resolution for higher altitudes? I have the feeling that your choice of Gaussian profile center heights and retrieval grid steps might deteriorate retrieval results for higher altitudes (as indicated in Fig. 9 and 11).

- Yes, this is correct. Since this is a feasibility study we first wanted to demonstrate that the method works before performing much more elaborate RT and retrieval modeling. We plan to expand the study including higher vertical resolution.

Section 5: I think it would be nice to add more information to this section to explain also the in-between steps and parameters of your CNN and LSTM.

- we have added more details about the ML in the text and supplement

P9, L251: "RMSprop was chosen...". Please explain.

- It is a consensus in the CS community that RMSprop works well on recurrent networks such as LSTM, but RMSprop is an unpublished optimization algorithm.

Figure 4: It would be interesting for the reader to see a similar plot describing the profile shape distribution. You could show 3 more plots for different partitioning, showing Gaussian center heights and width on x and y axis, respectively. Furthermore, the number of profiles with a certain total AOD would also be interesting (especially when looking at Figure 7). I fear that the reader might loose the connection to the actual profile shape due to the rather statistical analysis in the following paragraphs.

- while this is an interesting information we feel it does not add any additional insight into the results;

Figure 5, 6 and 8: It is interesting to see that mean error and standard deviation show areas with high or low values at certain geometries. I was wondering if this is a matter of the scattering angle (angle between incident and outgoing photon assuming single scattering)? Could you please create a plot showing the scattering angle versus the respective error/standard deviation? Since e.g. RAA = 30◦ and a low SZA is equivalent to a large scattering angle (e.g. Fig 5c) it might show issues for certain scattering geometries. In addition, you could check if there are certain profile shapes or aerosol parameters more frequent for areas with a large standard deviation or high mean errors compared to other geometries. I was also wondering about the outliers in all three histograms. Any reason for that?

- Thank you for the recommendation! Since Henyey-Greenstein approximation has a poor representation of the forward and backward scattering we will apply the suggested analysis to the future more realistic aerosol modeling.

P11, L292: I agree that OEM methods also struggle with data inversion measured at small RAA but I was wondering why your synthetic analysis fails?

- We believe this is due to the RT at small RAA, where the photon paths are very "direct" and MAX-DOAS is not "benefitting" from the low elevation angles as much.

P12, L295: "The total AOD retrieval..." or "The retrieval of total..."

- changed

P12, L297: In general, "the" ML algorithm

- changed

P12, L299: What is the reason for the second peak in the histogram?

- *we do not know*

Figure 7: Please explain all depicted quantities (mean, median, percentiles...) in the caption of this figure.

- added: "The central mark indicates the median, the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most

extreme data points not considered outliers, and the outliers are plotted individually using the '+' symbol."

Here, it would also be interesting to see if the largest underestimations correspond to certain profiles or parameters.

- figure 11 shows some of the worst cases that correspond to low AOD, RAA <= 10^o and large SZA >= 80^o.

Figure 9: Why is the error larger for 1.5 km than for 2 km? Since you also included Gaussian's with Peak heights around 2km, I would expected the largest error at higher altitudes. Especially when considering the higher sensitivity of MAX-DOAS measurements for aerosol loads closer to the surface (which can be seen for altitudes lower than 1.5km).

- This is potentially an artifact of the layer depths changes at 1 and 3 km

Figure 10: It appears that there is also an underestimation of the predicted AOD for all sub-figures with true partial AOD's larger than 0.2. For example in the upper left subfigure, but also in the second row (first figure), the third row (2nd and 3rd fig). Do these underestimations correspond to problematic scenarios/geometries/parameters?

- We have not explored the details.

P16, L353: Training and evaluation of "the" ML

- changed

P16-17, L365-372: Points 1 and 2 are valid but only a demand for near real-time applications. I doubt that there is a need for science to have profiles immediately after the measurement. For point 3, I was wondering if this is a major advantage. The dependence of profiling results on SSA is rather small and the Henyey-Greenstein approximation is in most cases a poor representation of the scattering distribution of aerosols. So why should the reader decide for your algorithm when an AERONET station nearby measures "real" phase functions and SSA?

- this is a feasibility study and by no means suggests that the presented algorithm should be used as is. However, what this study does suggest is that more elaborate RT modeling with more realistic RT settings and realistic profiles can open the possibilities to fast and potentially more accurate algorithms using multiple wavelengths and multiple species.

In point 4 you even diminish the potential of your approach by saying that it might be used as an initial guess for other algorithms. To me, this does not sound as if the authors are convinced of the capability of ML algorithms.

- The goal of this research is not to convert the entire community to use ML-based approach but rather to explore its feasibility and possibilities

If this is true, why not? If not, why are there no strong arguments in favor of your approach?

- We personally believe that physics based ML methods can be very effective in accurate inversions, especially of MAX-DOAS data. However, the quality of training data is very important. Ideally, 3D models should be used with complete physics and exhaustive

atmospheric conditions. Availability of such dataset is the part that we are mostly skeptical about. Another issue is the validation of the actual retrievals. There are no other profilers that “sample” in a similar way.