We thank the reviewer for taking the time to assess the manuscript and provide thoughtful, constructive comments. Below, the reviewer's comments are italicized in navy blue text followed by our response to each point.

> *As someone without experience in neural networks, I found the description in section 2.2 somewhat difficult to follow. This is probably standard in ML literature, but given the atmospheric journal/audience I think some context would be helpful here. For example, values for the various activation functions, epochs, batch sizes, cyclical and minimum learning rates are nicely provided, but what these terms mean and how these choices impact the retrieval are not clear except for a brief mention of overfitting. I don't mean to turn this into a primer on ML, but a bit more of a link between these parameters and the results would be nice.*

Indeed, it is standard for ML literature, but it is a great point that the audience for this is not ML experts but rather atmospheric scientists. We have added additional information in Sec. 2.2 that we hope better clarifies these terms and their importance for the retrieval methodology. Excerpts of these additions are provided below:

> "These parameters [layer types and number of nodes per layer] are related to the overall complexity that can be captured by the NN."
> "… activation functions, which introduce non-linearities into the model such that it can approximate complex behaviors."
> "… trained for 60 epochs (number of iterations over the data set; relates to model convergence) using the Adam optimizer (controls how the NN weights and biases are updated), a batch size of 256 (number of samples considered in each training iteration; relates to variance in the gradient and thereby how the model learns), and a cyclical learning rate (scaling factor for magnitude of model updates; cycling reduces number of epochs needed to train model to a certain performance, e.g., Smith, 2015; Himes et al., 2022) …"

> *A lot of emphasis is given to the computational cost of v2.1 and the inability to produce a near real time product using this approach. However, I don't think these conclusions are justified by the paper in its current form. For example:*
>
> *Line 4: "processes performed by traditional retrieval methods are too computationally expensive for near-real-time applications without simplifying assumptions."*
>
> *OMPS retrievals as implemented in v2.1 are an "embarrassingly parallel" problem across profiles, slits and wavelengths and seemingly could be sped up if needed. This may be cost prohibitive, but the paper does not discuss what hardware is currently required to process v2.1 or potential increases needed for a real time v2.1 algorithm vs NN. As it is,*

the OMPS-LP v2.1 data is processed at one day per day, so it seems the throughput of the current v2.1 system is not a problem, only potentially the lag.

Similarly, the cost of running the v2.1 algorithm and NN algorithms is discussed (line 185), but it's unclear what hardware was used for the comparison. Is the NRT approach ran on the same hardware as the v2.1 retrieval? Or are there special hardware requirements for this NRT retrieval (GPUs?) It is mentioned that 97% of the time is spent loading the NN into memory, does this require a machine with a large amount of memory (or more than the v2.1 uses)? Is the training factored into this analysis of computational cost?

Yes, it is an "embarrassingly parallel" problem. Yet, computational resources are limited, and the design of our processing system introduces additional constraints.

Near-real-time processing for OMPS limb data takes place in an automated scheduling and processing system. The automated processing system is a key component to producing NRT data in an on-going basis. Furthermore, the implementation of the non-ML NRT algorithms are essentially the same as the ones used to produce the publicly released standard products (our NRT aerosol algorithm presented here is the exception) but without the requirement that the processing system wait for delayed packets and auxiliary pressure/temperature data. To leverage the existing facilities for NRT processing, the limb granule size of one orbit per granule used in non-NRT processing is preserved.

The current retrieval algorithm has embarrassingly parallel characteristics in that each image in each slit can be retrieved independently and in parallel. However, the processing system limits us to running an entire granule on only one processing node, i.e., we do not break up a granule over multiple nodes for NRT because we do not do this in nominal processing. The processing nodes themselves consist of multi-core shared memory systems and so any parallelism is limited to farming the image-slit retrievals for the entire granule over only the cores on that node. This places a fundamental limit on how much the algorithm can be parallelized when using this processing system. This may change in the future, but it would require a significant investment of both money and worker-hours.

It might be concluded that the inadequacy of the physics-based algorithms for NRT processing will be overcome as nodes with more CPUs become available to process data within the NRT time constraint. However, there are developments that work against that. Experience with Suomi-NPP allowed the sample time for the follow-on mission, NOAA-21 OMPS LP, to be more than halved while maintaining sufficient signal-to-noise (i.e., the number of images per granule is more than doubled as is the concomitant retrieval processing time). Another is that improvements to the physics-based model can reduce the speed of model, potentially breaking compliance with NRT requirements. Perhaps more importantly when considering future developments, our current retrieval model is a 1D approximation, and 2D tomographic retrievals that leverage information from measurements before/after the event of interest add additional computational costs. For these reasons, it seems likely that as computational resources increase, so too will the computational complexity of the retrieval algorithm. Our NN-based methodology ensures that we continue to meet the NRT requirements regardless of how the computational resources and retrieval algorithm change over time.

The “60x faster” result we present is determined by comparing the average processing time (wall clock time) for a single orbit when using the same processing system, and thus it represents an apples-to-apples comparison. The NRT approach requires less RAM (by a factor of ~2) and less CPU time (by orders of magnitude) and so represents a more efficient algorithm in terms of computational resource requirements.

Training time is not factored into the “60x faster” metric reported. It requires around 13 hours to train each model using an NVIDIA V100 GPU, or just over 1 day in total. In the context of reprocessing even 1 year of OMPS LP data, this is a negligible amount of time. However, it is a good point that we did not address in the original manuscript.

We have added text into Section 1 to better clarify these points:

> “In the case of the NASA Atmospheric Composition Processing System used to produce the NASA OMPS LP aerosol product, the available computational resources result in just over 2 hours to process 1 SNPP orbit (and more than double that for NOAA-21 orbits), not including the time to downlink the data and process it into the Level 1 Gridded (L1G) radiance product. At present, this does not meet NASA’s prevailing NRT definition of within 3 hours of the observations. While these runtimes can be reduced by newer computing hardware, processing speed improvements can be offset by updates to the realism of the radiative transfer and retrieval models (e.g., tomographic retrievals as in Zawada et al., 2018).”

> *Line 46-48: Is the time to process the retrieval of a single profile in v2.1 so long that it precludes NRT applications? I would have guessed (maybe incorrectly) the downlink, attitude solution, L1 calibration, atmospheric reanalysis etc. would have been a larger contributor to any lag in NRT products than the retrieval itself. How “NRT” could the NN version be in practice, given this is proposed as a major benefit of the proposed system?*

Yes, much of the NRT processing time is spent on the downlink and L1 calibration. This leaves a very limited amount of time for the aerosol retrievals to meet NASA’s NRT definition (available within 3 hours of the measurements). Version 2.1 of the standard aerosol algorithm requires around 2 hours to process 1 orbit on our processing system. In an ideal world where we are not limited by computational resources, funding, and worker-hours, it could be sufficiently parallelized to meet that NRT definition, but this is not feasible for the reasons mentioned in the previous comment.

The NN-based algorithm requires around 2 minutes to process 1 orbit on ACPS, representing a ~2-hour speedup vs. the standard algorithm, and it ensures we meet NASA’s NRT definition.

Line 71: What are the outputs of these input-output pairs? Is it cloud top altitude, enhanced layer and PSC, as marked in Figure 4 as well as multi-wavelength extinction?

The outputs are the multi-wavelength aerosol extinction profiles, as mentioned on lines 80-81 in the original manuscript. For cloud/enhanced aerosol/PSC altitude, we apply the standard aerosol algorithm's approach, as it is very fast and doesn't require the speed benefits of ML. We have revised the text to better clarify these points:

> "Each case within the data set is comprised of the above listed inputs paired with the corresponding aerosol extinction coefficient reported in the OMPS LP aerosol retrieval version 2.1 data product …"
>
> "To determine the altitude of clouds, enhanced aerosols, and polar stratospheric clouds, we utilize an updated version of the detection algorithm of V2.1, since it is already sufficiently fast and does not require further speed improvements from ML."

Line 94: I would change to "no NASA OMPS LP aerosol retrieval version" as the University of Bremen and University of Saskatchewan OMPS-LP aerosol products both use version 2.6.

Thank you for the suggestion, this is a good point. We have updated the text accordingly.

Line 95: What is meant by "differences in correction methods are consistent"?

Versions 2.5 and 2.6 of the gridded radiances product use different approaches for tangent height and stray light corrections. We can view those correction algorithms as some transformation function applied to the same underlying data. Since each version applies those corrections consistently throughout the product, and the differences between versions are consistent (even if not readily apparent to a person's eyes), the NN can implicitly learn how to account for those consistent differences. This is an important assumption for our methodology because, as discussed in that section, there is not yet a NASA OMPS LP aerosol retrieval version that uses version 2.6 of the gridded radiances product, and only the version 2.6 radiance product is produced in near real time, so we must use that version out of necessity. It is thus a requirement that the NN is able to account for the differences between those versions, and indeed our results show that the NN is able to do so. We have revised the text here to better clarify this point:

> "Since the differences in correction methods can be viewed as different transformation functions applied to the same underlying data, our methodology ignores them and assumes that the NN will learn to perform the transformation from version 2.6 radiances to V2.1 aerosol extinction coefficients."

Line 125-130: Probably obvious for someone in the ML field, but how are results from these two NNs put together?

This is a good question, as there are multiple ways this could be performed. In our case, we make predictions using each of the two NNs, then select the subset of predictions that are relevant for each NN, and finally combine them into a single array. This approach was chosen because it is not only simple algorithmically, but more importantly a given wavelength, latitude combination will only have a relevant prediction from one NN. We have added some additional text in Sec. 2.2 to better clarify how this is performed:

> "To retrieve on 1 orbit, predictions are made with both models, and then the aforementioned relevant subsets of predictions from each model are combined."