David Stelter and Robert Sundberg “Diffusion Learning for Atmospheric Correction," Proc. SPIE 12688, Imaging Spectrometry XXVI: Applications, Sensors, and Processing, (20 October 2023).

DOI: https://doi.org/10.1117/12.2677771

Diffusion Learning for Atmospheric Correction

David Stelter and Robert Sundberg

Spectral Sciences, Inc., 30 Fourth Ave Suite 2, Burlington, MA 01803

ABSTRACT

Diffusion learning is a generative technique commonly applied to create new images or audio directly from sampled noise. The machine learning approach works by applying degrading signals, such as noise, continuously and learning the denoising process with a neural network. In place of noise, other operations can be performed, such as the addition of atmosphere effects using a physics-based radiative transport code. In this paper, we explore coupling the MODTRAN software to a diffusion learning framework. The goal is to apply atmosphere systematically for a variety of reflective surfaces and use diffusion learning to train models for atmospheric correction. To achieve this, we generate a scoped dataset containing randomized Lambertian surfaces with differing solar illumination and surface angles.

Keywords: Machine Learning, Diffusion Learning, Atmospheric Correction

1. INTRODUCTION

Diffusion learning is a generative artificial intelligence (AI) technique commonly employed to create synthetic imagery, and other types of data. Diffusion learning works by training a model, typically deep neural networks, to de-noise the source information.[1] The technique has far ranging applications, from image and document generation with enterprise AI systems to small molecule drug discovery.[2–4]

While application areas of diffusion learning vary, the key constant across them is twofold, 1) an initial seed state typically sampled from random noise and 2) a training process that can be split into incremental, Markovian processes representing the addition of noise systematically to pristine data. Recent work applied diffusion learning to generate novel protein structures by applying diffusion learning on the angles between amino acids that specify the proteins inherent structure.[5] This abstraction from direct prediction of structure enables the neural network to successfully learn a parameterization of an otherwise extensively large configuration space. In the following, we expand along this idea, instead generating parameters to drive a physics-based calculation used to train a neural network for atmospheric correction, rather than learning the direct process.

Atmospheric correction is a key processing step for overhead electro-optical/infrared (EO/IR) imagery. In this process, the detected sensor radiance is corrected with respect to the incident and scattered light due to the effect of observation through the atmosphere. The result is the true surface reflectivity measured by a remote sensor. Significant past effort has been dedicated to atmospheric correction by physics-based, in-scene, and recently data-driven machine learning approaches.[6–12, 12–14] Physics-based approaches are difficult primarily due to the substantial information requirements including the state of the atmosphere, calibration, and estimated optical depth, which are seldom known to the required accuracy. In-scene algorithms have dominated the atmospheric correction space due to no *a priori* information requirements.[11] Recent works with various machine learning approaches have indicated similar or improved performance to state-of-the-art in-scene models.[13–17]

While forward modeling of atmospheric correction can be difficult, the reverse process can be modeled accurately and robustly via physics-based radiative transport (RT) modeling.[18–20] In RT models, a line-of-sight (LoS) is constructed from the target (i.e.- surface material) to the observer (i.e.- sensor platform). For LoS through the atmosphere, the chemical composition, temperature, pressure, and other key quantities are input, rather than unknowns, and used to compute the effect on the observed radiance. This process shares a key similarity with successful applications of diffusion learning in that it is easily broken down to a near-continuous problem.

Further author information: (Send correspondence to D. S.)
E-mail: dstelter@spectral.com

Instead of directly specifying the LoS terminus at the observer, which may shoot through a large amount of atmosphere, diffusion learning can be applied to parameterize the observer altitude. At short diffusion times the observer is very near the sensed target, producing only a minor 'noise' effect when sensing the target that can be continuously strengthened by increasing the LoS column height. Using this abstraction, we develop a training framework that applies diffusion learning to key physical quantities (i.e.- sensor altitude) of a physics-based RT calculation using un-labeled training data.

In this paper, a neural network is trained within a novel diffusion learning framework. However, in place of generating random noise according to the diffusion noise schedule, as in conventional diffusion learning training algorithms, parameters to drive a physics-based RT modeling tool, MODTRAN,[18] are generated and used to directly compute the "noise" contribution due to the atmosphere directly during training. This enables trained models to accurately capture and de-noise the nontrivial, spectral contribution of the atmosphere to an observed sensor signal. The neural network results are presented and discussed. Finally, a synthetic scene is created and results for each pixel are compared to the in-scene QUAC atmospheric correction algorithm.[11]

2. METHODS

Training Procedure

Conventional diffusion learning algorithms learn to predict noise given an input noisy signal and noise schedule. Applications of these algorithms, such as popular image generation tools,[2,3] generate noise on-the-fly to enable a fully self-supervised training regime that requires no labeled data, only a collection of pristine images is required. While training a conventional diffusion learning model, the diffusion time is selected at random and used to generate a noisy image with precisely defined noise level via a 'noise schedule' mapping function. The choice of noise schedule function is important,[21] but herein we consider the simplest noise schedule, $n_s = 1 - t$, where t is the representative diffusion timescale, normalized between 0 (no noise) and 1 (only noise).

For atmospheric correction however, the noise is not known and cannot be directly applied to the pristine signal. Instead, we leverage radiative transfer (RT) calculations, using MODTRAN, to compute the truth noise, N_{truth}, at the specific noise schedule. To achieve this, we map the noise schedule directly to the column pressure of the atmosphere above the pixel. This implies an altitude which is used to drive the RT calculation.

Given a supplied input truth surface reflectance, R_{surf}, the radiance at the specified altitude, L_{alt} is computed by MODTRAN and converted to the altitude specific reflectance, R_{alt} by

$$R_{alt} = \frac{L_{alt}\pi}{I_{sol}\cos(\theta_{sol})} \tag{1}$$

where I_{sol} is the solar irradiance and θ_{sol} is the solar zenith angle. In the signal processing analogy, R_{alt} is the 'noisy pixel' to be de-noised by the neural network.

The truth noise component, N_{truth}, is computed from the MODTRAN result (R_{alt}) and input pixel (R_{surf}), rather than sampled, as

$$N_{truth} = (R_{alt} - \sqrt{n_s}R_{surf})/\sqrt{1 - n_s} \tag{2}$$

Normalization is performed to force the noises (range [-1,1]) into the same numeric range as the pixel reflectance data ([0,1]). The predicted noise is then computed from the neural network, $N^*_{pred} = NN(R_{alt}, n_s)$, and R_{alt} is de-noised using the predicted noises,

$$R^*_{surf} = R_{alt} - \sqrt{1 - n_s}N^*_{pred}/\sqrt{n_s} \tag{3}$$

where the * indices the predicted values.

Two loss functions are computed, the first is the training loss based on the mean squared error (MSE) of the noises, $MSE(N_{truth}, N^*_{pred})$, which is used to update the gradients and train the neural network. Second is just used as a convergence metric based on the de-noised pixel reflectance, $MSE(R_{surf}, R^*_{surf})$. The training step is summarized in Algorithm 1 for an arbitrary data batch.

Algorithm 1 Training Step with MODTRAN

function TRAIN_STEP(x)
  batch_size, nspectral_bins, channels = x.shape

  # sample diffusion time
  diffusion_time = random.uniform(0, 1, size = (batch_size, 1, 1))

  # compute noise schedule and generated altitude
  n_s = noise_schedule(diffusion_time) # 1−diffusion_time
  $h = -\frac{T_0 R}{g M_{air}} \log(n_s)$ # T_0, R, g, and M_{air} are physical constants

  # compute truth noise from MODTRAN result
  for data **in** batch_size **do**
    x_{noise} = run_MODTRAN(x, h)
    $N_{truth} = (x_{noise} - \sqrt{n_s}x)/\sqrt{1 - n_s}$

  # de-noise with neural network, compute training loss
  N_{pred} = neural_network(x_{noise}, n_s)
  $loss = (N_{pred} - N_{truth})^2$
  return $loss$

Neural Network Structure

A straightforward 1-D convolutional neural network (CNN) structure was chosen to demonstrate the training concept with MODTRAN. This structure performs convolutions across the spectral bins with padded sizing ensuring that at each layer, the number of spectral bins remains constant. This is in contrast to designed feature extraction networks, like U-Nets, common in applying diffusion learning to image data.

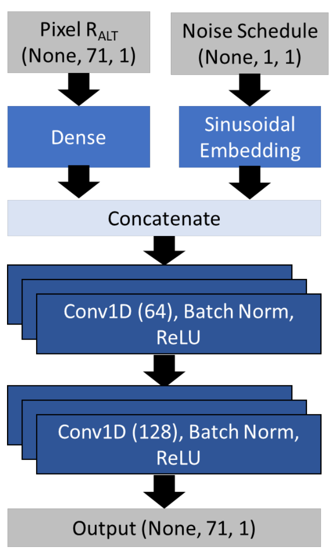

Figure 1. Neural network structure used for atmospheric correction with 71 spectral bins and a single feature channel

An overview of the network structure is shown in Figure 1. Two inputs are supplied to the neural network, just as in conventional diffusion learning approaches. The first is a noisy pixel, corresponding to either an arbitrary altitude and noise schedule, or top of atmosphere and noise schedule of 0. The spectral data is processed first by a dense layer while the noise schedule is processed by a sinusoidal embedding layer with 25 embedding dimensions[22] to ensure network sensitivity with respect to that input regardless of the amount of other input data. The layers are concatenated and processed by a 1-D CNN with two filter levels. The first includes three successive 64 filter layers with batch normalization and ReLU activation. The second set of 3 layers contain 128 filters, batch normalization, and ReLU activation. The final layer is a dense layer that outputs the predicted, de-noised pixel with same spectral resolution as the supplied input, R_{alt}.

As the primary goal of this paper is testing the training algorithm and incorporation of physics-based noise, little optimization of the network structure and its hyperparameters were performed. The batch size was set at 64 data samples and the learning weight was set initially to $1e-3$ with weight decay set to $1e-4$. Data for each training batch was sampled from a large, synthetic data set of pixel reflectances containing $21,000$ unique Lambertian spectral response functions. Validation data was obtained from MODTRAN's *spec_alb.dat* database which contains ~ 100 scene-averaged and measured spectral material responses.

Data Generation

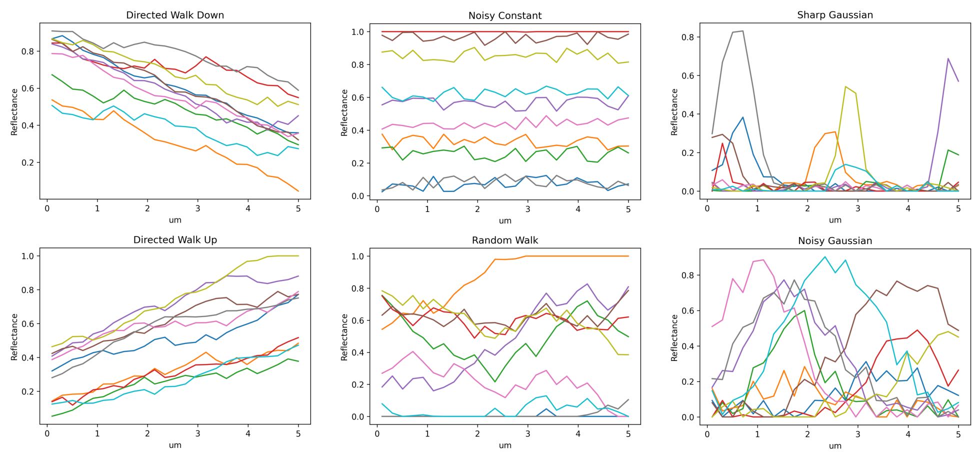

Figure 2. Synthetic training data examples of pristine pixel reflectance

Data was generated with six main motifs mimicking real material spectral response. A sampling of example training data is shown in Figure 2. Key motifs include random walks with biases both upward and downward in reflectance space as a function of microns, constants with noise added, as well as sharp and broad Gaussian-like spectral responses with shifted centers and randomized widths. Data was not checked to ensure physical consistency, such as a strong feature in an unlikely spectral region, but is guaranteed to be in the range of [0,1].

Data was generated for each motif $1,000$ times. For each motif, key properties such as Gaussian height or amount of sampled noise added to the constant signal, were also sampled. In total, $21,000$ unique spectral responses were created and drawn from during training. While each data is unique, similar data certainly exists within the generated training set. This was mitigated by using a limited number data for training initially to determine the minimal number of required data, approximately $3,200$, for predictive performance deemed adequate for this study.

MODTRAN Setup

The key idea behind the approach is to replace the generation of noisy images from Gaussian noise with known mean and variance with less characterized noise computed from physics. We employ MODTRAN[18] for radiative transfer calculations to accurately model the atmospheric contribution to the surface reflectance.

MODTRAN calculations were performed with atmospheric scattering and using a Lambertian surface reflectance model with a Nadir-viewing geometry with fixed overhead solar position. The atmosphere was held fixed to mid-latitude summer day with fixed, cold surface temperature of 252.166 K. The output spectral range was set to 1.2 to 5.0 microns with 0.002 step size and 0.005 spectral bin resolution.

The spectral range studied was initially focused on the short-wave infrared (SWIR), but several wavelengths in the mid-wave infrared (MWIR) were added to test the approach in that region, despite the lack of variation

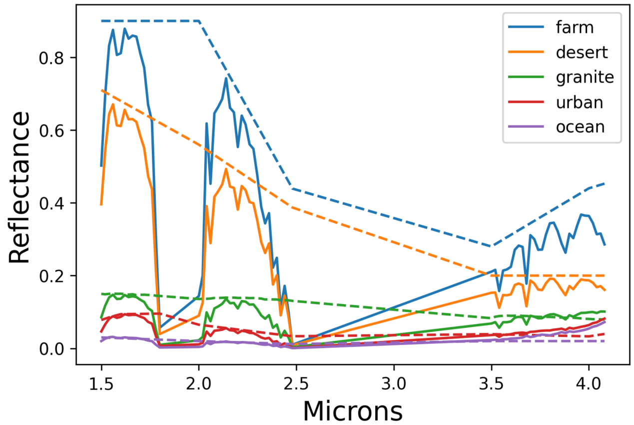

Figure 3. Example Validation Data (colors) with truth (dashes) and MODTRAN runs to produce NN input data with noise at top-of-atmosphere (lines).

in surface temperature for the training data. For each MODTRAN calculation, spectral data was extracted at 0.02 micron intervals between the ranges of 1.5 to 1.8, 2.0 to 2.5, and 3.5 to 4.1 microns. While this is a unusual spectral range, it was picked to be both close to strong atmospheric absorbers where the reflectance signal is inherently weak, and to extend the approach to alternative bands for atmospheric correction. Example truth data and processed MODTRAN data, to be used as NN inputs, is shown in Figure 3.

MODTRAN is run in parallel during training which increases the computational cost substantially. However, due to the relatively small network employed, the training time per-step is approximately ~ 5 minutes per batch on AMD EPYC 7642 processors.

3. RESULTS

Results for training the neural network are shown in Figure 4. The network converges quickly, with reasonable results obtained by Epoch 20. Generally, the procedure for recovering the de-noised image follows a reverse diffusion approach with use of the exponential moving average weights of the neural network to ensure smoothness. In place of that approach, we perform reverse diffusion across all noise levels and average the results using the normal network weights. This was done as it is currently unclear if with physically relevant noise present, there should be a baseline noise level in the image predictions. In practice, this modification to the reverse diffusion procedure yielded the best predictive result. 50 steps were used for all reverse diffusion pixel generation. Computationally, the requirements for reverse diffusion at this level of output data were low. A single pixel generation completed in ~ 0.73 seconds on average for 71 spectral bins.

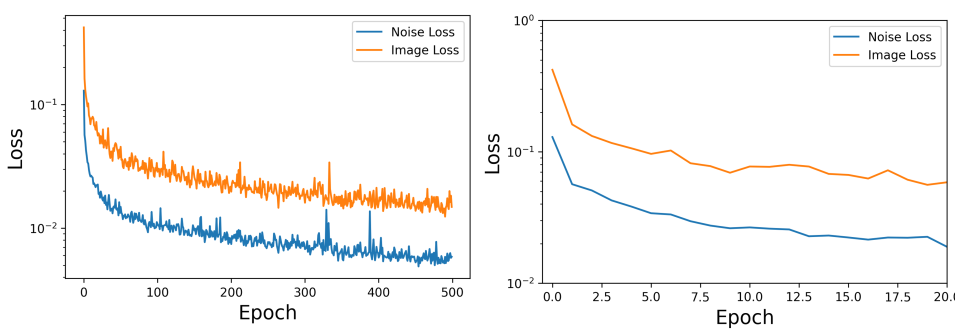

Figure 4. Convergence for noise training loss and image loss

Neural network predictions were validated against MODTRAN's *spec_alb.dat* database. Several data were omitted due to invalid spectral ranges, but the bulk of the database was used as-is. Five specific spectral albedos were selected (farm, desert, granite, urban, ocean) for a deeper investigation. Neural network predicted reflectance values and truth data is presented for these five materials in Figure 5. Agreement is largely excellent for these materials. Near the selected spectral window edge, and particularly near ~ 2.0 microns, performance is degraded. The selected wavelength range in that region goes very near large atmospheric spectral features that are optically thick. These wavelengths were selected as a stressing test for the approach. The trained network was not able to recover the truth signal in this regions. Further, conventional remote sensing strategies typically avoid these regions of the spectrum anyways.

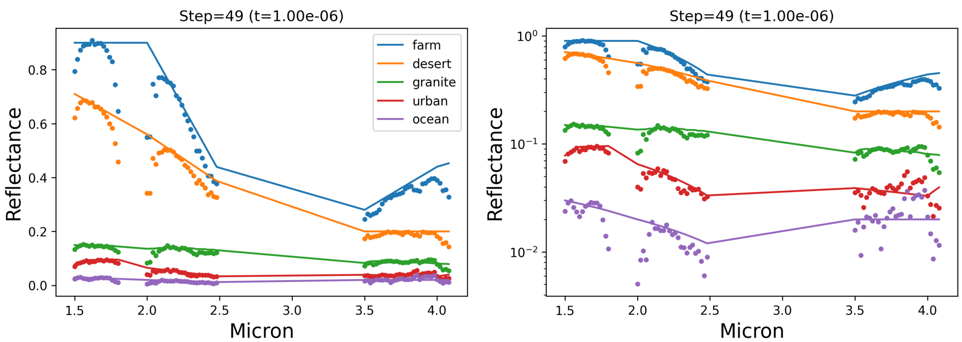

Figure 5. Neural network predicted, atmospherically corrected surface reflectance (dots) and truth data (lines) for selected validation data (colors)

Truth-normalized, $|pred - truth|/truth$, and absolute error for the entire validation database is shown in Figure 6. Validation data consists of both smoothed and approximate surface reflectances (i.e. 'farm') as well as measured properties (i.e.- 'granite') with numerical scatter present in the data.

Of immediate notice is the large normalized error for material 'constant 0%', due to the complete lack of signal. Further, other dark and near-black surfaces (ocean, black plastic) also exhibit high normalized error. However, when paired with the absolute error metric, it becomes clear that the neural network is correctly predicting these material properties, but with some scatter. In particular, 'constant 0%' has the lowest absolute error present of all validation data. For other materials, performance is good leading to low absolute errors for many materials. Error can also be shown as a function of spectral bin, as in Figure 7 for the five selected validation data. Most error is centralized to the window edges, where the atmospheric transmittance is small. This is an expected point of failure, as is the case for many other atmospheric correction algorithms. The 'ocean' material, which is nearly black, has the highest peaks in normalized error (not shown) near the strong H_2O atmospheric features. However, for absolute error, the 'farm' and 'desert' materials are the worst predicted, and brightest reflectors, yet still give adequate results.

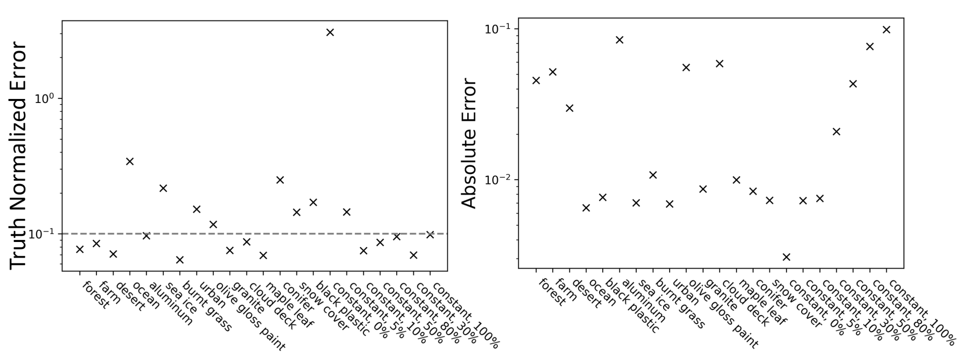

Figure 6. Errors for the validation dataset

Finally, we compare the neural network approach to the QUick Atmospheric Correction (QUAC) algorithm.[11] QUAC is an in-scene atmospheric correction algorithm that only requires information from the bulk scene. To

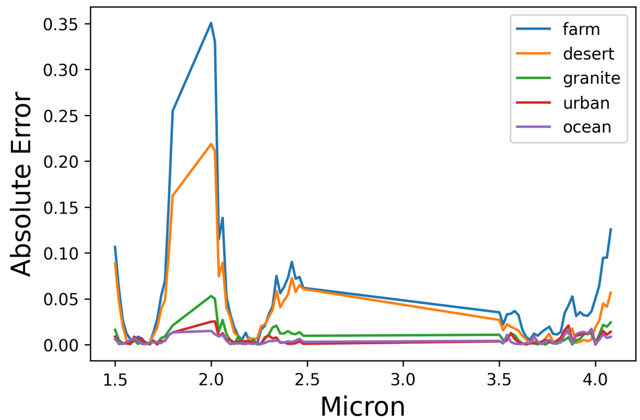

Figure 7. Spectral dependence of error

compare the diffusion learning neural network approach with QUAC we created a synthetic 1-D scene consisting of selected data from the validation database. In total, 16 pixels were added to the scene. It should be noted that this is a rather small scene with few endmembers for QUAC. Further, the scene pixels used to construct the QUAC scene utilized the same constant atmosphere used for training the neural network approach. A true comparison with QUAC on a larger scene with a different atmosphere than used during training should be a goal for future work. Nonetheless, we compare both algorithms on a handful of single-pixel materials, shown in Figure 8.

Performance of the neural network outperforms QUAC for both SWIR windows studied for all materials except 'ocean'. For 'ocean', the neural network predicts negative reflectance values nearest the windows. This is nonphysical and could be an area for follow-up study. Other materials, such as 'maple leaf', exhibit nearly exact shape matching in the spectral reflectance profile. While QUAC does a good job with the shape as well, it is unable to reproduce the correct reflectance magnitude.

4. CONCLUSIONS

A neural network was trained for atmospheric correction using a diffusion learning framework and the atmospheric radiative transport code MODTRAN. Once trained, the network requires only the top-of-atmosphere reflectance (or observed radiance with solar irradiance and solar zenith angle) as input and outputs the predicted noise to reconstruct the surface reflectance. A novel training procedure based on existing generative AI / diffusion learning techniques was discussed. In place of generating noise, parameters to drive a forward physics-based calculate are generated. Noise is then computed directly from the results of the physics calculation, given the pristine data as input, and used to train a 'de-noising' neural network. While above procedure is applied to atmospheric correction, the concept of replacing generated noise with generated simulation parameters is general and could be extended to other approaches and physics solvers.

The network was trained with unlabeled, synthetic data and validated against MODTRAN's spectral albedo database. The trained network was able to successfully predict atmospherically corrected spectral profiles for a range of different materials not used in training. Several atmospheric windows were considered in this pilot study, two SWIR regions and some wavelengths in the MWIR. The network was able to successfully learn to correct in each of these regions simultaneously.

Approximate errors for the approach were quantified and showed to have spectral dependence. The largest error is accumulated nearest any large atmospheric absorption features. This could likely be avoided by careful choice of wavelengths of interest. Darker materials have larger normalized errors, but lower overall absolute error than lighter materials.

A comparison was made to an industry-standard in-scene atmospheric correction code, QUAC. For all materials tested, the neural network produced more accurate spectral reflectance profiles than QUAC, except for

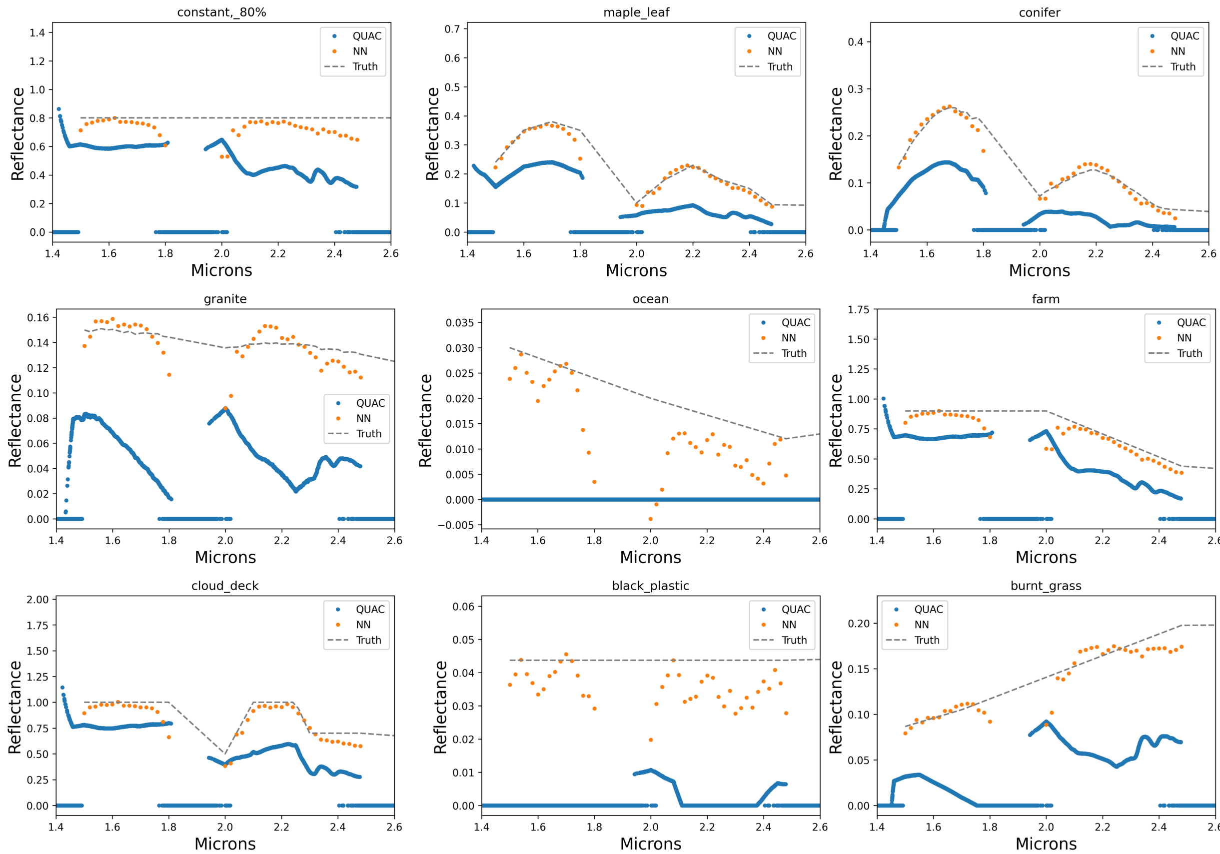

Figure 8. Comparison of atmospheric correction algorithms, QUAC (blue) diffusion learning neural network (orange), Truth (grey)

dark materials near the spectral window edge. One example was identified where the neural network produced a nonphysical result, but otherwise results successfully reproduced truth surface reflectance spectral profiles. This is attributed to the involvement of the physics-based MODTRAN RT solver directly integrated within the training loop. This enables the network to learn from the key atmospheric physics properties rather than from learning directly from data alone.

While the trained network successfully demonstrated the diffusion learning concept, the present study only a small demonstration of the techniques potential. Next steps include, for example, reducing the constraints on the training data (solar zenith and atmosphere composition), additional parameters to capture observable effects in alternative spectral bands, such as surface temperature in the mid-wave / adaption to specific sensor bands of interest, and application to full scenes with pixels containing multiple endmembers.

ACKNOWLEDGMENTS

The authors thank Timothy Perkins (SSI) for valuable discussion and insight throughout this effort.

REFERENCES

[1] Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H., "Diffusion models: A comprehensive survey of methods and applications," (2023).

[2] OpenAI, "Dall-e 2," (2022).

[3] Midjourney, "Midjourney user guide," (2022).

[4] Corso, G., Stärk, H., Jing, B., Barzilay, R., and Jaakkola, T., "Diffdock: Diffusion steps, twists, and turns for molecular docking," (2023).

[5] Wu, K. E., Yang, K. K., van den Berg, R., Zou, J. Y., Lu, A. X., and Amini, A. P., "Protein structure generation via folding diffusion," (2022).

[6] Ientilucci, E. J. and Adler-Golden, S., "Atmospheric compensation of hyperspectral data: An overview and review of in-scene and physics-based approaches," *IEEE Geoscience and Remote Sensing Magazine* **7**(2), 31–50 (2019).

[7] Berk, A., Adler-Golden, S., Ratkowski, A., Felde, G., Anderson, G., Hoke, M., Cooley, T., Chetwynd, J., Gardner, J., Matthew, M., et al., "Exploiting modtran radiation transport for atmospheric correction: The flaash algorithm," in [*Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.(IEEE Cat. No. 02EX5997)*], **2**, 798–803, IEEE (2002).

[8] Cooley, T., Anderson, G. P., Felde, G. W., Hoke, M. L., Ratkowski, A. J., Chetwynd, J. H., Gardner, J. A., Adler-Golden, S. M., Matthew, M. W., Berk, A., et al., "Flaash, a modtran4-based atmospheric correction algorithm, its application and validation," in [*IEEE international geoscience and remote sensing symposium*], **3**, 1414–1418, IEEE (2002).

[9] Miller, C. J., "Performance assessment of acorn atmospheric correction algorithm," in [*Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII*], **4725**, 438–449, SPIE (2002).

[10] Thompson, D. R., Gao, B.-C., Green, R. O., Roberts, D. A., Dennison, P. E., and Lundeen, S. R., "Atmospheric correction for global mapping spectroscopy: Atrem advances for the hyspiri preparatory campaign," *Remote Sensing of Environment* **167**, 64–77 (2015).

[11] Bernstein, L. S., Adler-Golden, S. M., Jin, X., Gregor, B., and Sundberg, R. L., "Quick atmospheric correction (quac) code for vnir-swir spectral imagery: Algorithm details," in [*2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*], 1–4 (2012).

[12] Vermote, E., Roger, J.-C., Franch, B., and Skakun, S., "Lasrc (land surface reflectance code): Overview, application and validation using modis, viirs, landsat and sentinel 2 data's," in [*IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*], 8173–8176, IEEE (2018).

[13] Basener, B. and Basener, A., "Gaussian process and deep learning atmospheric correction," *Remote Sensing* **15**(3) (2023).

[14] Koch, J., Forland, B., Doster, T., and Emerson, T., "A neural differential equation formulation for modeling atmospheric effects in hyperspectral images," in [*Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imaging XXIX*], Velez-Reyes, M. and Messinger, D. W., eds., **12519**, 125190E, International Society for Optics and Photonics, SPIE (2023).

[15] Shah, M., Raval, M. S., and Divakaran, S., "A deep learning perspective to atmospheric correction of satellite images," in [*IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*], 346–349 (2022).

[16] Rusia, P., Bhateja, Y., Misra, I., Moorthi, S. M., and Dhar, D., "An efficient machine learning approach for atmospheric correction," *Journal of the Indian Society of Remote Sensing* **49**, 2539–2548 (2021).

[17] Sun, J., Xu, F., Cervone, G., Gervais, M., Wauthier, C., and Salvador, M., "Automatic atmospheric correction for shortwave hyperspectral remote sensing data using a time-dependent deep neural network," *ISPRS Journal of Photogrammetry and Remote Sensing* **174**, 117–131 (2021).

[18] Berk, A., Conforti, P., Kennett, R., Perkins, T., Hawes, F., and Van Den Bosch, J., "Modtran® 6: A major upgrade of the modtran® radiative transfer code," in [*2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*], 1–4, IEEE (2014).

[19] Berk, A. and Hawes, F., "Validation of modtran® 6 and its line-by-line algorithm," *Journal of Quantitative Spectroscopy and Radiative Transfer* **203**, 542–556 (2017).

[20] Kotchenova, S. Y., Vermote, E. F., Matarrese, R., and Klemm Jr, F. J., "Validation of a vector version of the 6s radiative transfer code for atmospheric correction of satellite data. part i: Path radiance," *Applied optics* **45**(26), 6762–6774 (2006).

[21] Chen, T., "On the importance of noise scheduling for diffusion models," (2023).

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., "Attention is all you need," (2017).