9-2020

Physics-constrained Hyperspectral Data Exploitation Across Diverse Atmospheric Scenarios

Nicholas M. Westing

Physics-Constrained Hyperspectral Data Exploitation Across Diverse Atmospheric Scenarios

DISSERTATION

Nicholas M. Westing, Major, USAF

AFIT-ENG-DS-20-S-021

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This is an academic work and should not be used to imply or infer actual mission capability or limitations.

AFIT-ENG-DS-20-S-021

PHYSICS-CONSTRAINED HYPERSPECTRAL DATA EXPLOITATION ACROSS DIVERSE ATMOSPHERIC SCENARIOS

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Nicholas M. Westing, B.S.E.E., M.S.E.E.

Major, USAF

September 2020

AFIT-ENG-DS-20-S-021

PHYSICS-CONSTRAINED HYPERSPECTRAL DATA EXPLOITATION ACROSS DIVERSE ATMOSPHERIC SCENARIOS

Nicholas M. Westing, B.S.E.E., M.S.E.E.
Major, USAF

Committee Membership:

Brett J. Borghetti, PhD
Chairman

Kevin C. Gross, PhD
Member

Christine M. Schubert Kabban, PhD
Member

Robert B. Greendyke, PhD
Dean's Representative

ADEDEJI B. BADIRU
Dean, Graduate School of Engineering and Management

Abstract

Hyperspectral target detection promises new operational advantages, with increasing instrument spectral resolution and robust material discrimination. Resolving surface materials requires a fast and accurate accounting of atmospheric effects to increase detection accuracy while minimizing false alarms. This dissertation investigates deep learning methods constrained by the processes governing radiative transfer to efficiently perform atmospheric compensation on data collected by Long-Wave Infrared (LWIR) hyperspectral sensors. The atmospheric compensation problem is approached from the perspective of increasing operational demands, focusing on methodologies that accelerate data throughput while providing accurate data products. First, the importance of atmospheric compensation is illustrated from the standpoint of LWIR land cover classification followed by an investigation of atmospheric dimension reduction methods to support radiative transfer modeling. Next, two new atmospheric compensation algorithms are presented: DeepSet Atmospheric Compensation (DAC) and Multimodal DeepSet Atmospheric Compensation (MDAC). Both approaches depend on generative modeling techniques and permutation-invariant neural network architectures to predict atmospheric transmittance, upwelling radiance and downwelling radiance from in-scene data only.

Both DAC and MDAC utilized a worldwide database of atmospheric measurements forward modeled with MODerate resolution atmospheric TRANsmission (MODTRAN) 6.0 to operate across globally and temporally-diverse atmospheric conditions. Generative modeling techniques were used to project the forward modeled data to a low-dimensional data manifold. Data manifold sampling resulted in realistic, spectrally-resolved transmittance, upwelling radiance and downwelling radiance vectors. Additionally, MDAC also predicts the atmospheric measurements (T, H_2O, O_3) that would have produced the pre-

dicted transmittance and radiance vectors. The LWIR radiative transfer equation was used in all network loss functions to minimize at-sensor radiance mean-squared error rather than directly computing transmittance, upwelling and downwelling radiance mean-squared error. This modification to the network loss function provided lower reconstruction error across a diversity of materials, leading to better performance in the remote sensing task. Additionally, the MDAC algorithm employed a weighted atmospheric state loss function, driven by knowledge of atmospheric radiative transfer.

Both the DAC and MDAC methods were evaluated on collected LWIR hyperspectral data resulting in comparable performance to Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR) while reducing atmospheric compensation time. Target detection results were compared between MDAC and FLAASH-IR demonstrating comparable performance, however, MDAC resulted in an 8 times reduction in target detection time. This accelerated target detection pipeline is necessary for many real-world, time sensitive operations.

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

List of Abbreviations

1D-CNN One-Dimensional Convolutional Neural Network.

AAC Autonomous Atmospheric Compensation.

AAE Adversarial Autoencoder.

ACE Adaptive Coherence/Cosine Estimator.

AE Autoencoder.

AI Artificial Intelligence.

ANN Artificial Neural Network.

AR alpha residuals.

ASTER Advanced Spaceborne Thermal Emission and Reflection Radiometer.

ATREM atmospheric removal program.

AUC-BT brightness temperature RMSE area under the curve.

CAE Contractive Autoencoder.

CNN Convolutional Neural Network.

DAC DeepSet Atmospheric Compensation.

DAE Denoising Autoencoder.

ELU exponential linear unit.

EM electromagnetic.

FLAASH Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes.

FLAASH-IR Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared.

FPA focal plane array.

GAN Generative Adversarial Network.

GMM Gaussian Mixture Model.

GPU Graphical Processing Unit.

HEHR High Emissivity High Reflectivity.

HELR High Emissivty Low Reflectivity.

HSI Hyperspectral Imagery.

ICA Independent Component Analysis.

ILS instrument line shape.

ILSVRC ImageNet Large-Scale Visual Recognition Challenge.

ISAC In-Scene Atmospheric Compensation.

ISR Intelligence, Surveillance and Reconnaissance.

JPL Jet Propulsion Laboratory.

json JavaScript Object Notation.

KL Kullback-Leibler.

KNN K-nearest neighbors.

LBLRTM Line-by-Line Radiative Transfer Model.

LIDAR Light Detection and Ranging.

LUT look up table.

LWIR Long-Wave Infrared.

MDAC Multimodal DeepSet Atmospheric Compensation.

MMAE Multimodal Autoencoder.

MNF minimum noise fraction.

MODTRAN MODerate resolution atmospheric TRANsmission.

MSE Mean-Square Error.

MWIR midwave infrared.

NASA National Aeronautics and Space Administration.

NATO North Atlantic Treaty Organization.

NESR noise-equivalent spectral radiance.

PCA Principal Component Analysis.

PCRTM principal component radiative transfer model.

PCRTTOV principal component radiative transfer for TOVs.

RBF Radial Basis Function.

ReLU Rectified Linear Unit.

RMSE root mean square error.

ROC Receiver Operating Characteristic.

RT Radiative Transfer.

RX Reed-Xiaoli.

SAE Stacked Autoencoder.

SAM Spectral Angle Mapper.

SCR Signal to Clutter Ratio.

SEBASS Spatially Enhanced Broadband Array Spectrograph System.

SMF Spectral Matched Filter.

SNR Signal to Noise Ratio.

SPIE Society of Photonic Instrumentation Engineers.

SVD singular value decomposition.

SVM Support Vector Machine.

SWIR shortwave infrared.

TES Temperature-Emissivity Separation.

TIGR Thermodynamic Initial Guess Retrieval.

TUD Transmittance, Upwelling, and Downwelling.

VAE Variational Autoencoder.

VNIR visible and near-infrared.

PHYSICS-CONSTRAINED HYPERSPECTRAL DATA EXPLOITATION ACROSS DIVERSE ATMOSPHERIC SCENARIOS

I. Introduction

Remote sensing encompasses the science and art of collecting information about the Earth's surface without physically contacting the materials under investigation [1]. While the field of remote sensing includes multifarious technologies and methodologies, this research is focused on electro-optical data collected across hundreds of contiguous wavelength measurements, known as hyperspectral remote sensing. Hyperspectral data is passively collected with sensors receptive to electromagnetic (EM) energy between 0.4 - 14 µm. This research investigates thermal hyperspectral data, specifically the Long-Wave Infrared (LWIR) domain encompassing 7.5 - 14 µm. Infrared-active gases such as water vapor, ozone and carbon dioxide distort the collected LWIR data and must be accounted for, a process known as atmospheric compensation. Performing atmospheric compensation is a necessary data processing step to retrieve the detailed surface information measured by the hyperspectral sensor. The hundreds of contiguous bands provide information useful for fields of study from forestry and geology to search and rescue operations and target detection [2, 3]. This research leverages machine learning and deep learning approaches to accelerate this necessary data processing step, allowing for faster data exploitation.

The technological evolution of hyperspectral remote sensing can be traced to World War II and the Cold War where defense satellites were some of the first technologies to collect broadband images. Additionally, the advantages of spaceborne imaging were recognized in the 1960s as part of the manned space programs: Mercury, Gemini and Apollo [4]. The photographs taken by crew members aboard these missions proved detailed information

about the Earth could be collected from space. These results accelerated development of satellite programs such as Landsat, to collect multispectral information from unmanned satellites. The first Landsat satellite collected multispectral data across 7 bands spanning visible and near-infrared (VNIR) to LWIR spectrum [2]. This sensor provides repetitive measurements of the Earth in an easily accessible digital format, allowing researchers from diverse fields of study to leverage multispectral data. The small number of spectral channels provided by a multispectral sensor works well for land-cover classification tasks in which materials such as forest, cropland and urban areas are easily differentiated. Landsat continues to collect information about the Earth's surface with Landsat 7 and 8 currently in operation.

The success of multispectral sensors such as Landsat led to further development in improving sensor spectral resolution. Scientists in the 1980s at the National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL) are credited with creating the first sensor with sufficient spectral resolution to be designated hyperspectral imaging. Rather than tens of bands present in multispectral sensors with spectral resolutions approaching 100 nm, hyperspectral sensors collect across hundreds of bands with spectral resolution on the order of 10 nm [5]. This increased spectral resolution is necessary for differentiating similar materials within a scene, identifying concealed objects or performing change detection with temporally-varying hyperspectral data [6].

1.1 Motivation

Hyperspectral remote sensing is a key enabling technology to increase the information content in Intelligence, Surveillance and Reconnaissance (ISR) data products. These products must be generated at unrivaled speeds to address many defense applications. The North Atlantic Treaty Organization (NATO), European Defense Agency and U.S. Depart-

ment of Defense identified the following defense applications well-suited for hyperspectral technologies [7–10]:

- Battlespace situational awareness
- Discrimination between targets and decoys
- Defeating camouflage
- Early warning for long range missiles and space surveillance
- Detection of weapons of mass destruction
- International treaty monitoring
- Landmine detection.

Many of these applications can be addressed with monochromatic or multispectral data, but the additional information provided by hyperspectral sensors can accelerate the decision-making process. Additionally, VNIR/shortwave infrared (SWIR) sensors can now be found on small unmanned aerial vehicles to support a multitude of defense use-cases [11]. Miniaturization of LWIR sensors is limited by the size of cryogenic coolers, however, recent breakthroughs in cryocooler designs are leading to smaller thermal detectors [12], paving the way toward wider spread LWIR hyperspectral sensor use. These advancements can lead to data saturation, requiring efficient algorithms and methods to produce useful data products. As the number of LWIR data sources grow, the ability to quickly and accurately perform atmospheric compensation becomes more important since all other data exploitation relies on this processing step.

Atmospheric compensation converts measured at-sensor radiance to surface-leaving radiance by estimating the vertical distribution of aerosols, water vapor and atmospheric temperature between the sensor and target. If other meta data such as weather forecasts

or recent atmospheric measurements are available, this estimation problem becomes more tractable [13]. In many defense applications, meta data is not available and atmospheric compensation must be performed quickly to support time-sensitive data products. Atmospheric compensation errors can corrupt the recovered surface-leaving radiance, limiting the data product accuracy and slowing the decision-making process.

Defense-focused hyperspectral data processing requires both fast and accurate atmospheric compensation methods to address current and future security challenges. Model-based methods relying on computationally-expensive radiative transfer calculations are accurate, but often too time-consuming. Faster model-based atmospheric compensation is possible with precomputed lookup tables containing likely atmospheric conditions to be encountered during flight [14]. In-scene methods are ideal for real-time analysis, but are prone to errors if model assumptions are not satisfied. These assumptions include the presence of distinct materials in the scene and a clear band to estimate ground temperature [15]. Recent advances in machine learning promise solutions to the atmospheric compensation problem, but these methods must be validated to provide confidence in the generated data product. As this research will demonstrate, constrained machine learning solutions dependent on radiative transfer can provide both accurate and timely atmospheric compensation estimates. Applying finely tuned machine learning and artificial intelligence algorithms to portions of the image processing chain represents a significant step forward in advancing hyperspectral technology.

1.2 Objective

This research identifies suitable techniques from the fields of machine learning and deep learning to expedite atmospheric compensation and shorten target detection time. Unlike other in-scene atmospheric compensation techniques, this approach does not involve a manual pixel selection step performed by the user, further increasing data throughput for

current and future high data-rate LWIR hyperspectral systems. The algorithms investigated in this research are influenced by the equations governing transmission of EM radiation and evaluated using domain-specific metrics. This approach allows both training and evaluation performance to be placed in the context of the remote sensing task. Rather than using mean-squared error metrics between target and prediction signals, the methods presented here rely on domain-specific transforms for measuring model error. This results in models tuned with preferable properties, such as low error for reflective materials.

The deep learning architectures utilized in this research are derived from generative models and permutation-invariant networks. Breakthroughs in generative modeling have led to ultra-realistic image generation [16, 17]. These generative approaches are applied here to create representational atmospheric state estimates, useful for radiative transfer modeling and atmospheric compensation. Domain knowledge is applied to interrogate the generative model and reveal the physically-plausible predictions made as the model is tuned. Permutation-invariant neural networks are another recent advancement in deep learning, allowing networks to learn a fixed quantity of representative features from a set composed of identically-defined members. This research investigates permutation-invariant networks for predicting a single atmospheric output for a set of measured pixel spectra. Both the permutation-invariant network and generative model are trained and evaluated using domain-specific information.

To evaluate the efficacy of these approaches, models are tested on collected data and compared to established atmospheric compensation methods. Target detection is performed to demonstrate comparable performance, while reducing overall detection time. Demonstrating comparable performance on collected data is necessary to validate the limitations and assumptions outlined in the next section.

1.3 Limitations and Assumptions

The field of atmospheric compensation is extensive and well-studied in both the VNIR/SWIR and LWIR domains. Atmospheric compensation research typically employs several limitations and assumptions to make the problem tractable, while still maintaining an accurate solution for most scenarios. The following assumptions and limitations common to most atmospheric compensation research are applied:

- Target detection is performed in cloud-free conditions along the line of sight between the sensor and the ground. This limits the atmospheric conditions that must be considered while maintaining an operationally relevant remote sensing configuration. Additionally, cloud identification is not considered in this research.

- All imagery is collected or simulated in a nadir-viewing geometry with lambertian surfaces assumed for all materials. Atmospheric compensation in off-nadir geometries is more complex because of varying path lengths between the sensor and scene objects. The method presented in [18] investigates oblique in-scene atmospheric compensation. By applying a nadir-viewing assumption, this research remains applicable to a wide range of current sensors. Future work will consider off-nadir extensions to this research.

- Atmospheric conditions are assumed homogeneous for all pixels within a single hyperspectral cube. This assumption simplifies synthetic data generation allowing for a multitude of atmospheric conditions to be tested for an ensemble of materials. This assumption holds for many real-world scenarios and does not limit the applicability of this research. As sensor altitude increases, atmospheric variability across a scene will increase. This research primarily investigates sensor altitudes between 0.15-3.05 km.

- This research only considers LWIR data in the range 7.5-14.0 µm. Demonstrating the utility of deep learning approaches in the LWIR domain for atmospheric conditions is an important first step toward extending these methods to the midwave infrared (MWIR) and VNIR/SWIR domains. Additionally, the LWIR domain is of interest for many operational scenarios.

1.4 Summary of Research Objectives and Contributions

This section reviews the major research objectives and contributions for each research study in this dissertation, using the nomenclature studies A - D for the results presented in Chapters III - VI respectively. For clarity, a table outlining the objectives and contributions is presented for each study.

Study A investigated LWIR hyperspectral classification using collected data spanning a 24 hour period. This research found that both shallow and deep classifiers perform poorly when trained on at-sensor radiance data, rather than emissivity or surface-leaving radiance [19]. Intentionally biasing the training data such that validation data included pixel temperatures outside the training data range also led to lower classification performance. Under the biased training data configuration, the CNN classifier resulted in the highest classification performance, although still unacceptably low for the data set. The biased training data configuration was examined because of real-world data collection limitations, where it is difficult to guarantee a training data set that encompasses all atmospheric conditions, surface materials and material temperatures. By performing atmospheric compensation, all classification results improved with nearly perfect performance for most materials. This transformation was not easily understood by multiple neural network layers and demonstrates a useful preprocessing method for land cover classification problems.

Study B compared dimension reduction approaches to create faster radiative transfer models. These models can be used to support faster model-based atmospheric compen-

Table 1. Summary of objectives and contributions from Study A discussed in Chapter III

	Objectives
AO1	Compare classification performance using temporally-varying LWIR hyperspectral data
AO2	Evaluate how atmospheric compensation effects temporally-varying LWIR hyperspectral data classification
	Findings and Contributions
A1	Classification algorithms such as Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) are unable to generalize when the validation data contains material temperatures outside the training data surface temperature distribution.
A2	The CNN classifier demonstrated a 7% higher classification accuracy than SVM and ANN when evaluated on pixels with temperatures outside the training data range. The large convolutional filters extracted features across multiple bands to identify salient characteristics that were invariant to the pixel temperature biases.
A3	Performing any type of atmospheric compensation significantly improved all classifier results. This included scenarios where the classifier was trained on pixel temperatures not encountered in the validation set.

sation approaches relying on lookup table generation. A dimension reduction-based data augmentation technique was introduced in Study B significantly increasing the number of atmospheric measurements to forward model with MODerate resolution atmospheric TRANsmission (MODTRAN). Leveraging the augmented data reduced reconstruction errors, verifying the AE model required additional samples to generalize. Study B also presented a new loss function derived from the LWIR radiative transfer equation that reduced AE reconstruction error. This loss function measured model error for a range of possible materials providing a better measure of performance compared to mean-squared error. The radiative transfer model was created by sampling the pretrained AE latent space. This approach accelerated radiative transfer calculations by an order of 15 while minimizing at-sensor radiance errors.

Table 2. Summary of objectives and contributions from Study B discussed in Chapter IV

	Objectives
BO1	Utilize the LWIR radiative transfer equation to constrain Autoencoder (AE) latent space construction
BO2	Compare dimension reduction techniques for creating a fast radiative transfer model
	Findings and Contributions
B1	A novel loss function was created that relied on the LWIR radiative transfer equation to minimize AE reconstruction error. Compared to mean-squared error, this physics-based loss function proved more favorable for minimizing at-sensor radiance error.
B2	The Stacked Autoencoder (SAE) latent space was sampled with a small neural network, resulting in a 15 times faster radiative transfer model compared to correlated-k techniques.
B3	Utilizing the low-dimensional latent space, atmospheric state vectors could easily be estimated from a Transmittance, Upwelling, and Downwelling (TUD) vector. This involved optimizing the decoder inputs for the TUD vector output followed by encoder input optimization.
B4	Data augmentation strategies were investigated to increase the number of TUD vectors available for training. The augmentation strategy led to lower reconstruction error and was used throughout the dissertation research to increase the number of TUD vector training samples.

Study C demonstrated an in-scene atmospheric compensation method based on generative models and permutation-invariant networks. Rather than training on a spatially-resolved data cube, subsets of pixels were generated that could have been selected from a data cube. This small modification, combined with a permutation-invariant network and an efficient at-sensor radiance set generation algorithm resulted in the DeepSet Atmospheric Compensation (DAC) algorithm. The permutation-invariant network leveraged max pooling to transform the set of N pixel representations into a one-dimensional set representation. The DAC errors were explored both on synthetic and collected data, with comparable performance to Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR) while reducing inference time.

Table 3. Summary of objectives and contributions from Study C discussed in Chapter V

	Objectives
CO1	Identify neural network architectures useful for in-scene atmospheric compensation
CO2	Determine if neural network-based in-scene atmospheric compensation errors correspond to known physical limitations
	Findings and Contributions
C1	Permutation-invariant neural networks are useful for estimating the underlying TUD vector from a set of pixels. The set pooling operation must be carefully chosen such that predictions are stable as the number of pixels varies.
C2	At-sensor radiance data can be generated from a TUD library, emissivity library, pixel temperature sampling and scene emissivity. The set generation algorithm in [20] can be used for any LWIR experiment requiring many representations of at-sensor radiance.
C3	In-scene atmospheric compensation using permutation-invariant networks and a generative SAE reduces atmospheric compensation time from 67 s to 0.3 s. This reduced inference time supports accelerated LWIR target detection.

Study D investigated the ability of permutation-invariant networks to also estimate the atmospheric state vector (T, H_2O, O_3) from in-scene data. MODTRAN was used to forward model the atmospheric state vector prediction, forming a second TUD estimate. This new TUD estimate provided comparable compensation results to both DAC and FLAASH-IR. Rather than using max pooling as was investigated in Study C, Study D leveraged attention mechanisms to better understand what set features were dominant in atmospheric prediction. Reflective pixels typically received the highest attention scores, a conclusion supported by the LWIR radiative transfer equation. To understand the utility of the results of this research, a review of LWIR hyperspectral remote sensing is presented in Chapter II. This review also discusses methods for exploiting remote sensing data to include a variety of machine learning and deep learning approaches.

Table 4. Summary of objectives and contributions from Study D discussed in Chapter VI

	Objectives
DO1	Determine if atmospheric state vectors (T, H_2O, O_3) and TUD vectors can be recovered from in-scene data
DO2	Evaluate new set pooling operations to better understand what features the permutation-invariant network is sensitive to.
D03	Compare target detection performance between FLAASH-IR and the new atmospheric compensation methods developed in this dissertation
	Findings and Contributions
D1	A combination of weighted atmospheric state loss, at-sensor radiance loss and Kullback-Leibler (KL) divergence were used to create a joint atmospheric state and TUD vector representation. The combination of these loss functions results in lower model error, improving in-scene atmospheric compensation performance.
D2	Attention mechanisms in the set pooling operation are influenced by reflective materials in the scene. This functionality agrees with the LWIR radiative transfer equation as reflective materials are necessary for downwelling radiance prediction.
D3	A multimodal generative model is capable of producing physically-plausible atmospheric state vectors and their corresponding TUD vectors. Sampling the joint low-dimensional space identified latent components encoding the physical parameters such as total column water vapor content, resulting in an explainable latent code useful for deterministic generative modeling.
D4	Faster target detection is possible when using the Multimodal DeepSet Atmospheric Compensation (MDAC) method compared to FLAASH-IR without degrading detection performance on collected data.

1.5 Approach

All analysis in this dissertation is focused on accelerating LWIR atmospheric compensation using in-scene data to support faster target detection for current and future sensors. Chapter II describes necessary contextual information needed to develop novel atmospheric compensation methods. This includes a review of data sources such as atmospheric measurement libraries and material spectral libraries. A review of radiative transfer models and the mechanisms governing emission, absorption and scattering of EM radiation are described. Chapter II also provides necessary background on the neural network architectures

explored in this work, highlighting relevant research from other domains such as machine vision and image classification. A review of other compensation algorithms is provided to examine strengths and weaknesses compared to the approach investigated here. Finally, target detection methods are presented with a focus on techniques supporting near real-time detection.

Chapter III motivates the atmospheric compensation problem by applying well-studied deep and shallow classifiers to collected LWIR data. This research demonstrates that classification methods fail to generalize across temporally-varying cubes without first conducting atmospheric compensation.

Chapter IV investigates generative model techniques to create a faster radiative transfer model, useful for supporting model-based atmospheric compensation methods. An atmospheric generative model is created by forming a low-dimensional data manifold with smooth transitions between atmospheric states. Manifold components are mapped back to known atmospheric features such as atmospheric temperature or total column water vapor content. Through these visualizations and low-dimensional interpolations, we can verify the generative model creates physically-plausible modifications to the atmospheric state.

Chapters V and VI develop atmospheric compensation algorithms that sample the generative model data manifold. In Chapter V, less emphasis is placed on the generative model as it was already well defined by previous results, but a detailed error analysis is conducted on the permutation-invariant sampling network. Chapter VI revisits both the generative model and the sampling network to fuse multiple modalities into an in-scene atmospheric compensation algorithm and in-scene atmospheric state estimator. Finally, Chapter VI also demonstrates the model utility with respect to the target detection problem, showing comparable performance while reducing detection time.

II. Background

2.1 Hyperspectral Remote Sensing

In the past two decades, hyperspectral remote sensing has gained significant attention in areas such as forestry, geology, agriculture and defense because of its ability to discriminate materials using hundreds of narrow band spectral channels [21]. These sensors are carefully calibrated such that laboratory measurements contained in spectral libraries can be directly compared to field measurements allowing for material identification. Each pixel measurement represents a vector across all spectral channels of the sensor. After collecting measurements for all pixels in a scene, a three dimensional data cube is formed, width by height by spectral channel [22]. An example hyperspectral cube is shown in Figure 1 where the spectra from a group of pixels is plotted and an image generated from a single band is also displayed.

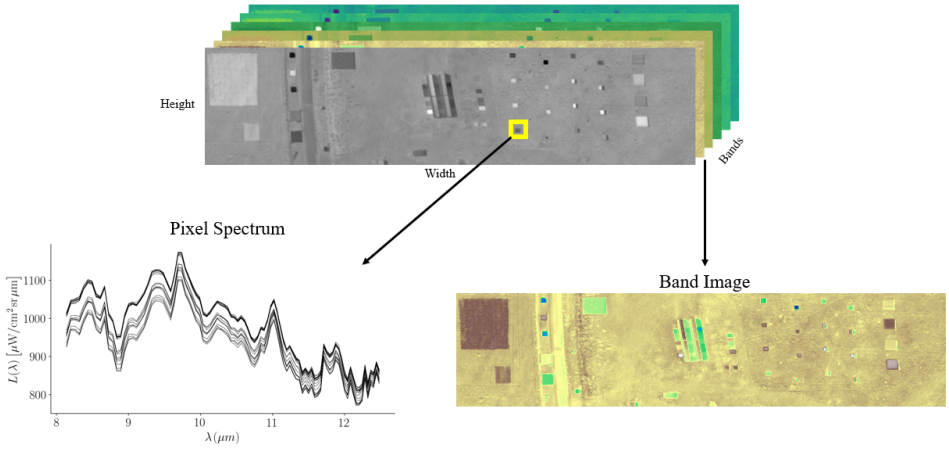


Figure 1. An example hyperspectral data cube is shown (top center) where plotting the spectral data for a single pixel results in a radiance curve (bottom left) and visualizing a single band for all pixels creates an image (bottom right).

To collect data across hundreds of spectral bands, the incident electromagnetic energy must be dispersed across the sensor's focal plane array (FPA) in such a way that both spectral and spatial information is encoded. This is achieved with a FPA consisting of a

spatial and spectral dimension rather than two spatial dimensions for panchromatic sensors. A single row or column of pixels within the scene is projected onto the two-dimensional FPA array. Along the FPA spectral axis, the electromagnetic (EM) radiation from a single pixel is measured at hundreds of contiguous bands. To gather information about all pixels in the scene, the sensor must scan in either a cross-track or along-track direction with respect to the platform's motion [2]. As the sensor scans, spectral and spatial information is combined forming a three-dimensional cube. The values recorded by the FPA are digital counts which must be converted to radiance values through sensor-specific radiometric calibration.

The at-sensor radiance observed by the sensor contains detailed material information across hundreds of bands. Fundamentally, the observed radiance is derived from quantum mechanics based on intrinsic material properties such as the absorption coefficient and the complex index of refraction of the material [21]. While these properties aren't directly measurable, the apparent spectral properties are. These properties consist of surface reflectance, transmittance and emissivity [2]. Incident EM radiation on a surface must either reflect off the material's surface, transmit through the material or be absorbed by the material by conservation of energy. Equivalently,

$$\rho(\lambda) + \alpha(\lambda) + \tau(\lambda) = 1 \tag{2.1}$$

where $\rho(\lambda)$ is apparent reflectance, $\alpha(\lambda)$ is apparent absorbance, $\tau(\lambda)$ is apparent transmittance measured at wavelength λ with each term ranging from 0 to 1. Spectral irradiance, $E(\lambda)$, is defined as the EM power per unit area, per spectral bandwidth received by a surface. The apparent reflectance is then a ratio between the reflected radiance, $L(\lambda)$, and the total irradiance:

$$\rho(\lambda) = \pi \frac{L(\lambda)}{E(\lambda)}. \tag{2.2}$$

By Kirchoff's Law, materials in thermodynamic equilibrium follow $\alpha(\lambda) = \varepsilon(\lambda)$. For opaque surfaces, $\tau(\lambda) \approx 0$, Equation 2.1 can be rewritten to relate reflectance to emissivity:

$$\rho(\lambda) = 1 - \varepsilon(\lambda) \tag{2.3}$$

In the Long-Wave Infrared (LWIR) domain, thermal emission of EM energy must be considered. A blackbody material absorbs all incident radiation making the material an ideal radiating surface since all absorbed energy is also emitted to maintain the same material temperature [21]. Planck's function provides the spectrum of a blackbody material for a given temperature:

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/\lambda kT} - 1} \tag{2.4}$$

where $c = 2.998 \times 10^8 \,\mathrm{m/s}$, $h = 6.626 \times 10^{-34} \,\mathrm{Js}$, $k = 1.381 \times 10^{-23} \,\mathrm{J/K}$ and T is the

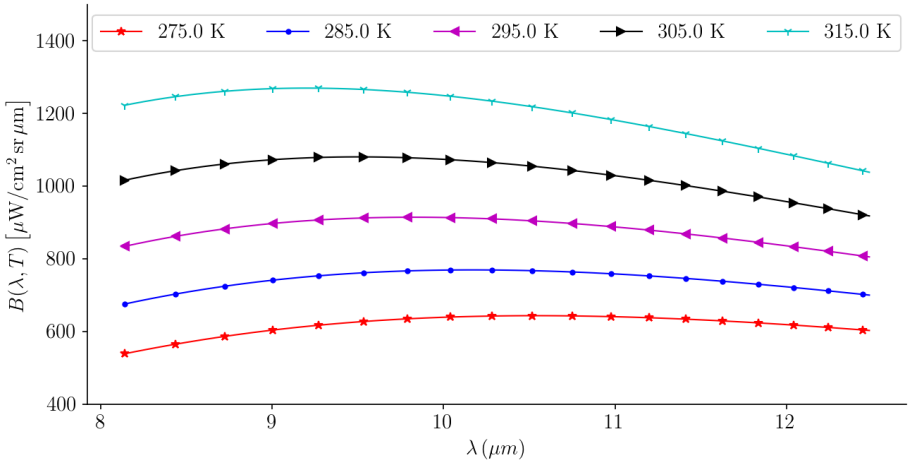


Figure 2. Planck's function for the LWIR domain at varying surface temperatures.

surface temperature. Planck's function in the LWIR domain is plotted in Figure 2 for varying surface temperatures commonly encountered. Using Planck's function, the spectral emissivity, $\varepsilon(\lambda)$, is defined as the ratio between the radiance emitted from a material at

temperature T, $L_T(\lambda)$ to that of a blackbody radiator at the same temperature

$$\varepsilon(\lambda) = \frac{L_T(\lambda)}{B(\lambda, T)}. \tag{2.5}$$

Given a material emissivity and temperature, the total at-sensor radiance for the LWIR domain consists of three distinct components: surface-emitted, surface-reflected and atmospheric upwelling radiance. Figure 3 describes at-sensor radiance, $L(\lambda)$, graphically in terms of these components. This configuration applies to diffuse, lambertian surfaces in thermal equilibrium with no angular dependence and is described by

$$L(\lambda) = \tau(\lambda)\Big[\varepsilon(\lambda)B(\lambda, T) + [1 - \varepsilon(\lambda)]\, L_d(\lambda)\Big] + L_a(\lambda), \tag{2.6}$$

where

$$\begin{aligned}
\lambda &: \text{Wavelength} \\
\tau(\lambda) &: \text{Atmospheric Transmission} \\
\varepsilon(\lambda) &: \text{Material Emissivity} \\
B(\lambda, T) &: \text{Blackbody Function} \\
L_d(\lambda) &: \text{Downwelling Radiance} \\
L_a(\lambda) &: \text{Path Radiance.}
\end{aligned}$$

The downwelling radiance, $L_d(\lambda)$ represents the thermal emission of the atmosphere downward toward the target pixel. If the material is reflective ($\varepsilon(\lambda) \neq 1$), some of this radiance is reflected to the sensor [2]. The path radiance, $L_a(\lambda)$, represents atmospheric thermal emission directly to the sensor. To characterize surface materials with LWIR sensors, the surface emissivity and surface temperature must be estimated from the at-sensor radiance signal. Extracting this data requires an accurate accounting of the atmosphere,

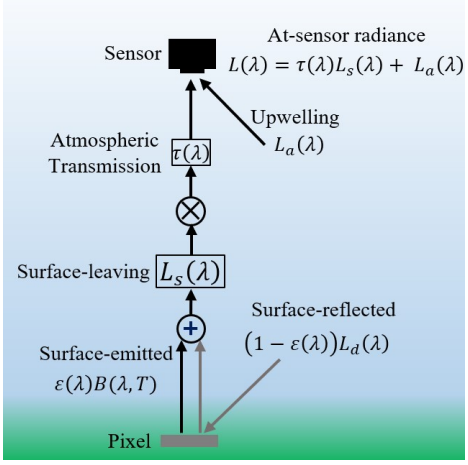


Figure 3. The LWIR at-sensor radiance broken down into the contributing components for a diffuse, lambertian surface in thermal equilibrium. This configuration assumes a nadir sensor viewing geometry. Figure based on Figure 11.31 in [21].

known as atmospheric compensation. This process is followed by Temperature-Emissivity Separation (TES) to estimate emissivity and temperature. Next, atmospheric compensation techniques are discussed followed by a review of TES methods.

2.2 Atmospheric Compensation

Atmospheric compensation techniques are typically based on one of two paradigms: scene-based compensation and model-based compensation [2]. Scene-based compensation attempts to use the wealth of information provided by the imaging spectrometer for a single scene to estimate scene visibility or atmospheric species such as water and ozone [13]. Often, scene-based methods will utilize *a priori* knowledge of materials present to assist atmospheric estimation. Model-based methods use radiative transfer modeling techniques to simulate scattering and absorption of atmospheric constituents in order to retrieve surface reflectance spectra [13]. Typically, model-based methods are computationally expensive and difficult to perform in real-time. Furthermore, estimates of various aerosol concentrations are needed to create accurate models. Measuring these atmospheric constituents using radiosonde data is costly and time-consuming. Profiles derived from climatology models

can be used in place of radiosonde data, but they typically miss temporal and spatial variations within a particular scene [23].

Significant effort has been placed on atmospherically compensating hyperspectral data using both paradigms. In some cases, a mix of scene-based and model based approaches are used to both efficiently and accurately estimate transmission and scattering. The spectral range under consideration also impacts the compensation algorithm formulation. The LWIR domain requires estimation of surface temperatures to arrive at material emissivity while the visible and near-infrared (VNIR)/shortwave infrared (SWIR) is dependent on scattering mechanisms to calculate surface reflectance values. In general, more work has been done in the VNIR/SWIR than the LWIR for atmospheric compensation [13, 21].

Assuming homogeneous atmospheric conditions between the sensor and all pixels in the scene, the terms strictly defining the atmosphere in Equation 2.6 are $\tau(\lambda)$, $L_a(\lambda)$ and $L_d(\lambda)$. These terms are often referred to as the Transmittance, Upwelling, and Downwelling (TUD) vector for the atmosphere. If the TUD vectors are estimated accurately, the surface-leaving radiance, $L_s(\lambda)$ can be calculated by

$$L_s(\lambda) = \frac{L(\lambda) - L_a(\lambda)}{\tau(\lambda)} = \varepsilon(\lambda)B(\lambda, T) + [1 - \varepsilon(\lambda)]\, L_d(\lambda). \tag{2.7}$$

An example of each term in the simplified LWIR radiative transfer equation is shown in Figure 4 for a 300 K foamboard material. In this case, the atmospheric transmission is very high with a corresponding low path radiance. Both the surface-leaving radiance and at-sensor radiance exhibit spectral features similar to the measured material emissivity because of the clear atmospheric conditions.

2.2.1 In-Scene Methods.

In-scene compensation methods rely only on the data available in the scene and are typically more computationally efficient compared to model-based methods. One of the

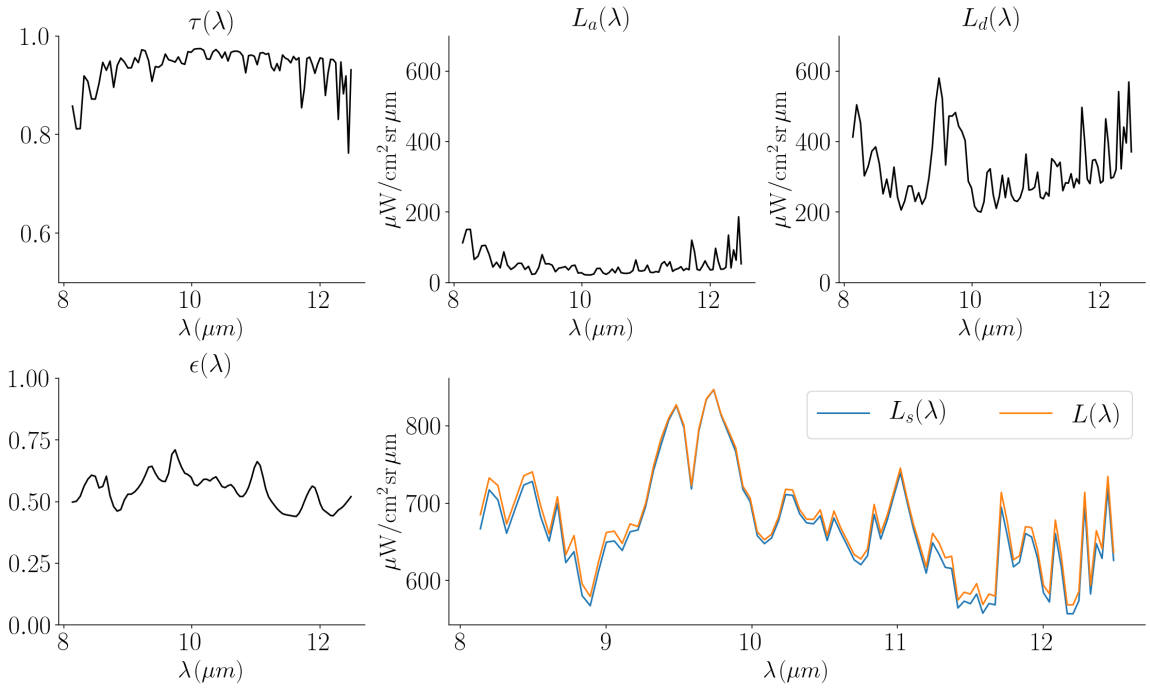


Figure 4. The at-sensor radiance $L(\lambda)$ is shown based on the TUD vector shown for a 300 K foamboard material.

most common LWIR compensation techniques is the In-Scene Atmospheric Compensation (ISAC) method. The ISAC method begins by identifying blackbody pixels ($\varepsilon(\lambda) \approx 1$) within a scene. Based on Equation 2.6 for blackbody pixels, the surface-reflected component is zero and the surface-emitted radiance is $B(\lambda, T)$ resulting in the blackbody pixel at sensor radiance:

$$L_{BB}(\lambda) = \tau(\lambda)B(\lambda, T) + L_a(\lambda). \tag{2.8}$$

To solve for $\tau(\lambda)$ and $L_a(\lambda)$ each blackbody pixel temperature must be estimated. Using a band with high transmission and low upwelling radiance ($\lambda_0 \approx 10\,\mu$m) results in $L_{BB} \approx B(\lambda_0, T)$. Inverting Planck's function recovers the pixel temperature estimate as described by the spectral apparent temperature or brightness temperature [2]:

$$\hat{T}_{BB}(\lambda_0) = B^{-1}\big(\lambda_0, L_{BB}(\lambda_0)\big) = \frac{hc}{\lambda_0 k \ln\left(\frac{2hc^2}{\lambda_0^5 L_{BB}(\lambda_0)} + 1\right)}. \tag{2.9}$$

The spectral apparent temperature is a spectral-varying quantity, however, for true blackbody pixels, this quantity should be constant across all wavelengths. Figure 5 shows an example of this process where λ_o is overlaid on $\tau(\lambda)$ and $L_a(\lambda)$ highlighting how close these values are to 1.0 and 0.0, respectively. Additionally, Figure 5 shows the brightness temperature for a high emissivity material, with a nearly constant temperature across all wavelengths. If λ_0 has less than unity transmission (as shown in Figure 5), $\hat{T}_{BB}(\lambda_0)$ will underestimate the true temperature. Radiative transfer models such as MODerate resolution atmospheric TRANsmission (MODTRAN) can be used to correct this scaling if atmospheric state information such as the vertical distribution of water vapor and ozone are known.

After estimating the blackbody pixel temperatures, a linear regression is performed across each band to determine $\hat{\tau}(\lambda)$ and $\hat{L}_a(\lambda)$. Identification of blackbody pixels within the scene can be challenging, however, materials such as vegetation and water bodies have low reflectance and provide reasonable estimates of these atmospheric terms.

This research only considers nadir-viewing geometries, but work in off-nadir geometries has shown significantly different performance results using standard atmospheric compensation algorithms. This is expected as path lengths vary for materials in the scene, resulting in non-homogeneous atmospheric terms. In-scene methods attempt to slice the

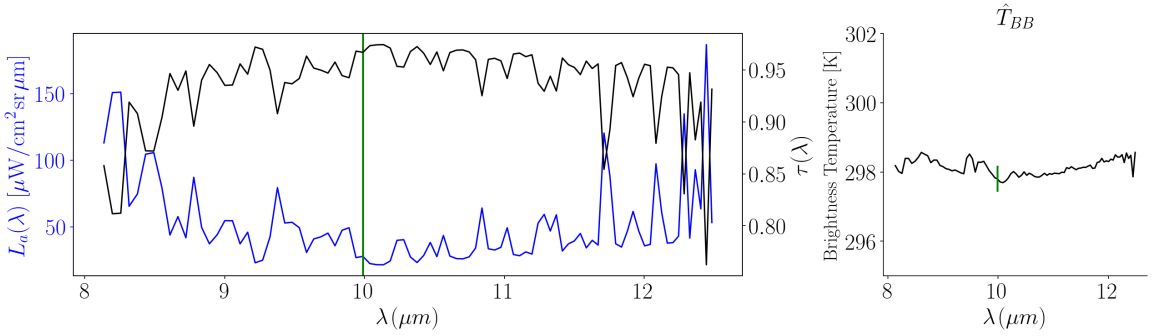


Figure 5. Left: The clear band, $\lambda_o \approx 10\,\mu$m is shown compared to transmittance and upwelling radiance. Right: The brightness temperature, $\hat{T}_{BB}$ for a high emissivity material is shown with λ_o overlaid result in a pixel temperature estimate of roughly 298 K.

image to reduce this problem, however, the number of horizontal image slices is scene dependent. An oblique in-scene atmospheric compensation algorithm is discussed in [18] that is similar to ISAC. However, this work also introduces radiance detrending which is an unsupervised classification of materials in the scene to identify the most probable atmospheric state. Radiance detrending allows for spectral signals to be tied to distances in the scene and therefore infer path radiance and transmission.

2.2.2 Model-Based Compensation.

Model-based atmospheric compensation methods convert at-sensor radiance values to surface reflectance values using radiative transfer models. MODTRAN is one of the most popular radiative transfer models and was developed by Spectral Sciences Inc. and the Air Force Research Laboratory for analyzing optical measurements through the atmosphere [24]. Line-by-line calculations can be performed over the ultraviolet to long wavelength infrared spectrum using predefined climatology data. Effects such as absorption/emission and scattering, surface reflections and solar illumination are all considered resulting in line-by-line resolutions as small as 0.2 cm^{-1} [24].

Applying MODTRAN, or an equivalent radiative transfer model, requires some prior knowledge of the scene, such as atmospheric constituents, but can provide highly accurate atmospheric corrections. Even gas plumes and thick clouds can be accurately modeled if enough information is known about the atmosphere of interest. Many of the assumptions underlying model-based compensation are reasonable in operational scenarios such as knowing the sensor location and time of collection. This type of information helps reduce the under determined estimation problem to a more tractable form.

Many atmospheric compensation algorithms have been developed for the VNIR/SWIR domain. One such atmospheric compensation algorithm is the Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH) algorithm, developed by the Air Force

Research Laboratory and Spectral Sciences, Inc. in the 1990s [25]. FLAASH is similar to other methods such as atmospheric removal program (ATREM) with one important difference: FLAASH accounts for adjacency effects between pixels, cloud cover and various scattering phenomenon. The FLAASH algorithm addresses cloud cover and hazy conditions by averaging the reflectivity of surrounding pixels, ρ_s, using Equation 2.10 which returns the at-sensor reflectance:

$$\rho_s = T_g \left(\rho_{atm} + \frac{t_d t_u \rho}{1 - \rho_{adj} S} + \frac{(\rho_{adj} - \rho) t_d t_u}{1 - \rho_{adj} S} \right) \tag{2.10}$$

where

T_g : Total atmospheric gas transmittance

ρ_{atm} : Atmospheric reflectance into the sensor

t_d : Scattering attenuation from TOA

t_u : Scattering attenuation from surface to sensor

S : Spherical albedo of the atmosphere

ρ_{adj} : Average reflectivity of surrounding pixels

Including adjacency effects in the overall reflectance calculation has been shown to improve atmospheric modeling performance [25]. The most significant performance increases have been observed in hazy conditions where scattering into the sensor field of view is most prevalent. Similar to other model-based atmospheric compensation algorithms, the FLAASH algorithm relies on MODTRAN for atmospheric modeling to determine atmospheric terms in Equation 2.10 [25].

The FLAASH algorithm was also extended to the infrared spectral range, appropriately named FLAASH-IR [26]. FLAASH-IR was validated with a Telops Hyper-Cam interfer-

ometric hyperspectral imager in ground to ground applications, where estimates for atmospheric transmission, path radiance and downwelling were calculated again using MODTRAN. Similar to other infrared compensation methods, FLAASH-IR assesses spectral smoothness to identify probable surface temperatures for a given material emissivity [26]. The FLAASH-IR was also tested on reflective objects such as bare-metal with acceptable results. Overcast conditions can contribute to lower apparent reflectance values, but overall the spectral shape is still preserved [26].

Regardless of the atmospheric compensation method employed, real-time material identification requires accurate and efficient compensation techniques. One of the primary goals of this research is to investigate how machine learning methods can be leveraged towards this goal. After validating the atmospheric compensation approach, material classification or detection can be performed for a wide range of atmospheric conditions with limited labeled training data. The next section reviews the fields of machine learning and deep learning to provide an algorithmic basis for hyperspectral classification techniques. This is followed by discussion on how these techniques can be applied to hyperspectral data for both classification and target detection scenarios focusing on necessary preprocessing steps such as atmospheric compensation or data standardization.

2.3 Temperature-Emissivity Separation

Assuming the atmospheric compensation approach provided an estimate of $\tau(\lambda)$ and $L_a(\lambda)$, temperature-emissivity separation recovers the pixel emissivity and temperature. For blackbody pixels, the brightness temperature will equal the surface temperature. For all other pixels, temperature-emissivity separation is an ill-posed problem because for K spectral bands there are $2K+1$ unknown variables contained in $L_d(\lambda), \varepsilon(\lambda), T$. To make this problem more tractable, $L_d(\lambda)$ can be assumed negligible in some situations and the pixel's maximum apparent spectral brightness temperature, T_{max}, can be used for pixel

temperature, resulting in the emissivity estimate [2]:

$$\hat{\varepsilon}(\lambda) = \frac{\hat{L}_s(\lambda)}{B(\lambda, T_{\max})}. \tag{2.11}$$

The temperature assumption in Equation 2.11 is straightforward to implement, but may provide insufficient results. Applying this technique to a foamboard pixel using the $\tau(\lambda)$ and $L_a(\lambda)$ estimates shown in Figure 4 results in the emissivity estimate shown in Figure 6. The overall spectral shape is similar between the reconstructed emissivity and measured values, but some atmospheric features are clearly present. Additionally, the overall emissivity estimate is higher than the measured value and $T_{\max} = 311\,\mathrm{K}$ using the method outlined in Equation 2.11.

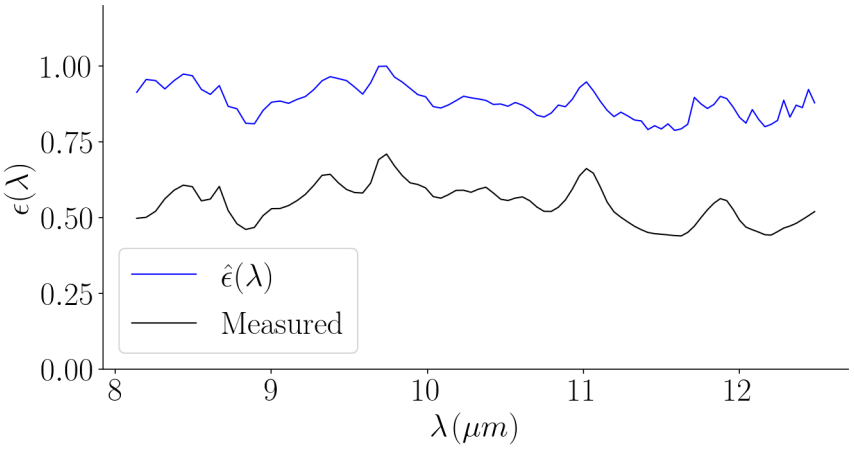


Figure 6. Applying Equation 2.11 to a foamboard pixel collected by a LWIR spectrometer results in an emissivity estimate with similar spectral features, but large overall error. This result may be sufficient for some applications, such as target detection, depending on the spectra of other materials within the scene.

To further improve on the results in Figure 6, an estimate of $L_d(\lambda)$ is necessary to improve the emissivity estimate. Lookup tables generated by MODTRAN can be used to determine the most likely $L_d(\lambda)$ for a given $\tau(\lambda)$ and $L_a(\lambda)$ or autoencoder networks can be used as presented in the results section of this research. With a complete TUD estimate,

emissivity can be expressed as [2]:

$$\hat{\varepsilon}(\lambda) = \frac{L(\lambda) - \hat{L}_a(\lambda) - \hat{\tau}(\lambda)\hat{L}_d(\lambda)}{\hat{\tau}(\lambda)\left[B(\lambda, T) - \hat{L}_d(\lambda)\right]}. \tag{2.12}$$

Solving for emissivity is still an ill-posed problem for a sensor with K bands, as there are $K+1$ unknowns $(\hat{\varepsilon}(\lambda), T)$ for K measurements. Considering the emissivity estimate in Figure 6, the recovered emissivity should be a smooth function of wavelength to remove residual atmospheric features [27]. To enforce this heuristic in the recovered signal, a range of temperatures are tested with a smoothed emissivity profile. Specifically, a smoothed emissivity, $\tilde{\varepsilon}(\lambda)$ and temperature estimate, $\hat{T}$ are used with the TUD estimate to create $\hat{L}(\lambda)$. The minimum error between the observed and predicted radiance recovers the emissivity and temperature [2]:

$$\hat{T} = \min_{T} \left[\sum_{k=1}^{K} \left(L(\lambda_k) - \hat{L}(\lambda_k) \right)^2 \right] \tag{2.13}$$

Applying this technique to the foamboard pixel using a 3-point averaging filter leads to

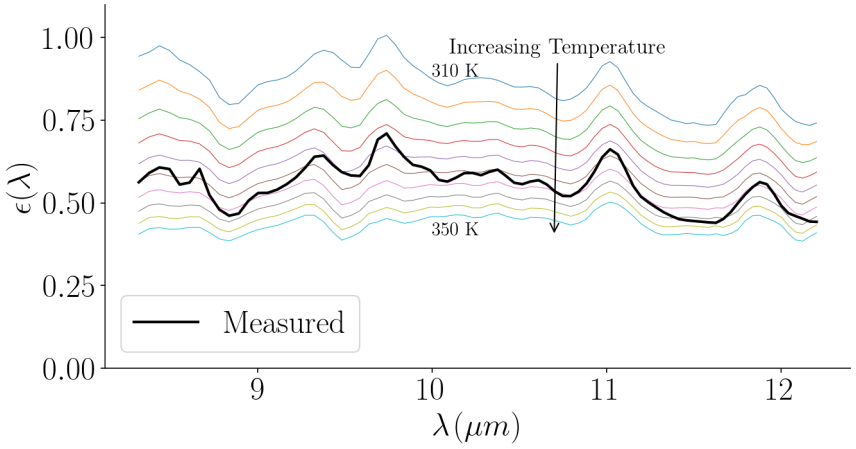


Figure 7. Enforcing a smoothness criteria on the recovered emissivity using a 3-point local averaging filter results in more accurate emissivity estimates compared to the result shown in Figure 6.

the emissivity estimates shown in Figure 7. Most atmospheric features have been removed from the signal, and will likely lead to better target detection or classification across a wider range of scenes.

In many target detection scenarios, accurate pixel emissivity recovery is more important than exact temperature estimation. Based on this requirement, other TES approaches have been implemented with less dependence on the recovered pixel temperature, referred to as alpha residual TES. Using Wien's approximation of the planckian function for a particular wavelength λ_j:

$$L_{BB}^{W}(\lambda_j) = \frac{C_1}{\lambda_j^5 \pi \left[\exp\left(\frac{C_2}{\lambda_j T}\right) \right]} \tag{2.14}$$

where $C_1 = 3.74151 \times 10^{-16}\,(\mathrm{Wm}^2)$ and $C_2 = 0.0143879\,(\mathrm{m \cdot K})$. Taking the natural logarithm of the surface emitted radiance term in the LWIR radiative transfer equation using Wien's approximation for the planckian results in [28]:

$$\ln\left(\varepsilon(\lambda_j) L_{BB}^{W}(\lambda_j)\right) = \ln \varepsilon(\lambda_j) + \ln C_1 - 5 \ln \lambda_j - \ln \pi - \frac{C_2}{\lambda_j T} \tag{2.15}$$

Following the derivation in [28], each side of Equation 2.15 is multiplied by λ_j and the mean over N spectral channels is taken resulting in N equations without a temperature dependence:

$$\alpha(\varepsilon(\lambda_j)) = \lambda_j \ln \varepsilon(\lambda_j) - \frac{1}{N} \sum_{j=1}^{N} \lambda_j \ln \varepsilon(\lambda_j) \tag{2.16}$$

The temperature independent calculation using the surface-leaving radiance, with an assumed downwelling radiance of zero is [28]:

$$\begin{aligned} \alpha(\varepsilon(\lambda_j)) =& \lambda_j \ln L_s(\lambda_j) - \frac{1}{N} \sum_{j=1}^{N} \lambda_j \ln L_s(\lambda_j) \\ & - \lambda_j \ln C_1 + \frac{\ln C_1}{N} \sum_{j=1}^{N} \lambda_j \\ & + \lambda_j 5 \ln \lambda_j - \frac{5}{N} \sum_{j=1}^{N} \lambda_j \ln \lambda_j \\ & - \lambda_j \ln \pi - \frac{\ln \pi}{N} \sum_{j=1}^{N} \lambda_j . \end{aligned} \tag{2.17}$$

With laboratory measured spectrum, Equation 2.16 can be directly compared to Equation 2.17 without recovering pixel temperatures. Figure 8 demonstrates the result of this process for a foamboard pixel showing comparable signal features between the transformed emissivity and recovered alpha residual emissivity. Atmospheric features are clearly present in the recovered alpha residual emissivity estimate and may be problematic for materials with less pronounced emissivity features.

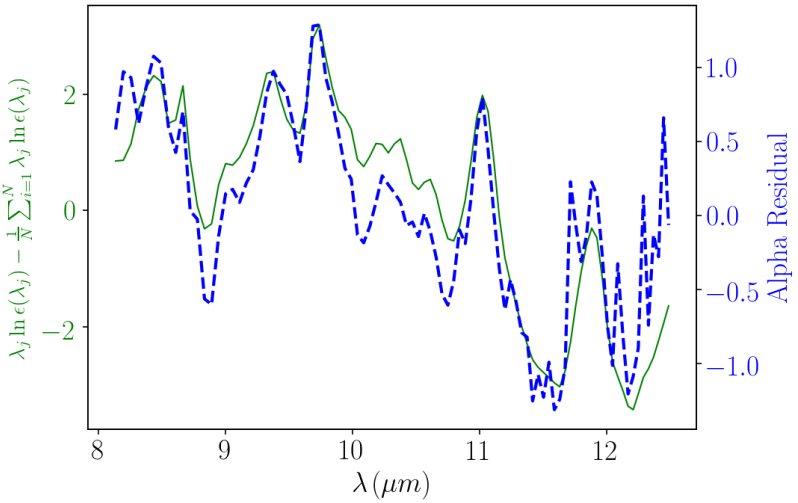


Figure 8. Applying Equation 2.17 to surface-leaving foamboard data results in the recovered alpha residual emissivity estimate shown. The recovered estimate contains atmospheric features but has comparable shape to the transformed emissivity.

A limitation in the derivation of Equation 2.17 is the assumption that downwelling radiance is zero resulting in $L_s(\lambda) = \varepsilon(\lambda)L_{BB}(\lambda)$. As shown in Figure 8, this results in residual atmospheric features in the recovered signal. This limitation was addressed in [29] where an initial temperature estimate is needed to recover the modified alpha residual estimate. Specifically, an α operator was defined as:

$$T_\alpha\left[x(k_j)\right] \triangleq \frac{48\ln x(k_j)}{k_j} - \frac{48}{N}\sum_{j=1}^{N}\frac{\ln x(k_j)}{k_j}, \tag{2.18}$$

where $T_\alpha[\varepsilon(k_j)] = \alpha(\varepsilon(\lambda_j))$ from Equation 2.16 with a change of variable $k_j = 48/\lambda_j$. Including the downwelling radiance, the modified surface-leaving radiance equation is :

$$L_s(\lambda_j) - L_d(\lambda_j) = \varepsilon(\lambda_j)[L_{BB}(\lambda_j, T) - L_d(\lambda_j)]. \tag{2.19}$$

After applying the α operator to Equation 2.19, the alpha residual calculation can be expressed as [29]:

$$\begin{aligned} \hat{\alpha}(\varepsilon(k_j,t)) =& T_\alpha[L_s(k_j) - L_d(k_j)] + T_\alpha\left[1 - e^{-k_j t}\right] \\ & - T_\alpha\left[1 - \frac{L_d(k_j)(e^{k_j t} - 1)}{C(k_j)}\right] - T_\alpha[C(k_j)], \end{aligned} \tag{2.20}$$

where $C(k_j) = C_1/\lambda_j^5$. The temperature dependence can be removed from Equation 2.20 by assuming $t = 300/T_o$ and using a fast TES method to estimate an initial temperature T_o. Figure 9 shows the result of applying Equation 2.19 to foamboard surface-leaving radiance data. By accounting for downwelling radiance, atmospheric features are reduced compared to results based on Equation 2.17. This approach is of interest to this research since downwelling radiance vectors will be estimated. TES methods that utilize the additional information from downwelling radiance will further highlight the utility of compensation methods derived in this research. To support a wide range of atmospheric compensation and target detection research, a TUD vector library is discussed next and will serve as the primary database for this research.

2.4 TUD Vector Data

Based on the previous discussion, it's clear that accurate TUD vector estimates are necessary for extracting pixel emissivity and temperature. Atmospheric compensation methods such as Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR), often rely on lookup tables of precomputed atmospheric data to accurately compensate a

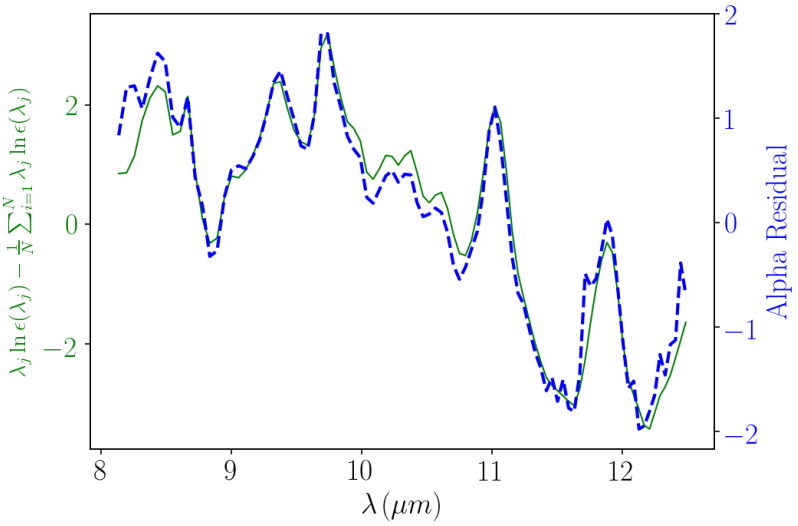


Figure 9. Applying Equation 2.19 to foamboard surface-leaving radiance data results in the recovered alpha residual estimate shown. Atmospheric features are less significant in this estimate compared to Figure8, showing the importance of correctly estimating and including downwelling radiance in the TES method.

data cube. This section discusses one particular database used throughout this research, the Thermodynamic Initial Guess Retrieval (TIGR) atmospheric state data.

The TIGR database consists of 2311 atmospheric measurements based on 80,000 world-wide radiosonde reports [30] [31]. The locations of the 2311 measurements are shown in Figure 10. The downselecting process employed to arrive at 2311 samples placed an emphasis on selecting both rare and frequent atmospheric states with equivalent frequency. The downselected atmospheric conditions represent a wide range of atmospheric conditions, useful for remote sensing models. Radiosonde collection dates are shown in Figure 11, highlighting that measurements cover a range of seasons and years. Each radiosonde consists of 43 discrete pressure level measurements ranging from the Earth's surface (1013 hPa) to $>$ 30 km ($<$ 1hPa). Additionally, the profiles are grouped by airmass category such as polar, tropical and mid-latitude to further inform or constrain modeling predictions. Temperature cumulative water vapor content and cumulative ozone vapor content are plotted against atmospheric pressure level in Figure 12, showing how these measurements change with altitude. Temperature and water vapor content have the largest

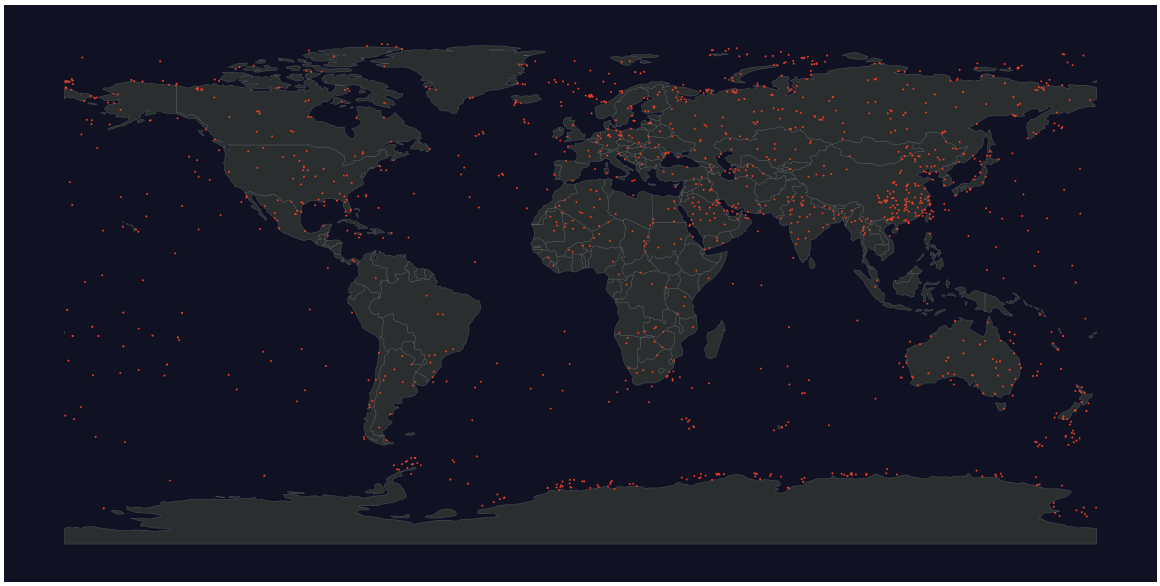


Figure 10. Radiosonde collection locations for the 2311 samples in the TIGR database. [30] [31]

impact on the observed at-sensor radiance for values close to the Earth's surface, while ozone only plays a significant role at higher altitudes.

The radiosonde data available within the TIGR database must be converted to TUD vectors for use in atmospheric compensation models. Radiative transfer models such as MODTRAN or Line-by-Line Radiative Transfer Model (LBLRTM) can be used with several scene assumptions to create a corresponding TUD database [32, 33]. In MODTRAN 6.0, JavaScript Object Notation (json) files can be used to specify all atmospheric parameters necessary for generating TUD vectors. This update also makes it very straightforward to run MODTRAN from other languages such as Python which is used in this research. The json used to run MODTRAN is provided with the source code, but constant parameters are outlined in Table 5. The 1976 U.S. Standard Atmosphere (Model 6) is used for all trace gases [34]. Column water vapor, ozone content and temperature are varied based on the TIGR measurements. Since all TIGR measurements are based on the same pressure grid, no modifications are made to the MODTRAN json pressure argument. MODTRAN requires a pressure level altitude which was not provided with the TIGR data but can be

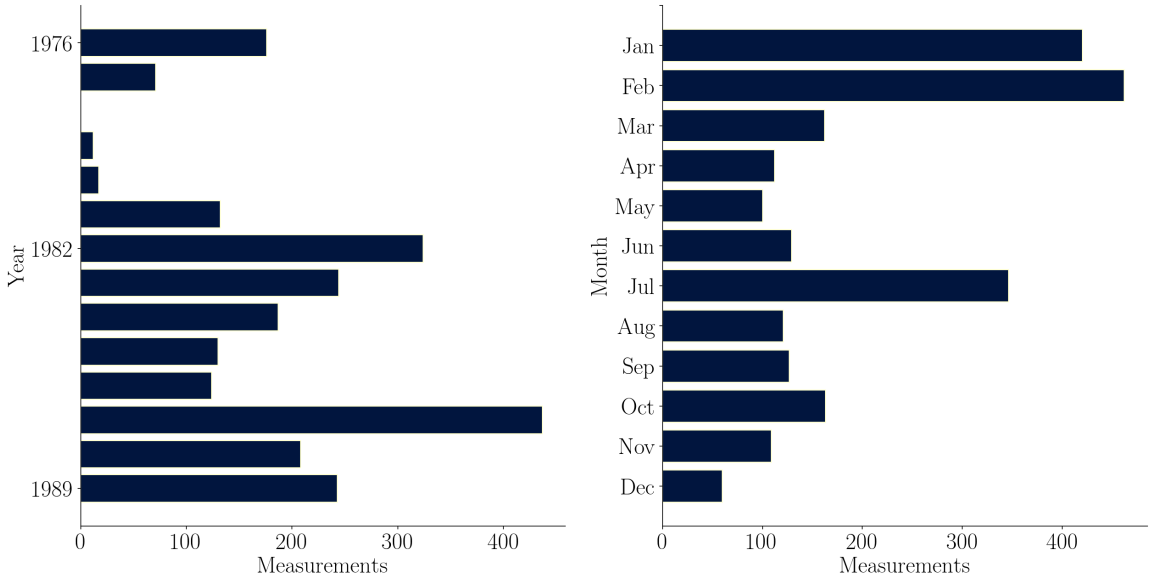


Figure 11. Time of radiosonde collection for the 2311 samples in the TIGR database.

calculated using the hydrostatic equation. Each layer thickness can be calculated by:

$$Z_2 - Z_1 = \frac{R_d}{g} \int_{p_2}^{p_1} T(p) \frac{dp}{p} \tag{2.21}$$

where $Z_2 - Z_1$ is the thickness between pressure levels p_1 and p_2, $R_d = 287\ \mathrm{J/(K\ kg)}$ and $g = 9.81 \mathrm{m/s^2}$.

Using MODTRAN with the constants specified in Table 5 and the varying atmospheric temperature, water vapor content ozone content and pressure level altitude, Figure 13 shows the TIGR-derived TUD vectors. These samples were originally created at 0.5 wavenumber spectral resolution and then downsampled to the Spatially Enhanced Broadband Array Spectrograph System (SEBASS) instrument line shape (ILS). Careful characterization of the sensor ILS is necessary before performing this downsampling. Characterization errors lead to misaligned band centers and band widths resulting in larger overall error when applying these vectors to collected data.

The $\tau(\lambda)$ and $L_a(\lambda)$ vectors are highly correlated based on the LWIR radiative transfer model formulation. Atmospheric states that can absorb more radiation (low $\tau(\lambda)$) can

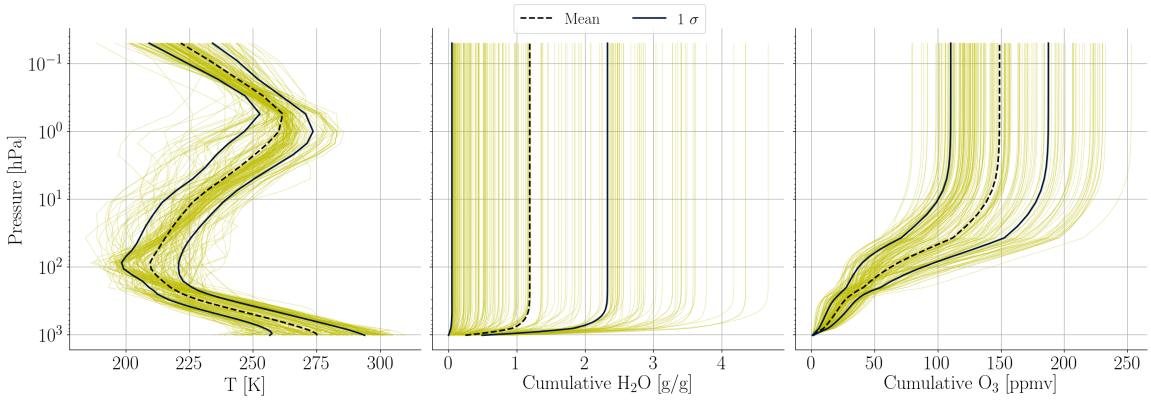


Figure 12. Radiosonde measurements making up the TIGR data. Atmospheric temperature and water vapor content have the largest impact on observed radiance at lower altitudes while ozone plays a significant role at higher altitudes.

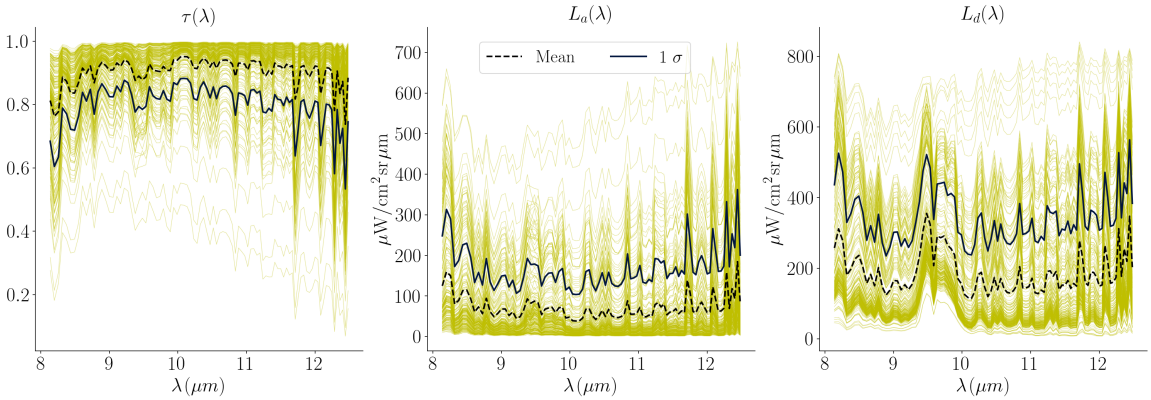


Figure 13. MODTRAN generated TUD vectors based on the TIGR radiosonde data. These vectors have been downsampled to match the SEBASS ILS.

also emit more radiation (high $L_a(\lambda)$). Conversely, atmospheric states with a low density of absorbing constituents (high $\tau(\lambda)$) also cannot emit as much radiation (low $L_a(\lambda)$). To visualize these relationships, the normalized integrated area under $\tau(\lambda)$ and $L_a(\lambda)$ are plotted in Figure 14 with atmospheric surface temperature, cumulative water vapor content and cumulative ozone content overlaid. The plots in this figure illustrate that atmospheric states consisting of warmer, humid conditions trend toward lower transmittance and higher upwelling radiance. Additionally, atmospheric states with higher ozone concentrations also have lower transmittance and higher upwelling radiance because of the additional molecules available to absorb and emit radiation.

Table 5. Constant MODTRAN parameters specified in json file. The json argument is shown in parenthesis under the parameter description.

Ground Altitude (GNDALT)	0.0	Sensor/Sun Azimuth (PARM1)	0.0	Solar Zenith (PARM2)	45.0
Target Altitude (H2ALT)	0.0	FWHM (FWHM)	1.0	Visibility (VIS)	50.0
Wavenumber Start (V1)	7.14×10^2	Haze Model (IHAZE)	AER_RURAL	H_2O Scaling (H2OSTR)	1.0
Wavenumber End (V2)	1.3×10^3	O_3 Scaling (O3STR)	1.0	CO_2 Scaling (CO2MX)	360.0
Sampling (DV)	0.5	Clouds (ICLD)	CLOUD_NONE	Day of Year (IDAY)	265
Scattering Function (IPH)	2	Target Temp (TPTEMP)	300	Background Temp (AATEMP)	295
Target Reflectance (SURFP:CSALB)	LAMB_CONST_50_PCT		CO_2, N_2O, CO, CH_4, O2, NO	6	
Background Reflectance (SURFA:CSALB)	LAMB_CONST_50_PCT		SO_2, NO_2, NH_3, HNO_3	6	

2.5 Deep Learning

Machine learning is now a part of daily life powering speech recognition systems, social media content filtering, facial recognition software and autopilot for autonomous vehicles. Only until recently were many of these breakthroughs realized by the data representations created through deep learning systems. Conventional machine learning systems utilize expert knowledge to create useful data representations while deep learning takes advantage of many nonlinear transformations to generate propitious data representations. The deep representations are learned through the training process, with deeper transformations specific to the training task and shallow transformations (closer to the input) capturing generic data transformations such as edge detection for image analysis [35]. Deep learning systems now exceed human performance on many tasks as shown by the recent success of Google's AlphaGo system beating the world Go champion or DeepMind's Starcraft Artificial Intelligence (AI) beating human professional players [36, 37]. In both cases, the deep representations created new techniques and strategies not seen by professional players.

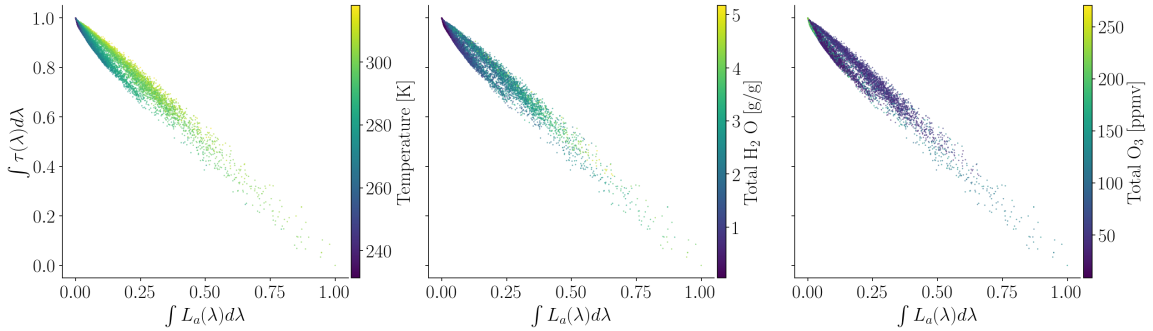


Figure 14. The integrated area under $\tau(\lambda)$ plotted against the integrated area under $L_a(\lambda)$ shows a linear relationship between these two terms. As the atmosphere becomes more transparent ($\int \tau(\lambda) d\lambda$ increases), upwelling radiance decreases. This is because there are less particles to absorb radiation so there are also less particles to emit radiation.

The breakthroughs in deep learning can be attributed to several important advancements: initialization, optimization and training. For years it was observed that deep networks were infeasible because poor initialization led to model convergence at poor local minimums. By performing a greedy layer-wise pretraining, the initialization problem could be avoided resulting in significantly better model results [38]. Furthermore, it was shown that layer-wise pretraining can be avoided if the weight initialization scheme is dependent on the size of the network. One such heuristic is called normalized standardization:

$$W_{i,j} \sim \mathrm{U}\left(-\sqrt{\frac{6}{m+n}}, \sqrt{\frac{6}{m+n}}\right) \tag{2.22}$$

where $W_{i,j}$ is the weight between node i and j, m is the number of inputs to the layer, n is the number of outputs and U denotes a uniform distribution [39]. This initialization scheme was shown to stabilize the magnitude of the gradients for deep networks allowing information to flow to lower layers, leading to better weight convergence. This weight initialization is only applicable to networks implementing activation functions symmetric about 0, such as the sigmoid and hyperbolic tangent. This is because the mean output of each layer is 0 and the standard deviation is 1. Functions such as Rectified Linear Unit (ReLU) require a different weight initialization since a mean output of 0 leads to vanishing

gradients with deeper layers. He et. al proposed a weight initialization for networks using rectifier activation functions such that the randomly chosen weights were multiplied by $\sqrt{2}/\sqrt{n}$ where n is the number of incoming weights [40].

The most significant optimization advancement accelerating the field of deep learning forward was the shift in activation functions from the sigmoid to ReLU or half-wave rectifier. It was observed that sigmoid activation functions tend to saturate as more layers were appended to the network [41]. This reduced the flow of information across all nodes and limited the model's ability to generalize. Additionally, using ReLUs reduced training time for deep networks, an important benefit for quickly iterating across many models. Other optimization improvements include improved methods for weight updates such as the Adam optimizer or RMSprop algorithm [42]. Both optimizers are standard approaches used in many applications with their own set of hyperparameters to be tuned for the task at hand.

The extended training times required to converge on acceptable solutions coupled with the need for large, labeled datasets had prohibited use of neural networks for many applications. With the advent of the internet, labeled data is freely available. This data ranges from labeled pictures and video to text information from Wikipedia [43]. Training times were reduced significantly when NVIDIA launched the CUDA programming interface for its Graphical Processing Units (GPUs). This allowed for parallel implementations of the many matrix multiplications needed to train a model [44]. Furthermore, the field of deep learning experienced an influx of researchers when packages such as Tensorflow and Keras allowed users without CUDA programming experience to quickly iterate through many models [43]. More recently, renewed attention has been placed on improved training algorithms as hardware limits are reached. Hashing functions have been investigated to accelerate training without the need for costly weight updates to every node in multi-million

parameter networks [45]. These algorithms are still under investigation, but may result in significantly more efficient model training.

One of the most well-known research breakthroughs applying these advancements in optimization, initialization and training was the best performing model at the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) competition [44]. The ILSVRC competition requires models to correctly label images from 1000 different categories, based on 1.2 million training images, 50,000 validation images and 150,000 test images. The best performing model utilized ReLU activation functions to reduce convergence time with 5 convolutional layers followed by 3 fully connected layers. Multiple GPUs were used to reduce training time and extensive use of regularization methods such as Dropout helped prevent overfitting. This approach reduced the Top-5 error rates by 10% over the next best performing model, proving that deep networks could be trained for image prediction without hand-engineered filters. This is especially important in the analysis of hyperspectral data where known physical phenomenon can motivate hand-engineered filters. If adequately sized datasets are available, this hand-engineering can be avoided allowing the model to learn the most salient features through gradient descent.

The ILSVRC competition provided images with labeled classes for the network to predict. This is an example of supervised learning in which a known outcome is used to direct the model's learning process [46]. Unsupervised learning has no known outcome, so the model can only determine informative ways to cluster or group the data. Since there is no known outcome, it is difficult to measure success and often the quality of the approach is dependent on the end user's judgement of the transformed data. Even with this limitation, unsupervised methods, such as Autoencoders (AEs), can be useful for finding informative low-dimensional embeddings to explain high-dimensional data [47]. Next, a brief review of AEs is presented for unsupervised learning applications related to hyperspectral imagery.

2.5.1 Autoencoders.

While many recent deep learning successes utilize labeled training samples, AEs are a unique type of network architecture for unsupervised learning. An AE consists of two separate networks: an encoder f and decoder g. The encoder transforms or encodes the input signal $\mathbf{x}$ to an embedded representation $\boldsymbol{z}$, typically with fewer dimensions than the input as shown in Figure 15. The decoder network reconstructs or decodes the embedded representation and error is measured using the loss function $\mathscr{L}$:

$$\mathscr{L}(\mathbf{x}, g(f(\mathbf{x}))) \tag{2.23}$$

When the loss function is mean-squared error and the decoder is linear, an AE can replicate results from Principal Component Analysis (PCA). The constraints placed on the embedded space allow AEs to learn a data manifold where most of the probability mass is centered [48]. These constraints can be introduced by forcing the embedded space $\mathbf{z}$ to have fewer dimensions than $\mathbf{x}$ (undercomplete) or by requiring $\mathbf{z}$ to be sparse. After applying one or more constraints and minimizing the loss, similar input signals will cluster in the embedded space as has been shown using the handwritten digits data MNIST [47]. The Contractive Autoencoder (CAE) provides an explicit regularization to encode small perturbations of $\mathbf{x}$ to similar locations within the embedded space, $f(\mathbf{x})$ [49]:

$$\mathscr{L}_{\text{CAE}}(\mathbf{x}, \hat{\mathbf{x}}) = \mathscr{L}\left(\mathbf{x}, g(f(\mathbf{x}))\right) + \gamma \sum_{ij} \left(\frac{\partial h_j(\mathbf{x})}{\partial x_i} \right)^2 \tag{2.24}$$

where h_j is node j in the hidden layer and γ is a regularization constant. Equation 2.24 penalizes sensitivity of the hidden representation, $h_j(\mathbf{x})$, to small changes in the input $\mathbf{x}$. CAEs can identify the most important variations in the data as smaller variations will be penalized resulting in meaningful hidden components [49]. Unfortunately, the CAE is expensive to compute as more hidden layers are added to the network architecture.

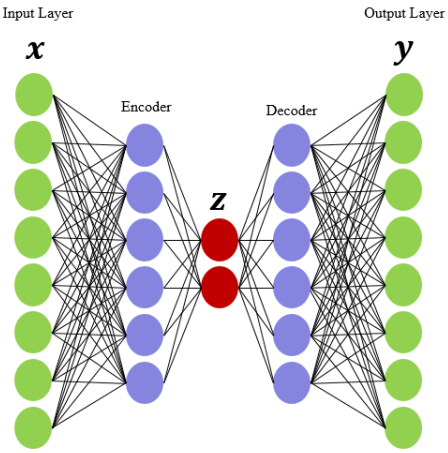


Figure 15. Standard AE Architecture

A computationally inexpensive regularization approach that shares many of the same properties as CAEs is the Denoising Autoencoder (DAE) where the input is corrupted before propagating through the AE network [50]. The DAE tries to reconstruct the original signal resulting in hidden components that are robust to small perturbations around each training sample.

While AEs have been extensively studied for many years, there have been recent modifications to the types of constraints placed on the embedded space. The Variational Autoencoder (VAE) model imposes a prior distribution, $p_\theta(\mathbf{z}) \sim N(\mathbf{0},\mathbf{I})$, on the AE embedded space [51, 52]. This constraint creates a continuous embedding space which can be sampled to create new samples not observed in the data. Specifically, the posterior is $p_\theta(\mathbf{z}|\mathbf{x})$ and for a given latent code sample $\mathbf{z}_i$, a new value $\mathbf{x}_i$ can be generated from the conditional distribution according to $p_\theta(\mathbf{x}|\mathbf{z} = \mathbf{z}_i)$. The entire data generated process can be described by

$$p(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})d\mathbf{z}$$

but unfortunately this is intractable to compute since we must integrate over all values of $\mathbf{z}$. Instead, in [52], an encoder model predicts distribution parameters ϕ, such that $q_\phi(\mathbf{z}|\mathbf{x})$ approximates the intractable distribution $p_\theta(\mathbf{z}|\mathbf{x})$.

To measure the difference between $p_\theta(\mathbf{z}|\mathbf{x})$ and $q_\phi(\mathbf{z}|\mathbf{x})$, the Kullback-Leibler divergence, D_{KL}, between the distributions is calculated:

$$D_{KL} = \left(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right) \tag{2.25}$$

and the goal is to minimize D_{KL} with respect to ϕ. Assuming the encoder model, $q_\phi(\mathbf{z}|\mathbf{x})$, predicts Gaussian distribution parameters, $\phi : \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ the posterior $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ can be sampled using $\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I})$ [52]. Additionally, D_{KL} can be rewritten as:

$$D_{KL} = \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right) \tag{2.26}$$

where j is the latent code dimension. Using D_{KL} allows $q_\phi(\mathbf{z}|\mathbf{x})$ to approximate $p_\theta(\mathbf{z}|\mathbf{x})$ but the probability of generating real data, $p_\theta(\mathbf{x})$, should be maximized while D_{KL} is minimized. Given a sample $\mathbf{x}_i$, the expected generated value is $\mathbb{E}_{\mathbf{z}_i \sim q_\phi(\mathbf{z}|\mathbf{x}_i)}\left[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)\right]$ and so the entire VAE loss function can be described by;

$$\mathscr{L}(\theta, \phi, \mathbf{x}) = -D_{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right) + \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|\mathbf{z})\right] \tag{2.27}$$

As discussed earlier, VAEs create a generative model by sampling the embedded space between training points. Since the embedded space follows the prior distribution, $p_\theta(\mathbf{z})$, sampling between training points is expected to generate results comparable to the training data with only minor changes based on the embedded space components.

Signals generated by VAEs typically do not contain high frequency features, leading to higher reconstruction error but a smoothly varying latent code. Sampling VAE latent codes fit with the ImageNet data results in understandable attribute vectors such as hair color or glasses, but introduces unacceptable image distortions. Generative models such as Generative Adversarial Networks (GANs) avoid such distortions and can create much

more detailed images difficult for humans to identify as real or fake [16]. The Adversarial Autoencoder (AAE) is another type of AE utilizing the properties of GANs to improve upon the generative ability of VAEs [53]. Both VAEs and AAEs shape the latent space with a prior distribution, but AAEs utilize adversarial training to enforce latent space distributions. AAEs consist of an encoder (generator), decoder and discriminator networks. The generator network, G, is similar to the VAE encoder as it predicts parameters for a probability density function. The discriminator network, D, tries to identify samples that originated from the true prior distribution and samples derived from the predicted distribution. This min-max formulation can be formalized as [53]:

$$\min_{G}\max_{D}\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log(D(\mathbf{x}))]+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1-D(G(\mathbf{z}))] \tag{2.28}$$

where $\mathbf{z}$ is the encoder predicted distribution parameters, p_{data} is the data distribution, $p_(\mathbf{z})$ is the prior and $G(\mathbf{z})$ is the sampling of the prior with these parameters. As training proceeds, the generator network predicts more realistic distribution parameters, resulting in latent space distributions close to the prior. The decoder still reconstructs the data into the original data space.

Since AAEs are optimizing both reconstruction error and discriminator accuracy, training is performed in two phases: reconstruction and regularization. In the reconstruction phase, only the encoder and decoder network weights are updated. The regularization phase uses the encoded data representation for discriminator training to identify samples from the true prior distribution (positive samples) and samples from the encoder generated distribution (negative samples). The encoder weights are then updated to create samples more representative of the prior distribution. Large changes during the reconstruction phase can adversely impact regularization phase results (i.e. the discriminator can easily identify positive and negative samples). AAEs also require more training iterations compared to VAEs because small learning rates must be used to ensure network training avoids mode

collapse. Mode collapse is a side-effect of the adversarial training process in which the generator and discriminator continue switching between modes in the data rather than using the entire multi-modal distribution.

Generative models such as VAEs, AAEs and GANs have been extensively applied to image data with applications ranging from data augmentation and style transfer to recommending merchandise based on a user provided picture. Application of generative models in hyperspectral classification research is limited. This research will utilize generative models for creating realistic TUD vectors such that surface leaving radiance can be estimated allowing for higher precision classification or target detection across diverse atmospheric conditions. Additionally, the generative models can also be used for data augmentation to increase classifier performance when labeled data is limited to a subset of atmospheric conditions.

2.5.2 Multimodal Representation Learning.

The AE networks discussed in the previous section focused on reconstructing a single vector input with variations in how the latent space was constrained. In cases where multiple measurement sources or modes are available, a combined representation should be formed to further constrain the latent space. Multimodal representation learning is a growing area of research, motivated by the observation that humans integrate multi-source information, such as audio-visual, to understand their environment [54]. Learning a combined representation can lead to improved results compared to learning two separate representations independently [55]. Some of the earliest research in learning multimodal representations combined audio of spoken words with video of people speaking to form a combined representation based on a Multimodal Autoencoder (MMAE) (Figure 16) [54].

In [54], they explored using one or both modes to learn a joint representation, allowing the network to generate the most likely video based on audio only and vice versa. Their

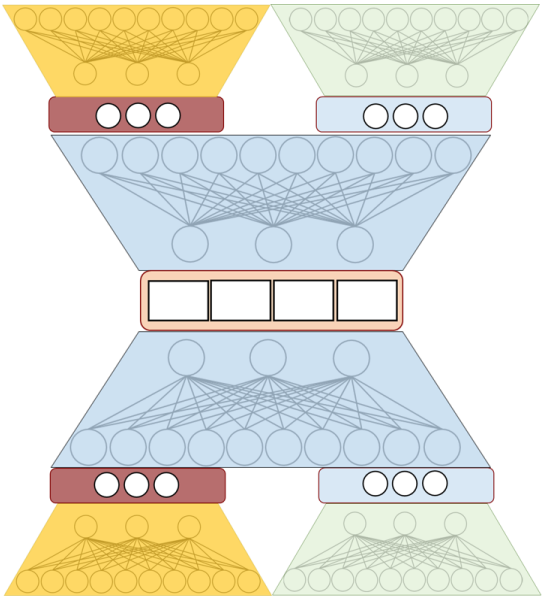


Figure 16. An example MMAE architecture is shown where two modes are presented to the network either simultaneously or one at a time. The MMAE must recover both modes when one or both modes are present at the input.

research demonstrated that generative modeling was possible with missing data modalities. A recent modification to the work presented in [54], found that unlabeled video data could be used to create a generative model for audio and images [56]. The model created in [56] could generate images of people playing guitar when presented the sound of a guitar or create realistic dog noises when presented pictures of dogs. Interestingly, the model was sensitive enough to distinguish bass guitars from acoustic guitars and create the correct corresponding imagery.

Multimodal representation learning can also be extended to textual information as shown in sentiment analysis research of online videos. In this domain, audio, video, text and human gesture information are combined to determine an overall video sentiment or opinion [57, 58]. Social media platforms such as Facebook and Twitter provide a nearly infinite data source for sentiment analysis research with many labeled datasets readily available [59, 60]. Healthcare research has also investigated multimodal representation learning,

combining medical test results, patient symptoms and imagery from multiple sources to predict the likelihood of a disease [61, 62].

Applications such as healthcare have heterogeneous modalities, creating challenging data fusion problems. Deterministic fusion techniques such as concatenation are too inflexible for many domains because this configuration only captures intra-modal relationships and misses any inter-modal dependencies [63]. Fusion techniques can be divided into three types: early, late and hybrid [64]. An example of early fusion is concatenation where the data is combined and then a joint representation is learned. Late fusion learns modal representations independently and then uses a voting or averaging to combine learned representations. Hybrid fusion is a dynamic technique that depends on early and late fusion results to produce the most useful representation. Numerous studies have investigated hybrid fusion techniques for combining heterogeneous data sources [65, 66]. The approaches include high order tensor pooling [64], adversarial training [66] and automatic architecture searches [67].

Recent advancements in generative modeling, such as VAEs [51, 52] have also been applied to multimodal data sources [68]. Creating a generative model for both modes at once is challenging because each mode must be independently conditioned on the same latent space. Given two observation variables (modes), $\mathbf{x}, \mathbf{w}$ and the latent space variable $\mathbf{z} \sim p_\theta(\mathbf{z}) = N(\mathbf{0}, \mathbf{I})$, the generating functions for the modes assuming an independent conditioning on the latent space:

$$\mathbf{X}, \mathbf{W} \sim p(\mathbf{x}, \mathbf{w}|\mathbf{z}) = p_{\theta_\mathbf{x}}(\mathbf{x}|\mathbf{z}) p_{\theta_\mathbf{w}}(\mathbf{w}|\mathbf{z}) \tag{2.29}$$

where the parameter θ represents the decoder network parameters for each mode, $\mathbf{X}$ and $\mathbf{W}$ respectively. Similar to the earlier VAE discussion, letting $q_\phi(\cdot)$ represent the encoder network, then the posterior distribution is $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{w})$ and the loss is calculated according to

[68]:

$$\mathscr{L}(\theta,\phi,\mathbf{x},\mathbf{w}) = -D_{KL}\Big(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{w})||p_\theta(\mathbf{z})\Big) + E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{w})}\Big[\log p_{\theta_\mathbf{x}}(\mathbf{x}|\mathbf{z})\Big] + E_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{w})}\Big[\log p_{\theta_\mathbf{w}}(\mathbf{w}|\mathbf{z})\Big] \tag{2.30}$$

The last two terms are reconstruction error over each output mode and the first term enforces the prior distribution over the latent space using the Kullback-Leibler divergence measure already discussed.

A multimodal representation learning approach is of interest to this research because the TUD vectors are derived from atmospheric state vectors: atmospheric temperature, water vapor content and ozone content. Both atmospheric state and TUD data can be used to create a joint, low-dimensional representation. Training models do not require fitting the model with missing modes, since the training data will have both atmospheric state data and the associated TUD vector. Applying a variational loss term to the joint representation as described in Equation 2.30 is another interesting way to constrain the latent space. Enforcing a continuous latent space across both atmospheric state and TUD vectors will improve sampling algorithm convergence for applications such as atmospheric compensation or radiative transfer modeling.

2.5.3 Permutation-Invariant Neural Networks.

Many machine learning problems rely on fixed input vectors to predict an output, known as instance-based learning [44, 69]. Classification and regression are common instance-based learning problems solved by models consuming fixed-input vectors [42]. Alternatively, parsing unstructured, variable length inputs remains a challenging problem limiting the cross-domain utility of many machine learning models. Domains such as 3D point cloud classification [70–72], scene classification [73] and outlier detection [69, 74] depend on sets of input data to predict a target value. This class of problems is referred to as set-

input learning or multi-instance learning where the the set order or the number of samples in the set should not impact algorithm performance [69]. These requirements are not easily met with standard neural network layers that expect fixed input dimensions. Permutation-invariant neural networks are specifically designed to meet these design requirements.

Given a set $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_M\}, \mathbf{x}_m \in \mathbb{R}^N$, a function $f(\cdot)$ is permutation-invariant if for any permutation π [75]:

$$f(\{\mathbf{x}_1, \ldots, \mathbf{x}_M\}) = f(\{\mathbf{x}_{\pi(1)}, \ldots, \mathbf{x}_{\pi(M)}\}). \tag{2.31}$$

The function $f(\cdot)$ can take on many forms such as sum-decomposition or max pooling across the set. In the case of sum decomposition the permutation-invariant function $f(\cdot)$ operating on the set $\mathbf{X}$ can be expressed by:

$$f(\mathbf{X}) = \rho\left(\sum_{m=1}^{M} \phi(\mathbf{x}_m)\right) \tag{2.32}$$

where (ρ, ϕ) are typically neural network layers [76]. If the operator $\phi(\cdot)$ is chosen such that $\Phi = \sum_{m=1}^{M} \phi(\mathbf{x}_m)$ is injective, then an operator $\rho(\cdot)$ can be chosen such that

$$\rho = f \circ \Phi^{-1},$$
$$f = \rho \circ \Phi,$$
$$\text{and } f(\mathbf{X}) = \rho\left(\sum_{m=1}^{M} \phi(\mathbf{x}_m)\right),$$

such that $f(\mathbf{X})$ is a valid set function [75, 77]. As stated in [75], this conclusion only extends to the countable case (M is finite). The function $\phi(\cdot)$ is applied to every input in the set $\mathbf{X}$, resulting in a low number of parameters when implemented as a neural network layer. When each member of the input set $\mathbf{X}$ is of dimension n and the output dimension

of $\phi(\cdot) \in \mathbb{R}^d$, the $\phi(\cdot)$ weight matrix contains $n \times d$ values, independent of the number of members in the set $\mathbf{X}$.

Additionally, in [76] it was shown the set max operator can be used to create a continuous, valid set function on $\mathbf{X}$. Specifically, the max operator returns the maximum value across the set at each of the vector values in $\phi(\mathbf{X})$ and can be described as:

$$f(\mathbf{X}) = \rho\left(\max_{m \in M} \phi(\mathbf{X})\right) \tag{2.33}$$

An advantage of the max-pooling operator over the sum-decomposition is a robustness to the number of elements in the set. As the set size, M, increases the max-pooling operator results in a better estimate of the true value.

A majority of the research in set-input learning is focused on the classification of Light Detection and Ranging (LIDAR) point clouds supporting fields such as autonomous driving, target detection and augmented reality [78]. One of the most well-known networks in this domain is PointNet where the authors utilized a max decomposition to simultaneously classify point clouds and segment individual points [76]. The PointNet research identified a subset of points C_s, known as the critical set, that were necessary for correct classification and segmentation. The number of critical points was directly related to the dimension of the set representation vector produced from the pooling operation. Visualizing the LIDAR point cloud critical points results in points spanning the shape of the object.

An improvement to PointNet was presented in [70], where a hierarchical structure was used to recursively apply PointNet to partitions of LIDAR point clouds. This proved beneficial for increasing point cloud classification by leveraging distance metrics to select a diversity of points. This intriguing idea is necessary for hyperspectral atmospheric compensation when data cubes consist of hundreds of thousands of pixels. Retooling PointNet for hyperspectral data requires careful data sampling techniques to verify the critical set, C_s, is extracted from the data cube.

Recently, attention mechanisms have pushed the state-of-the-art in neural language processing, image classification and neural machine translation by weighting portions of the input more heavily to achieve a task [79, 80]. Set-input learning has also investigated attention mechanisms to replace the sum or max pooling operations with a trainable set decomposition [69, 81]. Attention-based pooling also provides a set of weights to interpret the value of each member in the set.

Using the previous nomenclature, the transformed set is $\mathbf{H} = \phi(\mathbf{X})$ where $\mathbf{H} \in \mathbb{R}^{M \times d}$ assuming the network $\phi(\cdot)$ has d output nodes. The attention-based pooling in [81] utilizes a weighted dot product to measure the importance of samples within the set $\mathbf{X}$ [81]:

$$\mathbf{z} = \sum_{i=1}^{M} a_i \mathbf{h}_i$$

where the weighting term, a_i is:

$$a_i = \frac{\exp\left(\mathbf{w}^T \tanh(\mathbf{V}\mathbf{h}_i^T)\right)}{\sum_{i=1}^{M} \exp\left(\mathbf{w}^T \tanh(\mathbf{V}\mathbf{h}_j^T)\right)}$$

The learned weight vector, $\mathbf{w} \in \mathbb{R}^{1 \times l}$ contains l nodes and the weight matrix $\mathbf{V} \in \mathbb{R}^{l \times d}$ transforms the sample representations into an $l \times 1$ vector. Neither $\mathbf{w}$ or $\mathbf{V}$ are dependent on the number of samples M in the set $\mathbf{X}$. Applying an attention-based pooling operation to a permutation-invariant network provides insights into what features are important for creating the set representation vector $\mathbf{z}$. In LWIR atmospheric compensation scenarios, it is expected that pixels with blackbody-like characteristics will have high attention scores, a_i, to resolve $\tau(\lambda)$ and $L_a(\lambda)$ while lower emissivity pixels are also necessary to recover $L_d(\lambda)$.

2.5.4 Convolutional Neural Networks.

Convolutional Neural Networks (CNNs) are a type of network architecture useful for analyzing data taking on a grid form such as 1D spectral data, 2D imagery or 3D hyperspectral cubes [82]. These biologically-inspired networks have played an important role in tasks such as image classification [44], facial recognition [83], and address classification from Google Streetview imagery [84] often exceeding human-level performance [85]. Additionally, CNNs have been extensively studied for 1D data in applications such as speech recognition [86] and classification of spectral data within hyperspectral cubes [87].

A fully-connected network contains parameters for each input-output pair in each layer. This leads to an unacceptably large number of parameters to train if the input data has many dimensions. CNNs are sparsely connected at each layer, significantly reducing the number of trainable parameters needed for meaningful affine transformations of the data [42]. Each layer of a CNN consists of many kernels or filters significantly smaller than the input vector size. The kernels are applied at small local inputs and shared across all nodes of the input space creating unique feature maps.

During training on image data, kernel weights converge toward edge detectors for lower layers and abstract feature detectors at higher layers. In image classification tasks convolutional layers are followed by pooling layers to extract statistical summaries across large receptive fields of the input. Max-pooling is the most commonly applied pooling approach, reporting the maximum value from the nonlinear activation within a local neighborhood of input points. Pooling makes the entire transformation invariant to small shifts in the data, allowing features to be detected regardless of their location. Pooling can be conducted on individual feature maps or across all feature maps, known as global pooling. Applying CNNs to 1D spectral data from a single pixel in a hyperspectral image requires careful consideration of kernel size and pooling operations. Kernels will extract local relationships amongst neighboring spectral bands, creating feature maps tuned to specific material prop-

erties. Examples include the depth of water absorption bands in observed spectrum or the asymmetric band feature centered at 9 µm for silica oxygen bonding [21]. Adjusting the kernel size such that it covers specific regions of interest in the EM spectrum will result in understandable filters, but classification performance may not be optimized.

The spectral location where features are detected is critically important as material properties are wavelength dependent. Applying pooling to 1D spectral data would create an invariance to shifts in the spectral data. This isn't recommended and in practice pooling is found to reduce classification performance because of feature location importance when changing atmospheric conditions are considered [19]. Global pooling across all feature maps may still be appropriate if feature locations can be maintained.

Classifying data using CNNs requires a mapping from the convolutional feature maps to class labels. This is accomplished by flattening all feature maps into a high-dimensional feature vector and propagating this vector through one or more fully-connected layers. The final classification layer utilizes the softmax activation function to represent a probability distribution over n different classes [42]:

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_{j=1}^{N} \exp(z_j)} \tag{2.34}$$

where

$$z_i = \log(P(y = i|\mathbf{x})) \tag{2.35}$$

and $\mathbf{x}$ is the input vector. The goal of the training algorithm is to predict the correct class by maximizing $\log \text{softmax}(\mathbf{z})_i$. Most publicly available hyperspectral datasets contain labeled data for fewer than 20 classes. In reality, hundreds to thousands of materials must be identified or labeled as background for real-world image classification tasks. Image classification competitions commonly use datasets with thousands of possible image labels. Hyperspectral data classification will require larger datasets covering a wider range of ma-

terials for operational use. In the next section, a review of hyperspectral classification is provided detailing the current state-of-the-art in this field.

2.6 Hyperspectral Image Classification

Hyperspectral image classification has been an active area of research since the inception of hyperspectral sensors. Image classification consists of labeling all pixels in the scene as one of K predefined materials or classes creating an abundance or classification map. For each of the K materials, hundreds of pixels contain these materials providing replicates for classifier training. Hyperspectral classification is different from target detection as target detection scenarios may only contain a single pixel containing a material of interest [6]. Classification applications include land-use mapping, geology, forestry, urban development studies and many others [88].

Exploiting the high-resolution spectral information sampled across hundreds of contiguous spectral bands necessitates new approaches compared to gray scale and multispectral image analysis. Supervised classification approaches require labeled training samples that span the expected data variability to achieve stable classification performance. Collecting and labeling enough data to fill the high-dimensional space created by hundreds of spectral channels is time-consuming and costly. For a constant number of training samples, classification performance decreases as the number of spectral channels increases, known as the curse of dimensionality [89]. To address this limitation in training data availability, most supervised classification techniques use feature extraction or reduction techniques to reduce the input parameter space.

PCA is the most common feature reduction technique applied to hyperspectral data, capturing nearly all data variance in less than 20 components for sensors spanning hundreds of spectral channels [90]. An example of the component loadings is shown in Figure 17 for the Kennedy Space Center hyperspectral dataset where the first component clearly captures

loads strongly on water pixels against all other pixels in the scene. A drawback of PCA is that it only considers variance within the data, regardless if the variance is caused by the signals of interest or sensor noise. If sensor noise can be estimated, the components can be sorted based on their Signal to Noise Ratio (SNR) rather than data variance. This approach is known as minimum noise fraction (MNF). Another common feature reduction technique for hyperspectral data is Independent Component Analysis (ICA). ICA identifies the underlying components making up a signal based on the assumption that no more than one of these components are Gaussian distributed and all are statistically independent of one another [46] [91]. ICA is known for its ability to perform blind source separation in problems where an unknown number of underlying sources are present.

Neural networks, specifically AEs, have also been comprehensively studied for feature reduction. AEs can create low dimensional latent spaces or embeddings allowing for lower reconstruction error than PCA while also producing informative latent dimensions. When varied, these dimensions reveal interesting relationships in the data and provide insight of the high-dimensional structure [47]. Chen et al. were the first to use AEs to create informative low-dimensional feature vectors for hyperspectral classification [92]. They implemented an iterative approach where an AE was trained to convergence and then the decoder was removed, leaving an input to latent space representation. This process was repeated adding additional layers to create a Stacked Autoencoder (SAE). The output of the SAE was used to compare classification algorithms such as Support Vector Machine (SVM) and K-nearest neighbors (KNN) resulting in competitive performance for all techniques. The work performed by Chen et al. is also considered the first deep learning model for hyperspectral classification since the model is composed of multiple hidden layers.

Additional deep learning-based classification approaches have been investigated for hyperspectral image classification. Hu et al. applied 1D CNNs along the spectral axis to classify hyperspectral data [87]. They showed 1D CNNs could outperform Radial Basis

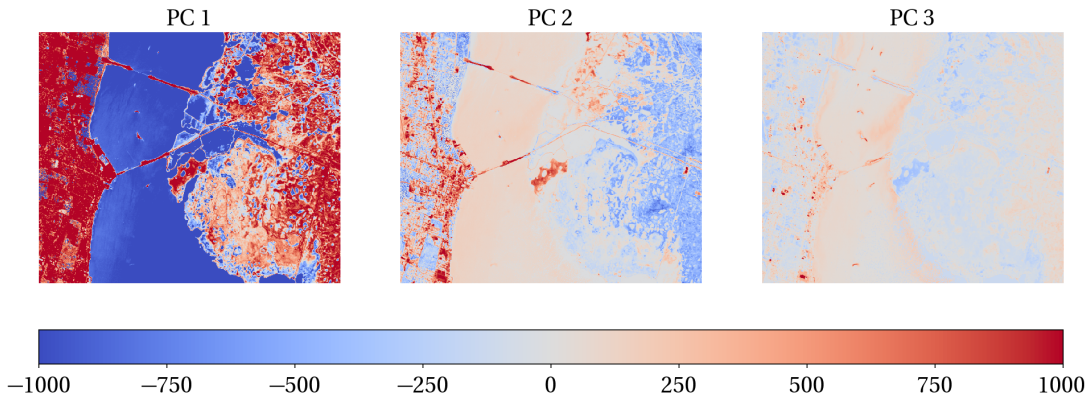
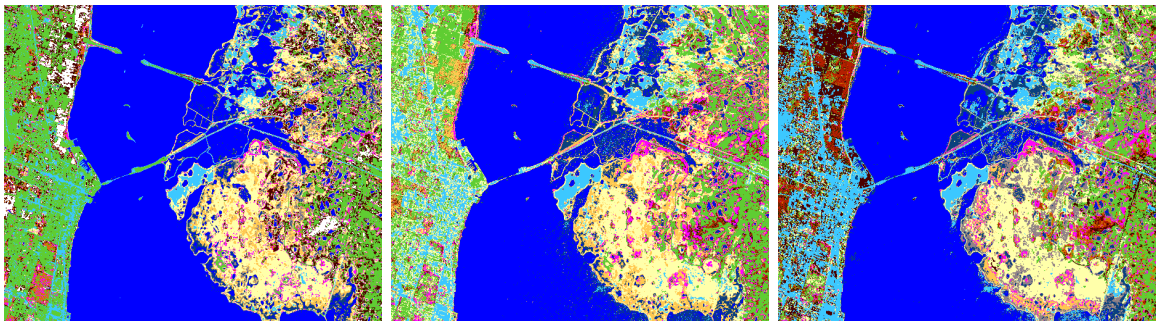


Figure 17. Intensity plots showing the first 3 components using PCA on a benchmark hyperspectral dataset (Kennedy Space Center) where the color represents the component loading. Most variance in the data is from water pixels in the center and the urban area on the left side of the image.

Function (RBF) SVM classifiers and fully-connected networks on benchmark hyperspectral data. Since convolution is performed directly on the spectral data, feature reduction is not necessary as the entire CNN will extract the most salient spectral features. A comparison of SVM, fully-connected networks, and 1D CNNs is shown in Figure 18 highlighting small differences in each of these approaches for the Kennedy Space Center dataset. These results were generated by this author to replicate the results of previous research in this field. Qualitatively, Figure 18 shows the high sensitivity of the 1D CNN compared to SVM and fully-connected networks. The results shown in Figure 18 do not consider spatial information, but this is an active area of research [88, 93].

The 1D CNN work performed by Hu et al. was extended to a 2D CNN implementation by Makantasis et al., using both spatial and spectral information to encode pixel data [94]. A neighborhood of 5 pixels was defined around each pixel and 3x3 convolutions were performed on each patch. PCA was performed along the spectral axis resulting in less than 30 components or transformed spectral channels. The 3x3 convolutions were performed on each of the components, achieving over 98% accuracy on all benchmark datasets. As shown in [95], small objects can be missed with 2D CNNs because of inappropriately



(a) (b) (c)

Figure 18. Visualizing classification results using (a) SVM, (b) fully-connected network, and a (c) 1D CNN for the Kennedy Space Center shows some small differences, but overall the methods are able to accurately predict most pixels in the scene. Truth data is only provided for a small number of pixels in the entire scene.

sized neighborhoods or single pixel or sub-pixel materials. The sensor spatial resolution is an important property to consider when applying 2D CNNs for hyperspectral classification.

Since hyperspectral data forms a three-dimensional data cube, 3D CNNs were also investigated for hyperspectral classification [96]. Similar to [87], Chen et al. defined small pixel patches around a pixel of interest such that small convolutional filters could be applied. The patch size was 27x27 pixels with filters of size 4x4x32 where the last dimension is the spectral axis. To prevent overfitting on the limited number of training samples, weight regularization and dropout were also used. Interestingly, both Hu et al. and Chen et al. used the same benchmark datasets, but report significantly different patch sizes for optimal CNN performance. In both cases, the reported overall and average accuracy is greater than 98% on the benchmark datasets.

More advanced deep learning algorithms such as recurrent neural networks [93] and GANs [97] have been applied to the same benchmark hyperspectral datasets considered in the papers outlined so far. These approaches offer additional advantages in network overfitting and data augmentation, however, it is difficult to determine an optimal classification technique since performance is saturated across most datasets. For example, in [93] a recurrent network is trained on the Pavia University dataset and compared with a random

forest classifier, SVM and a CNN. The overall accuracy of all these methods is between 82% and 86% with the CNN achieving the highest overall accuracy. Additionally, training and test sample distributions aren't equivalent across papers compounding the problem of identifying an optimal classification method.

Applying these techniques to datasets, real or synthetic, with atmospherically diverse data will provide a better measure of their practical use and help to identify the leading classification approach. Limited work has been performed comparing classifier performance across atmospherically varying data. In [98], an AE creates a low-dimensional embedding for pixels in shadow and sunlight for the VNIR and SWIR domains. The shadow invariant embedding allows all pixels to be projected to a sunlit representation, making classification easier as scene illumination changes throughout the day. Physics-based data augmentation was extensively used to provide the AE representative samples. Similar to [98], in [19], classification performance was compared across a 9 hour period in the LWIR domain to determine an optimal classification approach, however, no data augmentation was used, but a significant improvement in performance was observed using CNNs compared to fully-connected networks and SVMs.

2.7 Hyperspectral Target Detection

Distinctly different from hyperspectral land cover classification is hyperspectral target detection in which materials of interest occur with low probability. The low occurrence of target pixels limits the type of algorithms that can be used. The Neyman-Pearson criterion is used to maximize target detection probability while reducing false-alarm rates to an acceptable level [21, 99]. Target detection is a binary classification problem where pixels are labeled as either background or targets through two steps: anomaly detection and target identification [100].

Anomaly detection approaches do not consider target information, but instead identify pixels with significantly different spectral signatures from neighboring pixels. The Reed-Xiaoli (RX) detector is a benchmark method for anomaly detection that adapts mean and covariance estimates of background clutter for local regions around a pixel under test [101]. The RX detector is defined as:

$$r_{RX}(\mathbf{x}) = (\mathbf{x} - \hat{\boldsymbol{\mu}}_{local})^T \hat{\boldsymbol{\Sigma}}_{local}^{-1} (\mathbf{x} - \hat{\boldsymbol{\mu}}_{local}) \tag{2.36}$$

where the statistics $\hat{\boldsymbol{\mu}}_{local}$ and $\hat{\boldsymbol{\Sigma}}_{local}$ are estimated based on a local neighborhood of pixels.

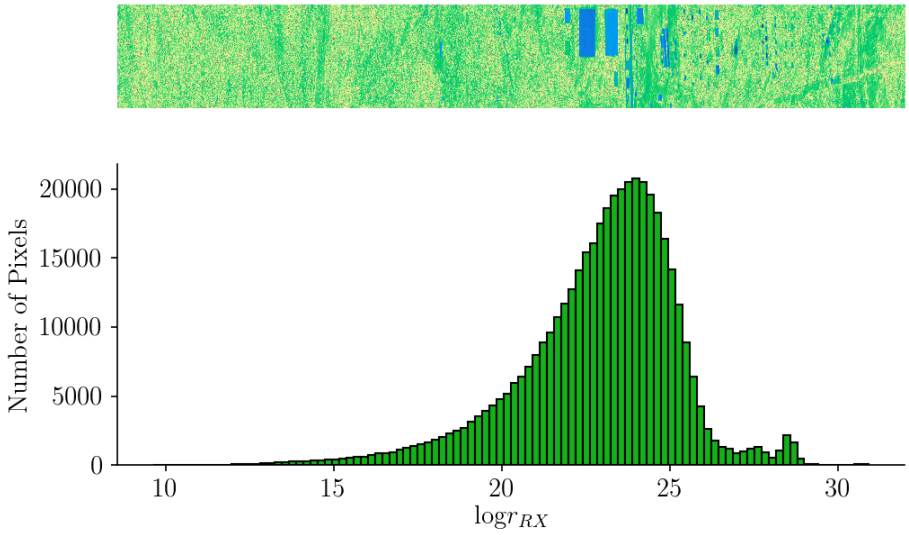


Figure 19. The RX detector described in Equation 2.36 is applied to a collected data cube, where the pixel detection statistics are shown spatially (top) and as a histogram (bottom). Applying a threshold at approximately 27 will remove most of the background pixels from further processing.

A guard band around the pixel under test can be used to ensure candidate target pixels don't interact with local background statistics. Using local windows to estimate background statistics around an individual pixel is computationally expensive and not recommended for real-time processing [21]. Instead, the entire data cube can be used to calculate $\hat{\boldsymbol{\Sigma}}^{-1}$ and $\hat{\boldsymbol{\mu}}$ assuming target pixels are rare in the scene. RX detection results are shown for a collected data cube in Figure 19. Applying a threshold to the histogram in Figure 19

at approximately 27 will remove most of the background pixels. Panels present in the scene are easily identified visually based on the RX detection statistic but smaller objects may be difficult to resolve. Commonly, PCA is applied along the spectral axis to remove redundancy in the collected data. Figure 20 shows the first two principal components with an RX decision boundary overlaid. Numerous false-alarms are observed, but most of the background pixels have been correctly identified.

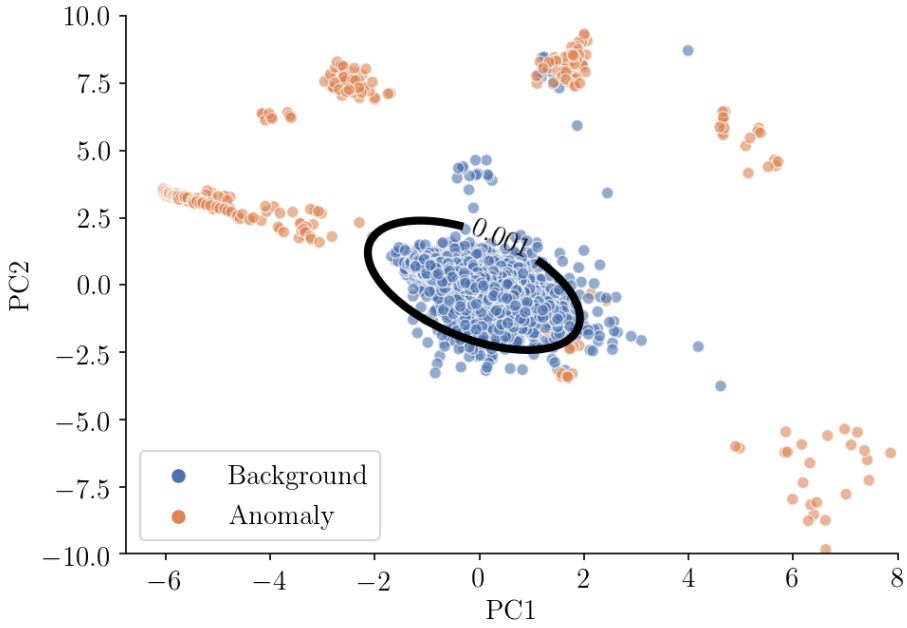


Figure 20. PCA was applied to a collected data cube along the spectral axis and the first two principal components are shown. Applying the RX detector on this reduced data results in the decision boundary shown. Point colors represent the truth data showing many false alarms.

The anomaly detection processing step is useful for many applications such as search and rescue or change detection. Change detection is an active area of research leveraging anomaly detection approaches to identify subtle differences between registered images [102, 103]. The detected differences are often lost in the background data and can only be detected using anomaly detectors with temporally-varying data.

Change detection scenarios require careful construction of background mean and covariance matrices. A nonlinear approach investigated in [104] showed how an AE model

could be used to perform anomaly detection. A Mahalanobis distance (MD) detector was compared to a denoising AE (DAE) for the anomaly detection task. Additionally, an RX detector was defined using a background covariance matrix derived from DAE background estimates, denoted as MD $\hat{\Sigma}^{-1}_{\text{DAE}}$.

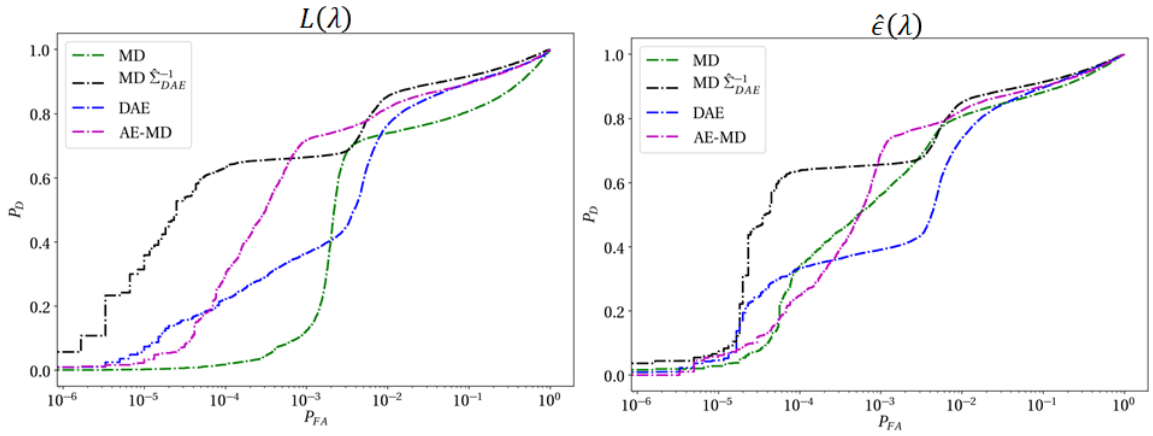


Figure 21. Receiver Operating Characteristic (ROC) curves are shown for various anomaly detection methods utilizing AE models in unique ways. MD stands for Mahalanobis distance or the RX detector. The MD $\hat{\Sigma}^{-1}_{\text{DAE}}$ results use an AE to define the background pixels and calculate background statistics for the RX detector. A denoising AE (DAE) and an RX detector applied to the low-dimensional AE latent space is compared (AE-MD).

After anomaly detection, signature-matched target detection approaches utilize an exemplar spectrum to identify remaining pixels as either targets or background. This conclusion depends on a hypothesis test where the null hypothesis, H_0, concludes the pixel is background material while the alternative, H_1, concludes the pixel is a target. Since targets are exceedingly rare, the Neyman-Pearson criterion is used to maximize the probability of detection at an acceptable probability of false-alarm level. The likelihood ratio for pixel spectra, $\mathbf{x}$ is:

$$l(\mathbf{x}) = \frac{p(\mathbf{x}|H_1)}{p(\mathbf{x}|H_0)} \tag{2.37}$$

where $p(\mathbf{x}|H_i)$ is the probability density function for the i^{th} distribution. Unfortunately, in most real-world scenarios these probability density functions are not known *a priori* and must be estimated. The generalized likelihood ratio test uses image data to identify the

maximum-likelihood estimates, θ_i, of the probability density function parameters such that this new likelihood ratio test can be expressed as:

$$l(\mathbf{x}) = \frac{\max_{\theta_1} p(\mathbf{x}|H_1)}{\max_{\theta_0} p(\mathbf{x}|H_0)} \tag{2.38}$$

The observed pixel spectra $\mathbf{x}$ can be described by the additive signal model [21]:

$$\mathbf{x} = \alpha\mathbf{s} + \mathbf{b}, \tag{2.39}$$

where $\mathbf{s}$ is the the known target signal and α accounts for deviations in the target spectrum from subpixel mixing or changes in illumination [2]. The background clutter model is described by $\mathbf{b} \sim \mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I})$. The Spectral Angle Mapper (SAM) uses the additive signal model described to create the hypothesis test:

$$H_0 : \mathbf{x} = \mathbf{b}$$

$$H_1 : \mathbf{x} = \alpha\mathbf{s} + \mathbf{b}.$$

Converting to the logarithmic form of the generalized likelihood ratio test, the detection statistic for the SAM detector is:

$$r(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{x}}{\sigma^2} - \frac{(\mathbf{x}-\alpha\mathbf{s})^T(\mathbf{x}-\alpha\mathbf{s})}{\sigma^2} = \frac{2\alpha\mathbf{x}^T\mathbf{x} - \alpha^2\mathbf{s}^T\mathbf{s}}{\sigma^2} \tag{2.40}$$

After substituting the maximum likelihood estimate for $\alpha = \mathbf{s}^T\mathbf{x}/\mathbf{s}^T\mathbf{s}$ the generalized likelihood ratio test for the SAM detector is [2]:

$$r_{SAM}(\mathbf{x}) = \frac{1}{\sigma^2}\frac{(\mathbf{s}^T\mathbf{x})^2}{(\mathbf{s}^T\mathbf{s})} = \frac{(\mathbf{s}^T\mathbf{x})^2}{(\mathbf{s}^T\mathbf{s})\mathbf{x}^T\mathbf{x}} \tag{2.41}$$

after estimating the background variance using $\mathbf{x}^T\mathbf{x}$.

The assumption of zero mean, white background is a significant drawback for the SAM detector. The Spectral Matched Filter (SMF) detector improves on this by allowing a background distribution to follow $\mathbf{b} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The SMF detection statistic can be formulated as:

$$r_{SMF}(\mathbf{x}) = (\mathbf{s}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}) - (\mathbf{x}-\alpha\mathbf{s}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\alpha\mathbf{s}-\boldsymbol{\mu}) \tag{2.42}$$

and after substituting in the maximum likelihood estimate for α and simplifying becomes [105]:

$$r_{SMF}(\mathbf{x}) = \frac{\left[\mathbf{s}^T\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}})\right]^2}{\mathbf{s}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{s}}. \tag{2.43}$$

The additive signal model is applicable for sensors with large ground sampling distances, but as sensor resolution improves, target pixels begin to fill a substantial portion of the pixel. In this case, a replacement model is more appropriately specified by:

$$H_0 : \mathbf{x} = \beta\mathbf{b}$$

$$H_1 : \mathbf{x} = \alpha\mathbf{s} + \beta\mathbf{b}$$

Subtracting the cube mean $\boldsymbol{\mu}$ from both the target spectrum and all pixel spectra results in the background distribution $\mathbf{b} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$. To estimate $\hat{\boldsymbol{\Sigma}}$, N training pixels are selected that do not contain targets. After simplifying the generalized likelihood ratio, the Adaptive Coherence/Cosine Estimator (ACE) detection statistic is [106]:

$$r_{ACE}(\mathbf{x}) = \frac{(\mathbf{s}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{x})^2}{(\mathbf{s}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{s})(\mathbf{x}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{x})} \tag{2.44}$$

Next, the entire target detection pipeline is demonstrated using the SMF and ACE detectors. The data considered was collected by the SEBASS LWIR sensor and are the same images used in [107, 108]. The target emissivity was converted to at-sensor radiance using the estimated surface temperature for the cube and the predicted TUD vector. Then, apply-

ing the RX detector results in the background pixel classification shown in Figure 22 in the second pane from the left. These pixels were used to create the background covariance $\mathbf{\Sigma}$ and mean $\boldsymbol{\mu}$ vector for each detection algorithm. The detection maps are shown in Figure 22 for SMF and ACE, where both methods correctly identify the labeled pixels. Additionally, ROC plots are shown in Figure 23 demonstrating high detection rates at low false-alarm rates. The target material used in these plots was foamboard, containing distinct spectral features making detection easier.

There are numerous metrics available for comparing detectors such as ROC curves, area under ROC curves, precision, recall or Signal to Clutter Ratio (SCR). SCR is commonly used in hyperspectral target detection and is described by:

$$\text{SCR} = \frac{\mu(r_t) - \mu(r_b)}{\sqrt{\sigma(r_t)^2 + \sigma(r_b)^2}} \tag{2.45}$$

where $\mu(r_t)$ is the mean detection statistic for target pixels while $\mu(r_b)$ is the mean detection statistic for background pixels. The standard deviation of these measures is represented by $\sigma(\cdot)$ where a higher SCR reflects a better target detection result.

Modeling the background clutter as a Gaussian distribution is not always practical since real data rarely follows this distribution. Modeling the background as a mixture of Gaussians is also problematic since it's unclear how many Gaussian mixtures are needed. By leveraging kernel machine learning [109], these problems can be overcome and detector performance improved. Specifically, the data is projected to a high-dimensional space with a kernel function enforcing data separability. There are numerous kernel functions available but the most commonly used function is the RBF described by:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2}{2\sigma^2}\right) \tag{2.46}$$

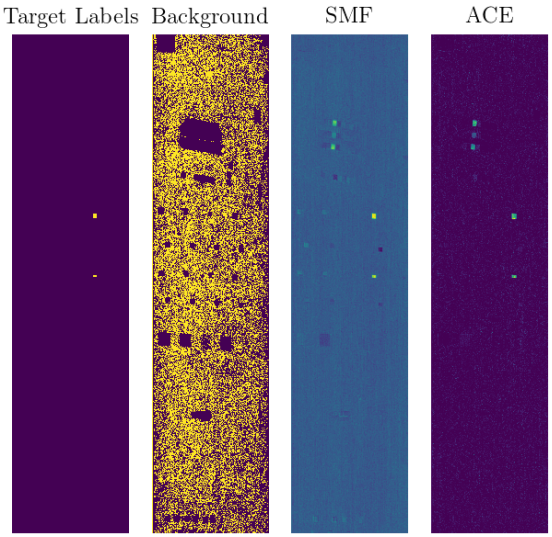


Figure 22. Background and target detection maps for SMF and ACE

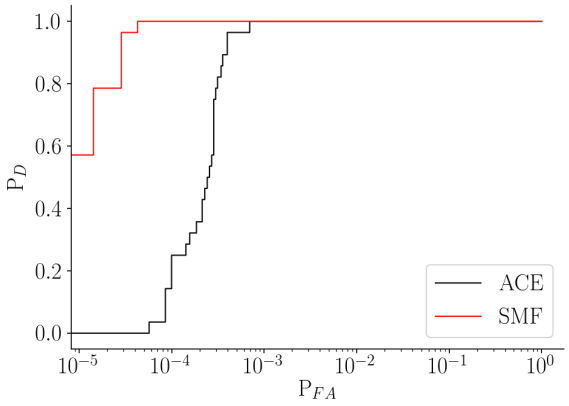


Figure 23. ROC plots based on the detection plots shown in Figure 22.

where σ is the Gaussian standard deviation that is adjusted for the classification task. This non-linear mapping is useful for a wide range of detection algorithms as shown by the comparison in [110] where the kernel-based detectors outperform standard detectors on several hyperspectral datasets.

SVM classifiers are a specific type of kernel machine learning methods which have been extremely successful for hyperspectral classification. Their use is limited in target detection because of an insufficient number of target training samples to define the non-linear classification boundary. This limitation can be avoided by creating a subspace of representative target samples using radiative transfer models such as MODTRAN to synthetically augment target data. A data augmentation approach was implemented in [111] to create an atmospherically invariant subspace detector for the VNIR-SWIR domain.

Neural networks are another machine learning approach that has resulted in state-of-the-art results for hyperspectral classification. Applying neural networks for target identification also requires a synthetically-augmented database of target samples to fit the network's nonlinear decision boundary. As shown in [112], small neural networks can be

used for target detection with performance exceeding standard detection algorithms such as ACE. The real-world accuracy of machine learning approaches, such as SVM and neural networks, will depend on the data augmentation strategy. Specifically, realistic noise must be included in example target spectra for correct identification. Sensor spatial resolution will also impact detection ability as low resolution sensors collect mixed-pixel spectra requiring data augmentation to also include this variability.

The following chapters will leverage the background material reviewed in this chapter to derive faster in-scene atmospheric compensation algorithms. The next chapter presents a case study on the importance of performing atmospheric compensation to support accurate material classification, comparing TES methods with a variety of machine learning classifiers. The following chapters derive generative modeling techniques and permutation-invariant network architectures to accelerate in-scene atmospheric compensation leading to faster target detection.

III. Analysis of Long-Wave Infrared Hyperspectral Classification Performance Across Changing Scene Illumination

3.1 Paper Overview

This conference paper was presented at the Society of Photonic Instrumentation Engineers (SPIE) Defense and Commercial Sensing conference 17 April, 2019. Hyperspectral image classification is considered across changing atmospheric conditions, demonstrating limitations in land cover classification techniques if the collected hyperspectral data is not corrected for atmospheric effects. This study compared classifier performance by modifying how training and test data were selected. Specifically, test data partitions considered cubes containing surface temperatures within and outside the training data range. This research found that One-Dimensional Convolutional Neural Network (1D-CNN) classifiers achieve the highest accuracy when surface temperatures are outside the training data range, emulating real-world conditions where an exhaustive training set is not available.

This paper was presented at the 2019 SPIE Defense and Commercial Sensing Conference and published in the conference proceedings.

3.2 Abstract

Hyperspectral sensors collect data across a wide range of the electromagnetic spectrum, encoding information about the materials comprising each pixel in the scene as well as atmospheric effects and illumination conditions. Changes in scene illumination and atmospheric conditions can strongly affect the observed spectra. In the long-wave infrared, temperature variations resulting from illumination changes produce widely varying at-aperture signals and create a complex material identification problem. Machine learning techniques can use the high-dimensional spectral data to classify a diverse set of materials with high accuracy. In this study, classification techniques are investigated for a long-wave

hyperspectral imager. A scene consisting of 9 different materials is imaged over an entire day providing diversity in scene illumination and surface temperatures. A Support Vector Machine classifier, feedforward neural network, and one-dimensional convolutional neural network (1D-CNN) are compared to determine which method is most robust to changes in scene illumination. The 1D-CNN outperforms the other classification methods by a wide margin when presented hyperspectral data cubes significantly different from the training data distribution. This analysis simulates real-world classifier use and validates the robustness of the 1D-CNN to changing illumination and material temperatures.

3.3 Introduction

Hyperspectral imaging is the science of simultaneously collecting spatial and spectral information to allow for detailed material characterization and identification [2]. Unique to hyperspectral imaging is the per-pixel collection of hundreds to thousands of continuous spectral channels across a wide bandpass in the electro-optical spectrum. As a hyperspectral imager scans a scene, a three-dimensional – two spatial, one spectral – data cube is formed. Slicing the hyperspectral cube across any spectral band produces a spatial image of the scene, and viewing any pixel across all spectral channels produces a spectrum encoding information about the material(s) within that pixel along with thermal, atmospheric, and illumination conditions.

The idea of hyperspectral imaging was first proposed in the 1980s at the NASA Jet Propulsion Laboratory [22]. Since that time, advances in spectrometer design, data processing, atmospheric compensation and spectral resolution have led to many hyperspectral imaging applications ranging from land cover mapping and crop health assessment to search and rescue operations [2, 21, 113]. Initially, these sensors were found only on airborne platforms, but as their size has continued to decrease they can now be found on small unmanned aerial vehicles.

Regardless of the platform and remote sensing application, these sensors can generate large volumes of data. For example, the Earth Observing-1 (EO-1) Hyperion spaceborne sensor can collect at a rate of 71.9 GB/h in the range of 0.4–2.5 µm across 220 spectral bands [114, 115]. As hyperspectral imaging sensors proliferate, robust and efficient methods to automate the reduction and exploitation of their high-bandwidth will be required. Machine learning approaches have already demonstrated great performance on hyperspectral data.

Recently, deep learning approaches have shown promising results on benchmark hyperspectral data sets in accurately identifying a wide range of materials. Unfortunately, most spectral-based classification research has only considered a single hyperspectral data cube at a time. A classification algorithm trained on data from a single scene is expected to perform poorly when presented new data collected under different illumination and/or atmospheric conditions. However, operational sensors require real-time performance across diverse atmospheric conditions. (While atmospheric correction is possible, it can be a time-consuming step which nonlinearly (and often imperfectly) transforms the at-aperture signal.) Algorithms which directly process the at-aperture signal and are robust to changes in atmospheric conditions would simplify the use of hyperspectral data products with minimal postprocessing.

To address changing at-aperture radiance as a function of atmospheric variability, this paper applies Support Vector Machine (SVM), Artificial Neural Network (ANN), and 1D-CNN to hyperspectral data cubes collected throughout an entire day. The hyperspectral sensor and the objects in the scene are stationary, but varying illumination causes the objects to move in and out of shadows. This is the first comparison of these machine learning algorithms on multi-cube Long-Wave Infrared (LWIR) hyperspectral data in order to identify architectures which are robust to changing scene illumination.

In the following sections a review of relevant research is presented, highlighting related machine learning methods applied to hyperspectral data. This is followed by an overview of the data collection, data preprocessing and classifier implementation details. Results are presented for all classification methods focusing on advantages and disadvantages of each method. Finally, we conclude with major takeaways from this study and areas where future work will be focused.

3.4 Background

To support a wide range of applications for hyperspectral imagery, significant research has been conducted to correctly identify a diverse set of materials within a particular scene [116]. These applications include land cover mapping, target detection and material classification. The material detection scheme is heavily dependent on the types of materials within a particular scene and target populations. Target detection is primarily focused on finding one object, possibly at the sub-pixel level, out of hundreds of thousands of possible pixels [2]. This is in contrast to land cover mapping applications, in which each scene often has hundreds of pixels for each material of interest.

Many classification algorithms have been implemented for hyperspectral data analysis. Methods such as Adaptive Coherence/Cosine Estimator (ACE), Spectral Angle Mapper (SAM) or Spectral Matched Filter (SMF) seek to identify pixels in a scene by defining a background spectral signature and a target spectral signature [117]. The background pixel spectra are typically assumed to fit a multivariate normal distribution; however, in practice, the tails of the background data distribution diverge from the multivariate normal distribution [6], leading to increased false-alarm rates and pixel misclassification. Additionally, these techniques are typically performed on emissivity or reflectivity and require atmospheric compensation and possibly temperature estimation before pixel classification.

A large number of machine learning methods have also been investigated for classifying hyperspectral data. These techniques can operate directly on the sensor data without atmospheric compensation. Machine learning methods such as SVM have been shown to increase material identification accuracy on several benchmark hyperspectral datasets, outperforming method such as ACE, SAM and SMF. [118], [119] SVMs are able to handle high-dimensional data with limited training samples, making them an excellent choice for hyperspectral classification [88]. A nonlinear SVM consisting of a Gaussian Radial Basis Function (RBF) is typically employed on most classification problems. This classification approach results in greater than 90% accuracy on many commonly investigated hyperspectral data sets [88]. While training time is very short for SVM classifiers, pixel inference can be time consuming compared to neural network classifiers.

Recently, deep learning techniques have achieved state-of-the-art performance in areas such as computer vision, machine translation and natural language processing [48]. Classifying hyperspectral data has many underlying similarities with these challenging big data problems, making deep learning a logical next step for hyperspectral classification. The stacked autoencoder implementation by Chen et al. (2014) was the first use of a deep model to classify hyperspectral data [92]. By continually training layers of the stacked autoencoder, deep representations of the spectral data were created leading to state-of-the-art classification at the time. Additionally, the pixel inference time is much shorter compared to SVM classifiers making deep learning approaches a viable technique for real-time classification scenarios.

Since many of the spectral bands are highly correlated, a 1D-CNN can be oriented to perform convolution along the spectral dimension, selecting salient features across small spectral windows. 1D-CNNs use a series of learned filters to detect specific patterns in the spectral data. The 1D-CNN has been shown to outperform SVMs on several benchmark hyperspectral data sets, motivating this research to revisit this algorithm with multiple hy-

perspectral cubes of training data [87]. Significant research has also been performed toward spatial-spectral classification. This area utilizes the spatial information in a scene to further improve classification accuracy over spectral only classifiers. The work presented in this paper does not consider spatial-spectral classifiers because we are primarily interested in how well spectral classifiers perform with slight changes to scene illumination. Additionally, spectral-only classifiers may be more useful in single pixel to sub-pixel target detection scenarios versus spectral-spatial classifiers.

Research using multiple hyperspectral data cubes to train and test classification algorithms is limited. All deep learning research discussed thus far have trained and tested on a single data cube - meaning they only contain information from one particular atmospheric scenario. Variations in the amount of direct sunlight, cloud cover and other atmospheric conditions all have direct effects on material surface temperatures and the at-aperture radiance. Atmospheric compensation is an entire field of study devoted to determining atmospheric effects so that the surface-leaving radiance can be estimated from the at-aperture signal. In spectral bands where thermal radiation is relevant, atmospheric correction is often followed by temperature-emissivity separation to extract unique pixel emissivity spectrum.

A simplified radiative transfer equation appropriate for describing the at-aperture radiance, $L(\lambda)$, in the LWIR is given by [2]

$$L(\lambda) = \tau(\lambda)\Big[\varepsilon(\lambda)B(\lambda,T) + \big[1-\varepsilon(\lambda)\big]L_d(\lambda)\Big] + L_a(\lambda), \tag{3.1}$$

where

λ : Wavelength

T : Material Temperature

$\tau(\lambda)$: Atmospheric Transmission

$\varepsilon(\lambda)$: Material Emissivity

$B(\lambda, T)$: Planckian Distribution

$L_d(\lambda)$: Downwelling Atmospheric Radiance

$L_a(\lambda)$: Atmospheric Path Radiance

Planck's blackbody distribution function is

$$B(\lambda, T) = \frac{2hc^2}{\lambda^5} \frac{1}{e^{hc/\lambda kT} - 1}, \tag{3.2}$$

where k is Boltzmann's constant, c is the speed of light and h is Planck's constant. Atmospheric compensation specifically estimates $\tau(\lambda)$, $L_a(\lambda)$ and $L_d(\lambda)$ such that the surface leaving radiance $L_s(\lambda)$ can be defined as:

$$\begin{aligned} L_s(\lambda) &= \frac{L(\lambda) - L_a(\lambda)}{\tau(\lambda)} \\ &= \varepsilon(\lambda) B(\lambda, T) + [1 - \varepsilon(\lambda)] L_d(\lambda). \end{aligned}$$

From Equation 3.1, it is apparent that training and testing on a single cube does not allow a classifier to learn relationships between $L_a(\lambda)$, $L_d(\lambda)$, or $\tau(\lambda)$ as these are constant for a single cube. Furthermore, omitting this diversity in training data reduces the classifier's ability to interpolate to new atmospheric scenarios.

Our work uses the same data collected and analyzed by Martin [112] where he showed a small neural network could outperform detection algorithms such as ACE when trained on multi-cube data encompassing changing atmospheric conditions. As atmospheric con-

ditions change throughout the day, so too will parameters in Equation 3.1, likely resulting in classification model degradation. Our work extends Martin's research to explore the effect of atmospheric change on classifier performance and compares several classification models' robustness against atmospheric change.

3.5 Methodology

Data was collected using a LWIR hyperspectral sensor at an angle of 68 degrees off-nadir and an approximate path length of 150 m [112]. Panels consisting of various materials were present in the scene to support several hyperspectral research projects. A subset of these panels were used in this work as shown by the labeled pixels in Figure 24. Sandpaper (36 and 320 grit), glass, tarp and canvas were placed in the scene in a vertical orientation and angled orientations. High Emissivty Low Reflectivity (HELR) and High Emissivity High Reflectivity (HEHR) targets were also placed in the scene to test the classifier's performance on two opposing reflectivity profiles.

Other materials are present in the scene, however, these are not considered for this work. Specifically, mixed pixel panels and the panels in the lower right portion and upper right portion of the image will not be used. Each scene is 486 by 1994 pixels.

Figure 24. Labeled pixels used for classification in each hyperspectral cube.

Table 6. Number of pixels used in for the training and test distributions.

Name	Training	Test
HELR	6000	7680
HEHR	6000	8400
36 Grit SP	6000	4860
320 Grit SP	6000	5140
Glass	6000	6840
Tarp	6000	7500
Canvas	6000	5380
Grass	6000	12500
Concrete	6000	13500
Total	54000	71800

Data collection was conducted across a 9 hour period, resulting in 26 hyperspectral cubes used in this analysis. In each cube, pixels were hand-labeled corresponding to the 9 different material types considered. Since the number of pixels per class was uneven, the total number of training samples per material was downsampled to 6000. Test data consisted of entirely separate cubes from the training data and the total number of samples are shown in Table 6. The spatial orientation of pixels within the scene are shown in Figure 24. The scene is partially shaded by nearby trees providing illumination diversity during the day-long collection. Since the path length is only 150 m and data was collected across only 9 hours, surface temperature is expected to account for most of the variation in the data over the samples.

A simple method was used to estimate an emissivity-like spectral quantity. First, In-Scene Atmospheric Compensation (ISAC) [15] was used to estimate transmittance and atmospheric path radiances and transform the at-aperture radiance into surface-leaving radiance $L_s(\lambda)$. The brightness temperature was then computed for each pixel according

to

$$T_B(\lambda) = \frac{hc}{\lambda k \ln\left(\frac{2hc^2}{\lambda^5 L_s(\lambda)} + 1\right)} \tag{3.3}$$

Finally, assuming a negligible $L_d(\lambda)$, a proxy for material emissivity, $\hat{\varepsilon}(\lambda)$, was computed via

$$\hat{\varepsilon}(\lambda) = \frac{L_s(\lambda)}{B(\lambda, T_{max})}, \tag{3.4}$$

where T_{max} corresponds to the pixel's maximum spectral brightness temperature. After transforming all at-aperture radiance values to $\hat{\varepsilon}$, the classification approaches will again be applied to compare performance. Applying ISAC to estimate material emissivity adds additional processing time. This additional time will be considered when evaluating classifiers.

This study examines the performance of different classification algorithms on multi-cube classification. Towards this goal, the training data partitioning is performed using two different approaches. The first approach, denoted Representative, creates training data representative of all 26 hyperspectral cubes. This is done by ensuring pixel surface temperatures in the test set are similar to the temperatures in the training set as shown in Figure 25. High classification accuracy is expected in this case since the training set represents the test set temperature variability.

The second approach, denoted Biased, examines classifier performance when test data surface temperatures are not contained in the training data set. This approach examines the robustness of classification algorithms when temperatures fall outside the training data distribution as shown by the temperatures in Figure 26. Specifically, HELR, HEHR, glass, grass, and concrete all contain test set temperatures outside the training set range. The two types of sandpaper have less training samples for temperatures between 310 and 325 K in the biased configuration leading to a more difficult modeling challenge.

After testing and comparing classification performance using the at-aperture radiance for both the representative and biased training sets, the biased set is converted to material emissivities. This is done by first applying the ISAC algorithm to estimate $\tau(\lambda)$ and $L_a(\lambda)$ from Equation 3.1 for each cube [15]. The total surface leaving radiance and material emissivity is estimated using Equation 3.4. This is only performed on the biased data set to test whether classification performance improves when using this data transformation. The representative training set case is not transformed to material emissivities because, as shown later, classification results are very high for all methods.

3.5.1 Classification Algorithms.

Three different classification techniques are considered in this study: SVMs, ANNs and 1D-CNNs. These methods have been extensively studied for hyperspectral classification, with SVM being the most common baseline approach. Only the spectral data of each pixel is used as input to each classifier with no information about neighboring pixels considered. The spectral data is normalized using the commonly applied z-score standardization, resulting in zero mean and unit standard deviation per spectral channel.

The SVM classifier investigated here utilizes a RBF kernel function described by

$$K(\mathbf{x}, \tilde{\mathbf{x}}) = e^{\left(-\gamma \|\mathbf{x}-\tilde{\mathbf{x}}\|^2\right)}, \tag{3.5}$$

where $\gamma = 1/2\sigma^2$ and $\mathbf{x}$, $\tilde{\mathbf{x}}$ represent two input samples. To train an SVM classifier, σ must be adjusted to determine an optimal Gaussian function for projection. Additionally, a complexity parameter must also be adjusted to allow for some misclassification leading to a more generalized classifier [120]. Successive, finer resolution grid searches are performed using validation metrics to determine these parameters.

The ANN used in this study consists of a single hidden layer with 20 neurons. Initially, much deeper networks with more parameters were considered. This configuration was

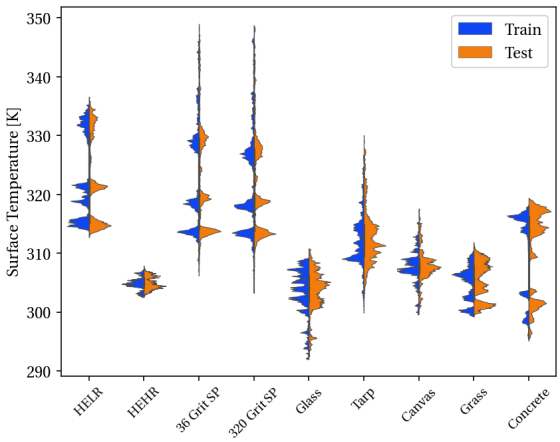


Figure 25. Representative Training Samples

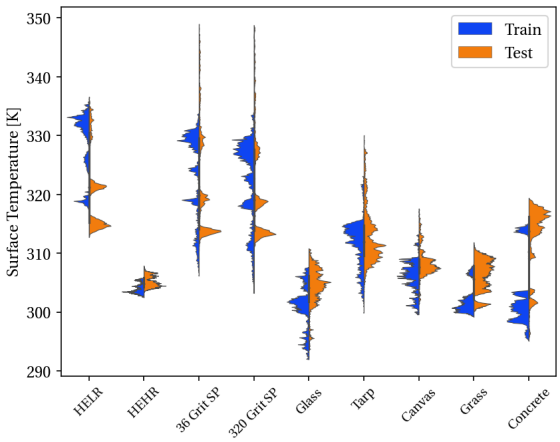


Figure 26. Biased Training Samples

selected after repeatedly pruning unnecessary connections while monitoring validation accuracy. Specifically, training a new model with 19 hidden neurons resulted in decreased validation accuracy while larger networks had nearly identical validation performance. All hidden layer neurons utilize the Rectified Linear Unit (ReLU) activation function which can be described as

$$\mathrm{ReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x < 0 \end{cases} \tag{3.6}$$

where α is a hyperparameter allowing additional information to flow through the network when negative inputs are provided to the function. This activation function is also known as a Leaky ReLU because of this feature.

The ANN output layer utilizes a softmax activation function to create class probabilities for each of the 9 materials labeled in the training data. The overall network structure consists of 20 hidden nodes and 9 output nodes. The maximum output probability represents the network's prediction for a particular sample. Ideally, network predictions are heavily weighted toward the correct class with minimum probabilities assigned to all other classes.

The 1D-CNN consists of two convolutional layers followed by a fully connected layer and an output layer for material prediction. The architecture used here does not contain

max-pooling layers as these were found to significantly reduce performance. Instead, a strided convolutional layer was used to reduce the number of network parameters and was motivated by the results presented by Springenberg et al [121]. The entire network structure is shown in Table 7 where 1DConv50-4 refers to a 1D-CNN layer consisting of 4 different kernels, each of width 50. The column names refer to the type of layer where C refers to a CNN layer, D_1 refers to a dropout layer and FC refers to a fully-connected layer. The dropout layer was introduced to improve network generalization after confirming the model had adequate capacity to overfit the training data. The hyperbolic tangent activation function was used for all nodes in the 1D-CNN.

The model shown in Table 7 contains 3,328 weights to be trained. This same architecture is evaluated for both the Representative data partition and the Biased partition. Similarly, the ANN model consists of 4,950 parameters and will also be evaluated on both partitions. The ANN model has slightly more parameters, however, reducing the number of nodes in the single hidden layer any further resulted in significantly worse performance.

Table 7. 1D-CNN Network Architecture

C_1	C_2	D_1	FC	Output
1DConv50-4 Stride: 1	1DConv40-4 Stride: 4	0.5	9	9

3.5.2 Classification Metrics.

Classifier performance is evaluated using several well-known metrics. First, the overall and average accuracy on the validation set is measured during each hyperparameter adjustment in the training phase. This allows for a quick overview of classifier performance to determine if the optimization step was correct. The overall accuracy (OA) is the ratio of the number of predicted correct labels to the total number of samples. This metric does not consider individual class accuracy. Average accuracy (AA) is the average of the individ-

ual class accuracies and can highlight instances when a classifier is performing poorly on particular classes.

The test accuracy will be biased since the classes are unbalanced in the validation and test sets. To remedy this, the Kappa score is used which considers the number of samples per class to compare algorithm performance. The Kappa score is

$$k = \frac{p_o - p_e}{1 - p_e}$$

where p_o is the class accuracy ratio and p_e is the expected accuracy ratio given random labeling. Additionally, the maximum F1 score is also calculated:

$$\text{F1} = 2\frac{P_D}{1 + P_D + P_{FA}}$$

where P_D is the probability of detection and P_{FA} is the probability of false-alarm. The F1 score represents the average between classifier precision and recall. Since neural network approaches rely on a random weight initialization, both the Convolutional Neural Network (CNN) and ANN models are repeatedly trained from different weight initializations. Results reported using these method will include a standard deviation, denoted as $\pm$. The SVM classifier does not rely on a random weight initialization and so no standard deviation results are reported.

In the next section results are presented first for the training data partition encompassing all surface temperature variability. This is followed by results for the biased training data partitioning where the test set contains samples with surface temperature outside the training data distribution. The three classification algorithms are compared for both configurations and strengths and weaknesses between the methods are identified.

3.6 Results

The Representative training partition is expected to result in higher classification performance since this is the ideal case where the collected training data matches all possible test conditions. The expectation of being able to collect data under a large enough set of atmospheric conditions is a difficult assumption to satisfy when parameters such as atmospheric transmittance, upwelling and downwelling radiance play a significant role in the at-aperture signal. The Biased training data represents this case when training samples do not span the test data distribution.

As shown in Table 8 for the Representative training column, all three classification algorithms achieve over 90% average classification accuracy. The ANN and 1D-CNN have nearly perfect classification across all classes. The kernels used in the 1D-CNN are intentionally very large to capture interband dependencies across portions of the spectrum. The ANN is able to simultaneously view all interband dependencies at once, and this likely makes the classification problem more difficult since some spectral bands are more sensitive to changing atmospheric conditions than others.

The Biased training partition represents a more realistic situation in which the training data does not fully span all possible test scenarios. All three classification techniques showed degraded performance when applied to the Biased partition. The SVM classifier is unable to generalize beyond the training distribution resulting in significantly lower classification accuracy for all materials. The ANN maintains high classification accuracy for concrete, canvas and tarp materials. The tarp and canvas test sets are generally not biased since the training set nearly spans the test set surface temperature range. Compared to the other materials in the scene, concrete has a unique spectral profile making this material easy to identify for this scenario. The 1D-CNN shows the best classification performance on the biased data partition with maximum classification accuracy for nearly all materials. High emissivity materials such as grass show a significant classification improvement over

the ANN results. The at-aperture radiance for these materials is quite sensitive to the surface temperature, resulting in highly variant spectral profiles across cubes. The 1D-CNN kernels appear less sensitive to surface temperature and create informative feature maps for material classification.

The 1D-CNN has the highest F1 score on 6 out of 9 classes as shown in Table 9. The lowest F1 scores are on the two types of sandpaper. This was unsurprising since these materials have similar spectral profiles. Next, the at-aperture radiance values are converted to material emissivity to determine if this transformation improves classifier performance in the biased partition scenario.

Table 8. Classification Accuracy for the data partitions shown in Figures 25 and 26. The bold font represents the maximum classification accuracy per material for the Representative and Biased data partitions.

	Representative Training			Biased Training		
Material	SVM	ANN	1D-CNN	SVM	ANN	1D-CNN
HELR	87.19	99.21 ± 0.39	$\mathbf{99.90 \pm 0.01}$	23.54	85.31 ± 6.37	$\mathbf{86.52 \pm 2.79}$
HEHR	99.76	99.66 ± 0.12	$\mathbf{99.91 \pm 0.01}$	64.34	80.82 ± 6.41	$\mathbf{99.07 \pm 0.59}$
36 Grit SP	84.01	$\mathbf{95.80 \pm 0.35}$	92.17 ± 0.28	63.42	$\mathbf{83.25 \pm 2.39}$	61.81 ± 12.48
320 Grit SP	85.74	$\mathbf{94.87 \pm 0.91}$	88.30 ± 0.22	47.03	57.87 ± 3.79	$\mathbf{75.87 \pm 10.70}$
Glass	94.58	95.95 ± 0.52	$\mathbf{98.27 \pm 0.001}$	84.50	88.31 ± 4.93	$\mathbf{96.70 \pm 0.96}$
Tarp	95.85	99.34 ± 0.08	$\mathbf{99.57 \pm 0.001}$	91.95	$\mathbf{99.02 \pm 0.07}$	97.19 ± 1.04
Canvas	98.09	98.82 ± 0.65	$\mathbf{99.90 \pm 0.001}$	79.44	94.24 ± 0.07	$\mathbf{98.70 \pm 0.73}$
Grass	95.51	98.32 ± 0.35	$\mathbf{99.95 \pm 0.001}$	42.24	76.19 ± 3.70	$\mathbf{88.70 \pm 1.64}$
Concrete	98.85	99.99 ± 0.001	$\mathbf{99.99 \pm 0.001}$	57.75	99.96 ± 0.03	$\mathbf{99.99 \pm 0.001}$
OA	89.70	97.15 ± 0.22	$\mathbf{98.03 \pm 0.01}$	59.63	76.87 ± 2.18	$\mathbf{87.32 \pm 1.98}$
AA	92.05	$\mathbf{97.68 \pm 0.21}$	97.44 ± 0.01	61.40	82.43 ± 1.90	$\mathbf{88.41 \pm 0.94}$
Kappa	91.16	$\mathbf{97.42 \pm 0.23}$	97.16 ± 0.01	57.11	80.48 ± 2.10	$\mathbf{87.12 \pm 1.04}$

Having verified that the ANN and 1D-CNN models have an adequate number of parameters to correctly classify the data, we then employed the same architectures on the emissivity proxy $\hat{\varepsilon}$ computed for the biased set. The classification accuracies are shown in Table 10 where all classifiers show significant performance improvements over the results

Table 9. Maximum F1 Score for Biased Training Data Configuration. The bold font represents the maximum score per material.

Material	SVM	ANN	1D-CNN
HELR	0.27	0.68	**0.75**
HEHR	0.59	0.87	**0.94**
36 Grit SP	0.47	0.45	**0.55**
320 Grit SP	0.38	**0.63**	0.60
Glass	0.88	0.85	**0.94**
Tarp	0.56	0.90	**0.98**
Canvas	0.82	**0.91**	0.89
Grass	0.45	0.85	**0.92**
Concrete	0.71	**1.00**	0.98

shown in Table 8 for the biased data. The ANN model does contain 1,622 more parameters than the 1D-CNN model and results in the highest OA, AA and Kappa performance. Based on the high classification performance using $\hat{\epsilon}$ rather than at-aperture radiance, the ANN and 1D-CNN models can likely be further pruned leading to faster inference times.

The results shown in Table 10 demonstrate that an SVM or very small neural network can be used to achieve high classification, however, these results don't consider the total inference time on new data. Transforming at-aperture radiance to emissivity is a time-consuming operation which may be unacceptable to real-time, remote sensing scenarios. Each classification method was tested on a single hyperspectral cube to compare inference time. The timing includes all data cube preprocessing such as z-score standardization and required array reshaping for matrix multiplications. The inference times reported in Table 11 are average times and were conducted using an Intel Xeon E5-2660 and Nvidia Titan V graphics card for an entire 486 by 1,994 pixel scene.

The highest accuracy model when considering the biased radiance data was the 1D-CNN, but after converting to emissivity space the ANN showed the best performance. If near real-time classification is necessary, the 1D-CNN model coupled with the at-aperture

Table 10. Biased Data Partition Classification Accuracy using Pixel Emissivity

Material	SVM	ANN	1D-CNN
HELR	71.19	73.50 ± 2.44	$\mathbf{73.54 \pm 3.88}$
HEHR	**99.02**	96.75 ± 0.15	97.98 ± 0.46
36 Grit SP	**83.65**	80.79 ± 2.38	72.94 ± 5.36
320 Grit SP	72.18	87.92 ± 3.48	$\mathbf{89.22 \pm 1.52}$
Glass	97.55	$\mathbf{99.50 \pm 0.25}$	99.24 ± 0.19
Tarp	**98.32**	95.58 ± 0.32	93.79 ± 0.55
Canvas	99.69	$\mathbf{99.92 \pm 0.04}$	98.90 ± 1.12
Grass	**99.67**	99.56 ± 0.01	98.50 ± 0.47
Concrete	**100.0**	99.99 ± 0.01	99.99 ± 0.01
OA	92.35	$\mathbf{94.09 \pm 0.21}$	93.23 ± 0.41
AA	91.20	$\mathbf{92.77 \pm 0.42}$	91.73 ± 0.71
Kappa	90.22	$\mathbf{91.97 \pm 0.46}$	90.81 ± 0.79

radiance values, $L(\lambda)$, is the best choice since emissivity calculations are not necessary to achieve high classification accuracy. For applications which may not require real-time classification and accurate atmospheric estimates can be made, the ANN model using emissivity data is the best choice. The data used in this analysis was collected across a short path length leading to minimal atmospheric effects. Longer path lengths, such as those encountered with air- and space-based platforms, will further distort the at-aperture signal requiring highly-accurate estimates of the atmosphere to estimate material emissivity.

Table 11. Total inference time for one hyperspectral cube

	$L(\lambda)$			$\varepsilon(\lambda)$		
	SVM	ANN	1D-CNN	SVM	ANN	1D-CNN
Inference Time (s)	239.78	27.19	39.49	303.54	80.82	98.91

3.7 Conclusions and Future Work

This study has compared the performance of three hyperspectral classification algorithms on data collected across an entire day. Different from other studies, this paper has highlighted how these methods perform when tested on new hyperspectral data independent from the training data cubes. Training and testing on independent cubes showed the ability of classification methods to adapt to changing atmospheric conditions and surface temperatures. Additionally, the training and testing data sets were intentionally partitioned such that the test data contained surface temperatures not encountered in the training set. All methods performed well when the training data was representative of the test data, but the 1D-CNN showed the best generalization results when surface temperatures extended beyond the training data range.

The training and test sets were also converted to pixel emissivity to train the three classification methods. All methods showed nearly perfect classification performance when using the proxy for emissivity; however, the short path length likely introduced minimal atmospheric effects on the at-aperture signal.

Total inference time for each technique was also investigated. While classification performance can be improved when using pixel emissivity estimates, this comes with additional computational costs and time. The 1D-CNN using at-aperture radiance values allows for very fast pixel classification without significant reductions in accuracy.

It is unclear how these classifiers will behave when applied to space-based imagery across widely varying atmospheric conditions. Future work in this area will focus on the atmospheric parameter estimation problem for space-based platforms. Classification algorithms will again be trained on a subset of all possible conditions. Moving towards real-time classification, in-scene atmospheric compensation approaches will be explored to maximize classifier speed.

The high accuracy results using $\hat{\varepsilon}$ are dependent on the atmospheric compensation approach. Future work in this area will investigate the use of generative models, such as those presented by Kingma *et al* [52], to create highly-accurate and efficient atmospheric estimates, allowing for real-time classification with improved performance.

IV. Fast and Effective Techniques for LWIR Radiative Transfer Modeling: A Dimension Reduction Approach

4.1 Paper Overview

This paper investigated Autoencoder (AE) networks for compressing Transmittance, Upwelling, and Downwelling (TUD) vectors resulting in faster radiative transfer calculations to support a wide range of remote sensing applications. A new loss function was introduced, dependent on the underlying Long-Wave Infrared (LWIR) radiative transfer equation. This loss function outperformed standard mean-squared error for minimizing at-sensor radiance error. Additionally, a sampling network was developed to determine the low-dimensional components necessary for making radiative transfer model predictions. After fitting the complete radiative transfer model, the network was optimized from the output to the input allowing for estimates of atmospheric state to be made based on a known TUD vector.

This paper was published in the Journal of Remote Sensing on 9 August, 2019.

4.2 Abstract

The increasing spatial and spectral resolution of hyperspectral imagers yields detailed spectroscopy measurements from both space-based and airborne platforms. These detailed measurements allow for material classification, with many recent advancements from the fields of machine learning and deep learning. In many scenarios, the hyperspectral image must first be corrected or compensated for atmospheric effects. Radiative Transfer (RT) computations can provide look up tables (LUTs) to support these corrections. This research investigates a dimension reduction approach using machine learning methods to create an effective sensor-specific LWIR RT model. The utility of this approach is investigated emulating the Mako LWIR hyperspectral sensor ($\Delta\lambda \simeq 0.044\,\mu\text{m}$, $\Delta\tilde{\nu} \simeq 3.9\,\text{cm}^{-1}$).

This study employs physics-based metrics and loss functions to identify promising dimension reduction techniques and reduce at-sensor radiance reconstruction error. The derived RT model shows an overall root mean square error (RMSE) of less than 1 K across reflective to emissive grey body emissivity profiles.

4.3 Introduction

Next generation hyperspectral imagers continue to improve in both spatial and spectral resolution with increasingly lower noise-equivalent spectral radiance (NESR) values, presenting unique opportunities in efficiently characterizing pixel materials [122]. A pixel in a hyperspectral image can be represented as a vector across all spectral channels, producing a three dimensional data cube for an entire image, width by height by spectral channel [22]. Hyperspectral imagers have been deployed in both airborne and space-based platforms with uses ranging from precision agriculture to search and rescue operations [2]. The spectral bands making up a hyperspectral cube can span from the visible to the LWIR, sampled across hundreds of narrow spectral channels [21]. The visible to shortwave infrared (SWIR) (0.4 - 3.0 µm) is dominated by scattering while the LWIR (5.0 - 14.0 µm) is dominated by material emission [13]. The atmospheric state — which includes the altitude-dependent temperature and pressure; how column water vapor content, carbon dioxide, ozone, and other trace gases are distributed vertically; the kind and size distributions of various aerosols — has a significant impact on the at-sensor radiance. Understanding and accounting for these atmospheric effects is critical for quantitative exploitation of hyperspectral imagery, especially in the domain of material identification.

RT calculations convert the atmospheric state parameters — temperature, water, and ozone values as functions of altitude or pressure — into spectral radiances observed at the sensor by discretizing the atmosphere into thin, homogeneous layers. At each layer, high spectral resolution RT calculations (*e.g.*, LBLRTM) are performed, or approxima-

tions thereof (*e.g.*, MODTRAN). Due to the large number of discrete absorption lines of the many trace gases in the atmosphere, millions of calculations are required to model a sensor's entire spectral range with high fidelity [123]. This computational complexity is the primary bottleneck in remote sensing retrieval problems, often limiting the use of RT models in real-time data analysis.

To avoid the high computational cost of line-by-line RT calculations, approximate RT models are used to increase computational speed while trading off accuracy [124]. One of the most widely used approaches to improve RT computation time is the correlated-k method, which divides opaque spectrum into a subset of b bands and then applies a weighting k to these bands, dependent on the opacity distribution of the b bands [125].

Similar to the weighting scheme employed in the correlated-k method, Principal Component Analysis (PCA) has also been implemented to reduce RT computation time [123]. PCA can be applied on the input space (atmospheric state parameters) and/or on the output space (spectral radiances) to reduce RT computational time. In [126], PCA was applied to atmospheric state parameters for quickly estimating spectra in the O_2 A-band with an error of 0.3% compared to multi-stream methods with a 10-fold reduction in computation time. In [123], PCA was applied to a database spectral radiances identifying a lower dimensional space of only a few hundred components compared to the thousands of dimensions in the original data space. Implementations such as principal component radiative transfer model (PCRTM) [123] or principal component radiative transfer for TOVs (PCRTTOV) [127] perform RT computations for a subset of bands and map these to the low-dimensional space to create the highly-efficient RT model. In [128], PCA was considered on both the input atmospheric parameter space and the output spectral radiance space to further reduce computational time with an overall error of 0.05%.

This study focuses on efficient conversion of atmospheric state parameters into spectral radiances for the LWIR domain using neural network approaches. This is achieved by

performing dimension reduction on the output spectral radiance space (TUD vectors) and then fitting a neural network to sample the low-dimensional space. Our approach is similar to PCRTM [123], however, we utilize autoencoder networks for the dimension reduction step instead of PCA.

The most salient contributions and findings of this research include:

- Employing machine learning techniques which: (1) are computationally faster than correlated-k calculation methods; (2) reduce the dimension of both the TUD and atmospheric state vectors; (3) produce the desirable latent-space-similarity property such that small deviations in the low-dimension latent space result in small deviations in the high-dimension TUD
- Developing a data augmentation method using PCA and Gaussian Mixture Models (GMMs) on real atmospheric measurements that lead to improved model training and generalizability
- Improving machine learning model training by introducing a physics-based loss function which encourages better fit models than traditional loss functions based on mean squared error
- Demonstrating an effective AE pre-training strategy that leverages the local-similarity properties of the latent space to reproduce TUDs from atmospheric state vectors

Together, these contributions form the basis of a novel method for efficient and effective RT modeling, using a small number of parameters.

4.3.1 Background.

Atmospheric compensation techniques estimate the atmospheric effects imposed on the at-sensor signal, leading to atmospherically-corrected data for material classification and

identification. In the LWIR, the simplest RT model for describing the at-sensor radiance, $L(\lambda)$, from a diffuse (lambertian) thermal emitter and scatterer, can be expressed as [2]:

$$L(\lambda) = \tau(\lambda)\Big[\varepsilon(\lambda)B(\lambda,T) + \big[1-\varepsilon(\lambda)\big]L_d(\lambda)\Big] + L_a(\lambda), \tag{4.1}$$

where

λ : wavelength

T : material temperature

$\tau(\lambda)$: atmospheric transmission

$\varepsilon(\lambda)$: material emissivity

$B(\lambda,T)$: Planckian distribution

$L_d(\lambda)$: downwelling atmospheric radiance

$L_a(\lambda)$: atmospheric path (upwelling) radiance

Both τ and L_a are specific to the line of sight between the sensor and the surface, whereas L_d represents a cosine-weighted average of the downwelling radiance for the hemisphere above the surface. Planck's blackbody distribution function, $B(\lambda,T)$, is given by

$$B(\lambda,T) = \frac{2hc^2}{\lambda^5}\frac{1}{e^{hc/\lambda kT}-1}, \tag{4.2}$$

where k is Boltzmann's constant, c is the speed of light and h is Planck's constant. Atmospheric compensation recovers the surface leaving radiance $L_s(\lambda)$ by estimating $\tau(\lambda)$ and $L_a(\lambda)$ in Equation 4.1 as shown in Equation 4.3.

$$L_s(\lambda) = \varepsilon(\lambda)B(\lambda,T) + [1-\varepsilon(\lambda)]\,L_d(\lambda). \tag{4.3}$$

One of the most popular LWIR atmospheric compensation techniques is the In-Scene Atmospheric Compensation (ISAC) method which first identifies blackbody pixels within the scene to estimate $\tau(\lambda)$ and $L_a(\lambda)$ [15]. By using only pixel spectra from blackbodies, the surface leaving radiance is equivalent to Planck's blackbody distribution and the simplified LWIR at-sensor radiance can be expressed as

$$L(\lambda)_{\varepsilon(\lambda)\to 1} = \tau(\lambda)B(\lambda,T) + L_a(\lambda). \tag{4.4}$$

Under the assumption that distinct blackbody pixels can be identified and their temperatures known, a linear fit can be performed across all spectral channels to identify $\tau(\lambda)$ and $L_a(\lambda)$. In practice, temperature estimates are made in the most transmissive part of the at-sensor spectral radiance, but they are often systematically biased since $\tau(\tilde{\lambda})$ and $L_a(\tilde{\lambda})$ are unknown and assumed to be 1 and 0, respectively, for that particular spectral channel $\tilde{\lambda}$. A common method to remove the biases introduced into τ and L_a by inaccurate surface temperatures relies on spectral analysis near the isolated water absorption feature near 11.73 µm. This method is very similar to the Autonomous Atmospheric Compensation (AAC) method, which estimates a transmittance ratio and an upwelling radiance parameter derived from the off- and on-resonance spectral values at the same isolated water band [129]. By assessing this water feature, both transmittance and path radiance contributions can be independently estimated, allowing the biased ISAC estimates of τ and L_a to be fixed, or under further assumptions about the atmosphere, allowing full estimates of τ and L_a to be made. To ensure algorithmic efficiency for both ISAC and AAC, precomputed look-up tables of $\tau(\lambda)$ and $L_a(\lambda)$ are forward modeled with a RT model over a wide range of possible atmospheric water and temperature profiles. In the LWIR portion of the spectrum, Temperature-Emissivity Separation (TES) follows atmospheric compensation to estimate material emissivity and surface temperature from $L_s(\lambda)$ [130].

In this study, we conduct dimension reduction on the TUD vectors $(\tau(\lambda), L_a(\lambda), L_d(\lambda))$ in Equation 4.1 which span a wide range of global atmospheric variability. Specifically, this research uses the Thermodynamic Initial Guess Retrieval (TIGR) database comprising a myriad of atmospheric conditions in the form of temperature, water vapor, and ozone profiles on a fixed pressure grid. The 2311 atmospheric profiles provided in the TIGR database are based on 80,000 radiosonde measurements collected worldwide [30, 31]. The TIGR atmospheric profiles are first filtered for cloud-free conditions and then forward modeled using the Line-by-Line Radiative Transfer Model (LBLRTM) (version 12.8) to create realistic TUD vectors which also span a broad range of atmospheric conditions.

Conducting dimension reduction on the TIGR-derived TUD vectors creates a low-dimensional representation that can be sampled to create new TUD vectors without the need of costly RT calculations. Research performed in [131] specifically considered a low-rank subspace of $\tau(\lambda)$ and $L_a(\lambda)$ for atmospheric compensation in the LWIR spectrum. They performed a singular value decomposition on representative $\tau(\lambda)$ and $L_a(\lambda)$ vectors generated by MODTRAN for a given seasonal model and flight altitude. Blackbody pixels were identified within a scene based on their projection onto these subspaces, thus providing a way to directly estimate transmittance and upwelling radiance for a scene. The neural network approach we take for TUD vector compression is not invertible, therefore we cannot directly apply the approach outlined in [131] for atmospheric compensation. The RT model can assist the atmospheric compensation in [131], by quickly providing a wide range of transmittance and upwelling vectors to construct the low-rank subspaces.

Creating a low-dimensional TUD representation is also important for data augmentation applications, distinctly different from atmospheric compensation. To identify a material of interest, many augmented representations of that material through diverse atmospheric conditions can be created using a low-dimensional TUD representation. By providing many of the commonly investigated classification techniques augmented at-sensor data represen-

tative of diverse and realistic TUD vectors, atmospherically-robust classification can be improved. This was the approach employed in [112], where a small neural network was trained to detect specific materials in the LWIR across varying atmospheric conditions. The neural network-based approaches investigated here offer a highly efficient method to generate realistic TUD vectors to support data augmentation.

Additionally, the utility of the low-dimensional representation is explored by mapping the atmospheric state (temperature, water vapor and ozone profiles) to the low-dimensional space, thus creating a highly-efficient RT model. This RT model can be used for data augmentation as discussed above or to support model-based compensation techniques where hundreds of possible transmittance and upwelling vectors can be computed in real-time, avoiding the use of precomputed look-up tables. By performing dimension reduction prior to fitting this mapping, similar atmospheric conditions cluster together in the low-dimensional space. Additionally, based on this clustering, small deviations in the low-dimensional space correspond to small changes in generated TUD vectors, further improving the mapping from atmospheric measurements to the low-dimensional space.

In the next section, dimension reduction techniques are reviewed and the TIGR dataset is explained in further detail. Metrics based on Equation 4.1 are also derived to ensure dimension reduction techniques are correctly evaluated. Sampling of the low-dimensional TUD representation is also outlined to demonstrate the utility of these techniques and highlight the importance of latent-space-similarity where deviations in the latent space correspond to similar deviations in high-dimension TUD space.

Following the methodology section, results are presented comparing dimension reduction performance derived from the TIGR data and an augmented version of the TIGR data. After confirming improved performance with the augmented data, a novel physics-based loss function is compared to Mean-Square Error (MSE) to further improve dimension reduction reconstruction error. Finally, a RT model is formulated from the dimension re-

duction algorithms, showing the importance of the dimension reduction pre-training step toward reduced TUD prediction error.

4.4 Methodology

This section first reviews the atmospheric measurement data and corresponding forward modeled TUD vectors used for dimension reduction. Metrics for comparing the performance of each technique are also reviewed, with a focus on incorporating properties from the simplified RT model in Equation 4.1. A unique data augmentation scheme is also discussed to increase the number of TUD samples for model fitting.

4.4.1 Data.

The TIGR database consists of 2311 atmospheres selected from over 80,000 worldwide radiosonde reports. These atmospheric conditions represent a broad range of conditions favorable for capturing atmospheric variations in remotely sensed data. Each sample contains temperature, water content and ozone at 43 discrete pressure levels ranging from the Earth's surface (1013 hPa) to $>$ 30 km ($<$1 hPa) [30, 31]. Cubic interpolation was used to upsample these profiles to 66 pressure levels, with finer sampling in the lower, most dense part of the atmosphere. Additionally, the profiles are grouped by air mass category such as polar, tropical and mid-latitude. The entire TIGR data matrix shape is 2311 atmospheric profiles by 198 measurements, where the 66 pressure level measurements for temperature, water content and ozone are concatenated.

By using atmospheric profiles that span nearly all expected atmospheric variability, RT can be conducted to generate TUD vectors encapsulating nearly every possible atmospheric scenario. The LBLRTM was used to create high resolution TUD vectors, however, the spectral resolution must be downsampled for a particular sensor to ensure the dimension reduction techniques are applicable to real-world sensor resolutions.

The LWIR Mako hyperspectral sensor is a high-performance, airborne sensor imaging across $7.8-13.4\,\mu m$ into 128 spectral channels with a noise-equivalent temperature difference of 0.02 K at $10\,\mu m$ and 300 K [122, 132]. The high-resolution LBLRTM generated TUD vectors (11,513 spectral channels) are downsampled according to the Mako instrument line shape creating representative TUD vectors for this sensor. Additionally, the TUD vectors are generated to represent a sensor altitude of 3.3 km. The result of this process is shown in Figure 27, where after downsampling, the TUD data matrix shape is 2311 samples by 384 spectral measurements ($\tau(\lambda), L_a(\lambda), L_d(\lambda)$ concatenated). The goal of the dimension reduction algorithms discussed next is to project the length 384 TUD vectors to a lower dimensional space such that reconstruction error is minimized.

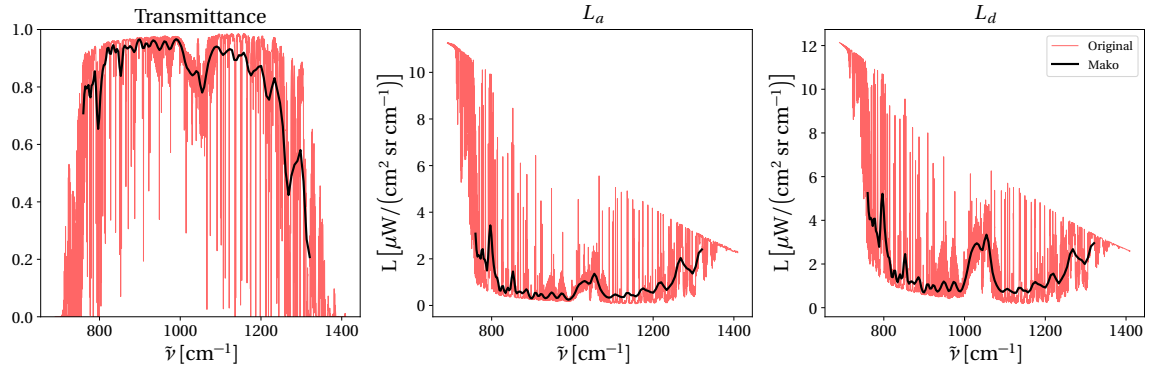


Figure 27. The high-resolution LBLRTM transmittance, upwelling and downwelling vectors are shown with their downsampled counterparts for the Mako LWIR sensor. The downsampled vectors are the data used in the remainder of this study.

4.4.2 TUD Dimension Reduction Techniques.

PCA removes correlation from data by projecting it onto a new coordinate system which maximizes data variance. Let $\mathbf{x}_i$ be a single measurement with P features and $\mathbf{X}$ be the data matrix containing N measurements such that $\mathbf{X} \in \mathbb{R}^{N\times P}$. To apply PCA to data matrix $\mathbf{X}$, first an eigendecomposition is performed on the data covariance matrix $\mathbf{C}_x$ such that

$$\mathbf{C}_x = \mathbf{A}\mathbf{D}\mathbf{A}^T \tag{4.5}$$

where the matrix $\mathbf{A}$ is an orthonormal square matrix consisting of P eigenvectors and $\mathbf{D}$ is a diagonal matrix consisting of the corresponding eigenvalues [46]. The eigenvalues are sorted in descending order and the L eigenvectors corresponding to the largest eigenvalues are kept, where $L << P$. This selection is based on the cumulative sum of data variance explained with the eigenvectors. The smallest eigenvalue components are considered noise and have little impact on reconstructing the original signal. In this study, we will vary the number of L components to minimize reconstruction error. The subset of selected eigenvectors are used to linearly transform the data to the low-dimensional representation $\mathbf{y} \in \mathbb{R}^L$ by:

$$\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i.$$

An AE is a neural network designed for performing nonlinear compression by projecting data to a low-dimensional latent space, followed by nonlinear reconstruction from the latent space. An AE is composed of two networks to perform this operation: an encoder network and a decoder network. The encoder compresses the input data, $\mathbf{x}$, into a lower dimensional latent space, $\mathbf{z}$, and the decoder reconstructs the data based on the latent space mapping into $\mathbf{y}$ [47]. Equations for these two transformations are

$$\begin{aligned} \mathbf{z} &= f(\mathbf{W}_z \mathbf{x} + b_z) \\ \mathbf{y} &= f(\mathbf{W}_y \mathbf{z} + b_y) \end{aligned} \tag{4.6}$$

where $\mathbf{x} \in \mathbb{R}^d$ is the input data, $\mathbf{z} \in \mathbb{R}^l$ is the latent space representation with $l \ll d$. The reconstructed data is $\mathbf{y} \in R^d$ and b_y and b_z are the biases of the hidden and output layer layers respectively. $\mathbf{W}_z$ and $\mathbf{W}_y$ are the weight matrices from the input to hidden layer and hidden layer to output layer, respectively. An AE diagram is shown in Figure 4.6 specifically for TUD compression and decompression using the encoder, decoder and latent

space nomenclature. This figure is only notional, and does not represent the number of nodes actually used in this study.

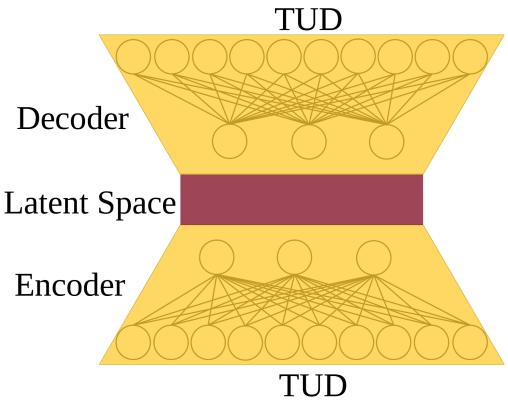


Figure 28. An example AE model where the TUD vectors are compressed through one or more encoder layers to a low-dimensional latent space. The decoder transforms the low-dimensional latent space back to the original TUD vector.

The AE predicted TUD vectors are compared to the LBLRTM generated TUD vectors through a loss function to determine model performance and update the weight matrices. The loss function used for measuring reconstruction error between TUD vectors is an important design variable influencing how the AE structures the latent space and ultimately what the network understands about TUD reconstruction. A commonly used loss function is MSE, calculated according to:

$$\text{MSE} = \frac{1}{K}\sum_{i=1}^{K}(\mathbf{x}_i - \mathbf{y}_i)^2 \tag{4.7}$$

where K equals the number of dimensions in the TUD vector, and $\mathbf{x}_i$ and $\mathbf{y}_i$ are the predicted and truth TUD vectors respectively. MSE will be used in this study, but an additional loss function will be derived later based on the underlying LWIR RT model. In many applications, a series of hidden layers are used to create a latent space representation of the data. This architecture is commonly referred to as a Stacked Autoencoder (SAE), where the functions shown in Equation 4.6 are nested to include additional layers. The activation

function, f, can be linear or non-linear. A comprehensive search of activation functions found the Rectified Linear Unit (ReLU) [133] yielded the best results over functions such as hyperbolic tangent, sigmoid, exponential linear units, and scaled exponential linear units. The ReLU function used in this study is

$$\text{RELU}(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha x, & \text{if } x \leq 0 \end{cases}$$

where α controls how much information is passed through the network for negative inputs. This small slope increases information flow during backpropagation allowing more weights to be influenced by training samples [134].

The number of latent components is an important design parameter controlling model complexity and reconstruction performance. Using two latent components allows for visualizations of the latent space by overlaying measurement parameters such as surface temperature and total atmospheric water vapor content. For the PCA model, the first two components capture 99.50% of the data variance. Plotting just these two components shows a smoothly varying relationship between the components and these physical parameters as shown in Figure 29. Both the validation set (158 samples) and test set (176 samples) are plotted to highlight this underlying dependence on atmospheric conditions.

Similar plots are shown in Figure 30 when considering a 2 component SAE model. Interestingly, the SAE disperses the validation and test set points throughout the latent space which is beneficial for sampling the low dimensional representation. Small changes in latent space components should result in small changes in the generated TUD vectors. This latent-space-similarity is an important property when fitting a sampling method to correctly identify a small number of components to generate a TUD vector. Small sampling errors should not result in large TUD deviations. Since it's difficult to visualize higher

dimension latent spaces, this property can be observed by fitting a small neural network to correctly predict the latent components for a known TUD.

The PCA model has little change in components 1 and 2 for cold, dry atmospheric conditions as shown by the tight clustering of these points. There appears to be a stronger dependence on component 2 for cold, dry atmospheres, while hot, humid conditions are dependent on both components. The SAE components are both influenced by surface temperature, while component 2 appears more dependent total water vapor content. Based on the preliminary results shown in Figures 29 and 30, sampling the two-dimensional AE latent space will result in lower reconstruction error because of the tight clustering of cold dry atmospheric conditions shown in the PCA latent space. Using only two components results in large reconstruction error, therefore, we will consider additional latent components to cluster similar atmospheric conditions together in a low-dimensional space while minimizing reconstruction error.

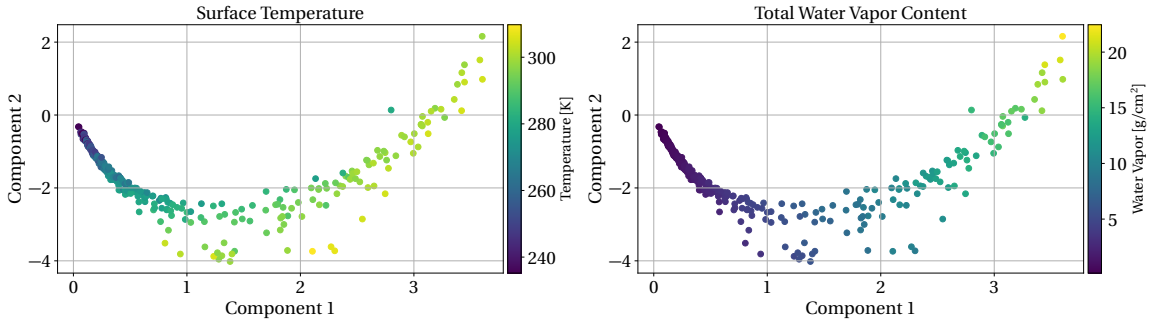


Figure 29. The first two principal components using a 6 component PCA model with the augmented TIGR data. Hot, humid atmospheric conditions vary with component 1 and 2 while cold, dry atmospheres are more dependent on component 2

4.4.3 Metrics.

For all methods considered, the reconstruction error must be placed in context of the at-sensor radiance to provide meaningful reconstruction performance. At-sensor radiance errors are dependent on the material emissivity as shown in Equation 4.4 where down-

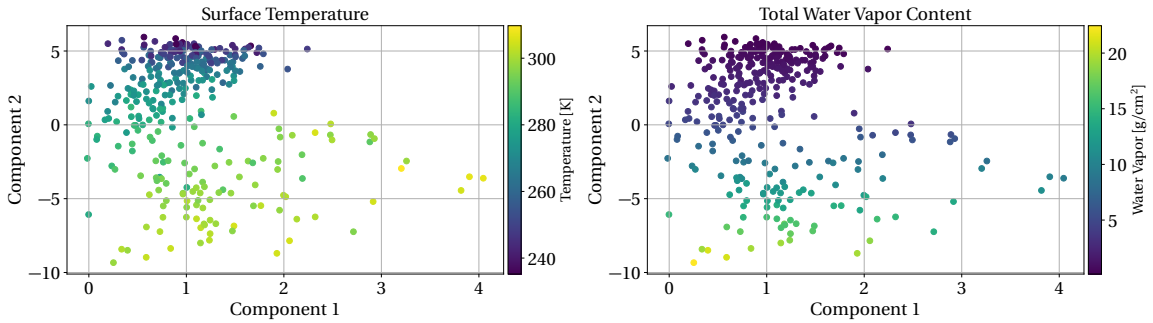


Figure 30. Autoencoder latent space when trained using 2 components. The points are scattered throughout the latent space with an overall clustering of similar atmospheric conditions. Both components appear dependent on surface temperature. Component 1 also appears more dependent on total water content versus component 2.

welling radiance does not play a role in the total error. However, if the surface material is reflective ($\varepsilon(\lambda) = 0$), the simplified LWIR RT equation becomes

$$L(\lambda)_{\varepsilon(\lambda)\to 0} = \tau(\lambda)L_d(\lambda) + L_a(\lambda) \tag{4.8}$$

where errors in $\tau(\lambda)$ and $L_d(\lambda)$ are now exaggerated. Using a standard metric, such as MSE, does not capture this dependence on material emissivity and provides misleading model performance for reflective versus emissive materials. A more appropriate metric for this domain considers the material emissivity in the at-sensor radiance error calculation.

For a test emissivity, $\varepsilon_t(\lambda)$, the estimated at-sensor radiance, $\hat{L}(\lambda)$ is calculated based on the reconstructed TUD vector. Additionally, the original TUD vector is used in conjunction with $\varepsilon_t(\lambda)$ to calculate the true at-sensor radiance $L(\lambda)$. The RMSE, E_t, is calculated across all spectral channels such that

$$E_t = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(L(\lambda_i) - \hat{L}(\lambda_i)\right)^2} \tag{4.9}$$

where K represents the number of spectral channels and E_t is now in radiance units representing the emissivity dependent RMSE. For the LWIR domain, errors are typically expressed in terms of temperature where conversion of radiance to brightness temperature is

defined as [2]:

$$T_{BB}(\lambda) = \frac{hc}{\lambda k \ln\left(\frac{2hc^2}{\lambda^5 L(\lambda)} + 1\right)} \tag{4.10}$$

By transforming at-sensor radiance to brightness temperature, the at-sensor error between $\hat{L}(\lambda)$ and $L(\lambda)$ can now be expressed in Kelvin. In general, reconstruction performance improves as $\varepsilon_t(\lambda)$ approaches 1.0 based on Equation 4.4. The actual emissivity values used are assumed grey bodies (spectrally flat) and linearly sampled between 0 and 1. Calculating the emissivity dependent RMSE provides additional information over standard MSE between predicted and truth TUD values. Model selection is performed based on performance across the entire emissivity domain resulting in lower error for reflective materials.

Constructing a model with low error across a range of emissivity values requires modifications to model training. As shown by the emissivity dependent RMSE metric, standard MSE will not provide sufficient information to properly update model weights. Instead, the loss function for training the SAE must include emissivity dependent information. The loss function must still be differentiable and result in stable training performance. To achieve this, we use the TUD MSE calculation for stablized training, but also include an at-sensor radiance MSE dependent on material emissivity:

$$\mathscr{L}(\mathbf{x},\mathbf{y}) = \frac{1}{3K}\sum_{i=1}^{3K}(\mathbf{x}_i - \mathbf{y}_i)^2 + \frac{\gamma}{MK}\sum_{j=1}^{M}\sum_{i=1}^{K}\left(L_{\mathbf{x}}(\lambda_i,\varepsilon_j) - \hat{L}_{\mathbf{y}}(\lambda_i,\varepsilon_j)\right)^2, \tag{4.11}$$

where $\mathbf{x}$ is the truth TUD vector, $\mathbf{y}$ is the reconstructed TUD vector and K is the number of spectral bands. The terms $L_{\mathbf{x}}(\lambda_i,\varepsilon_j)$ and $\hat{L}_{\mathbf{y}}(\lambda_i,\varepsilon_j)$ represent the at-sensor radiance using the truth and predicted TUD vectors respectively. The at-sensor radiance loss is calculated using a linear sampling of M emissivity values between 0 and 1, noted by ε_j in the loss calculation. A regularization term, γ, is included in Equation 4.11 to trade-off at-sensor radiance error and the TUD reconstruction error. In this study, we only consider

$\gamma = 1$, however, future work will consider this additional hyperparameter in the network optimization.

By including a loss component for the at-sensor radiance, network weights are updated to minimize at-sensor radiance error rather than strictly TUD reconstruction error. This is an important additional constraint since material emissivity impacts the difficulty of the reconstruction problem as shown by Equation 4.8. The MSE component in Equation 4.11 is necessary to stabilize training since errors in one component of the TUD vector can cause a reduction in overall loss depending on material emissivity. In practice, training networks without the MSE component caused large deviations in loss values as the network weights tried to simultaneously optimize for a range of emissivity values.

4.4.4 Radiative Transfer Modeling.

We consider the utility of the low-dimension TUD representation by applying it to the problem of RT modeling. Specifically, this section considers how to map atmospheric state vectors to the previously fit AE latent space. Our approach for creating the RT model is similar to pre-training performed in other domains such as AEs to create useful feature maps for classification [92]. Figure 31 displays an overview of the entire RT model training process. The first step in Figure 31 is the fitting of the TUD dimension reduction technique already discussed.

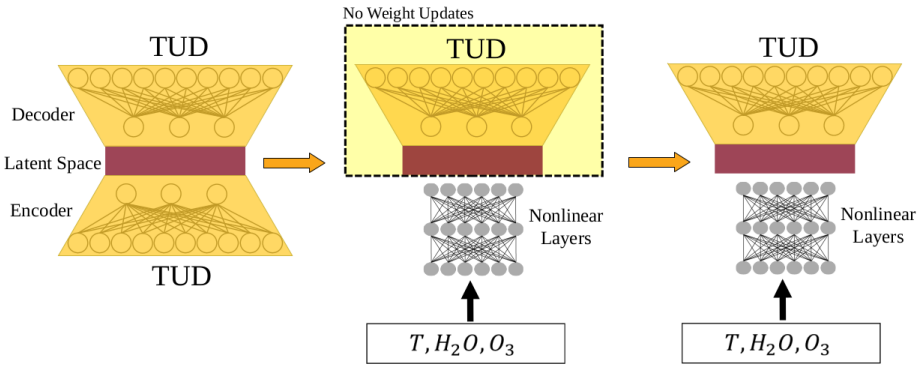


Figure 31. The RT model is created by first creating a low-dimensional representation of the TUD vectors with acceptable at-sensor radiance reconstruction errors. The latent space and decoder parameters are locked and a sampling model is fit to correctly identify the low-dimensional components to map atmospheric measurements to their corresponding TUD vectors. This diagram is specific for the SAE approach, but the encoder and decoder can be replaced with equivalent PCA transformations.

Next, a sampling network is trained to correctly predict the latent space components using atmospheric state vectors as shown by the second step in Figure 31. During this step of the training process, no updates are made to the previously trained decoder network. Once the sampling network weights have converged, the entire RT model is trained end-to-end (third step in Figure 31), allowing small weight updates in both the sampling network and the decoder. We observed less than 200 iterations are needed for the final training step as network weights are nearly optimized for the TUD regression task. As shown later, we compare the results of this process to a fully-connected neural network without the two pre-training steps shown in Figure 31.

Creating a RT model also highlights differences in the latent space construction among differing dimension reduction techniques. Ideally, similar atmospheric conditions will form clusters in the low-dimensional space. Sampling anywhere within these clusters should result in similar TUD vectors reducing the impact of sampling errors. Additionally, small changes in generative model components should lead to small deviations in the generated TUD vectors. We found that pre-training an AE to reconstruct TUD vectors was useful for enforcing a similarity between generative model components and their corresponding TUD vectors. Both of these properties allow a sampling method to quickly learn a relationship

between atmospheric measurements and the corresponding generative model components. Difficulties in sampling the latent space to generate TUD vectors may be the result of one or both of these properties not being satisfied.

The loss function for updating the nonlinear sampling layers is dependent on the dimension reduction method used to form the latent space. For SAE dimension reduction, the loss function is simply the MSE calculated between predicted components and truth components. In this case, truth components are derived from the encoder model. For PCA, components are ordered according to the variance they capture, therefore, it is important for the sampling method to correctly predict components capturing higher variance. The loss function used in this case is a weighted MSE described as:

$$\mathrm{MSE}_{\mathrm{PCA}} = \frac{1}{K}\sum_{i=1}^{K} \mathbf{w}_i(\mathbf{x}_i - \mathbf{y}_i)^2, \tag{4.12}$$

where $\mathbf{w}_i$ corresponds to the percentage of variance (expressed as a fraction) captured by component i during the PCA fitting process. This scaling ensures a weighted reconstruction reflecting component importance. Next, TIGR data augmentation is discussed as more atmospheric measurement samples are needed to fit the large number of dimension reduction model parameters.

4.4.5 Atmospheric Measurement Augmentation.

The 2311 TIGR atmospheric measurements span expected atmospheric variability, providing a set of basis measurements to fit dimension reduction models. To accurately fit the thousands of weights within a neural network, additional TUD vectors are needed to interpolate between the TIGR samples. In this section, a data augmentation approach is introduced, resulting in over 11,000 new TUD vectors derived from the TIGR database.

This study will only consider cloud free conditions requiring a relative humidity calculation to be performed on each TIGR measurement. Using a threshold of 96% relative

humidity, we downselect the TIGR data to 1755 samples. Each remaining temperature, water vapor content and ozone measurement is concatenated forming vectors of length 198. A weighted PCA approach is employed by air mass type (Polar, Tropical, etc.) on the concatenated measurement vectors such that reconstruction error is minimized at low-altitudes. Low-altitude atmospheric dynamics have a larger impact on the resulting TUD vector, requiring more accurate reconstruction at these altitudes to generate realistic TUD values. Here, 15 components were used to capture nearly all variance ($> 99.9\%$) using the weighted scheme.

Next, a 10 mixture GMM is fit to the 15 dimensional latent space created by the weighted-PCA approach. Sampling the multivariate normal distribution results in new latent space samples that are inverse transformed using the weighted-PCA model. This creates new temperature, water content and ozone measurement vectors for a particular air mass category. The relative humidity of the generated measurements is again calculated, removing new measurements exceeding 96%. Measurement vectors exceeding 10% of filtered TIGR bounds are removed and any measurements with pressure level gradients larger than the TIGR data are also removed. By filtering the generated results, the generated measurements closely match the statistics of the original data as shown in Figure 32. These measurements are forward-modeled with LBLRTM increasing the number of samples in the TUD training data. This data augmentation step is important as the number of parameters in most AE models is significantly higher than the number of samples in the original TIGR database. Model validation and testing will only consider held out sets of original TIGR samples to ensure model performance isn't based on possible errors in the augmentation process.

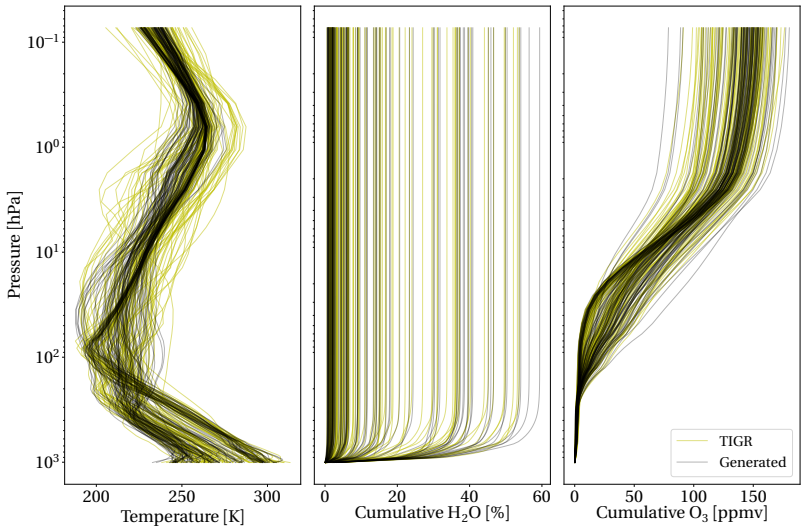


Figure 32. Generated atmospheric measurements based on sampled PCA components from the GMM.

Using the metrics, models and augmented data outlined in this section, algorithm performance is compared in the next section. The best performing methods are considered for the RT modeling problem where we show the overall effectiveness of using SAE pretraining to improve RT performance.

4.5 Results and Discussion

In this section, we first consider the impact of including the augmented atmospheric measurements in fitting the dimension reduction algorithms. After validating the augmented data improves model performance, we next compare the loss function described in Equation 4.11 against MSE. Finally, the latent space created by each dimension reduction technique is sampled following the process outlined in Figure 31 to compare RT model performance.

4.5.1 Atmospheric Measurement Augmentation.

Using the data augmentation approach outlined in Section 4.4.5, over 11,000 new atmospheric measurements were created from the original 1755 filtered TIGR measurements.

These measurements were forward-modeled through LBLRTM to create high-resolution TUD vectors. The augmented TUD vectors were downsampled to the Mako instrument line shape (ILS) resulting in 128 spectral channels for each component of the TUD vector. To test the validity of this augmentation strategy, we consider dimension reduction performance with and without the use of the augmented TUD samples.

All results are reported on test TUD vectors derived from the original TIGR database to verify the models generalize to real measurement data. The validation and test TIGR data points were selected based on total optical depth. Optical depth, $OD(\lambda)$, is related to transmittance by $\tau(\lambda) = e^{-OD(\lambda)}$. The validation and test sets contain the entire range of optical depth encountered in the TIGR data, ensuring these sets are not biased toward a particular atmospheric condition. To extract average performance information, 5 fold cross validation was used for the PCA model where the training and validation sets were still configured to contain the entire range of total optical depths in the data. For the SAE model, random weight initialization was used to derive performance statistics.

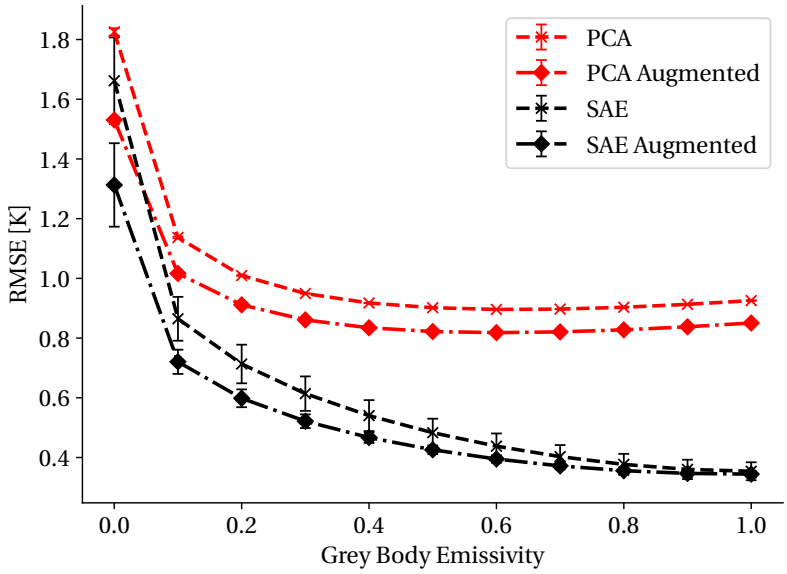


Figure 33. Dimension reduction techniques show improved results when using the augmented TIGR data. All models reduce the input data to 5 components in this plot, however, the number of components is an additional hyperparameter that will be considered later.

As shown in Figure 33, reconstruction performance improves when using the augmented data to train both the SAE and PCA algorithms. Since the SAE has many parameters to fit, the additional information encoded in the augmented data allows this technique to extract these underlying relationships with lower error. This additional information also improves PCA reconstruction performance by enforcing the axes of maximum variance within the data.

The SAE encoder and decoder have mirrored configurations with the encoder consisting of 40 nodes followed by 15 nodes connecting to the latent space. The overall network node structure is 384-40-15-N-15-40-384 where N represents the number of latent components and 384 corresponds to the TUD vector dimension. This configuration was found by conducting a hyperparameter sweep across the number of layers, nodes per layer, learning rate, batch size, and activation functions. In Figure 33, all models use 5 components to evaluate the utility of the augmented data. A learning rate of 0.001 was found to achieve acceptable results when training for 500 epochs. Additionally, the ReLU activation function was used for all nodes, except the output layer which consisted of linear activation functions. MSE loss was used for all SAE models in Figure 33. Next, the augmented data is used to evaluate the utility of the physics-based loss function described in Equation 4.11.

4.5.2 At-Sensor Loss Constraint.

The same SAE architecture used to create Figure 33 is used again in this analysis where we evaluate the utility of the loss function in Equation 4.11. To compare the MSE against our new loss function, two identical networks were trained. Specifically, each network was initialized with the same weights and samples presented in identical order, where the only difference between the networks is in the loss calculation. Both networks utilize 5 components in the latent space for dimension reduction to demonstrate differing loss characteristics for a particular network configuration. As shown in Figure 34, the physics-based

loss function provides lower reconstruction error for reflective materials ($\varepsilon(\lambda) \leq 0.5$). This is the designed behavior of the loss function since the at-sensor radiance error for reflective materials increases based on the relationship shown in Equation 4.8. As the material emissivity trends toward one, the at-sensor radiance can be described by Equation 4.4, where errors are no longer multiplicative. In this regime, MSE and the physics-based loss function converge. The error bars in Figure 34 are based on random weight initialization of the networks for repeated training trials. MSE shows significantly less variance across repeated training, but is unable to reach the lower RMSE values for reflective materials observed with the physics-based loss function. MSE outperforms the physics-based loss function for emissive materials since the reconstruction problem no longer benefits from this additional information and inhibits the model training process.

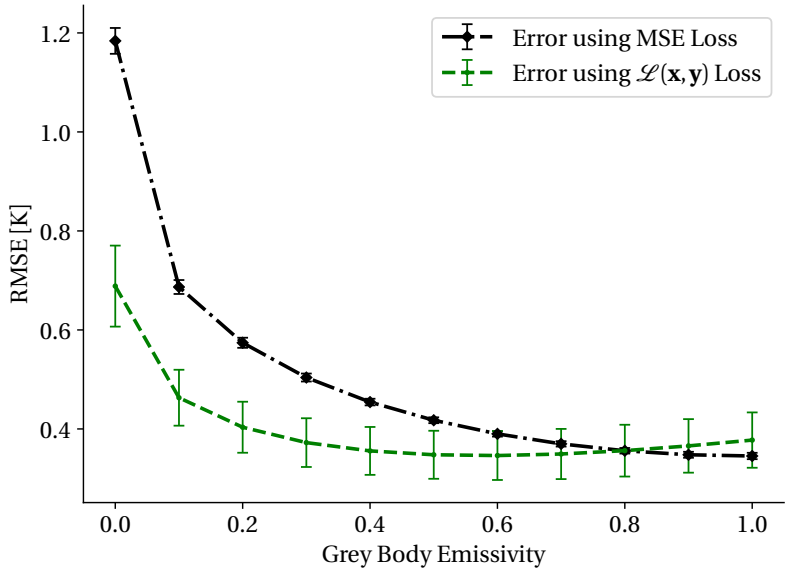


Figure 34. Comparison of SAE performance when using strictly MSE loss or the loss function described in Equation 4.11. Updating the model using information from the at-sensor radiance error improves reconstruction performance for reflective materials. The error bars represent the performance standard deviation when training multiple networks with identical architectures and random weight initialization.

4.5.3 Dimension Reduction Performance.

Using the same SAE structure discussed in Section 4.5.1, the augmented training data and the physics-based loss function, the number of components were adjusted to compare dimension reduction performance. Rather than creating plots similar to Figure 33 for each component configuration, the area under the RMSE curve was calculated (smaller is better), creating the plot shown in Figure 35. When the area under the curve is similar for multiple methods, we cannot determine which method is better without further analysis. This is because the individual curves demonstrate different performance characteristics for emissive and reflective materials. For example, we cannot say which method is best when using 8 components without also considering the material emissivity.

From Figure 35, it is clear that lower reconstruction error can be achieved with the SAE when using a low number of components. While PCA can achieve overall lower error with greater than 8 components, this isn't ideal for sampling the low-dimensional space as additional components complicate the sampling process. For the SAE model, 4 components is adequate for reconstructing the data, when considering the 176 test samples used to create these results.

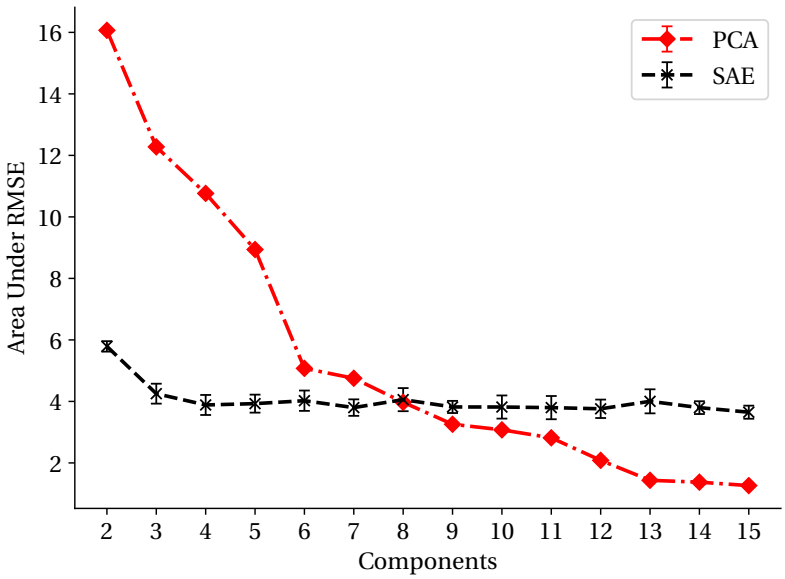


Figure 35. Varying the number of latent components and calculating the area under the RMSE curve shown in Figure 34 shows how many components are necessary to reconstruct the TIGR data. Results are plotted for the validation set consisting of 158 samples using the augmented data for training and the loss function outlined in Equation 4.11 for SAE training. The PCA error bars correspond to performance standard deviation when using 5 fold cross-validation. The SAE errors bars show the performance standard deviation when random weight initialization is used.

4.5.4 Radiative Transfer Modeling.

The low-dimensional representations created by PCA and SAE can be used for efficient radiative transfer modeling by mapping TIGR atmospheric measurements to the encoder-predicted latent components. This mapping is more difficult if diverse atmospheric conditions are closely grouped in the latent space, or similarly, if small deviations in the latent space create large TUD vector differences. The same metrics used for developing the dimension reduction methods are also used to compare RT models as we are primarily concerned with at-sensor radiance reconstruction error across a range of material emissivities.

The 66 pressure level measurements for air temperature, water content and ozone interpolated from the TIGR database are concatenated together forming a 198 dimensional input vector for latent space prediction. A two layer, fully-connected neural network (58-29) is used to map the atmospheric measurement vector to the latent space. This network utilizes

ReLU activation functions, a learning rate of 0.001, and a batch size of 16. This network configuration was identified by performing a hyperparameter sweep across number of layers, nodes per layer, activation functions, batch sizes, and learning rates resulting in over 1,400 model comparisons. The atmospheric measurements were z-score standardized by feature and the latent space components were normalized between 0 and 1. This network configuration was used for the 6 component PCA model and the 4 component SAE model. The 6 component PCA model was selected for this analysis because of the large reduction in RMSE error from 5 to 6 components. In both cases, the network was trained for 500 epochs with validation and training loss stabilizing between 200 and 300 epochs. This model configuration was found by performing a hyperparameter sweep for the PCA model and identifying the configuration with minimum validation set at-sensor reconstruction error for a range of emissivity values. The sampling network weights were updated based on latent space component prediction errors with SAE utilizing standard MSE loss. For sampling the PCA latent space, weighted MSE loss was used as outlined in Equation 4.12.

The resulting RMSE for each RT model is shown in Figure 36 where the RT model derived from the SAE decoder has the lowest error across all emissivity values. The Artificial Neural Network (ANN) model results shown in Figure 36 represent a baseline approach where an end-to-end neural network was trained with the same network configuration as the SAE RT model (198-58-29-4-15-40-384) using MSE for the model loss function. The performance difference between the SAE RT model and the ANN model highlight the advantages of first using an AE to initially fit network weights before training the RT model. Initially, training the SAE weights clusters similar atmospheric conditions together, limiting the impact of RT model sampling errors, improving overall RT model performance. Additionally, the at-sensor radiance loss function used in the SAE approach significantly improves model performance for reflective materials. Without the at-sensor radiance loss

function or the weight initialization imposed by training an AE model, the ANN model is unable to reconstruct TUD vectors with the accuracy of the SAE RT model.

The SAE RT model was trained by first fitting encoder and decoder networks to minimize at-sensor radiance reconstruction error, followed by training a small sampling network to correctly predict latent components. This training methodology did not allow updates to the decoder network after training of the sampling network. The SAE RT tuned model result shown in Figure 36, has the same configuration as the SAE RT model, but the decoder weights were also updated after the sampling network training converged. This final training step utilized the same physics-based loss function used for the initial SAE training, improving reconstruction performance for reflective emissivity values. Since the previous training steps had already created weight matrices resulting in high performance, only small changes were needed to further reduce reconstruction error.

Additionally, the 15 component PCA model results are shown. In this case, even higher RMSE error is observed since correctly sampling the 15 components is a more complex task. While the 15 component PCA model has the lowest reconstruction error during the TUD reconstruction training phase, the added complexity in latent space fitting limits the utility of this model. In all cases, the highest errors are observed for reflective materials since errors in transmittance and downwelling radiance are multiplicative in this region.

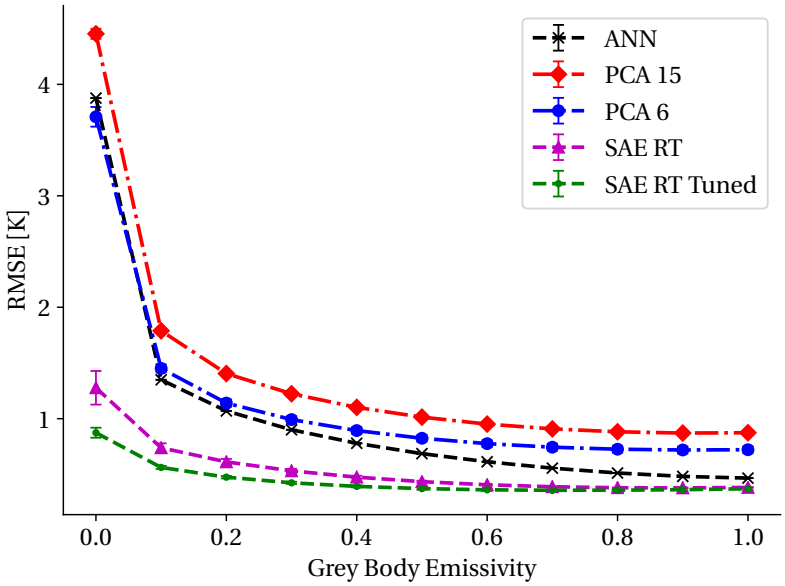


Figure 36. The performance of the RT models is shown as a function of emissivity where it is clear the SAE derived RT models create a latent space that is easier to sample with a small neural network. In all cases performance improves as materials become more emissive since downwelling radiance plays a less significant role in these cases. The 15 component PCA model is also shown, where sampling the 15 components correctly becomes a complex problem resulting in lower overall performance.

Considering only the SAE RT Tuned model, the at-sensor radiance RMSE as a function of wavenumber and emissivity is shown in Figure 37. These results are the average RMSE for the 176 test TIGR samples at each emissivity level. For emissive materials, RT model errors are below 0.5 K for most bands. The ozone absorption bands between 1050 and 1100 cm^{-1} lead to larger errors because of limited transmittance in this domain of the spectrum resulting in large deviations in the at-sensor radiance. The challenge of correctly identifying a TUD vector for reflective materials is seen by the high RMSE for $\varepsilon = 0.01$ in Figure 37. While these errors appear large on the scale shown in Figure 37, these errors are significantly larger using other models based on the results for $\varepsilon = 0$ in Figure 36.

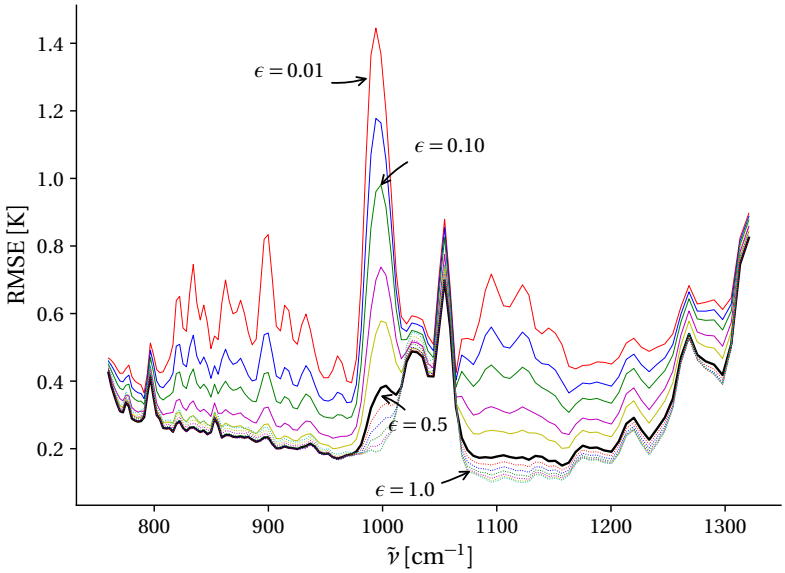


Figure 37. The average at-sensor radiance RMSE RT model errors for 176 test TIGR samples as a function of surface emissivity expressed in spectral brightness temperature. Model errors decrease with increasing emissivity, consistent with the findings in Figure 36.

Based on the results shown in Figure 37, the SAE-based RT model can estimate TUD vectors with errors below 0.5 K for most spectral bands and a range of emissivity values. These generated TUD vectors are useful for estimating surface leaving radiance described in Equation 4.3 if estimates of atmospheric conditions can be provided.

Also, the RT model can be used to quickly estimate TUD vectors. The RT model developed here is approximately fifteen times faster than the correlated k method. On average, a single TUD vector can be predicted in 0.1 seconds, however, this increase in performance is amplified when considering batch processing as multiple TUD predictions can be performed in parallel. By reducing TUD prediction time, this method is useful for quickly generating augmented representations of emissivity profiles based on a multiple atmospheric state vectors. The data was constructed such that this method could be used for the Mako LWIR hyperspectral sensor, however, resampling of the high-resolution LBLRTM data can be performed for other sensors.

4.5.5 Atmospheric Measurement Estimation.

Finally, we consider estimation of the most likely atmospheric measurements for a given TUD vector using the formulated RT model. Instead of propagating inputs forward through the RT model, this section considers estimation of the model's inputs for a given output. Since the RT model is composed of a sampling network (atmospheric measurements to latent components) and a decoder network (latent components to TUD vectors) the estimation problem can be partitioned into two steps.

First, the latent space components are identified that correspond to the TUD vector. Since the latent space only consists of 4 components, finding these 4 components using an optimizer takes little time and results in predicted TUD vectors closely matching the given TUD vector. This optimization is performed with respect to the decoder network of the RT model. As an example of this process, we select a TUD vector with a 50^{th} percentile reconstruction error from the test data set. Figure 38 shows the LBLRTM generated TUD components and the predicted TUD from optimizing the 4 latent components.

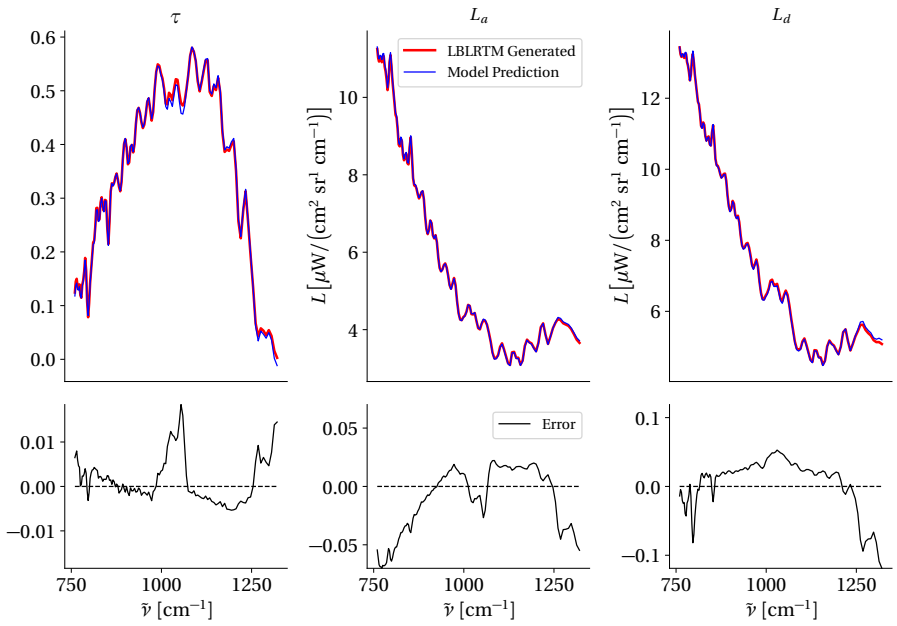


Figure 38. The top 3 panels are the predicted TUD components plotted against the LBLRTM generated TUD components. The predicted TUD components were generated by optimizing the 4 latent components. The bottom 3 panels are the TUD component residual curves, showing low error across most spectral channels.

After identifying the latent components, inputs to the sampling network (atmospheric measurements to latent components) must be optimized to identify measurements that will produce the estimated latent components. Unfortunately, the input measurement vector contains 198 values (temperature, water content and ozone at 66 pressure levels) and optimizing this large number of values for 4 components is a time-consuming and difficult task.

To make this problem more tractable, a PCA transform was applied to the training data atmospheric measurements. Using 10 components to represent the atmospheric measurements captures 90% of data variance and simplifies the optimization problem as only 10 values must be optimized to predict the 4 latent components. For the same TUD vector used in Figure 38, the result of the atmospheric measurement estimation process is shown in Figure 39. Interestingly, the largest errors occur at high altitudes, where deviations in these measurements have less impact due to lower air density.

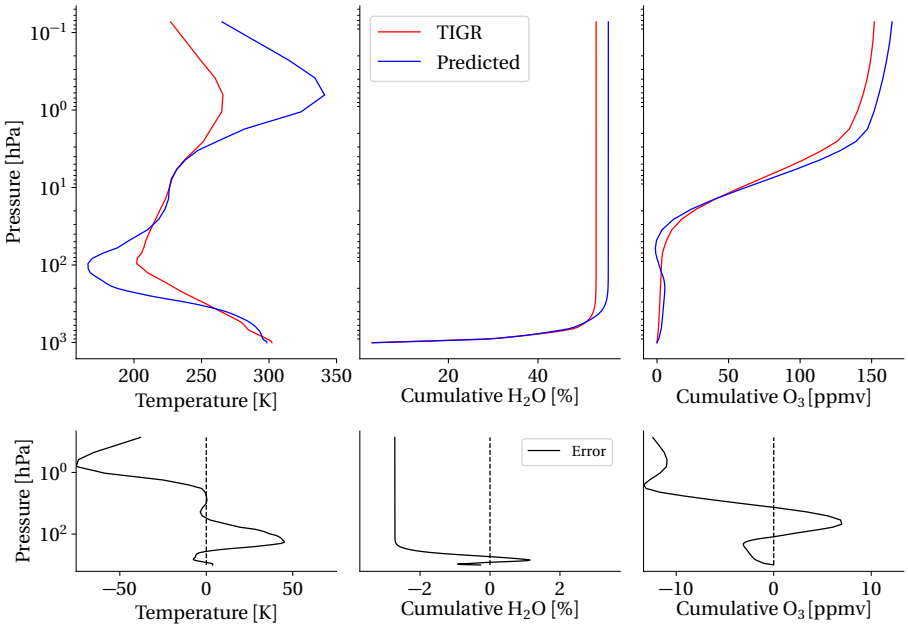


Figure 39. Predicted atmospheric measurements compared to the TIGR atmospheric measurements. The predicted atmospheric measurements will produce a close match to a given TUD vector but do show some deviations from the original TIGR measurement, specifically at high altitudes. Fortunately, high altitude error has less impact on the at-sensor radiance error because of lower air density.

The results shown in Figures 38 and 39 are for a single TUD vector. This process was repeated for all TUD vectors in the test data set to determine overall performance metrics. Errors between predicted temperature and water content measurements and the corresponding TIGR atmospheric measurements are weighted by the density at the discrete pressure levels as errors at high altitudes will have a lower impact on the TUD vector prediction. For all 176 test set TUD vectors, we observe an average error of 2.61 K for predicted temperature profiles and 0.45 cumulative H_2O % for predicted water content profiles. For ozone measurements, an atmospheric density weighting was not applied. The observed average error for ozone measurement estimation was 0.79 ppmv.

Overall, the goal of the RT model is predict TUD vectors, based on known or estimated atmospheric measurements. By showing the model's ability to estimate atmospheric measurements from a given TUD vector (inverse problem), we have demonstrated the utility of low-dimension representations of the TUD vectors. Additionally, since the latent space

clusters similar atmospheric conditions together (latent-space-similarity), an ensemble of likely atmospheric measurements can be generated for a given TUD by applying small deviations to latent components.

4.6 Conclusions

This study has leveraged SAEs with a novel physics-based loss function to reduce TUD vector dimensionality such that fast and effective LWIR RT models could be constructed by sampling the low-dimensional TUD representation. By using an AE pre-training step, the low-dimensional TUD representation clustered similar atmospheric conditions together reducing sampling errors. Additionally, the pre-training step verified that small deviations in the low dimensional TUD representation corresponded to small deviations in the high dimensional TUD vector. These approaches were shown to reconstruct at-sensor radiance with errors below 0.5 K for most emissivity values.

The dimension reduction results utilized real atmospheric measurements from the TIGR database and augmented data derived from this same database. Using augmented atmospheric measurements improved both PCA and SAE performance for a range of material emissivities. PCA was shown to reconstruct data with lower error than the SAE when using beyond 8 components. The SAE performance did not improve significantly when using more than 4 components, demonstrating adequate capacity for the augmented TIGR data.

Sampling the low-dimensional representations created by these methods highlighted significant differences in TUD reconstruction. This study found the SAE latent space easier to sample, resulting in lower RT model errors. Additionally, training the entire RT model after pre-training the sampling network and decoder networks improved RT performance. The RT model was further explored to identify the most likely atmospheric measurements for a given TUD vector. This analysis revealed that RT model inputs could easily be optimized resulting in predicted atmospheric measurements with some agreement to the TIGR

measurements. While this was not the goal of this work, we will explore the utility and limitations of SAEs in the inverse problem of estimating atmospheric conditions from spectral measurements.

Optimizing the latent components for a particular TUD vector is a straightforward process, however, no known physical quantities are directly correlated with these components. No constraints were placed on the formulation of this latent space other than the overall network loss function. This unconstrained representation creates a disentanglement problem limiting the utility of the SAE as a generative model when limited atmospheric information is available. Future work in this area will consider additional constraints on the latent space to improve upon this disentanglement problem, offering a means for creating TUD vectors with properties representative of specific atmospheric conditions. Specifically, if complete atmospheric measurements are not available, this analysis will determine what information is required to predict the most likely TUD vector using a small number of components. Methods such as Variational AEs, multi-modal AEs and Generative Adversarial Networks coupled with physical constraints will be investigated toward this goal.

V. Learning Set Representations for LWIR In-Scene Atmospheric Compensation

5.1 Paper Overview

Previous research investigated methods for compressing worldwide Transmittance, Upwelling, and Downwelling (TUD) data allowing for faster radiative transfer calculations. Similarly, the low-dimensional latent space created by an Autoencoder (AE) network can be used to support in-scene atmospheric compensation. This paper utilizes previous research in AE models and new network architectures dependent on permutation invariant neural network layers to estimate a scene's TUD vector. A novel data generation algorithm was created to fit an in-scene atmospheric compensation neural network without the need for spatially-resolved hyperspectral data cubes. Results are presented for both synthetic and collected hyperspectral data demonstrating comparable or better performance to current Long-Wave Infrared (LWIR) atmospheric compensation approaches.

This paper was published in the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

5.2 Abstract

Atmospheric compensation of LWIR hyperspectral imagery is investigated in this paper using set representations learned by a neural network. This approach relies on synthetic at-sensor radiance data derived from collected radiosondes and a diverse database of measured emissivity spectra sampled at a range of surface temperatures. The network loss function relies on LWIR radiative transfer equations to update model parameters. Atmospheric predictions are made on a set of diverse pixels extracted from the scene, without knowledge of blackbody pixels or pixel temperatures. The network architecture utilizes permutation-invariant layers to predict a set representation, similar to work performed in point cloud

classification. When applied to collected Hyperspectral Imagery (HSI) data, this method shows comparable performance to Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR), using an automated pixel selection approach. Additionally, inference time is significantly reduced compared to FLAASH-IR with predictions made on average in 0.24 s on a 128 pixel by 5000 pixel data cube using a mobile graphics card. This computational speed-up on a low-power platform results in an autonomous atmospheric compensation method effective for real-time, on-board use, while only requiring a diversity of materials in the scene.

5.3 Introduction

Hyperspectral sensors continue to improve in both spatial and spectral resolution, allowing for a wide range of applications such as land cover mapping, search and rescue operations and target detection [2, 21, 135]. Each HSI pixel contains information sampled across hundreds of narrow spectral channels creating a three-dimensional data cube: width by height by spectral channel. HSI data collected in the LWIR region of the electromagnetic spectrum (8 - 14 μm) contains surface emissivity and temperature information. These measurements are important for atmospheric modeling, climate change studies and urban heat island analysis [136, 137]. Efficiently and accurately extracting emissivity and temperature information remains a challenging problem. The goal of this paper is, with limited assumptions on material content in a scene, to develop and evaluate an efficient method for extracting atmospheric information from a LWIR HSI data cube.

Compared to the reflective region of the electromagnetic spectrum (0.4 - 2.5 μm), the LWIR domain is dominated by emission of surface materials and atmospheric constituents. The at-sensor radiance, $L(\lambda)$, for a lambertian surface can be described by the simplified

LWIR radiative transfer model [2]:

$$L(\lambda) = \tau(\lambda)\Big[\varepsilon(\lambda)B(\lambda,T) + \big[1-\varepsilon(\lambda)\big]L_d(\lambda)\Big] + L_a(\lambda), \tag{5.1}$$

where

λ : wavelength

T : material temperature

$\tau(\lambda)$: atmospheric transmission

$\varepsilon(\lambda)$: material emissivity

$B(\lambda,T)$: Planckian distribution

$L_d(\lambda)$: downwelling atmospheric radiance

$L_a(\lambda)$: atmospheric path (upwelling) radiance.

The Planckian distribution is:

$$B(\lambda,T) = \frac{2hc^2}{\lambda^5}\frac{1}{e^{hc/\lambda kT}-1}, \tag{5.2}$$

where c is the speed of light, k is Boltzmann's constant and h is Planck's constant. Transmittance and path radiance are spectrally-varying quantities which depend on the spatially-varying temperature and constituent concentrations in the atmosphere [13]. Water vapor, ozone, and carbon dioxide are among the most important infrared-active gases affecting the remotely sensed spectrum. Path radiance represents atmospheric emission directly into the sensor line-of-sight, while downwelling radiance represents atmospheric emission toward the surface. Assuming a lambertian surface, downwelling radiance is a cosine-weighted average over the entire hemisphere above the surface. Downwelling radiance reflected off

the surface enters the sensor line of sight, requiring reflective materials to estimate this term.

Emissivity retrieval from $L(\lambda)$ can be divided into two steps: atmospheric compensation and Temperature-Emissivity Separation (TES). Atmospheric compensation methods estimate the TUD $(\tau(\lambda), L_a(\lambda), L_d(\lambda))$ vector allowing estimation of the surface-leaving radiance $L_s(\lambda)$:

$$L_s(\lambda) = \frac{L(\lambda) - L_a(\lambda)}{\tau(\lambda)} = \varepsilon(\lambda)B(\lambda, T) + \big[1 - \varepsilon(\lambda)\big]L_d(\lambda). \tag{5.3}$$

Next, TES algorithms are applied to simultaneously estimate $\varepsilon(\lambda)$ and T. Separating these terms is complicated by their coupling in the emissive portion of the surface-leaving radiance. For a sensor with K spectral bands, estimating $\varepsilon(\lambda)$ and T is an under-determined problem as there are $K+1$ unknowns $(\varepsilon(\lambda), T)$ and only K observed radiance values. TES algorithms apply constraints to $\varepsilon(\lambda)$ making the estimation problem more tractable. Typically, $\varepsilon(\lambda)$ is an assumed smoother function of wavelength than the observed atmospheric features [27]. Additionally, if the downwelling radiance can be estimated, emissivity can be expressed as [14]:

$$\varepsilon(\lambda) = \frac{L_s(\lambda) - L_d(\lambda)}{B(\lambda, \hat{T}) - L_d(\lambda)}, \tag{5.4}$$

where $\hat{T}$ is the estimated pixel temperature. Pixel temperature is determined by minimizing atmospheric features in the estimated emissivity profile, resulting in a smoother spectral emissivity. More recent methods such as subspace-based TES [138], project the original data to a lower-dimensional subspace to determine maximum-likelihood estimates of both temperature and emissivity.

This study presents a new method for in-scene LWIR atmospheric compensation using a neural network approach with minimal assumptions on scene material content, while efficiently producing comparable results to other compensation methods on collected HSI

data. The DeepSets network introduced in [75] is the basis of our approach and so our method is named DeepSet Atmospheric Compensation (DAC). The DAC algorithm relies on a non-linear TUD vector dimension reduction performed using an AE network to reconstruct spectrally-resolved TUD vectors. This low-dimensional TUD representation is utilized with permutation-invariant neural network layers to fully reconstruct the TUD vector for a given set of pixels. No blackbody pixel assumptions are made and pixel temperature estimates are not necessary to predict the underlying TUD vector. In the next section a review of various atmospheric compensation methods is presented, highlighting differences between previous methods and our new compensation approach.

5.3.1 Atmospheric Compensation Methods.

Atmospheric compensation algorithms can be divided into two paradigms: model-based methods and in-scene methods. Radiative transfer models such as MODerate resolution atmospheric TRANsmission (MODTRAN) support model-based compensation methods using the known or estimated atmospheric state information (column water vapor, trace gas content) to calculate the TUD vector in Equation 5.1 [24, 26]. Model-based methods are computationally more expensive than in-scene methods, but can be implemented efficiently if look-up tables encompassing expected conditions are computed before collecting data [139].

One model-based approach considered in this study is FLAASH-IR, which retrieves scene atmospheric parameters based on a look-up table of precomputed TUD vectors from MODTRAN [26]. The look-up table is generated by varying atmospheric surface temperature, water vapor column density and an ozone scaling factor. Typically 10-20 pixels must be selected consisting of varying brightness and emissivity profiles. High reflectivity materials are useful for downwelling radiance estimation and should be included in pixel selection. Mean-Square Error (MSE) between the observed radiance and predicted radi-

ance is minimized by varying surface temperature and atmospheric scaling parameters to recover the TUD vector. As will be shown later, our approach also benefits when reflective materials are present in the scene, but selects these materials automatically through a spectral angle measurement.

Another LWIR atmospheric compensation approach utilizing a MODTRAN look-up table and a coupled subspace model is presented in [131]. This approach utilizes singular value decomposition (SVD) to form basis matrices of transmittance and upwelling radiance. Blackbody pixels are identified using the basis matrices to retrieve surface leaving radiance. The DAC algorithm utilizes an AE model to perform dimension reduction on the TUD vectors, similar to the SVD approach employed in [131]. Nonlinear dimension reduction using an AE model allows for lower reconstruction error compared to linear approaches, but requires additional training data to properly fit the network weights.

In-scene methods typically do not rely on look-up tables to estimate atmospheric parameters, but some material information is required to make the atmospheric compensation problem tractable. One of the most common approaches is the In-Scene Atmospheric Compensation (ISAC) method that first identifies blackbody pixels ($\varepsilon(\lambda) \approx 1$), where at-sensor radiance can be described by [15]:

$$L(\lambda) = \tau(\lambda)B(\lambda,T) + L_a(\lambda). \tag{5.5}$$

A linear fit is performed on each spectral channel to estimate $\tau(\lambda)$ and $L_a(\lambda)$. Each pixel temperature must be determined prior to this fitting procedure. Temperature estimates are made in the most transmissive spectral bands, but can be systematically biased. Water absorption features near 11.73 μm are used to reduce biases introduced by inaccurate surface temperature estimates. Treating $\tau(\lambda)$ and $L_a(\lambda)$ as independent fit parameters, when in fact they are strongly correlated, can also exaggerate fit errors [140].

5.4 Methodology

Training the DAC algorithm requires a library of worldwide atmospheric measurements, forward modeled with MODTRAN, forming a training set of diverse TUD vectors. Additionally, a low-dimensional representation of these TUD vectors is used to reduce model fitting complexity [141]. Next, the TUD vector dimension reduction process is reviewed and how this method fuses with DAC is explained.

5.4.1 TUD Vector Dimension Reduction.

The Thermodynamic Initial Guess Retrieval (TIGR) database is derived from over 80,000 radiosonde measurements collected worldwide and consists of 2311 atmospheric temperature, water vapor and ozone measurements on a fixed pressure grid [30, 31]. These measurements are filtered for cloud free conditions with 96% relative humidity threshold, reducing the number of atmospheric measurements to 1755.

To increase the number of training samples used for neural network fitting, a data augmentation strategy is employed on the remaining 1755 atmospheric measurements. First, Principal Component Analysis (PCA) is applied to the measurements using 15 components for each air mass category (Tropical, Polar, etc.). Reconstructing low altitude atmospheric measurements is weighted more heavily in the PCA fitting process, since these measurements have a greater impact on the resulting TUD vector. A Gaussian Mixture Model (GMM) is fit to the 15 dimensional space, and sampling this GMM creates new measurement components that are transformed back into pressure level measurements. After filtering these measurements for relative humidity, they are included in the training data.

All atmospheric measurements are forward modeled using MODTRAN 6.0 to create high resolution TUD vectors based on a nadir sensor zenith angle. In this study, the Spatially Enhanced Broadband Array Spectrograph System (SEBASS) instrument line shape (ILS) is applied to the high resolution output, creating sensor-specific TUD vectors [142].

Training, validation and test data partitioning is based on total optical depth of the original TIGR measurements resulting in 161 validation TUD vectors, and 179 test TUD vectors. Combining the remaining TIGR samples with the augmented data results in 8,450 training TUD vectors. Each training sample is considered at 17 different altitudes (0.15 km - 3.05 km), leading to a training set of 143,650 TUD vectors. This altitude range was used because previously collected SEBASS data spanned these altitudes. Validation and test samples are generated at altitudes not considered in the training set. Similar to the work performed in [141], a low-dimensional representation of the generated TUD vector library is created using an AE network (Figure 40). An AE consists of two networks, an encoder and decoder, to perform nonlinear data compression. The encoder transforms input data $\mathbf{y} \in \mathbb{R}^d$ to the latent space representation $\mathbf{z} \in \mathbb{R}^l$ where $l \ll d$. The decoder reconstructs the input from $\mathbf{z}$ to produce $\hat{\mathbf{y}}$ and weights are updated based on the error between $\mathbf{y}$ and $\hat{\mathbf{y}}$. The entire AE data transformation can be expressed with:

$$\begin{aligned} \mathbf{z} &= f(\mathbf{W}_z \mathbf{y} + \mathbf{b}_z), \\ \hat{\mathbf{y}} &= f(\mathbf{W}_y \mathbf{z} + \mathbf{b}_y), \end{aligned} \tag{5.6}$$

where $\mathbf{W}_z$ and $\mathbf{W}_y$ are the encoder and decoder weight matrices respectively. Additionally, a bias term is also used at each node, represented by $\mathbf{b}_z$ and $\mathbf{b}_y$. The function f is a nonlinear transform used throughout the network. In this work, we consider two activation functions: the leaky Rectified Linear Unit (ReLU) and the exponential linear unit (ELU) described by:

$$\text{Leaky RELU}(x) = \begin{cases} x, & \text{if x} > 0 \\ \alpha x, & \text{if x} \leq 0 \end{cases}$$

$$\text{ELU}(x) = \begin{cases} x, & \text{if x} > 0 \\ \alpha(\exp(x) - 1), & \text{if x} \leq 0 \end{cases}$$

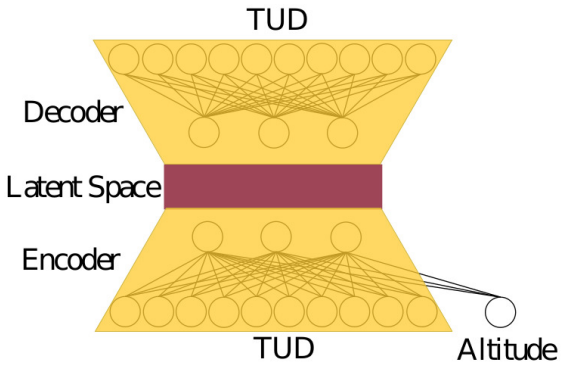


Figure 40. TUD vectors are compressed by the encoder into the latent space and then reconstructed by the decoder network. Reconstruction error is minimized through weight updates during the training process. Additionally, a scalar altitude input is also presented with the TUD vector allowing the model to scale to multiple altitudes.

where α allows information to flow through the network when the activation function output is negative.

Networks weights are updated using their individual contribution to overall network error, measured by the loss function. The loss function, which was presented previously [141], features both a standard reconstruction error term as well as an at-sensor apparent radiance error:

$$\mathscr{L}(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{3K}\sum_{i=1}^{3K}(\hat{y}_i - y_i)^2 + \frac{\gamma}{MK}\sum_{j=1}^{M}\sum_{i=1}^{K}\left(L_{\hat{\mathbf{y}}}(\lambda_i,\varepsilon_j) - L_{\mathbf{y}}(\lambda_i,\varepsilon_j)\right)^2 \tag{5.7}$$

Here, $\mathbf{y}$ is the truth TUD vector and $\hat{\mathbf{y}}$ is the reconstructed vector, K is the number of spectral channels, $L_{\hat{\mathbf{y}}}(\lambda_i,\varepsilon_j)$ and $L_{\mathbf{y}}(\lambda_i,\varepsilon_j)$ are the at-sensor radiance values for the vectors $\hat{\mathbf{y}}$ and $\mathbf{y}$. Additionally, a linear sampling of M grey-body emissivity values between 0 and 1 are used to calculate this loss term, improving reconstruction error for reflective and emissive materials [141].

The hyperparmeter γ controls the relative importance between the contribution of the TUD MSE and the at-sensor radiance MSE within the loss function, and is set to $\gamma = 1$ in this study. As γ approaches zero, more emphasis is placed on TUD MSE resulting in

higher reconstruction error for reflective materials as shown in [141]. Similarly, when $\gamma > 1$ more emphasis is placed on at-sensor radiance error. The TUD MSE term is necessary to stabilize training and in our experience, increasing γ to a large value can lead to unstable AE training.

The TUD AE presented here differs from the work of [141] because sensor altitude is also included in the model. This scalar input allows the AE to correctly reconstruct TUD vectors at a range of altitudes making the model more applicable to real-world scenarios where sensor altitude varies. For sensors operating at a constant altitude, this model can be retooled to consider a small range of altitudes in the sensor's operating range, or a single altitude can be considered as was previously demonstrated in [141]. Validation and test sets consist of hold out samples where neither the atmospheric state nor the sensor altitude were observed in the training set. Performance on validation and test sets explain the network's ability to generalize to new samples or highlights models that are overfit to the training data.

5.4.1.1 Autoencoder Metrics.

TUD vector reconstruction error must be placed in context of the overall remote sensing goal to select AE models with the best performance. Predicted at-sensor radiance is calculated using the predicted TUD vector, a range of grey-body emissivity values, and an assumed pixel temperature. Since this study is focused on the LWIR domain, spectral radiance values were transformed to brightness temperature $T_{BB}(\lambda)$ for conveniently representing model errors. $T_{BB}(\lambda)$ is computed by inverting Planck's function:

$$T_{BB}(\lambda) = \frac{hc}{\lambda k \ln\left(\frac{2hc^2}{\lambda^5 L(\lambda)} + 1\right)}. \tag{5.8}$$

The root mean square error (RMSE) in Kelvin can be calculated with:

$$E_t = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(T_{BB}(\lambda_i) - \hat{T}_{BB}(\lambda_i)\right)^2}, \tag{5.9}$$

where index t corresponds to a test grey body, $\varepsilon_t(\lambda)$, used in Equation 5.1 to compute $L(\lambda)$. The test emissivity values range from 0 to 1 producing an RMSE describing overall performance between reflective and emissive materials for the AE model. Next, the entire in-scene atmospheric compensation method is introduced utilizing the fit AE model.

5.4.2 In-Scene Atmospheric Compensation.

Numerous methods are available for LWIR in-scene atmospheric compensation, typically relying on the selection of blackbody pixels to determine $\tau(\lambda)$ and $L_a(\lambda)$. Rather than following this paradigm, the DAC algorithm relies on an automated selection of diverse pixels, without knowledge of $\varepsilon(\lambda)$ or pixel temperature to estimate atmospheric effects on LWIR HSI data.

Given N diverse pixels $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, extracted from a data cube collected across K bands, $\mathbf{x}_i \in \mathbb{R}^K$, the DAC network, $f(\mathbf{X})$, must predict the AE low-dimensional representation $\mathbf{z}$. Since this function operates over a set of extracted pixels, $f(\mathbf{X})$ must be permutation-invariant to the pixel selection order. Additionally, this network must also provide similar predictions as the number of pixels varies within the set $\mathbf{X}$. After predicting $\mathbf{z}$, the previously trained decoder network, $d(\cdot)$ can be used to reconstruct the full TUD vector.

This class of problems is referred to as set-input learning where a single target corresponds to a set of input samples [69, 75, 143]. Recently, new network architectures have been investigated for domains such as point cloud classification, anomaly detection and image tagging to address set-input learning, referred to as DeepSets or PointNet for point cloud classification [75, 78]. These architectures utilize a permutation-invariant function,

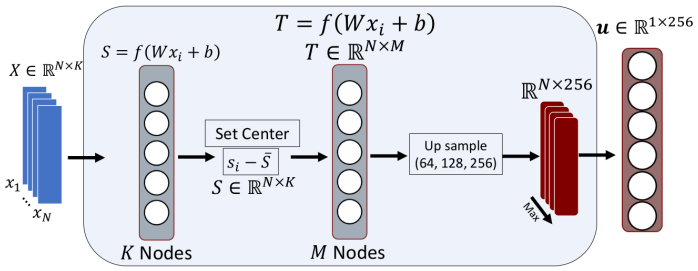


Figure 41. The permutation invariant transform $\phi(\mathbf{X})$ used in this study consists of a neural network applied to all pixels in the set X followed by a centering operation of the transformed pixel representations. The set S is transformed by a dense layer and upsampled. The $N \times 256$ set representation is collapsed into a single permutation-invariant set representation through a max operation.

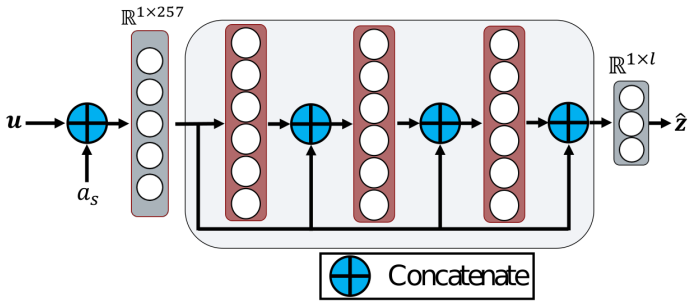


Figure 42. The $\rho(\cdot)$ network is shown highlighting the use of skip connections to propagate the set representation to deeper layers. The input to this network is the result of the permutation-invariant set extraction concatenated with the sensor altitude, a_s. Each layer inside the network block contains 50 nodes and the predicted latent space contains 4 components.

$\phi(\cdot)$, to extract a set feature vector. This operation can be broken down into two steps: set transformation and set decomposition. Set transformation is performed by $\phi(\cdot)$ with U output nodes to produce the set $\mathbf{V}$:

$$\mathbf{V} = \phi(\mathbf{X}), \quad \mathbf{V} \in \mathbb{R}^{N\times U} \tag{5.10}$$

Next, set decomposition can be performed using any permutation invariant function. In this study, the maximum value is taken across the N pixels, resulting in the set feature vector $\mathbf{u}$:

$$\mathbf{u}_j = \max_{i\in N} \mathbf{V}_{ij} \quad \mathbf{u} \in \mathbb{R}^{1\times U} \tag{5.11}$$

The entire permutation invariant function can be written as:

$$f(\mathbf{X}) = \rho\left(\max_{i \in N} \phi(\mathbf{X})\right), \tag{5.12}$$

where ρ is another transformation (neural network) applied to the set feature vector $\mathbf{u}$. Additionally, $\phi(\mathbf{X})$ can be applied to any number of N inputs, since the max operator pools all N samples into a set feature vector.

The max decomposition of $\phi(\mathbf{X})$ is shown in Figure 41. First, a K node neural network layer transforms the at-sensor radiance pixels into the set $\mathbf{S}$. Next, $\mathbf{S}$ is centered to encode overall set information in the learned representation improving convergence during training. This operation is similar to batch normalization [144], however, we apply this normalization across the transformed set, rather than the batch. After set centering, a layer containing M nodes is used to transform the centered representation before upsampling. The number of nodes M is a hyperparameter we vary during model evaluation. Upsampling layers are required to provide enough information in the set feature vector for the following $\rho(\cdot)$ network to predict the target value. The upsampling layers create the set $\mathbf{V} \in \mathbb{R}^{N \times 256}$.

The $\rho(\cdot)$ network predicts the low-dimensional TUD representation $\hat{\mathbf{z}}$ using the set information extracted by the max decomposition. The previously fit decoder network, $d(\cdot)$, returns the fully spectrally resolved TUD vector such that $(\hat{\tau}(\lambda), \hat{L}_a(\lambda), \hat{L}_d(\lambda)) = d(\hat{\mathbf{z}})$.

Predictions made by $\rho(\cdot)$ are altitude dependent. To include this information in $f(\mathbf{X})$, the sensor altitude, a_s, is concatenated to the input of $\rho(\cdot)$ (Figure 42). This allows $\rho(\cdot)$ to modify its low-dimensional prediction, $\hat{\mathbf{z}}$, to changes in altitude, ultimately making the model more applicable for real-world conditions. The $\rho(\cdot)$ network shown in Figure 42 makes extensive use of skip connections, allowing the extracted 1×257 vector to propagate to deeper network layers. Finally, combining the max decomposition and $\rho(\cdot)$ networks,

the DAC algorithm can be expressed as:

$$f(\mathbf{X}) = \rho\left(\max_{i \in N}\left[\phi(\mathbf{X})\right], a_s\right) \tag{5.13}$$

To identify the best network architecture, hyperparameter sweeps were performed on the number of nodes M in $\phi(\mathbf{X})$ and the number of nodes per layer in $\rho(\cdot)$. Additionally, batch size, learning rates and activation functions were also varied. The network architecture used in this study sets $M = 90$ in the $\phi(\mathbf{X})$ network utilizing the ELU activation function. Additionally, the $\rho(\cdot)$ network contains 3 layers containing 50 nodes, all using the ELU activation function. The network was trained with $N = 50$, however, this can be varied during model evaluation since the max decomposition is along the pixel axis. A learning rate of 0.001 and a batch size of 64 was used to fit network weights. The Adam optimization algorithm was used for calculating weight updates [17]. Networks were constructed using Python 3.6.8, Keras version 2.2.4, Tensorflow 1.15 and hyperparameter sweeps were conducted across 20 Graphical Processing Units (GPUs) using Ray Tune version 0.7.6 [145] [146]. Since the AE and DAC model only use 132 MB of memory, multiple models can be trained in parallel on a single GPU. The model contains 109,026 weights fit through the training process discussed next.

5.4.3 Algorithm Training.

Each training example consists of a set of at-sensor radiance spectra, $\mathbf{X}$, and the low-dimensional TUD representation, $\mathbf{z}$, generated by the encoder network. Creating the at-sensor radiance spectra requires a library of emissivity spectra, TUD vectors and a method for assigning pixel temperatures.

Pixel emissivity spectra are selected from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) database and downsampled using the SEBASS ILS. To verify model performance on new data, 200 emissivity spectra are held out leaving

Algorithm 1 Set Generation

Input: $\boldsymbol{\varepsilon}, \boldsymbol{\tau}, \boldsymbol{L_a}, \boldsymbol{L_d}, t_o, N$
Output: $\boldsymbol{L}$

Emissivity Selection :
$\varepsilon_t = \mathrm{U}(0.75, 1.0)$
$\boldsymbol{\varepsilon}_f = \boldsymbol{\varepsilon}$ s.t. $\bar{\boldsymbol{\varepsilon}} < \varepsilon_t$
$\boldsymbol{\varepsilon}_R = \boldsymbol{\varepsilon}_f$ s.t. $\bar{\boldsymbol{\varepsilon}}_f < \varepsilon_t - 0.10$
$\boldsymbol{\varepsilon}_E = \boldsymbol{\varepsilon}_f$ s.t. $\bar{\boldsymbol{\varepsilon}}_f \geq \varepsilon_t - 0.10$
$P_E = \mathrm{U}(0.5, 0.95)$
$N_E = \mathrm{int}(P_E \cdot N)$
$N_R = N - N_E$
$\boldsymbol{\varepsilon}_S$ = [N_R samples from $\boldsymbol{\varepsilon}_R$, N_E samples from $\boldsymbol{\varepsilon}_E$]
At-Sensor Radiance Generator
$w = \mathrm{U}(2, 20)$
for $i = 0$ to N **do**
    T = $\mathrm{U}(t_o - w, t_o + w)$
    $\boldsymbol{L_s} = \boldsymbol{\varepsilon}_S[i]B(T) + (1 - \boldsymbol{\varepsilon}_S[i])\boldsymbol{L_d}$
    L[i] = $\boldsymbol{\tau L_s} + \boldsymbol{L_a}$
end for
return $\boldsymbol{L}$

978 profiles for training. The held out emissivity spectra contain a range of reflective and emissive materials to evaluate model performance.

Selecting the N emissivity spectra for a particular training set begins by dividing the ASTER database into emissive and reflective samples. To model a wide range of scenes, an initial emissivity threshold is calculated, $\varepsilon_t \sim \mathrm{U}(0.75, 1.0)$, where U is a uniform distribution. Emissivity spectra with means exceeding this threshold are removed, resulting in a filtered emissivity database, $\boldsymbol{\varepsilon}_f$. The filtered database is divided into emissive samples, $\boldsymbol{\varepsilon}_E$, and reflective samples, $\boldsymbol{\varepsilon}_R$, based on a threshold of $\varepsilon_t - 0.10$.

Next, the percent of emissive samples, P_E, in the set N is sampled according to the distribution $P_E \sim \mathrm{U}(0.5, 0.95)$. The number of reflective, N_R, and emissive, N_E, materials in the scene are determined by P_E. Emissivity spectra are sampled from the emissive and reflective portions of the filtered ASTER database forming the set emissivity spectra, $\boldsymbol{\varepsilon}_S$. This process is outlined in Algorithm 1 under Emissivity Selection, where $\boldsymbol{\varepsilon}$ corresponds

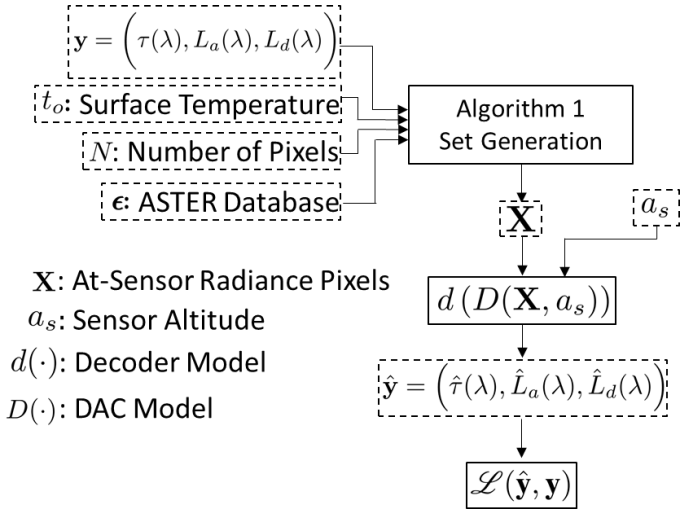


Figure 43. The entire DAC network training pipeline is shown highlighting the inputs to the Set Generation algorithm resulting in the at-sensor radiance pixels X. Using the known sensor altitude, a_s, the DAC model, $D(\cdot)$, predicts the low-dimensional TUD representation such that the decoder model, $d(\cdot)$ can reconstruct the TUD vector. The loss function, $\mathscr{L}(\hat{\mathbf{y}}, \mathbf{y})$ (Equation 5.7) directs weight updates within the DAC network.

to the training or validation ASTER database samples. By varying P_E and ε_t, training sets dominated by blackbody pixels with little diversity can be created, and highly diverse scenes can be created containing both reflective and emissive materials.

After selecting a set of N emissivity spectra, N pixel temperatures must be assigned to calculate at-sensor radiance. The surface-level temperature measurement, t_o, from the TIGR data is used to assign pixel temperatures, T_i, such that $T_i \sim \mathrm{U}(t_o - w, t_o + w)$, where $w \sim \mathrm{U}(2, 20)$. By allowing the T_i uniform distribution width to vary, scenes containing little temperature variation and high temperature variation can be generated. After initializing the emissivity spectra and pixel temperatures, a set of at-sensor radiance values, $\mathbf{L}$, are calculated for a single TUD vector as shown by the entire set generation process in Algorithm 1. Figure 43 shows the entire training process to include the role of the DAC model, decoder model and overall network loss calculation. Figure 44 shows the result of this process for 50 sampled emissivity spectra, where $\varepsilon_t = 0.85$, $P_E = 0.75$ and the surface temperature was 296 K.

Considering the 978 emissivity training spectra and the set size $N = 50$, the number of possible emissivity training sets is $\binom{978}{50} = 3 \times 10^{84}$. Additionally, each pixel emissivity temperature is randomly sampled following the strategy outlined in Algorithm 1, further increasing the number of training samples for a single TUD vector. Using 8,450 TUD vectors sampled at 17 altitudes results in a large training data set to fit the DAC network. Based on the large number of training samples possible, spectral noise was not added to the ASTER emissivity spectra. Similarly for the validation set, the number of emissivity spectra combinations based on 200 hold out emissivity spectra is $\binom{200}{50} = 4.5 \times 10^{47}$, and 161 TUD vectors are considered across 2 altitudes, none of which were a part of the training data.

While the number of possible training and validation samples is very large, we find training only requires 150 iterations for network performance to converge. During each training iteration, 50 batches are randomly generated from the training TUD database. Specifically, using a batch size of 64, the set generation algorithm shown in Algorithm 1 is executed 64 times to generate a single training batch. During each training iteration, weight updates are made based on 3,200 TUD vectors.

The encoder network of the previously trained AE model is used to map the underlying TUD vector to the low-dimensional representation $\mathbf{z}$. The DAC algorithm predicts $\hat{\mathbf{z}}$ such that the decoder network, $d(\cdot)$, can fully reconstruct the TUD vector. The same loss function used in the AE model training (Equation 5.7) is used for the DAC network, $\mathscr{L}(d(\hat{\mathbf{z}}), d(\mathbf{z}))$, where the inputs are the decoder transformed TUD vectors. Again, M grey-body emissivity values, e_j, between 0 and 1 are used calculate DAC loss, allowing network weight updates to minimize at-sensor radiance error for reflective and emissive materials.

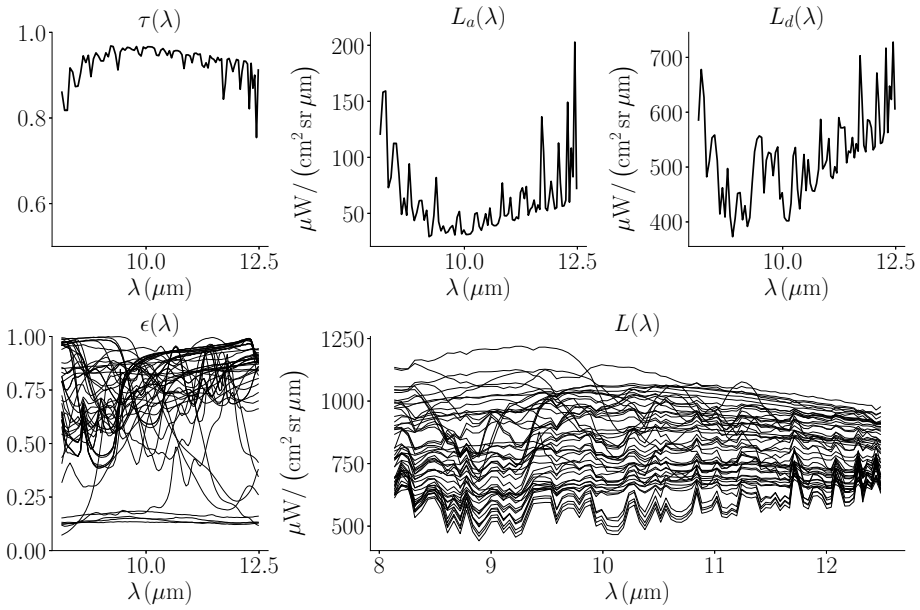


Figure 44. An example of set X is shown in the bottom right plot where $N = 50$. The lowest atmospheric temperature measurement was 296 K for the given TUD vector. The emissivity threshold, ε_t was 0.85, sampled P_E was 0.75, resulting in a mean emissivity less than 0.75 for 25% of the materials and a mean emissivity between 0.75 and 0.85 for 75% of the materials.

5.4.4 Pixel Selection.

Applying the trained model to real data requires selection of N pixels to predict $\hat{\mathbf{z}}$ and ultimately the cube TUD vector. This selection process should be automated, increasing data throughput, while providing reliable results. As shown in Figure 41, the set $\mathbf{X}$ must contain some pixel diversity to extract a set representation. Specifically, if all N pixels are identical, $\phi(\mathbf{X})$ will converge to zero after centering the set $\mathbf{S}$.

To extract N pixels from a real data cube, the mean at-sensor radiance spectrum, $\bar{L}(\lambda)$, is calculated. Next, the spectral angle, θ_i, between pixel i and $\bar{L}(\lambda)$ is calculated:

$$\theta_i = \cos^{-1}\left(\frac{L_i(\lambda)\cdot\bar{L}(\lambda)}{\|L_i(\lambda)\|\,\|\bar{L}(\lambda)\|}\right), \tag{5.14}$$

where $\|\cdot\|$ denotes the l_2 norm and $L_i(\lambda)$ is the at-sensor radiance for pixel i. After sorting all pixels by spectral angle from the mean radiance spectrum, the 10% largest spectral angles are used for pixel selection. First, the lowest spectral angle pixel (90th percentile) is

selected and a one pixel guard band is applied spatially. Any pixels within this guard band are removed from the sorted array and no longer considered for selection. This process is repeated by linearly sampling N pixels across the sorted spectral angle array. Endmember extraction techniques were investigated for pixel selection, but added significant computational overhead without noticeable improvement in algorithm performance. Anomaly detection approaches such as Mahalanobis distance were also considered, but did not yield noticeable improvements while also requiring data covariance calculation.

5.5 Results

This section first presents the AE model results applied to the TIGR data across a range of altitudes. The trained AE model is then used to create the training data samples for fitting the DAC model. After reporting training results for both methods, several different measured hyperspectral data sets are used to verify DAC performance is comparable to FLAASH-IR.

5.5.1 Autoencoder Results.

The relative humidity filtered TIGR data and augmented samples are used to fit the AE model. A hyperparameter sweep was performed across the number of nodes per layer, number of layers, batch size, learning rate, activation functions, latent components and loss functions. The network with minimium brightness temperature RMSE on the validation TUD vectors was selected. Additionally, each model was trained 10 times starting from a random weight initialization and the model with the best mean performance was selected as the best overall architecture. The selected model consisted of a two layer encoder with 4 latent components: 276-48-16-4 where 276 is the TUD vector dimension and 48-16 are the encoder layer dimensions. The decoder is the reverse order of the encoder: 4-16-48-276. The leaky ReLU activation function was used with the at-sensor radiance loss described in

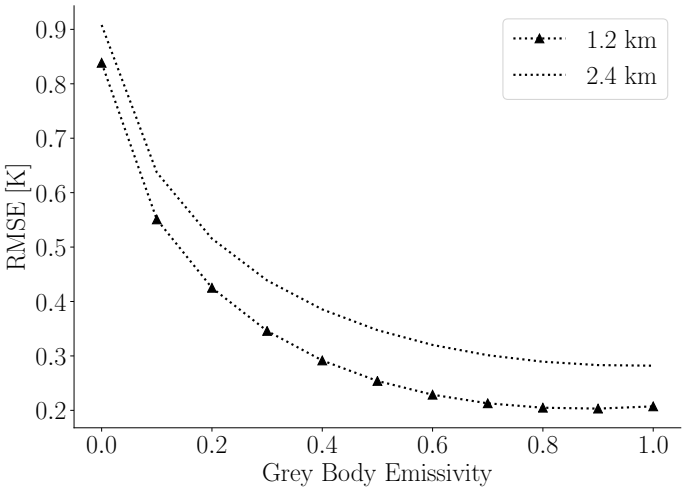


Figure 45. Brightness temperature RMSE is reported for TUD samples never used in training. The sensor altitudes also were never observed in the training data. The AE model generalizes well to these new samples with most errors below 1 K **except for reflective materials (**$\varepsilon(\lambda) = 0$**).**

Equation 5.7. Model training executed for 300 iterations, using a batch size of 64 and a learning rate of 1×10^{-4}.

The RMSE in brightness temperature of the AE model is shown in Figure 45. These results are based on TUD vectors and altitudes not included in the training data, representing model performance when presented new data. Brightness temperature RMSE increases for lower emissivity (higher reflectivity) materials, where errors in transmission and downwelling radiance are mutiplied in the simplified LWIR radiative transfer equation. The errors reported in Figure 45 represent the lowest achievable error of the DAC method since all DAC low-dimensional predictions are transformed through the decoder network.

5.5.2 Synthetic Data Results.

The DAC network results are shown in Figure 46 for the validation TUD samples, emissivity profiles and altitudes. These results highlight the DAC network ability to accurately predict the underlying TUD vector from a set of at-sensor radiance values on new samples. The largest observed errors are in the downwelling radiance because estimating this component is dependent on reflective materials in the scene. To determine how these errors

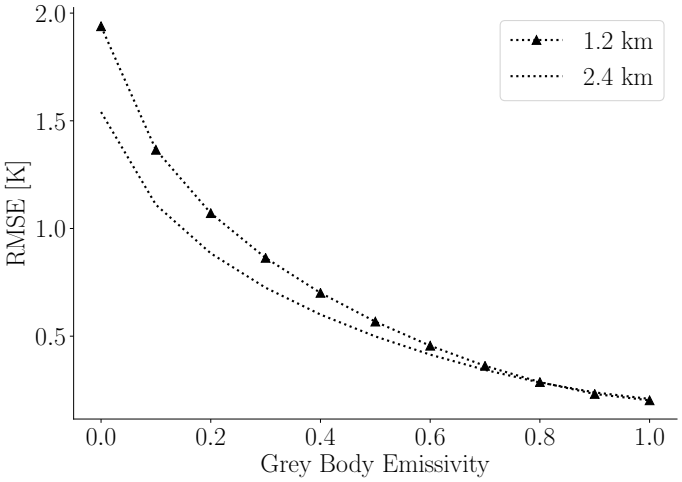


Figure 46. DAC brightness temperature RMSE for the hold out synthetic data is reported as a function of grey body emissivity. The hold out data consisted of TUD vectors never observed in the training data. Additionally, these hold out samples were tested at new altitudes not included in the training set showing the model interpolates to new TUD vectors and altitudes.

impact overall at-sensor error, next we consider scenes with varying scene emissivity and temperature statistics.

To explore the DAC algorithm's dependency on a diversity of pixels, sets of N pixel sets were randomly sampled from the ASTER database with varying scene statistics. The spectrally-averaged emissivity $\bar{\varepsilon}_i$ for a selected emissivity, ε_i, measured across K bands is $\bar{\varepsilon}_i = \frac{1}{K}\sum_{j=1}^{K}\varepsilon_i(\lambda_j)$ and the set mean emissivity of N selected emissivity spectra is $\varepsilon_\mu = \frac{1}{N}\sum_{i=1}^{N}\bar{\varepsilon}_i$. Additionally, the set standard deviation, ε_σ, is calculated according to:

$$\varepsilon_\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\varepsilon_\mu - \bar{\varepsilon}_i}. \tag{5.15}$$

The set mean and standard deviation were calculated for each randomly sampled set. Sampling continued until a range of set means and standard deviations were recorded. The standard deviation represents the diversity of pixels within the scene, while the set mean corresponds to reflective versus emissive scenes.

These N pixel sets were used with varying temperature distributions to determine DAC error. Figure 47 shows the at-sensor error in brightness temperature as a function of set

mean and standard deviation for the TUD validation set. The two plots in Figure 47 correspond to different temperature uniform distribution widths w. Errors decrease with increased mean emissivity because errors in downwelling radiance play a less significant role as emissivity approaches 1.0. Additionally, errors are also reduced as the pixel diversity increases within a scene, supporting that DAC relies on diverse pixels to estimate the TUD vector. These trends are consistent for low temperature variance ($w = 2$) and high temperature variance ($w = 20$) and overall performance is better as temperature variance increases. Additionally, a portion of the error shown in Figure 47 is derived from the AE errors shown in Figure 45. This is because all DAC predictions are transformed through the AE decoder network. The errors shown in Figure 47 represent at-sensor error but don't fully explain individually how $\tau(\lambda)$, $L_a(\lambda)$ and $L_d(\lambda)$ errors vary. Next, additional N pixel sets are created to further identify trends in these errors.

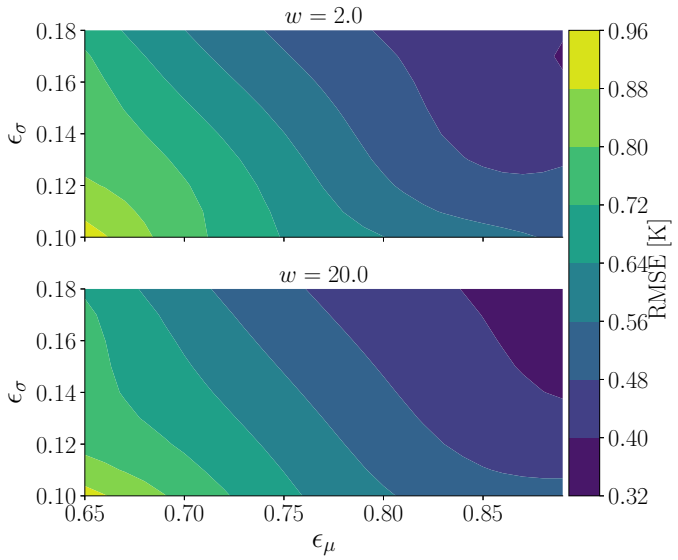


Figure 47. At-sensor brightness temperature error contours are shown for the DAC model for randomly sampled sets of pixel emissivity spectra with overall mean and standard deviations shown. Here w is the uniform distribution width for sampling pixel temperatures. Errors decrease with increasingly diverse sets of pixels and increased mean emissivity as expected from the LWIR radiative transfer equation.

From the simplified LWIR radiative transfer equation, it is expected that downwelling radiance prediction error will increase when reflective materials aren't present in the scene.

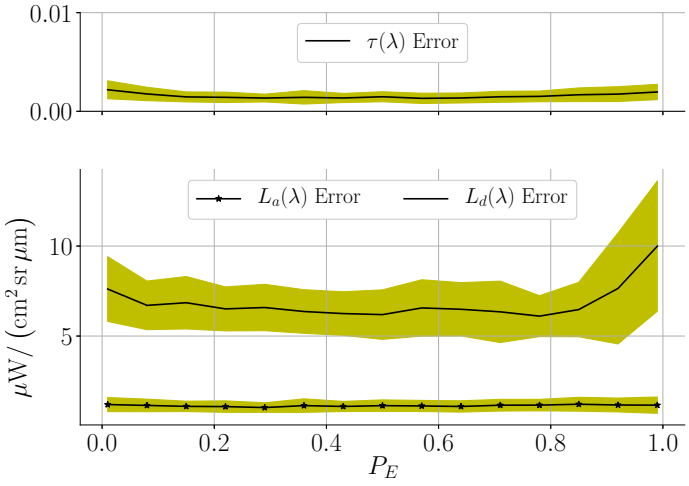


Figure 48. Increasing the percentage of emissive samples, P_E, leads to larger error in the model's downwelling prediction. This error is expected from Equation 5.1, since blackbody materials provide little information to resolve $L_d(\lambda)$. The sets used to generate this plot held $\varepsilon_t = 1.0$ in Algorithm 1.

To verify this observation with the DAC model, synthetic data sets were created containing an increasingly higher percentage of emissive materials, P_E. Additionally, the emissivity threshold in Algorithm 1 was set to 1.0 for set generation. As shown in Figure 48, the downwelling radiance error increases significantly when scenes consist of materials with a mean emissivity greater than 0.9. Also, transmittance and upwelling radiance errors are unaffected by scenes consisting of nearly all blackbody pixels as expected from Equation 5.1. Next, the trained model is applied to collected data cubes to evaluate atmospheric compensation performance in a real-world scenario.

5.5.3 Real HSI Data Results.

This study uses the same data cubes reported in [108] and [107], collected at altitudes ranging from 0.45 km to 2.7 km with the SEBASS LWIR imager. First, we consider a 128 by 5000 pixel cube collected at 0.45 km under clear sky conditions. The collected data contains varying size material panels at different tilt angles and surface roughness. Only flat panels within the scene are considered to evaluate downwelling radiance prediction accuracy. The labeled materials are: Foam Board, Low Emissivity Panel (LowE), Glass,

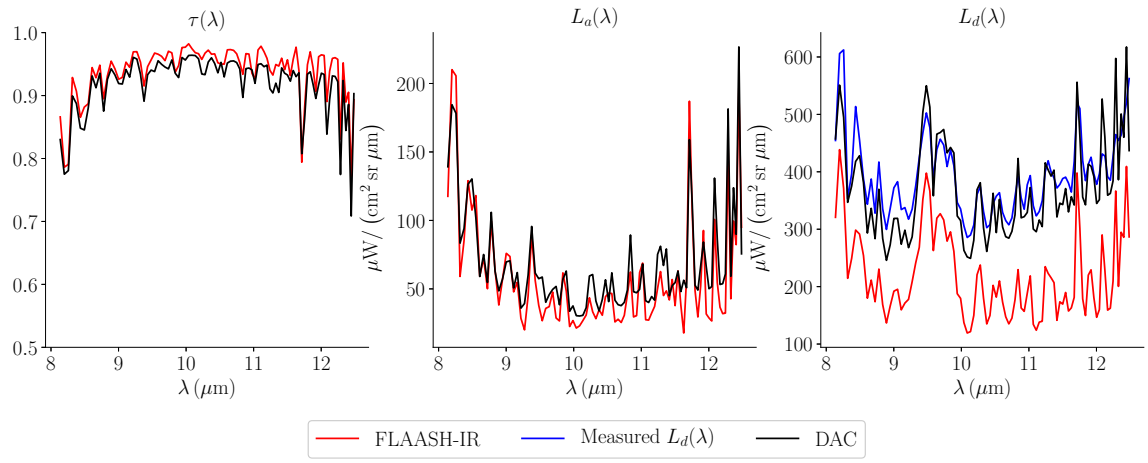


Figure 49. Real data TUD predictions where close agreement is observed for the atmospheric terms, $\tau(\lambda$ and $L_a(\lambda)$, while larger deviations are seen for $L_d(\lambda)$. This cube was collected at 1856L from an altitude of 0.45 km **under clear sky conditions.**

Medium Emissivity Panel (MedE) and Sandpaper. The ground truth emissivity for each material was measured with a D&P spectrometer. Measured emissivity spectra are shown in Figure 51 with emissivity predictions using TES to be discussed later. Additionally, downwelling radiance was also measured with a D&P spectrometer by measuring radiance from an infragold sample.

The first hyperspectral data cube considered was collected at 1856L from an altitude of 0.45 km under clear sky conditions. Predictions from DAC and FLAASH-IR are shown in Figure 49, where the largest difference is in the downwelling radiance component. The DAC $L_d(\lambda)$ prediction closely aligns with the D&P spectrometer measurement demonstrating the ability of this method to extract information from reflective pixels in the scene. Truth data for $\tau(\lambda)$ and $L_a(\lambda)$ are not available, however, these predictions are considered in the total at-sensor radiance error discussed next.

Using the pixel labels to assume a known emissivity, at-sensor radiance error can be calculated if pixel temperatures can be estimated. Pixel temperature, T_i, is determined by

minimizing the MSE between the measured and predicted emissivity:

$$\min_{T_i} \frac{1}{K} \sum_{j=1}^{K} \left(\varepsilon(\lambda_j) - \hat{\varepsilon}_i(\lambda_j) \right)^2, \tag{5.16}$$

where the predicted emissivity is also a function of pixel temperature:

$$\hat{\varepsilon}_i(\lambda)(T_i) = \frac{L_i(\lambda) - \hat{L_a}(\lambda) - \hat{\tau}(\lambda)\hat{L_d}(\lambda)}{\hat{\tau}(\lambda)\left[B(\lambda, T_i) - \hat{L_d}(\lambda)\right]}. \tag{5.17}$$

Radiance predictions are made using the estimated TUD, estimated pixel temperature and measured emissivity. Figure 50 shows the resulting errors in brightness temperature where materials are organized in increasing emissivity from left to right. The number of pixels per material are shown in parenthesis and violin plots are used to display the distribution of errors. For this data cube, the DAC predictions result in lower error across a range of material emissivity spectra, with only small improvements for the highest emissivity material, sandpaper. A log-scale is used in Figure 50 to highlight differences in LowE and MedE errors that approach 0.4 K. Additionally, the estimated temperatures from each compensation method are shown in Table 12 with close agreement observed between both methods.

The previous results used the known pixel emissivity to estimate pixel temperature, however, this is unrealistic in real-world conditions, since pixel emissivity isn't known beforehand. Next, TES is applied to the HSI data to compare compensation performance. A total of 2048 temperatures between 280 K and 350 K are considered to maximize the smoothness of the estimated emissivity spectra with a seven-point local averaging filter based on the method presented in [27]. The mean TES estimated emissivity spectra are shown in Figure 51, where both DAC and FLAASH-IR provide similar estimates. The FLAASH-IR estimates are derived from the TES method described to compare TUD predictions, rather than using the reported emissivity within the FLAASH-IR software.

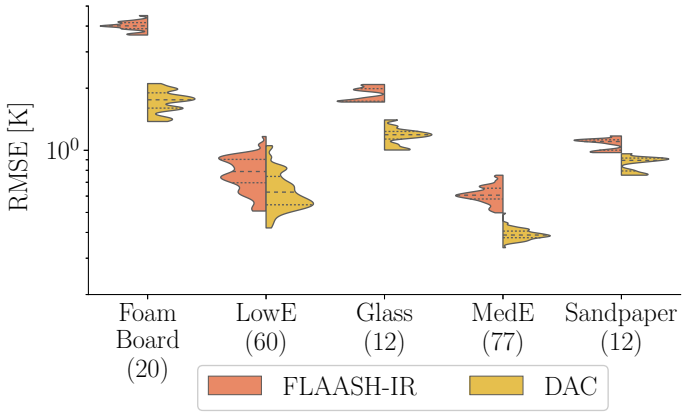


Figure 50. Brightness temperature error is shown when using the measured pixel emissivity and pixel labels to estimate individual pixel temperatures. The materials are organized from left to right in increasing emissivity. The dashes within each plot are the inner quartile range where the thick dashes are the median error. The number of pixels per material are shown in parentheses.

Table 12. Predicted Material Temperatures [K] using pixel labels to minimize emissivity error.

	Foam Board	LowE	Glass	MedE	Sandpaper
Measured	318.27	330.32	321.85	341.79	338.27
FLAASH-IR	335.34	335.81	330.44	344.95	341.15
DAC	330.29	334.17	329.90	344.85	342.12

Numerous target detection algorithms exist differing in background clutter modeling, subpixel replacement strategies and detection statistic calculation. For many algorithms, detection statistics are based on a spectral angle measurement between a known emissivity measurement and the extracted emissivity from TES. Using the TES emissivity estimates, spectral angles are calculated using the measured emissivity spectra with spectral angle error, θ, shown in Figure 52. Lower spectral angle errors for the DAC algorithm are observed across all materials supporting the utility of this approach for target detection scenarios.

The $\phi(\mathbf{X})$ network can use any number of pixels since the max decomposition is performed along the pixel axis. Varying the set size from 5 to 200 pixels and calculating the spectral angle error after conducting TES is shown in Figure 53. Using only 5 pixels does not contain enough information to accurately predict the scene TUD vector leading to

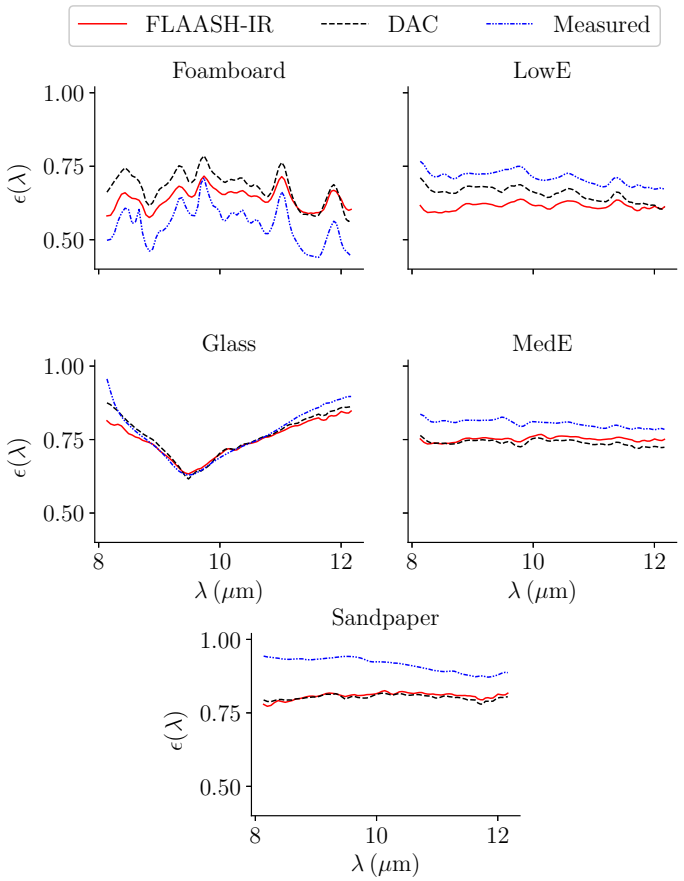


Figure 51. Predicted emissivity curves are shown for both FLAASH-IR and DAC. Emissivity estimates were made using the maximum-smoothness TES technique [27].

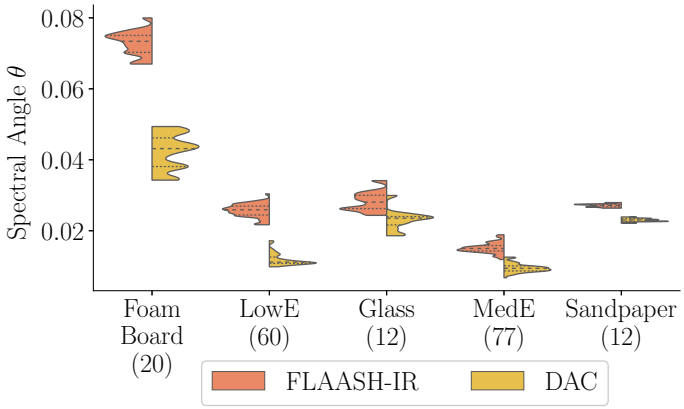


Figure 52. Brightness temperature errors are shown between the two methods where TES was used with each TUD prediction to determine $\varepsilon(\lambda)$ and T. These estimates were forward modeled to determine the at-sensor radiance. Comparable performance is observed, but the DAC method operates in under one second including automated pixel selection.

higher spectral angle error. The model was trained using sets of 50 pixels, however, from Figure 53 only 10 pixels are needed for this cube. For more diverse scenes such as urban areas, additional pixels are expected to further improve TUD prediction accuracy.

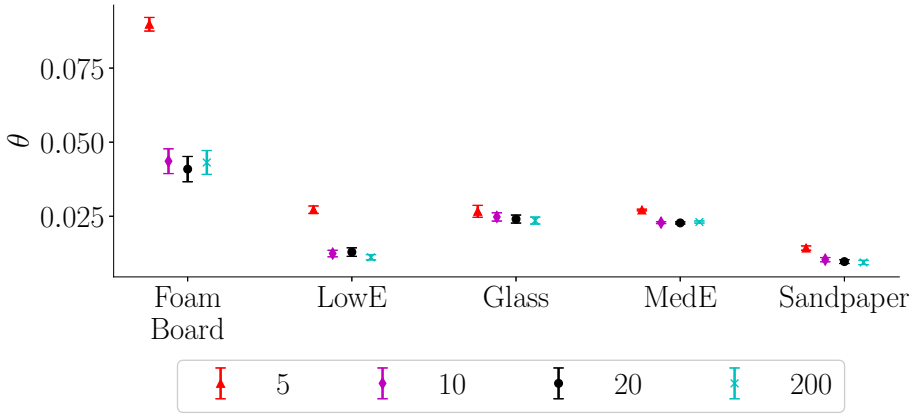


Figure 53. Varying the input set size N between 5 and 200 pixels and performing TES using the predicted TUD vectors demonstrates how set size impacts overall spectral angle error. For this cube 5 pixels is not adequate to correctly predict the TUD vector, but increasing to 10 pixels captures the necessary diversity in the data.

Next, a data cube collected at an altitude of 1.2 km is considered to demonstrate DAC performance at a new altitude. This cube was collected in the afternoon five days after the previous cube collect. Weather was noted as clear sky conditions during this collection. Figure 54 shows the predicted TUD vector and the resulting spectral angle error after applying TES. While radiosonde data is not available to compare atmospheric state vectors between the data cubes, a significant change in $\hat{\tau}(\lambda)$ and $\hat{L_a}(\lambda)$ is noted between the collects.

Finally, inference time is another important factor to consider when deploying these methods in real-world scenarios. The DAC algorithm benefits from accelerated computation using a graphics card, however, to compare inference time between FLAASH-IR and DAC, both methods were tested on an Intel i7-4710MQ processor. Inference time for DAC was on average 0.35 s while FLAASH-IR took approximately 67 s not including lookup table generation. Running DAC on an Nvidia RTX 2060 mobile graphics card reduced

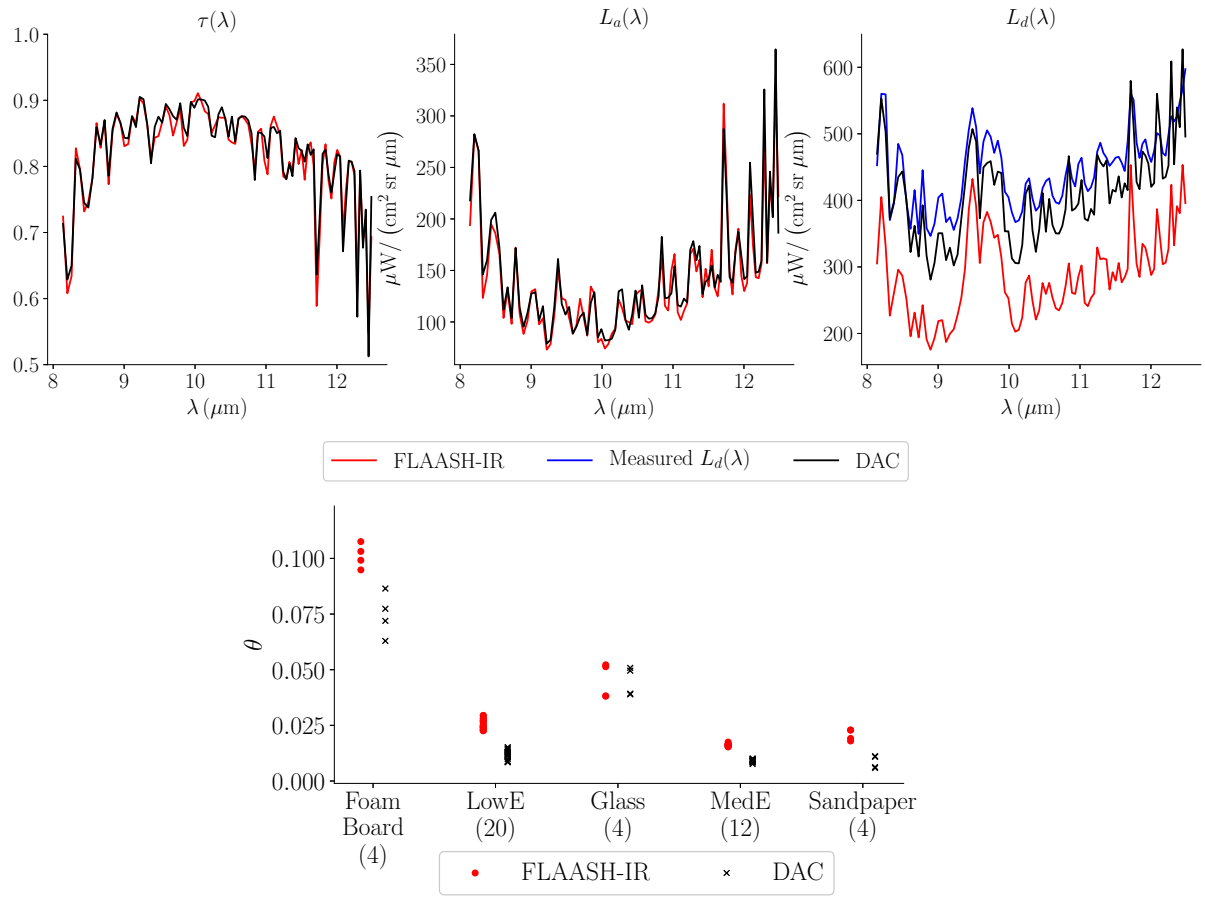


Figure 54. Applying DAC and FLAASH-IR to a data cube collected at 1.2 km **shows good agreement between the two approaches. This data cube was collected at a different time of day from the cube results reported in Figure 49 and 52. The spectral angle errors are based on applying max smoothness TES [27] using the predicted TUD vectors. This cube was collected at 1638L under clear sky conditions 5 days after the previous cube was collected. Violin plots are not shown because the number of pixels per material is significantly smaller at this altitude.**

inference time to 0.24 s. The DAC inference times include automatic pixel selection using the spectral angle method detailed in Equation 5.14.

5.6 Conclusion

The use of in-scene atmospheric compensation algorithms allows for efficient estimation of key components in the LWIR radiative transfer equation, but typically with higher error versus their model-based counterparts. This study has presented a hybrid approach, dependent on previously generated MODTRAN data, but applicable to a wide range of

conditions and altitudes. The inference step only requires in-scene data, without the need for lookup table generation, making this method applicable for real-time predictions. We demonstrated comparable performance to FLAASH-IR with an inference time of 0.24 s using a mobile graphics card. This computational speedup is important for efficiently dealing with the large volumes of data generated by modern LWIR sensors.

A key enabler of the DAC algorithm presented here was the use of permutation-invariant neural network layers. This approach allowed the model to estimate the underlying TUD vector from in-scene data without generating spatially resolved hyperspectral data cubes. Additionally, permutation-invariant layers were necessary to handle the diversity of possible at-sensor radiance pixel sets, derived from varying materials, material temperatures and atmospheric conditions.

The results and analysis presented included both synthetic data and collected HSI confirming this method generalizes to real-world conditions. The entire training pipeline can be retooled for a particular sensor, only requiring a modified ILS for training data generation. There is a wide range of future work in this area, including testing against additional measured HSI data sets, varying types of AE models, pixel selection strategies and modifications to the DAC network. Future work will also consider off-nadir sensor zenith angles and modifications to the neural network architecture to support this additional information in the data compression and TUD estimation steps.

VI. Multimodal Representation Learning and Set Attention for LWIR In-Scene Atmospheric Compensation

6.1 Paper Overview

This paper extends the research in the previous chapter by implementing a Multimodal Autoencoder (MMAE) to combine atmospheric state vector information (T, H_2O, O_3) with Transmittance, Upwelling, and Downwelling (TUD) vector data. The atmospheric compensation algorithm derived in this research can predict both outputs using only in-scene data. The MMAE depends on three unique loss functions to create a smoothly varying latent space. Sampling the MMAE low-dimensional components shows clearly defined attribute vectors such as total column water vapor content. While the MMAE is useful on its own for radiative transfer modeling, this paper also considers new set pooling operations to improve the permutation-invariant network approach presented earlier. The set pooling operation in this paper utilizes an attention mechanism to display which pixels in the scene are most informative for the atmospheric compensation prediction. This provides additional confidence in the model since more attention is paid to reflective pixels to recover the downwelling radiance term. Atmospheric compensation results are compared against Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR) through a target detection study.

This paper has been submitted to the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

6.2 Abstract

A multimodal generative modeling approach combined with permutation-invariant set attention is investigated in this paper to support Long-Wave Infrared (LWIR) in-scene atmospheric compensation. The generative model can produce realistic atmospheric state

vectors (T, H_2O, O_3) and their corresponding TUD vectors by sampling a low-dimensional space. Variational loss, LWIR radiative transfer loss and atmospheric state loss constrain the low-dimensional space, resulting in lower reconstruction error compared to standard Mean-Square Error (MSE) approaches. A permutation-invariant network predicts the generative model low-dimensional components from in-scene data, allowing for simultaneous estimates of the atmospheric state and TUD vector. Forward modeling the predicted atmospheric state vector results in a second atmospheric compensation estimate. Results are reported for collected LWIR data and compared to FLAASH-IR, demonstrating commensurate performance when applied to a target detection scenario. Additionally, an approximate 8 times reduction in detection time is realized using this neural network-based algorithm compared to FLAASH-IR. Accelerating the target detection pipeline while providing multiple atmospheric estimates is necessary for many real-world, time sensitive tasks.

6.3 Introduction

Long wave infrared hyperspectral sensors collect data between 8 - 14 μm across hundreds of contiguous bands, providing detailed information about the Earth's surface and material temperatures. Accurate characterization of surface constituents is important for a wide range of applications such as urban heat island analysis, search and rescue operations and target detection [23, 135, 136]. Fully leveraging thermal hyperspectral data for these applications requires precise atmospheric compensation algorithms for accurate material characterization. Additionally, these compensation methods should be efficient and require minimal user input to operate on the large volumes of data collected by modern sensors. This paper extends previous research in efficient LWIR atmospheric compensation [20], investigating new architectures to form a joint representation of atmospheric measurements and their corresponding radiometric quantities. The major contributions of this paper are:

- The Multimodal DeepSet Atmospheric Compensation (MDAC) architecture is introduced, predicting both atmospheric state ($\mathrm{T}, \mathrm{H_2O}, \mathrm{O_3}$) and the $\tau(\lambda)$, $L_a(\lambda)$, and $L_d(\lambda)$ vectors to support in-scene atmospheric compensation.

- Variational loss and weighted atmospheric state loss are shown to further improve in-scene atmospheric compensation performance against the results presented in [20].

- Set attention pooling is investigated to convey reflective pixels' role in the MDAC prediction. Emphasis of reflective pixels in the atmospheric compensation prediction is in agreement with the LWIR radiative transfer equation.

- Atmospheric compensation errors are compared from the target detection perspective using collected LWIR data, demonstrating comparable performance to FLAASH-IR while reducing total detection time.

In the next section, a review of permutation-invariant neural networks for LWIR atmospheric compensation is discussed. This is followed by an overview of LWIR hyperspectral data processing, necessary for evaluating model errors and characteristics.

6.4 Background

Given N pixels $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, extracted from a data cube collected from an altitude a_s across K bands, $\mathbf{x}_i \in \mathbb{R}^K$, the DeepSet Atmospheric Compensation (DAC) network, $D(\mathbf{X}, a_s)$, predicts a low-dimensional representation, $\mathbf{z}$, of the estimated TUD $(\hat{\tau}(\lambda), \hat{L}_a(\lambda), \hat{L}_d(\lambda))$ vector, $\mathbf{y}_T$ [20]. A decoder network, $d(\cdot)$, transforms $\mathbf{z}$ to $\mathbf{y}_T$, such that

$$\hat{\mathbf{y}}_T = d(D(\mathbf{X}, a_s)) \tag{6.1}$$

The pixel set $\mathbf{X}$ corresponds to a single $\mathbf{y}_T$ vector and the DAC network should provide the same $\mathbf{y}_T$ prediction regardless of the order of the pixels in $\mathbf{X}$. To achieve this functionality,

the DAC network is permutation-invariant to pixel order in $\mathbf{X}$, relying on a set transformation operation $\phi(\cdot)$ and a max pooling operation to form a one-dimensional set feature vector. The low-dimensional $\mathbf{z}$ prediction is made with another prediction network, $\rho(\cdot)$, such that the entire DAC network can be expressed by:

$$D(\mathbf{X}, a_s) = \rho\left(\max_{i \in N}\left[\phi(\mathbf{X})\right], a_s\right). \tag{6.2}$$

Instead of max pooling the transformed set representations created by the $\phi(\cdot)$ network, this research leverages recent advancements in set attention pooling to perform the set decomposition operation [69, 81]. Attention mechanisms are loosely based on how human vision operates: focusing on objects of high importance while blurring background objects. By focusing or attending to the most salient data aspects for a particular task, model performance can be improved while also increasing interpretability [147]. These advantages are achieved through a weighted average where the weights are attention scores that highlight feature importance.

Set attention pooling is a modified attention mechanism used in cases where multiple instances correspond to a single output value [69, 81]. Some samples in the set will contain more information, captured by the set attention scores, and have a stronger influence on the set decomposition operation. Set attention pooling is of interest to the LWIR atmospheric compensation problem because pixels receiving higher attention scores can be further investigated to identify unique spectral properties. This additional interpretability is necessary for validating model performance on a wide range of conditions.

In addition to set attention pooling, this research also extends [20] by investigating a multimodal representation. The decoder network $d(\cdot)$ in [20] utilized the TUD vector data to create the low-dimensional data manifold $\mathbf{z}$, however, this research also utilizes the atmospheric state vector, $\mathbf{y}_A$, creating a MMAE to constrain the data manifold $\mathbf{z}$. Evaluating

the benefits of these modifications requires a review of LWIR hyperspectral data analysis discussed next.

The observed at-sensor radiance, $L(\lambda)$, consists of two factors: surface-leaving radiance, $L_s(\lambda)$, attenuated by atmospheric transmission, and atmospheric emission directly to the sensor. Assuming a lambertian surface, the simplified LWIR radiative transfer equation can be described as [2]:

$$L(\lambda) = \tau(\lambda)L_s(\lambda) + L_a(\lambda) \tag{6.3}$$

where $L_s(\lambda)$ consists of emissive and reflective contributions:

$$L_s(\lambda) = \underbrace{\varepsilon(\lambda)B(\lambda,T)}_{\text{Emissive}} + \underbrace{\big[1-\varepsilon(\lambda)\big]L_d(\lambda)}_{\text{Reflective}}. \tag{6.4}$$

Based on these definitions, the entire simplified at-sensor radiance equation can be described by:

$$L(\lambda) = \tau(\lambda)\Big[\varepsilon(\lambda)B(\lambda,T) + \big[1-\varepsilon(\lambda)\big]L_d(\lambda)\Big] + L_a(\lambda), \tag{6.5}$$

where

λ : wavelength

T : material temperature

$\tau(\lambda)$: atmospheric transmission

$\varepsilon(\lambda)$: material emissivity

$B(\lambda,T)$: Planckian distribution

$L_d(\lambda)$: downwelling atmospheric radiance

$L_a(\lambda)$: atmospheric path (upwelling) radiance

The Planckian distribution is:

$$B(\lambda,T) = \frac{2hc^2}{\lambda^5}\frac{1}{e^{hc/\lambda kT}-1}, \tag{6.6}$$

where c is the speed of light, k is Boltzmann's constant and h is Planck's constant.

The signal of interest in LWIR target detection is the material emissivity defined as a ratio between the radiance emitted at temperature T and the radiance emitted by a blackbody ($\varepsilon(\lambda) = 1$) at the same temperature [14]:

$$\varepsilon(\lambda) = \frac{L(\lambda,T)}{B(\lambda,T)}. \tag{6.7}$$

Retrieving emissivity consists of two steps: atmospheric compensation and Temperature-Emissivity Separation (TES). Atmospheric compensation methods estimate the TUD vector, such that surface leaving radiance can be recovered. Model-based atmospheric compensation approaches rely on radiative transfer models such as MODerate resolution atmospheric TRANsmission (MODTRAN) to predict TUD vectors based on known or estimated atmospheric state information (column water vapor, trace gas content, air temperature) [24, 26]. By generating a look-up table of TUD vectors from expected atmospheric conditions, model-based methods can be implemented efficiently for real-time use [139]. Specifically, methods such as FLAASH-IR modify the surface temperature, water vapor column density and the ozone scaling factor to minimize the error between observed and predicted radiance [26].

In-scene atmospheric compensation methods rely on blackbody pixels to make the compensation problem tractable. The In-Scene Atmospheric Compensation (ISAC) method identifies blackbody pixels allowing at-sensor radiance, $L_{BB}(\lambda)$, to be described by [15]:

$$L_{BB}(\lambda) = \tau(\lambda)B(\lambda,T) + L_a(\lambda). \tag{6.8}$$

Pixel temperature is estimated through clear bands ($\tau(\lambda) \approx 1$), such that the only remaining unknowns are $\tau(\lambda)$ and $L_a(\lambda)$. A linear fit is performed on each spectral channel to determine these terms. The ISAC procedure does not recover the downwelling radiance, important for accurately characterizing reflective materials.

Next, TES is typically performed to estimate both $\hat{\varepsilon}(\lambda)$ and $\hat{T}$. For a sensor with K spectral bands, decoupling these terms is an under-determined problem as there are only K measurements but $K+1$ unknowns $(\hat{\boldsymbol{\varepsilon}}, \hat{T})$. A common approach to this under-determined problem is to assume $\varepsilon(\lambda)$ is a smooth function of wavelength compared to the atmospheric features [27]. Assuming downwelling radiance was estimated during the atmospheric compensation process, emissivity can be estimated as [14]:

$$\hat{\varepsilon}(\lambda) = \frac{\hat{L}_s(\lambda) - \hat{L}_d(\lambda)}{B(\lambda, \hat{T}) - \hat{L}_d(\lambda)}. \tag{6.9}$$

Unfortunately, TES methods recover material temperatures with limited accuracy, leading to increased errors in $\hat{\varepsilon}(\lambda)$ [148]. Unique from TES procedures, researchers have investigated methods to determine $\hat{\varepsilon}(\lambda)$ with less dependence on $\hat{T}$. The alpha residuals approach introduced in [28] and extended in [149] converts a target emissivity, $\varepsilon_t(\lambda)$ to $\alpha_{\varepsilon_t}(\lambda)$ by:

$$\alpha_{\varepsilon_t}(\lambda_i) = \lambda_i \ln\left[\varepsilon_t(\lambda_i)\right] - \frac{1}{K}\sum_{j=1}^{K} \lambda_j \ln\left[\varepsilon_t(\lambda_j)\right]. \tag{6.10}$$

The alpha residual formulation presented in [28] and [149] omits the reflective component in the surface leaving radiance. In [29], the reflective component was included allowing improved emissivity estimation for reflective and emissive materials. In both [149] and [29] an estimate of pixel temperature is needed, but target signal estimation is robust to temperature estimation errors.

Both TES and alpha residual approaches rely on TUD vector estimates derived from the atmospheric compensation process. This study presents an efficient method for in-

scene LWIR atmospheric compensation and compares this method's performance using both TES and alpha residuals from a target detection perspective.

6.5 Methodology

The MDAC model, $D_m(\cdot)$, predicts a low-dimensional representation, $\hat{\mathbf{z}}$, of both the scene atmospheric state vector, $\hat{\mathbf{y}}_A$ and the TUD vector, $\hat{\mathbf{y}}_T$. A multimodal decoder, $d_m(\cdot)$, is used to reconstruct both outputs from $\hat{\mathbf{z}}$ such that the atmospheric compensation and atmospheric state estimation problem can be described by:

$$\hat{\mathbf{y}}_{\mathrm{A}}, \hat{\mathbf{y}}_{\mathrm{T}} = d_m(D_m(\mathbf{X}, a_s)). \tag{6.11}$$

This result depends on the ability of the MDAC model to predict the latent space components $\mathbf{z}$ from the set $\mathbf{X}$ and the decoder model to reconstruct $\mathbf{y}_A$ and $\mathbf{y}_T$ from $\mathbf{z}$. The decoder model is a part of the overall MMAE that is trained prior to fitting the MDAC network. The MMAE model architecture and training is explained in the next section.

6.5.1 Multimodal Generative Models.

This research utilizes the same TUD database and corresponding atmospheric state vectors used in [20]. Specifically, we use the Thermodynamic Initial Guess Retrieval (TIGR) database after filtering for cloud free conditions based on a 96% relative humidity threshold [30, 31]. Following the data augmentation strategy outlined in [20, 141], a total of 8,450 atmospheric state vectors are created. These vectors were forward modeled with MODTRAN 6.0 assuming a nadir sensor zenith angle at altitudes between 0.15 km - 3.05 km resulting in 143,640 TUD vectors. This altitude range spans previously collected data altitudes, allowing for model comparisons with real data. The high resolution TUD vectors created by MODTRAN were downsampled to the Spatially Enhanced Broadband Array Spectrograph

System (SEBASS) instrument line shape (ILS) to create a sensor-specific TUD database [142]. Validation samples were created from held out atmospheric state vectors at unique altitudes from the training data.

A MMAE (Figure 55) is used to compress both the atmospheric state vector and the TUD vector into a joint latent space, $\mathbf{z}$. MMAEs have been investigated in other domains such as audio and video where it is possible to generate one mode from the other [54]. In this research, both modes are always present during training since only the MMAE decoder is used for atmospheric compensation. The MMAE architecture is leveraged to improve feature fusion compared to concatenating the TUD and atmospheric state vectors.

Independent input and output branches combined through joint encoder and decoder networks are used to form the MMAE. The $\mathbf{y}_T$ encoder consists of two layers of 25 and 10 nodes and the $\mathbf{y}_A$ encoder consists of two layers of 20 and 15 nodes. The joint encoder takes the concatenated 10 and 15 node encoder outputs and transforms this representation to the latent space using two layers of 16 and 10 nodes. The latent space is the bottleneck in the representation learning problem, with 6 dimensions considered in this research based on previous results from TUD vector compression [19, 20]. This compression operation is reversed as shown in Figure 55 to create the decoder model.

Interpolations across the latent space should lead to semantically smooth variations in both atmospheric state and TUD outputs. This is a necessary property to support MDAC latent space sampling and is achieved by enforcing a prior distribution on the latent space. This research applies a Gaussian prior such that $\mathbf{z} \sim p(\mathbf{z}) = \mathrm{N}(\mathbf{0}, \mathbf{I})$. This constraint is used in Variational Autoencoders (VAEs) [17] and was extended in [68] for multiple modalities to define a joint multimodal VAE. Given the atmospheric state vector $\mathbf{y}_A$ and the TUD vector $\mathbf{y}_T$, the joint multimodal VAE generative processes for these modes are [68]:

$$\mathbf{y}_A, \mathbf{y}_T \sim p(\mathbf{y}_A, \mathbf{y}_T \mid \mathbf{z}) = p_{\theta_A}(\mathbf{y}_A \mid \mathbf{z}) p_{\theta_T}(\mathbf{y}_T \mid \mathbf{z}) \tag{6.12}$$

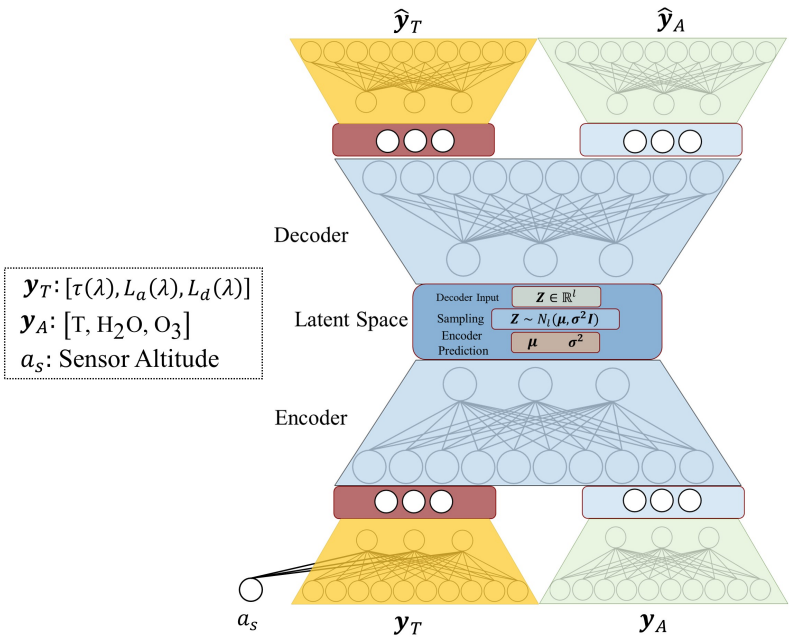


Figure 55. TUD vectors are compressed by the encoder into the latent space and then reconstructed by the decoder network. Reconstruction error is minimized through weight updates during the training process. Additionally, a scalar altitude input is also presented with the TUD vector allowing the model to scale to multiple altitudes.

where the parameter θ represents the decoder network for each mode. The encoder network, q_ϕ, predicts distribution parameters $\boldsymbol{\mu} \in \mathbb{R}^{1\times c}$, $\boldsymbol{\sigma} \in \mathbb{R}^{1\times c}$ for a latent space with c components. Using the reparameterization trick introduced in [17], the posterior $\mathbf{z} \sim q_\phi(\mathbf{z} \,|\, \mathbf{y}_A, \mathbf{y}_T)$ can be sampled according to $\boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\varepsilon}$ where $\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$. To enforce the prior distribution on the latent components, the Kullback-Leibler (KL) divergence is calculated according to [17]:

$$\mathscr{L}_{KL}\left(q_\phi(\mathbf{z} \,|\, \mathbf{y}_A, \mathbf{y}_T) \parallel p(\mathbf{z})\right) = \frac{1}{2} \sum_{j=1}^{d} \left(1 + \log\left(\sigma_j^2\right) - \mu_j^2 - \sigma_j^2\right). \tag{6.13}$$

While $\mathscr{L}_{KL}$ enforces a prior distribution on the latent components, atmospheric state and TUD vector reconstruction error must also be minimized to provide a useful model.

Similar to previous work [20, 141], the TUD vector reconstruction error is minimized using

$$\mathscr{L}_T(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{3K}\sum_{i=1}^{3K}(\hat{y}_i - y_i)^2 + \frac{\gamma}{MK}\sum_{j=1}^{M}\sum_{i=1}^{K}\left(L_{\hat{\mathbf{y}}}(\lambda_i,\varepsilon_j) - L_{\mathbf{y}}(\lambda_i,\varepsilon_j)\right)^2, \tag{6.14}$$

where $\mathbf{y}$ is the truth TUD vector and $\hat{\mathbf{y}}$ is the reconstructed vector. K is the number of spectral channels, $L_{\hat{\mathbf{y}}}(\lambda_i,\varepsilon_j)$ and $L_{\mathbf{y}}(\lambda_i,\varepsilon_j)$ are the at-sensor radiance values for a grey-body emissivity ε_j. A linear sampling of M grey-body emissivity values between 0 and 1 are used to calculate loss, improving reconstruction error for reflective and emissive materials. The hyperparameter γ is a regularization term controlling the relative importance between the TUD MSE and the at-sensor radiance MSE within the loss function.

Atmospheric state error is minimized using a weighted MSE loss function described by:

$$\mathscr{L}_A(\hat{\mathbf{y}},\mathbf{y}) = \frac{1}{3p}\sum_{i=1}^{3p} w_i(\hat{y}_i - y_i)^2 \tag{6.15}$$

where the weights $\mathbf{w} \in \mathbb{R}^{1\times 3p}$ are derived from the atmospheric pressure levels leading to the largest deviation in at-sensor radiance. To identify these pressure level dependent deviations, a Jacobian matrix is calculated between at-sensor radiance and each measurement vector. Each pressure level measurement is modified by 1% of the training data mean value resulting in the Jacobian matrix described in Equation 6.16:

$$\mathbf{J_L}(\mathbf{M}) = \begin{bmatrix} \frac{\partial L(\lambda_1)}{\partial M(a_1)} & \cdots & \frac{\partial L(\lambda_K)}{\partial M(a_1)} \\ \vdots & \ddots & \vdots \\ \frac{\partial L(\lambda_1)}{\partial M(a_p)} & \cdots & \frac{\partial L(\lambda_K)}{\partial M(a_p)} \end{bmatrix} \tag{6.16}$$

where $\mathbf{M}$ represents the particular measurement $(\mathrm{T}, \mathrm{H_2O}, \mathrm{O_3})$. The mean absolute change in at-sensor radiance across all bands for a particular pressure level, p, and measurement

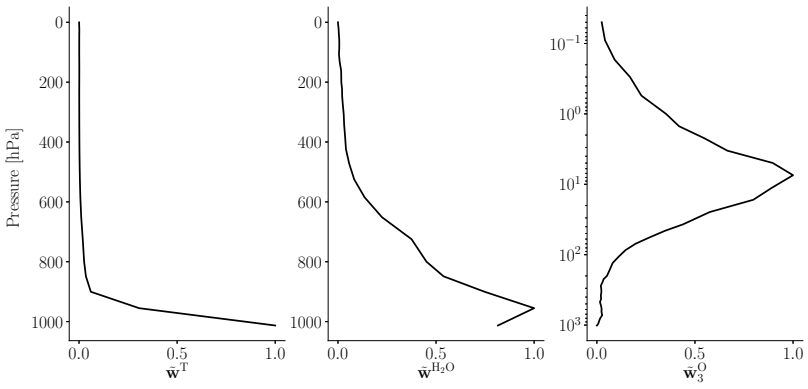


Figure 56. The atmospheric state weighted MSE loss function utilizes the concatenated weight vectors shown. These weights allow the model to accurately predict atmospheric measurements that have the largest impact on the generated TUD vector. Both temperature and water vapor content must be reconstructed correctly at low altitudes (high pressure levels), while ozone concentration has the largest impact at high altitudes.

vector M is calculated according to:

$$w_p^{\mathrm{M}} = \frac{1}{K}\sum_{i=1}^{K} |\mathbf{J}_{L_i}(\mathbf{M}_p)|. \tag{6.17}$$

Next, $\mathbf{w}^M$ is normalized between 0 and 1 across p pressure levels to form $\tilde{\mathbf{w}}^M$. Each normalized measurement weight vector is concatenated to create $\mathbf{w}$ in Equation 6.15 such that $\mathbf{w} = [\tilde{\mathbf{w}}^T, \tilde{\mathbf{w}}^{\mathrm{H_2O}}, \tilde{\mathbf{w}}^{\mathrm{O_3}}], \mathbf{w} \in \mathbb{R}^{1\times 3p}$. The result of this process is shown in Figure 56 agreeing with typical concentration variation of water vapor and ozone at the altitudes shown. Similarly, temperature profiles can often be fit using only surface temperature and lapse rate [15]. The weight $\tilde{\mathbf{w}}^T$ captures this behavior by emphasizing only the measurements closest to the surface.

The total MMAE network loss is calculated by combining each mode loss and the latent space KL loss:

$$\mathscr{L}(\hat{\mathbf{y}}_A, \mathbf{y}_A, \hat{\mathbf{y}}_T, \mathbf{y}_T) = \mathscr{L}_A(\hat{\mathbf{y}}_A, \mathbf{y}_A) + \mathscr{L}_T(\hat{\mathbf{y}}_T, \mathbf{y}_T) + \beta \mathscr{L}_{KL}\big(q_\phi(\mathbf{z}\,|\,\mathbf{y}_A, \mathbf{y}_T) \parallel p(\mathbf{z})\big) \tag{6.18}$$

where β is used to trade off reconstruction accuracy against enforcing the prior distribution. The inclusion of β is based on [150] where interpretable latent space components can be recovered if the data generating processes are understood. This research leverages this modification to create an interpretable latent space, capturing variables such as atmospheric water vapor content and atmospheric temperature, allowing new samples to be generated with known properties.

Each layer in the MMAE performs a transform with the function $y = f(\mathbf{wx} + \mathbf{b})$ where $f(\cdot)$ is the activation function, $\mathbf{w}$ is the layer weight matrix and $\mathbf{b}$ is the layer bias vector. The MMAE implemented here utilizes the exponential linear unit (ELU) activation function:

$$\mathrm{ELU}(x) = \begin{cases} x, & \text{if x} > 0 \\ \alpha(\exp(x) - 1), & \text{if x} \leq 0 \end{cases}$$

The activation for predicting $\boldsymbol{\mu}$ is linear and the activation for predicting $\boldsymbol{\sigma}$ is $\mathrm{ELU}(x) + 1$ to guarantee positive variances. Additionally, each mode's output layer utilizes a linear activation function.

6.5.1.1 Generative Model Metrics.

Evaluating the MMAE performance on hold out samples is necessary to determine if the model has generalized to the underlying relationships in the data or over fit to the training

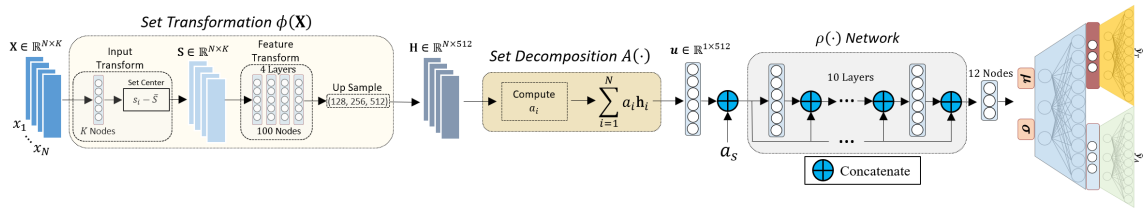


Figure 57. The MDAC network consists of a set transformation, set decomposition and a network $\rho(\cdot)$ for predicting the MMAE latent components. The set transformation converts the input set X to the set H using the input transform and feature transform shown. The set H is converted into the set representation vector u with the attention pooling operation $A(\cdot)$. Sensor altitude, a_s, is concatenated to u before entering the $\rho(\cdot)$ network.

samples. The hold out samples considered here consist of TUD vectors and atmospheric state vectors never encountered in the training data. Additionally, the validation sensor altitudes were never observed in the training set. To measure hold out sample performance with respect to at-sensor radiance error, a range of grey-body emissivity values, ε, with an assumed pixel temperature of 300 K are used to create simulated at-sensor radiance spectra, $L(\lambda,\varepsilon)$. Since this study is focused on the LWIR domain, spectral radiance values were converted to brightness temperature, $T_{BB}(\lambda,\varepsilon)$:

$$T_{BB}(\lambda,\varepsilon) = \frac{hc}{\lambda k \ln\left(\frac{2hc^2}{\lambda^5 L(\lambda,\varepsilon)} + 1\right)}. \tag{6.19}$$

Using $\mathbf{y}_T$ and $\hat{\mathbf{y}}_T$ to create $L(\lambda,\varepsilon)$ and $\hat{L}(\lambda,\varepsilon)$ respectively, the root mean square error (RMSE) in degrees Kelvin can be calculated with:

$$E(\varepsilon) = \sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(T_{BB}(\lambda_i,\varepsilon) - \hat{T}_{BB}(\lambda_i,\varepsilon)\right)^2} \tag{6.20}$$

The grey body emissivity is varied from 0 to 1 producing an RMSE curve describing overall performance between reflective and emissive materials. Additionally, MODTRAN [24] can be used to convert $\hat{\mathbf{y}}_A$ to a TUD vector, resulting in the same error metric for the atmospheric state prediction. When multiple models are compared at once, the brightness temperature RMSE area under the curve (AUC-BT) is reported to capture reflective to emissive performance with a single scalar value:

$$\text{AUC-BT} = \int_{0.0}^{1.0} E(\varepsilon)d\varepsilon \tag{6.21}$$

Since the AUC-BT metric measures RMSE across reflective to emissive materials, lower values represent better reconstruction performance with perfect reconstruction represented by AUC-BT $= 0$.

6.5.2 Set Attention for In-Scene Atmospheric Compensation.

The MDAC model utilizes the MMAE decoder model to predict $\hat{\mathbf{y}}_A$ and $\hat{\mathbf{y}}_T$, from a set of pixels, $\mathbf{X}$. This set-input learning has been investigated in domains such as point cloud classification where a set of points correspond to a single target value or class label [69, 75, 143]. An important characteristic of methods solving set-input learning problems is permutation-invariance to the points in the set. Regardless of pixel selection order, the MDAC algorithm must still provide the same TUD and atmospheric state prediction.

Permutation-invariant predictions are made by the MDAC network using two operations: set transformation and set decomposition. In this study, the set transformation operation is a neural network consisting of an input transform and feature transform as shown in Figure 57. The input transform consists of a K node layer to transform each pixel identically, followed by a set centering operation. The weights in the K node layer are shared across all pixels, maintaining permutation invariance. The feature transform utilizes 4 layers each with 100 nodes, again sharing weights across all pixels. The set transformation concludes with pixel representation upsampling to create the set $\mathbf{H}$:

$$\mathbf{H} = \phi(\mathbf{X}), \quad \mathbf{H} \in \mathbb{R}^{N \times M} \tag{6.22}$$

where $M = 512$ from the upsampling layer. The rows of $\mathbf{H}$ correspond to transformed pixel representations $\mathbf{h}_i$ which must be pooled together by the set decomposition operation. To understand the role each pixel plays in the overall model prediction, this study investigates set attention pooling [81]:

$$\mathbf{u} = \sum_{i=1}^{N} a_i \mathbf{h}_i, \quad \mathbf{u} \in \mathbb{R}^{1 \times M} \tag{6.23}$$

where $\mathbf{u}$ is the set representation vector and a_i is the attention score for pixel i calculated according to:

$$a_i = \frac{\exp\left(\mathbf{w}^T\left(\tanh\left(\mathbf{V}\mathbf{h}_i^T\right) \odot \text{sigm}\left(\mathbf{U}\mathbf{h}_i^T\right)\right)\right)}{\sum_{j=1}^{N}\exp\left(\mathbf{w}^T\left(\tanh\left(\mathbf{V}\mathbf{h}_j^T\right) \odot \text{sigm}\left(\mathbf{U}\mathbf{h}_j^T\right)\right)\right)} \tag{6.24}$$

The trainable parameters are $\mathbf{w} \in \mathbb{R}^{1\times L}$, $\mathbf{V} \in \mathbb{R}^{L\times M}$ and $\mathbf{U} \in \mathbb{R}^{L\times M}$, where L corresponds to the attention pooling dimension. The value of L is varied as part of the overall network hyperparameter sweep with the results in this study using $L = 512$. In Equation 6.24, $\tanh(\cdot)$ corresponds to the hyperbolic tangent function, $\text{sigm}(\cdot)$ is the sigmoid function and $\odot$ is an Hadamard product. The set pixel representations are initially transformed by matrix $\mathbf{V}$ which is learned through the training process. The $\tanh(\cdot)$ operation is approximately linear between -1 and 1 and so the $\text{sigm}(\cdot)$ function is used as a gating function to model more complex dependencies [81, 151]. The matrix $\mathbf{U}$ controls the gating mechanism and is also learned through the training process. The vector $\mathbf{w}$ converts the pixel representation into a scalar value that is used in the overall softmax function to create the attention weights a_i which sum to 1.

The set representation vector $\mathbf{u}$ captures information necessary to predict $\hat{\mathbf{y}}_A$ and $\hat{\mathbf{y}}_T$, however, to create a multi-altitude model the sensor altitude a_s is concatenated to $\mathbf{u}$. This concatenated vector forms the input to the $\rho(\cdot)$ network, which predicts the low dimensional components of the MMAE model, $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$. The $\rho(\cdot)$ network consists of 10 layers each with 100 nodes utilizing skip connections to propagate the set representation vector to deeper layers as shown in Figure 57. Similar to the MMAE model, the $\rho(\cdot)$ output layer utilizes a linear activation for predicting $\boldsymbol{\mu}$ and $\text{ELU}(x)+1$ for predicting $\boldsymbol{\sigma}$. The output layer has 12 nodes because the first 6 outputs are for $\hat{\boldsymbol{\mu}}$ and the last 6 are for $\hat{\boldsymbol{\sigma}}$. Denoting the attention weighted sum in Equation 6.23 as A, the MDAC network can be specified as:

$$D_m(\mathbf{X}, a_s) = \rho\left(A(\phi(\mathbf{X})), a_s\right). \tag{6.25}$$

The network configuration shown in Figure 57 was the result of a hyperparameter sweep over possible set transformation networks, $\rho(\cdot)$ networks and the number of attention nodes in the set decomposition. Additionally, batch size, learning rate, and activation functions were varied in the hyperparameter sweep. The results presented here utilize a learning rate of 1×10^{-3} and a batch size of 512. The number of pixels in each training set was $N = 50$ and so for a single batch, 512 sets were presented to the network (25,600 pixels). The Adam optimization algorithm was used for calculating weight updates [17]. Networks were constructed using Python 3.6.8, Keras version 2.2.4, Tensorflow 1.15 and hyperparameter sweeps were conducted across 20 Graphical Processing Units (GPUs) using Ray Tune version 0.7.6 [145] [146].

6.5.3 Algorithm Training.

The MDAC algorithm is trained using sets of at-sensor radiance data $\mathbf{X}$ created from an underlying TUD vector and atmospheric state vector. The same TUD and atmospheric state data are used to fit MMAE and MDAC models. Training the MDAC algorithm follows the strategy outlined in [20], with the exception that MDAC has multiple outputs requiring additional loss calculations. Emissivity profiles are sampled from the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) database with 200 emissivity samples held out for model validation and 978 different material profiles used during training. Emissivity selection and pixel temperature assignment follows the set generation algorithm outlined in [20]. During training, the at-sensor radiance set $\mathbf{X}$ contains $N = 50$ pixels resulting in $\binom{978}{50} = 3 \times 10^{84}$ possible training emissivity sets.

Only the MDAC weights are updated during training, leaving the MMAE weights unchanged. The MDAC weights are updated based on the $\mathbf{y}_A$ and $\mathbf{y}_T$ error using the loss functions $\mathscr{L}_A$ and $\mathscr{L}_T$, respectively. The same atmospheric weights, w_i, are again used

to calculate the loss on $\mathbf{y}_A$ reinforcing atmospheric state reconstruction at pressure levels impacting the predicted TUD vector.

6.5.4 Pixel Selection.

Accurate MDAC prediction is predicated on access to a set of diverse pixels with respect to emissivity and temperature. To select N diverse pixels from a collected data cube, this study follows the pixel selection strategy outlined in [20] where the spectral angle, θ_i, between pixel i and the cube mean, $\bar{L}(\lambda)$, is calculated according to:

$$\theta_i = \cos^{-1}\left(\frac{L_i(\lambda)\cdot\bar{L}(\lambda)}{\|L_i(\lambda)\|\,\|\bar{L}(\lambda)\|}\right) \tag{6.26}$$

An iterative pixel selection strategy is employed starting with the 90^{th} percentile pixel with respect to sorted cube spectral angles. A one pixel guard band is applied spatially, removing all neighboring pixels from being included in the set $\mathbf{X}$. A uniform sampling of the 10% highest spectral angles is conducted following this procedure resulting in N diverse pixels with respect to the cube mean. Prior to pixel selection, anomalous pixels such as those from dead pixels, are removed from the sorting process. These noisy pixels may not follow the simplified radiative transfer model leveraged in this work and are eliminated from atmospheric compensation consideration.

6.5.5 Target Detection Analysis.

After sampling a collected data cube using the method presented in Equation 6.26, the MDAC predictions can be used to compensate a data cube and perform target detection. The target detection method used in this study is the Adaptive Coherence/Cosine Estimator

(ACE) detector defined by [106]:

$$r_{ACE}(\mathbf{x}) = \frac{(\mathbf{s}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{x})^2}{(\mathbf{s}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{s})(\mathbf{x}^T\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{x})}, \tag{6.27}$$

where $\mathbf{x}$ is a sample pixel, $\mathbf{s}$ is the target, and $\hat{\boldsymbol{\Sigma}}$ is the estimated background covariance. To estimate $\boldsymbol{\Sigma}$, a Mahalanobis anomaly detector is applied to filter background pixels from possible targets. The Mahalanobis detector can be described by:

$$r_{MD}(\mathbf{x}) = (\mathbf{x}-\hat{\boldsymbol{\mu}})^T\hat{\boldsymbol{\Sigma}}^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}), \tag{6.28}$$

where $\hat{\boldsymbol{\mu}}$ is the cube mean and $\hat{\boldsymbol{\Sigma}}$ is the cube covariance. The detection statistic, $r_{MD}(\mathbf{x})$, is sorted and pixels below the 90^{th} percentile are classified as background. These background pixels are used to form $\hat{\boldsymbol{\Sigma}}$ for the ACE detector. Target detection results can be compared using the Signal to Clutter Ratio (SCR) defined as:

$$\text{SCR} = \frac{\mu(r_t)-\mu(r_b)}{\sqrt{\sigma(r_t)^2+\sigma(r_b)^2}}, \tag{6.29}$$

where $\mu(r_t)$ is the mean detection statistic for target pixels and $\mu(r_b)$ is the mean detection statistic for background pixels. Similarly, the standard deviations of these two classes are calculated with $\sigma(\cdot)$. Large SCR values imply higher detection statistics on target pixels compared to background pixels with little variance among both classes.

6.6 Results

This section first presents the MMAE results and demonstrates the model's ability to generate new atmospheric states and TUD vectors. Next, the MMAE is used as part of the overall MDAC algorithm to perform in-scene atmospheric compensation and atmospheric state estimation. Results are presented for synthetic data to demonstrate model character-

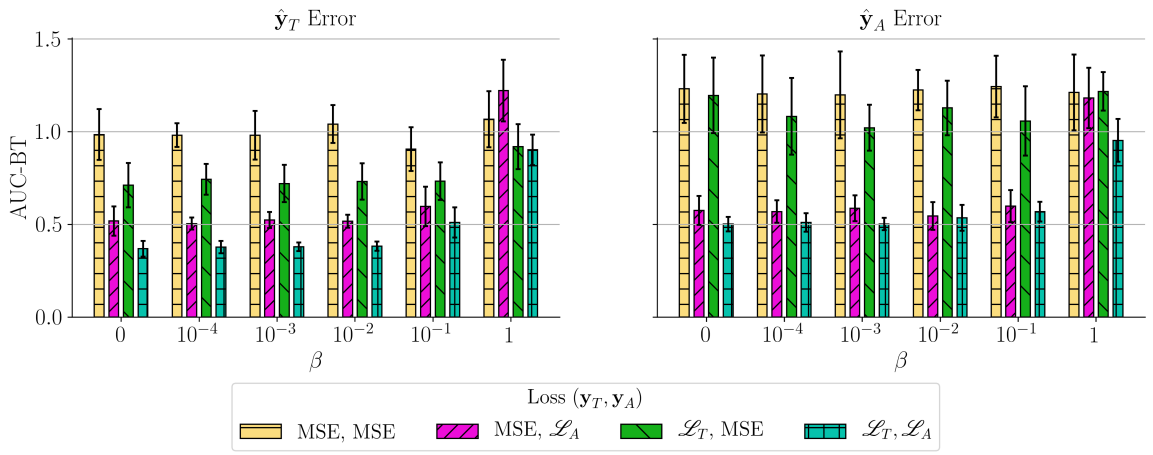


Figure 58. Loss configuration results are shown using the AUC-BT reconstruction error (lower is better) where the best performance is achieved when both $\mathscr{L}_T$ and $\mathscr{L}_T$ are used. As β is increased beyond 0.01, reconstruction error increases because the latent space is overconstrained and no longer has adequate capacity to capture data variability.

istics followed by analysis on SEBASS collected data cubes spanning multiple days and sensor altitudes. Atmospheric compensation results are compared to FLAASH-IR through a target detection comparison.

6.6.1 Multimodal Generative Model Results.

Models utilizing $\mathscr{L}_{KL}$, $\mathscr{L}_A$, $\mathscr{L}_T$ are first compared against models using MSE to demonstrate the benefit of these loss functions in minimizing model reconstruction error. The pairwise model comparisons considered for the MMAE network outputs $(\mathbf{y}_A, \mathbf{y}_T)$ respectively are: (MSE, MSE), (MSE, $\mathscr{L}_A$), ($\mathscr{L}_T$, MSE), ($\mathscr{L}_T$, $\mathscr{L}_A$). Additionally, for each model configuration, $\mathscr{L}_{KL}$ is investigated by varying β from 0.0 to 1.0. Each loss and β configuration result is based on 10 randomly initialized models to provide estimates of model mean performance.

The AUC-BT results for each MMAE output are shown in Figure 58 for all loss configurations and considered β values. Reconstruction errors on $\mathbf{y}_T$ are reduced by using either $\mathscr{L}_A$ or $\mathscr{L}_T$ compared to MSE with the lowest reconstruction error observed when both $\mathscr{L}_A$ and $\mathscr{L}_T$ are used. The $\mathbf{y}_A$ error is not reduced for the ($\mathscr{L}_T$, MSE) case compared to the

baseline MSE model. This is driven by the observation that similar TUD vectors can be created from significantly different atmospheric state vectors. While atmospheric state to TUD vectors is a one-to-one function, TUD vectors to atmospheric state is not.

Figure 58 also highlights the role KL divergence plays in reconstruction accuracy. Increased reconstruction error is observed when $\beta > 10^{-2}$ because the latent components are over-constrained, reducing modeling capacity. From Figure 58, it is not clear which β value should be selected or if KL divergence should even be used since $\beta = 0$ has comparable reconstruction error. Next, latent space continuity is evaluated for each β model, an important attribute for latent space sampling.

Latent space continuity is evaluated by modifying N latent space representations, $\mathbf{z} \in \mathbb{R}^{N \times c}$, and measuring the output deviation in terms of AUC-BT, denoted as ΔAUC-BT. This process is outlined in Algorithm 2, where e is the encoder model, d is the decoder model, $\mathbf{y}_T$ and $\mathbf{y}_A$ are the validation data containing N samples, $\mathbf{\Delta} \in \mathbb{R}^{N \times c}$ is the latent space deviation matrix and ε is a grey body emissivity. The rows of matrix $\mathbf{\Delta}$ are formed by randomly picking points on a hypersphere using [152]:

$$\mathbf{\Delta}_i = \frac{r}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \; x_l \sim \mathrm{N}(0,1) \tag{6.30}$$

where $\|\mathbf{\Delta}_i\| = r$. To make comparable changes to each β model latent space $\mathbf{z}$, Algorithm 2 applies Principal Component Analysis (PCA) whitening to $\mathbf{z}$ resulting in $\tilde{\mathbf{z}}$. After adding $\mathbf{\Delta}$ to $\tilde{\mathbf{z}}$, the whitening process is reversed and the decoder transforms the new latent samples to $\mathbf{y}'_T$ and $\mathbf{y}'_A$. Output deviations are measured according to AUC-BT as shown in Algorithm 2.

Algorithm 2 Latent Space Variation

Input: e, d, $\mathbf{y}_T$, $\mathbf{y}_A$, $\mathbf{\Delta}$, ε
Output: ΔAUC-BT
Modify latent components :
$\mathbf{z} \leftarrow e(\mathbf{y}_T, \mathbf{y}_A)$
$\hat{\mathbf{y}}_T, \hat{\mathbf{y}}_A \leftarrow d(\mathbf{z})$
$\mathbf{\Sigma} \leftarrow \mathrm{E}[(\mathbf{z} - \mathrm{E}[\mathbf{z}])(\mathbf{z} - \mathrm{E}[\mathbf{z}])^T]$
$\mathbf{U}, \mathbf{\Lambda} \leftarrow \text{s.t. } \mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$
$\tilde{\mathbf{z}} = (\mathbf{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{z})$
$\tilde{\mathbf{z}}_\Delta = \tilde{\mathbf{z}} + \mathbf{\Delta}$
$\mathbf{z}' = \mathbf{U}\mathbf{\Lambda}^{1/2}\tilde{\mathbf{z}}_\Delta$
Measure output deviation
$\mathbf{y}'_T, \mathbf{y}'_A \leftarrow d(\mathbf{z}')$
$E(\hat{\mathbf{y}}_T, \mathbf{y}'_T, \varepsilon) \leftarrow \sqrt{\frac{1}{K}\sum_{i=1}^{K}\left(T_{BB}(\lambda_i, \varepsilon) - \hat{T}_{BB}(\lambda_i, \varepsilon)\right)^2}$
$\Delta\text{AUC-BT} \leftarrow \int_{0.0}^{1.0} E(\hat{\mathbf{y}}_T, \mathbf{y}'_T, \varepsilon) d\varepsilon$
return ΔAUC-BT

Applying Algorithm 2 to each β model results in the ΔAUC-BT shown in Figure 59 where smaller output deviations are observed for larger β values. The right axis of Figure 59 shows the validation reconstruction error for the $(\mathscr{L}_T, \mathscr{L}_A)$ loss configuration from Figure 58. When $\beta > 10^{-2}$, KL divergence loss begins to negatively affect reconstruction error as the latent space is over-constrained. In this research, $\beta = 10^{-2}$ is selected to trade off a continuous latent space and low reconstruction error.

Many generative model studies have investigated latent space attribute vectors allowing for new samples to be generated with certain properties such as images of faces wearing sunglasses or smiling [153, 154]. Varying the MMAE latent space components reveals analogous attribute vectors allowing atmospheric state conditions to be precisely controlled. The MMAE model using $\beta = 10^{-2}$, $\mathscr{L}_A$ and $\mathscr{L}_T$ is used to identify one such attribute vector. A single latent component is varied from -3.0 to -1.0 while all other components are unchanged resulting in the atmospheric measurements and TUD vectors shown in the Appendix. The predicted atmospheric measurements show significant changes in the total water vapor content as a single component is varied with corresponding changes in the

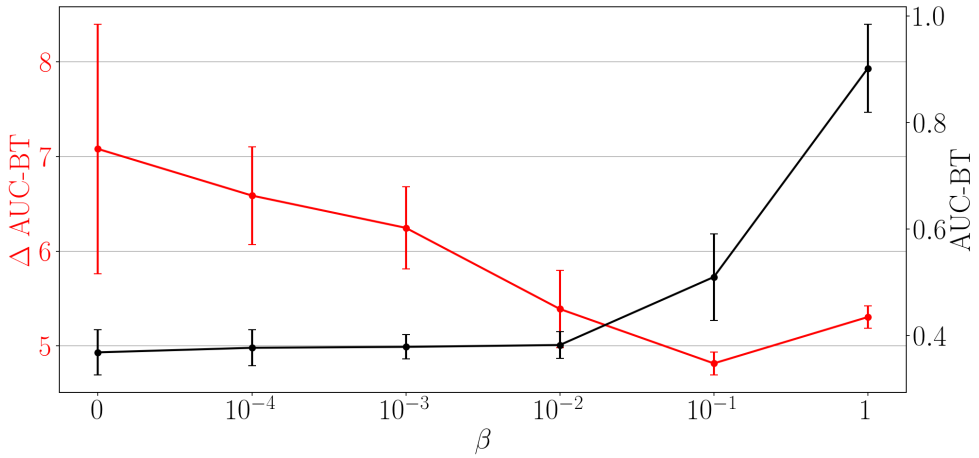


Figure 59. Making small changes to latent components and measuring the change in the $\hat{\mathbf{y}}_T$ is plotted on the left axis. The model validation performance is shown on the right axis, also shown in Figure 58 for the $(\mathscr{L}_T, \mathscr{L}_A)$ configuration. Increasing β results in a more continuous latent space as shown by the decreasing ΔAUC-BT values. However, increasing β beyond 10^{-2} over-constrains the latent space resulting in poor validation performance (right axis). By selecting $\beta = 10^{-2}$, the MMAE has both a continuous latent space and low reconstruction error.

predicted TUD output. Interestingly, as water vapor content increases, atmospheric temperature also increases, supporting the relative humidity threshold set as part of the training data selection. This particular range in the latent space varies outputs from cold, dry atmospheric conditions to warmer, humid conditions. Sampling additional points in this region of the data manifold is useful for a range of applications such as radiative transfer modeling and data augmentation. Next, the joint, low-dimensional representation created by the MMAE will be used for in-scene atmospheric compensation.

6.6.2 Atmospheric Compensation with Synthetic Data.

Using the previously fit MMAE network, the MDAC network was trained to predict the low-dimensional representation $\mathbf{z}$ from a set of at-sensor radiance samples, $\mathbf{X}$. At-sensor radiance sets were generated based on the set generation algorithm presented in [20]. Using a batch size of 512 and set size of $N = 50$, training executed for 50 epochs using training data created from the set generation algorithm. At the conclusion of 50 epochs, new training data was generated, with this process repeated 60 times. During each

50 epoch training iteration, error was gradually reduced as the model was fit to the new data. We found that 60 iterations of this training process resulted in stable errors, even when the model was presented new at-sensor radiance sets.

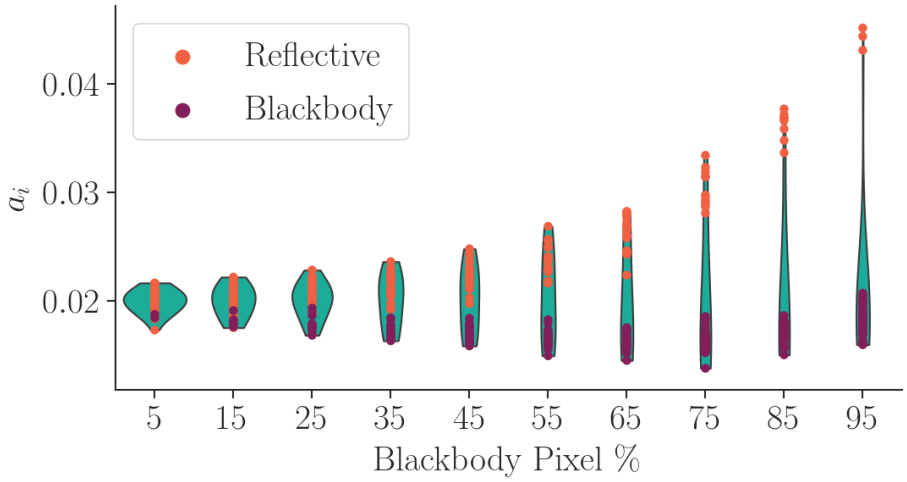


Figure 60. At-sensor radiance sets were created with an increasing percentage of blackbody pixels. The attention scores for reflective scenes (low blackbody pixel %) are small and clustered together while scenes containing only a few reflective pixels have larger attention scores to emphasize the importance of the reflective pixels.

The MDAC network relies on attention pooling to convert the pixel set $\mathbf{X}$ into the set representation vector, $\mathbf{u}$. The attention weights, a_i, represent the importance of each pixel in forming the set representation. To evaluate data characteristics the attention pooling operation has learned, at-sensor radiance sets were generated with varying blackbody pixel percentage within the scene. These synthetic scenes were used to evaluate the attention weights with the results shown in Figure 60.

Reflective material dominated scenes (low blackbody percentage), result in low magnitude, tightly clustered, attention scores because multiple pixels contain information necessary for recovering the scene TUD vector. For low blackbody percentage scenes, the blackbody pixels are unique with respect to the set, however, the attention score is still low for these pixels because no additional downwelling information is provided. As the generated scenes change from reflective material dominated to emissive material dominated

(large blackbody percentage), the overall attention magnitude increases. The remaining reflective pixels are important for downwelling radiance estimation and receive a larger attention score.

6.6.3 Collected HSI Data Results.

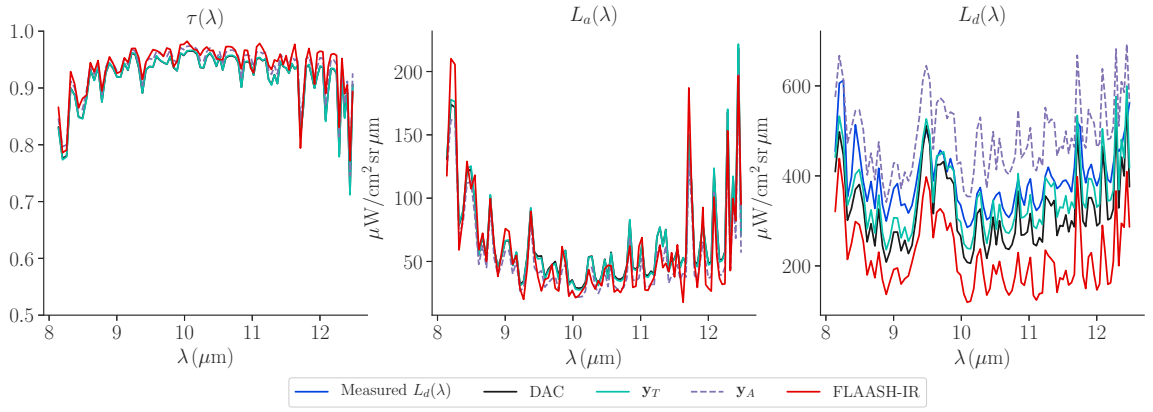


Figure 61. Applying MDAC to a collected data cube results in the two TUD predictions $\mathbf{y}_A$ and $\mathbf{y}_T$ shown. The $\tau(\lambda)$ and $L_a(\lambda)$ estimates are comparable for all methods. As expected, the largest model discrepancy is in the downwelling estimate, which relies on the selection of reflective pixels to estimate this term. This cube was collected at 1856L from an altitude of 0.45 km **under clear sky conditions.**

This study uses the same data cubes reported in [20, 107, 108], collected at altitudes ranging from 0.45 km to 2.7 km with the SEBASS LWIR imager. The collected data contains varying size material panels at different tilt angles and surface roughness, however, only flat panels within the scene are considered to evaluate downwelling radiance accuracy. The labeled materials are: Foam Board, Low Emissivity Panel (LowE), Glass, Medium Emissivity Panel (MedE) and Sandpaper. The ground truth emissivity for each material was measured with a D&P spectrometer and downwelling radiance was measured using an infragold sample.

The first cube considered was collected at 0.45 km under clear sky conditions. Predictions for FLAASH-IR, DAC [20], and each output of MDAC are shown in Figure 61. It is important to note the $\mathbf{y}_A$ prediction in Figure 61 is based on the model's atmospheric state prediction (T, H_2O, O_3) converted to a TUD vector using MODTRAN. While no ra-

diosonde data is available to directly compare the atmospheric state prediction, this atmospheric state estimate does result in a similar TUD vector using only in-scene data. Next, the TUD estimates are compared from a target detection perspective, using both TES [27] and improved alpha residuals (AR) [29].

6.6.4 Target Detection Results.

In many scenarios, the object temperature is of less interest than the object emissivity, such as geological studies or military target detection. To support these applications, the improved AR approach outlined in [29] is used for comparing detection performance. In domains where material temperature is important, the commonly used maximum smoothness TES approach [27] is investigated. For Foam Board and LowE materials, the recovered signals are shown in Figure 62 based on the TUD predictions shown in Figure 61. Close agreement is observed between all AR results for this data cube, while the TES results contain some biases. These biases are derived from incorrect temperature estimates made during the TES process, but the distinctive signal features are still clearly evident. The results presented thus far are for a single data cube. To further compare performance, two additional data cubes are considered and aggregated target detection results are reported.

For each of the three investigated data cubes, the ACE background covariance matrix, $\mathbf{\Sigma}$, was estimated using the Mahalanobis anomaly detector with a threshold of 90% to classify pixels as background or anomaly. Applying the ACE detector in AR space or emissivity space results in the average ROC curves shown in Figure 63 for Foam Board and LowE materials. The $\mathbf{y}_A$ output has the lowest ROC curve for TES, however, this is expected since estimating the atmospheric state vector from in-scene data using only 92 spectral bands is a challenging problem. Comparable detection performance is observed for all methods using improved AR, because this method is less dependent on pixel temperature estimation. Mean and standard deviation results of the SCR metric described in

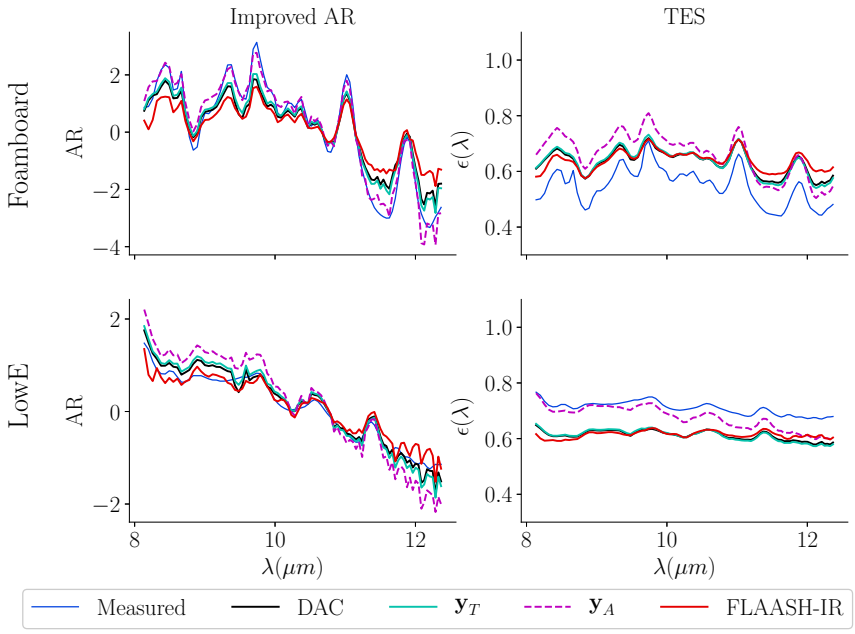


Figure 62. Predicted alpha residual curves and emissivity spectra are shown for FLAASH-IR, DAC and the two MDAC outputs. Alpha residual estimates were made using the improved alpha residual method discussed in [29] and the emissivity estimates were made using the maximum smoothness TES procedure from [27].

Equation 6.29 are shown for each material across all three cubes in Figure 64. Using TES or improved AR results in consistent SCR performance for all atmospheric compensation approaches, however, improved AR is faster to compute because temperature estimation can be performed on a coarser grid.

Many target detection scenarios are time-sensitive, requiring an efficient data pipeline to convert measured at-sensor radiance to a detection statistic. Atmospheric compensation with MDAC takes on average 0.3 s including pixel selection. Combining MDAC with the improved AR approach and the Mahalanobis anomaly detector for background statistic estimation allows for target detection in 8.5 s using the data cubes reported in this study. Replacing MDAC with FLAASH-IR in this processing chain results in 75 s target detection, which may be significant for some detection applications.

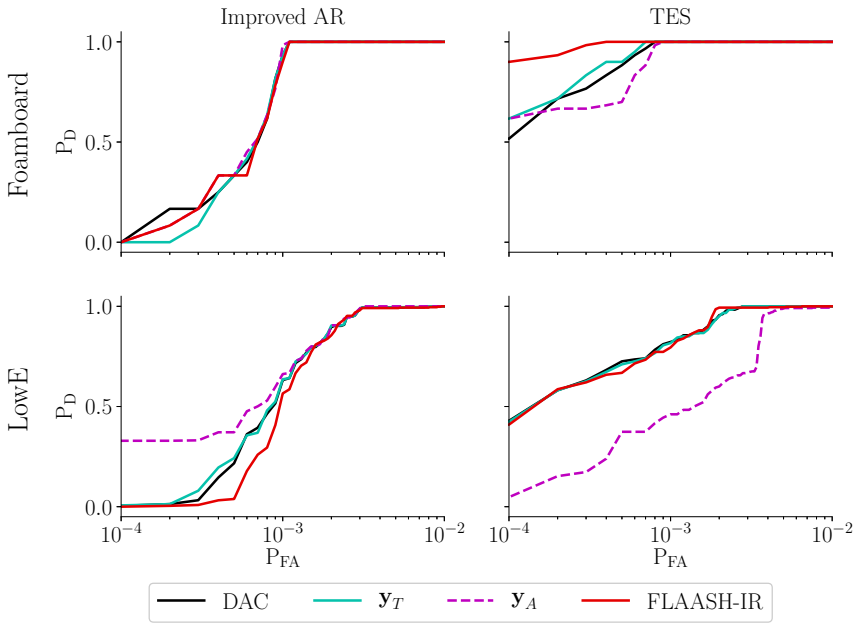


Figure 63. Mean Receiver Operating Characteristic (ROC) curves are shown for DAC, each MDAC output and FLAASH-IR for two materials across three different collected cubes. The probably of false alarm axis utilizes a logarithm scale because of the low false alarm rates for all methods and materials.

6.7 Conclusion

This study has presented a new LWIR in-scene atmospheric compensation approach, producing both an atmospheric state vector and TUD vector from in-scene data only. The compensation approach takes advantage of a pretrained generative model that jointly maps atmospheric state vectors and TUD vectors to a low-dimensional space using LWIR radiative transfer loss, variational loss and a weighted atmospheric state loss. Sampling the generative model yields physically plausible outputs with correct dependencies between atmospheric constituents, transmission and radiance. Given a set of in-scene data, the permutation-invariant MDAC method produces low-dimensional components which map through the generative model to compensate the data cube.

Both of the MDAC predictions were compared against FLAASH-IR and DAC on collected data cubes, demonstrating commensurate detection performance, with a significant reduction in processing time. The use of attention set pooling in the MDAC network re-

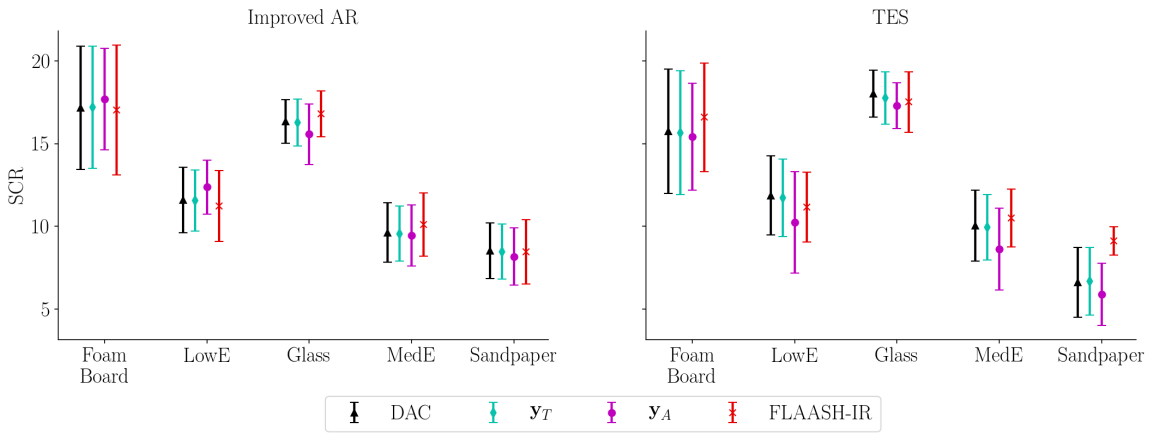


Figure 64. Considering three collected data cubes, the SCR results are shown based on multiple atmospheric compensation approaches. Similar performance is observed for all compensation methods, however, DAC and the MDAC outputs $\mathbf{y}_A$ and $\mathbf{y}_T$ reduce the compensation time allowing for faster target detection.

vealed the model's use of reflective pixels, agreeing with the LWIR radiative transfer equation. This is an important model property, as fully understanding the mechanisms governing network prediction is necessary for dealing with diverse data. While not a primary goal of this study, the atmospheric state predictions of the MDAC network demonstrated that limited atmospheric sounding can be performed. The comparable detection results using the atmospheric state vector prediction suggest the model prediction was a reasonable estimate of the actual atmospheric state.

Applying this approach to higher resolution sensors is an area of future work that will identify how increased sensor resolution impacts target detection performance. Increasing sensor resolution is expected to improve the atmospheric state estimate, supporting the in-scene atmospheric sounding results presented in this study. Also, applying this atmospheric compensation method to additional data cubes is necessary to better understand how emissivity and temperature diversity affects target detection results.

6.8 Appendix

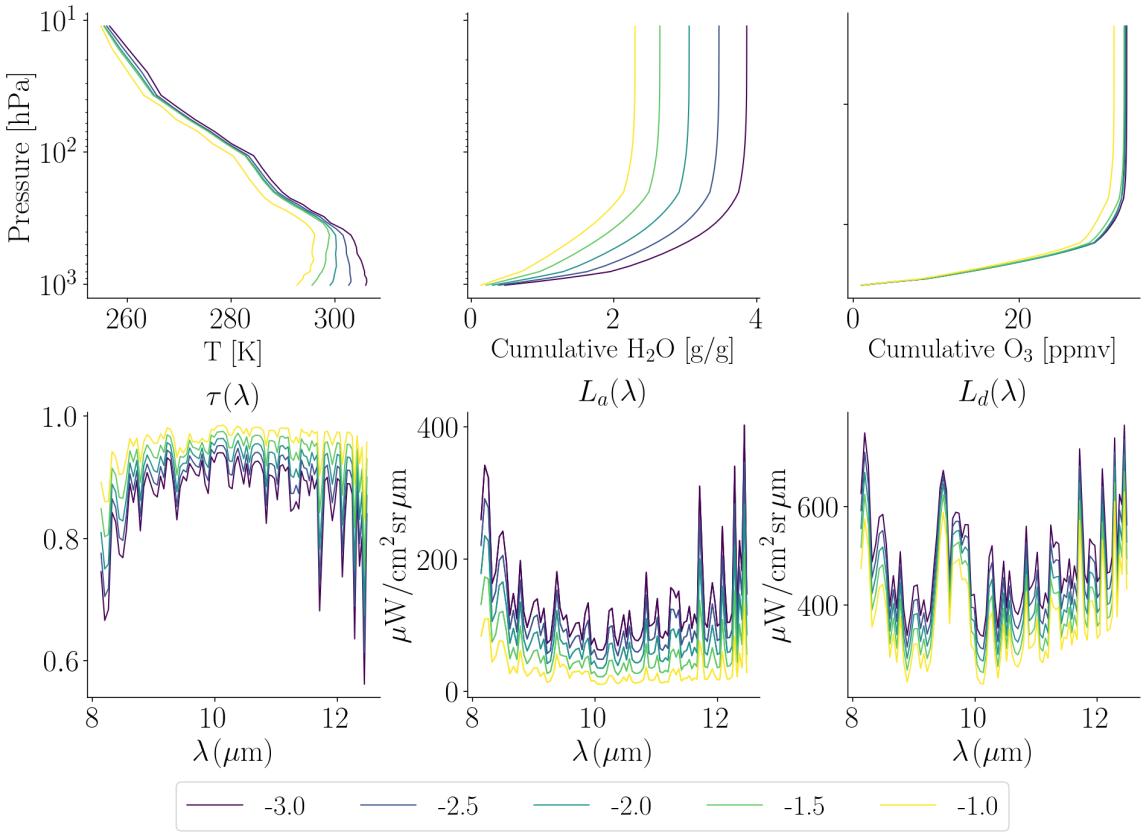


Figure 65. Modifying a single latent component from -3.0 to -1.0 results in the generated atmospheric state vectors and TUD vectors. Warping the latent space in this range allows samples to be created varying from cold, dry atmospheric conditions (-1.0) to warmer, humid conditions (-3.0). By increasing the total water vapor content, more radiation can be absorbed (lower transmittance) and more radiation can be emitted (higher path and downwelling radiance).

VII. Conclusions and Future Work

This dissertation has presented new methods for in-scene atmospheric compensation using novel deep learning approaches constrained by the physical processes governing radiative transfer. Generative modeling approaches were applied to atmospheric databases, creating low-dimensional atmospheric data manifolds, useful for supporting the overall atmospheric compensation algorithm.

Chapters I and II reviewed the challenges and opportunities of Long-Wave Infrared (LWIR) hyperspectral target detection. This included an emphasis on atmospheric compensation within the image processing chain and the need for faster, automated methods. Many methods require prior knowledge of materials within the scene to solve the atmospheric compensation problem. The methods presented here avoid this assumption by employing a neural network approach over a set of collected pixels. Additionally, applying domain-specific knowledge to network construction and training proved beneficial for minimizing model errors. Next, these contributions are discussed in greater detail by chapter in the following section.

7.1 Contributions and Findings

Chapter III compared classification performance using Support Vector Machine (SVM), Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) models on time-varying LWIR hyperspectral data. This research evaluated classifier ability to adapt to changing scene surface temperatures. Validation set partitioning was modified to evaluate classifier performance on surface temperatures exceeding those observed in training. This research motivated further investigations in atmospheric compensation based on the following findings:

A1: Classification algorithms such as SVM, CNN and ANN are unable to generalize when the validation data contains material temperatures outside the training data surface temperature distribution.

A2: The CNN classifier demonstrated a 7% higher classification accuracy than SVM and ANN when evaluated on pixels with temperatures outside the training data range. The large convolutional filters extracted features across multiple bands to identify salient characteristics that were invariant to the pixel temperature biases.

A3: Performing any type of atmospheric compensation significantly improved all classifier results. This included scenarios where the classifier was trained on pixel temperatures not encountered in the validation set.

Performing atmospheric compensation is necessary to minimize false-alarms during target detection, but model-based methods often rely on radiative transfer codes to generate the relevant transmittance and radiance terms. Chapter IV explored dimension reduction techniques to accelerate radiative transfer modeling. Dimension reduction using Principal Component Analysis (PCA) was compared against a uniquely constrained Autoencoder (AE) approach to create a low-dimensional Transmittance, Upwelling, and Downwelling (TUD) vector representation. Visualizing the low-dimensional representations showed clustering based on surface temperature and total column water vapor content. This research demonstrated using AE models to support radiative transfer modeling with the following findings:

B1: A novel loss function was created that relied on the LWIR radiative transfer equation to minimize AE reconstruction error. Compared to Mean-Square Error (MSE), this physics-based loss function proved more favorable for minimizing at-sensor radiance error.

B2: The Stacked Autoencoder (SAE) latent space was sampled with a small neural network, resulting in a 15 times faster radiative transfer model compared to correlated-k techniques.

B3: Utilizing the low-dimensional latent space, atmospheric state vectors could be estimated from a TUD vector. This involved starting at the network output and optimizing backwards to the input.

B4: Data augmentation strategies were investigated to increase the number of TUD vectors available for training. The augmentation strategy led to lower reconstruction error and was used throughout the dissertation research to increase the number of TUD vector training samples.

Chapter V utilized the metrics, loss functions and network architectures from Chapter IV to build a complete in-scene atmospheric compensation method. The DeepSet Atmospheric Compensation (DAC) model was compared against Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR) showing comparable or better performance on collected hyperspectral data. Additionally, DAC errors increased when reflective materials were absent, supported by the LWIR radiative transfer equation. The contributions and findings of this research were:

C1: Permutation-invariant neural networks are useful for estimating the underlying TUD vector from a set of pixels. The set pooling operation must be carefully chosen such that predictions are stable as the number of pixels varies.

C2: At-sensor radiance data can be generated from a TUD library, emissivity library and careful sampling of pixel temperature and scene emissivity. The set generation algorithm in [20] can be used for any LWIR experiment requiring many representations of at-sensor radiance.

C3: In-scene atmospheric compensation using permutation-invariant networks and a generative SAE reduces atmospheric compensation time from 67 s to 0.3 s. This reduced inference time supports accelerated LWIR target detection.

After establishing the benefits of an in-scene atmospheric compensation method derived from a generative model and permutation-invariant networks, the previous results were further extended in Chapter VI to include the atmospheric state vector. The atmospheric state, defined as temperature, column water vapor content and column ozone content, is typically used with a radiative transfer model to generate a TUD vector. The Multimodal DeepSet Atmospheric Compensation (MDAC) model derived in Chapter VI estimates an atmospheric state vector and TUD vector using only in-scene data resulting in the following findings:

D1: A combination of weighted atmospheric state loss, at-sensor radiance loss and Kullback-Leibler (KL) divergence were used to create a joint atmospheric state and TUD vector representation. The combination of these loss functions results in lower model error, improving in-scene atmospheric compensation performance.

D2: Attention mechanisms in the set pooling operation are influenced by reflective materials in the scene. This functionality agrees with the LWIR radiative transfer equation as reflective materials are necessary for downwelling radiance prediction.

D3: A multimodal generative model is capable of producing physically-plausible atmospheric state vectors and their corresponding TUD vectors. Sampling the joint low-dimensional space identified latent components encoding the physical parameters such as total column water vapor content, resulting in an explainable latent code useful for deterministic generative modeling.

D4: Faster target detection is possible when using the MDAC method compared to FLAASH-IR without degrading detection performance on collected data.

Together these contributions and findings demonstrate the utility of combining generative modeling techniques with permutation-invariant networks and domain-specific knowledge to create enhanced atmospheric compensation methods. Next, areas of future work are discussed to further extend this research.

7.2 Future Work

Both the DAC and MDAC algorithms were created specifically for the Spatially Enhanced Broadband Array Spectrograph System (SEBASS) LWIR hyperspectral sensor, but can quickly be retooled for any sensor with a known instrument line shape (ILS). Future hyperspectral and ultraspectral sensors with increasing spectral resolution should be investigated to validate this approach. Specifically, the MDAC algorithm showed promising results as both an atmospheric compensation method and atmospheric sounding algorithm. Sensors with increased spectral resolution should reduce the atmospheric sounding error within MDAC, but this must be verified with additional experimentation. This area of future work may also provide information on sensor spectral resolutions necessary for specific remote sensing applications.

A nadir viewing geometry was assumed in this research, however, there are real-world scenarios where off-nadir atmospheric compensation is needed. Both the DAC and MDAC methods must be investigated for off-nadir geometries to support a wider range of applications. Sensor zenith angle should be included as a model input, just as sensor altitude is currently leveraged. When considering off-nadir geometries, path length varies throughout the scene. This violates the assumption that a homogeneous atmosphere can be assumed for the entire scene as transmittance and path radiance will vary across pixels. Scene segmentation approaches can be applied to make atmospheric estimates over small homogeneous regions. Segmentation is challenging because a trade off is made between creating seg-

mented images with enough data for accurate atmospheric compensation while minimizing atmospheric variation within the image [18, 155].

Both the DAC and MDAC algorithms should be further extended for the visible and near-infrared (VNIR)/shortwave infrared (SWIR) domain. VNIR/SWIR sensors are more common with smaller form factors, allowing wider spread use for civilian and military applications. In the current implementation, the MDAC atmospheric state vector estimate can be forward modeled with MODerate resolution atmospheric TRANsmission (MODTRAN) to estimate VNIR/SWIR atmospheric terms, however, zenith angle must also be known. Additionally, some atmospheric constituents will have a larger effect in the VNIR/SWIR domain, requiring careful standard model selection. A VNIR/SWIR at-sensor radiance training data set can be created for training the MDAC method by sampling over zenith angle, sensor altitude and a range of TUD vectors and emissivity spectra. It is unclear what modifications will be needed to the MDAC network to adapt to these additional variations or how much additional data will be needed.

Further analysis is still needed on a wider range of collected LWIR hyperspectral data. The collected data used in this dissertation was from the same region and was collected over a period of days. A globally diverse set of images is needed to validate both DAC and MDAC performance. The training and validation data was derived from the Thermodynamic Initial Guess Retrieval (TIGR) data, however, conditions may exist where these algorithms perform poorly. Identifying reasons why atmospheric compensation performance degrades will be an important area of research guiding future modifications to the network.

This research relied on the TIGR atmospheric measurement database because all measurements were on a constant pressure grid and the measurements encompassed a diversity of weather conditions. Other atmospheric measurement databases are available and should be explored in future work. Changes to surface altitude, pressure grid and noisy mea-

surements must be considered when forming a larger atmospheric database. This research utilized a data augmentation technique on the TIGR data to avoid pressure axis alignment and data parsing challenges. Additional collected measurements may reveal limitations of the model for specific atmospheric conditions and help inform training data construction.

Appendix A. Estimating Model Uncertainty

Quantifying uncertainty in neural network predictions is necessary for embedding the methods presented in this research with operational systems. The models trained in this research were optimized to produce point estimates, but no information on model uncertainty was included. This section discusses a bootstrap method to estimate model uncertainty and then applies this approach to the Multimodal DeepSet Atmospheric Compensation (MDAC) results presented in Chapter VI for collected and synthetic data.

In classification problems, the neural network model typically utilizes a softmax output activation function to predict the probability of each possible class. These probabilities provide some insight into the model's confidence and can be used as one type of uncertainty measure. Regression problems rely on linear outputs to provide a point estimate, $\hat{y}$ of the target value y. Unfortunately, this point estimate does not contain uncertainty information but modifications to the training and evaluation process can ameliorate this problem.

One of the most common approaches for measuring neural network uncertainty is the bootstrap method [156–158]. This method relies on an ensemble of trained models to produce a mean point estimate, $\hat{y}_i$ for B models such that [156]:

$$\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B} \hat{y}_i^b \tag{1.1}$$

and a variance estimate $\sigma_{\hat{y}_i}^2$ such that:

$$\sigma_{\hat{y}_i}^2 = \frac{1}{B-1}\sum_{b=1}^{B} \left(\hat{y}_i^b - \hat{y}_i\right)^2 \tag{1.2}$$

This approach depends on B well-trained models to estimate $\sigma_{\hat{y}_i}^2$ and $\hat{y}_i$ but poorly trained models can result in large biases in these estimates. Error metrics during training must be analyzed to remove models that did not converge to limit the effect of biased models in ensemble predictions.

Constructing a prediction interval for new samples requires an estimation of the error variance, $\sigma^2_{\hat{\varepsilon}}$, where [156]:

$$\sigma^2_{\hat{\varepsilon}} \simeq \mathbb{E}\left[(y-\hat{y})^2\right] - \sigma^2_{\hat{y}} \tag{1.3}$$

and a set of variance squared residuals computed by:

$$r_i^2 = \max\left(\sigma^2_{\hat{\varepsilon}_i}, 0\right) \tag{1.4}$$

An additional neural network can be trained to predict r_i^2 from input values using the ensemble predictions to create r_i^2. This bootstrap approach is expensive during training since $B+1$ models must be fit, but inference is efficient because neural network forward passes are only required.

This bootstrap approach was applied to the MDAC algorithm letting $B = 10$ to form the network ensemble. Increasing the ensemble size will result in better variance estimates, but $B = 10$ was selected because of hardware limitations. Each MDAC model utilized an independent Multimodal Autoencoder (MMAE) model and both the MMAE and MDAC models were trained until validation loss stabilized. Each model was trained on unique subsets of the training data such that no two models were fit with the same samples. A permutation-invariant network was trained to predict the ensemble squared residuals, utilizing max pooling to compress the set of at-sensor radiance values into a one-dimensional set representation. Figure 66 shows the ensemble mean prediction and one standard deviation, while Figure 67 shows the prediction interval based on Equation 1.4. The Fast Line-of-Sight Atmospheric Analysis of Hypercubes - Infrared (FLAASH-IR) estimate is within the ensemble prediction interval with close agreement to the ensemble mean estimate for all Transmittance, Upwelling, and Downwelling (TUD) components. The permutation-invariant network used to determine the error variance, $\sigma^2_{\hat{\varepsilon}}$, was not comprehensively tuned.

Additional training and modifications to this network may lead to smaller prediction intervals.

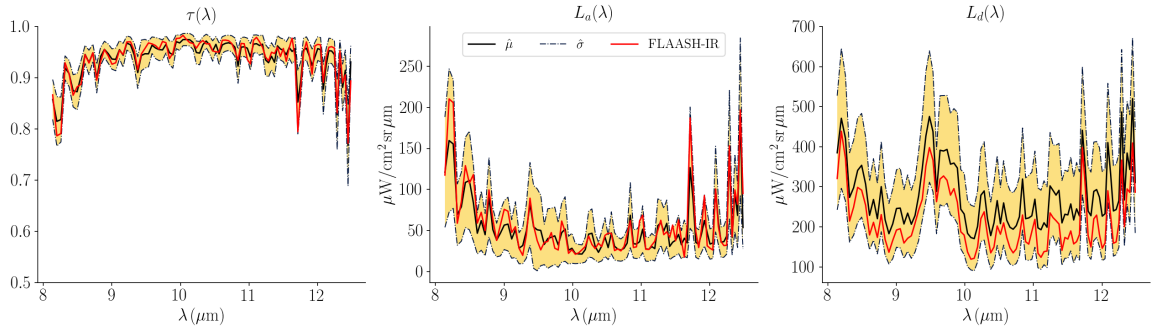


Figure 66. The mean and one standard deviation of ensemble predictions for a collected data cube demonstrate comparable performance for all model random initializations.

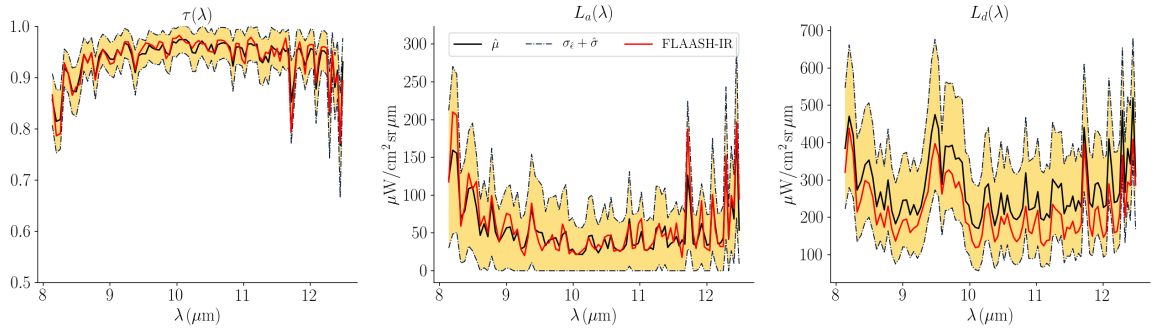


Figure 67. The prediction interval produced using Equation 1.4 and an additional neural network to create the estimate $\sigma^2_{\hat{\varepsilon}}$.

Appendix B. Increased Sensor Resolution

This research investigated the Mako and Spatially Enhanced Broadband Array Spectrograph System (SEBASS) Long-Wave Infrared (LWIR) hyperspectral sensors for radiative transfer modeling and in-scene atmospheric compensation. Specifically, the Multimodal DeepSet Atmospheric Compensation (MDAC) results presented in Chapter VI utilized SEBASS data to recover both an atmospheric sounding and Transmittance, Upwelling, and Downwelling (TUD) vector estimate. Next generation LWIR hyperspectral sensors continue to increase spectral resolution, gathering more information about the atmospheric state and surface elements. This section compares MDAC results using a fictitious higher resolution sensor against the 92 spectral band SEBASS results to demonstrate the research methodology presented in this dissertation can also benefit future hyperspectral sensors.

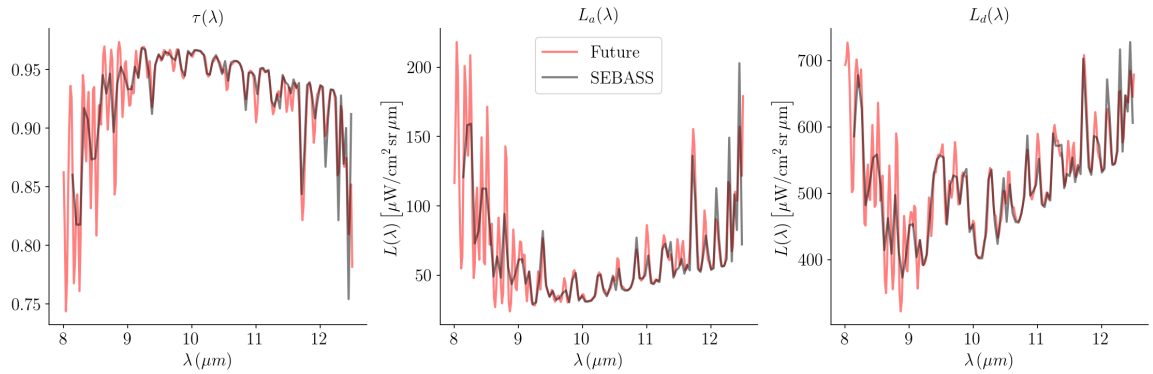


Figure 68. An example TUD vector is shown for the SEBASS sensor with 92 spectral bands and a future sensor using 256 spectral bands.

The higher resolution sensor instrument line shape (ILS) consists of 256 spectral bands sampled from 8 µm to 12.5 µm. A guassian lineshape was used with a constant full-width half max for all bands of 0.017 µm. This assumption is unrealistic as actual instrument lineshapes vary with wavelength, however, this analysis is only interested in how the MDAC algorithm handles additional spectral information. An example TUD vector is shown in Figure 68 for both the SEBASS ILS and this future sensor ILS. The additional spectral bands capture more atmospheric features as shown by the high frequency content between

8 µm to 9 µm. This additional information content is useful for methods such as MDAC since the low-dimensional latent space can be restructured to include this information.

The MDAC latent space was typically varied between 3 and 6 components, resulting in acceptable reconstruction error and atmospheric compensation performance using the SEBASS sensor. Increasing the input dimension may require additional latent components without changing any other network architecture parameters. In this analysis, increasing the spectral dimension from 92 channels to 256 channels required a corresponding increase from 6 latent components to 8 latent components in the Multimodal Autoencoder (MMAE) configuration. The MMAE errors are shown in Figure 69 for both the SEBASS resolution and the notional 256 channel instrument resolution.

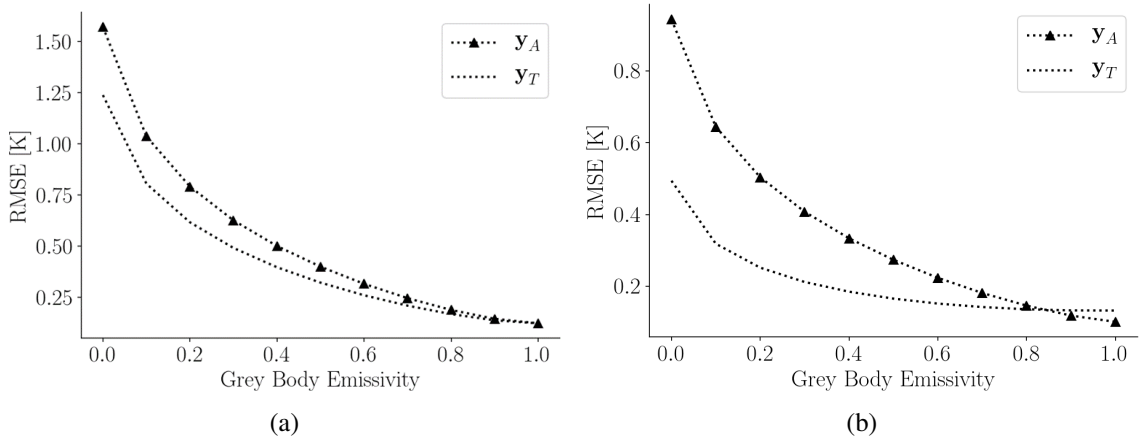


Figure 69. At-sensor brightness temperature error for (a) SEBASS sensor using 92 spectral bands, (b) notional 256 spectral band sensor. The increased information content provided by the notional 256 channel sensor results in lower reconstruction error.

Appendix C. Disentangled Latent Components

The latent space formed by compressing atmospheric state vectors and Transmittance, Upwelling, and Downwelling (TUD) vectors is useful for generative modeling if proper constraints are applied during training. Modified training approaches based on β-Variational Autoencoders (VAEs) were investigated in this dissertation showing improved latent space smoothness when Kullback-Leibler (KL) divergence loss was applied to the latent space. Models leveraging KL divergence loss can produce a disentangled latent representation, revealing the underlying data generating processes. When single latent components are sensitive to known generating processes, while being invariant to others, new samples can be generated with known properties.

The Long-Wave Infrared (LWIR) TUD vectors used in this research were derived from a set of atmospheric measurements, T, H_2O and O_3 as a function of pressure level. Since MODerate resolution atmospheric TRANsmission (MODTRAN) was used to convert these measurements to TUD vectors without modifications to any other MODTRAN parameters, these measurements can be considered the data generating processes governing the TUD vector structure. Latent component sensitivity to these parameters can be identified, resulting in a controllable generative model.

Identifying latent component sensitivity requires modifications to the Multimodal Autoencoder (MMAE) input vectors that are physically plausible. First, temperature profiles are modified starting with a Thermodynamic Initial Guess Retrieval (TIGR) atmospheric temperature profile. A new surface temperature, T_s' is sampled according to $T_s' = \mathrm{N}(T_s, 7)$, where T_s is the TIGR atmospheric temperature profile surface measurement. To create a new atmospheric temperature profile, this research follows the strategy outlined in [138]:

$$T' = (1+g)(T - T_s') + T_s' + \partial T, \tag{3.1}$$

where g is a slope parameter sampled according to $g = \mathrm{U}(-0.5, 0.5)$ and ∂T is an offset sampled according to $\partial T = \mathrm{U}(-7, 7)$. This process creates a new temperature profile, however, only the first 6 measurements (starting at the surface) are used to modify the existing measurement, T. After updating the temperature measurement, relative humidity is measured at each pressure level to verify a 96% threshold is not violated.

Next, the water vapor density profile is modified based on the research presented in [140]. Specifically, it was observed in [140] water vapor density as a function of altitude, z, could be fit with the following form:

$$\mathrm{H_2O}(z) = ae^{-bz}, \tag{3.2}$$

creating a new water vapor density profile requires modifying the fit parameters a and b. These values are sampled according to $a' = \mathrm{N}(a, a/2)$ and $b' = \mathrm{N}(b, b/2)$. These sampled fit parameters create a new water vapor density profile, where only the first three pressure levels are used to modify the existing profile, $\mathrm{H_2O}(z)$. Again, relative humidity calculations are performed to verify the 96% threshold is not violated. After sampling either a new temperature or water vapor density profile, the new atmospheric state vector is used with MODTRAN to create a corresponding TUD vector. The result of this process is shown in Figures 70 and 71. These two vectors form the input to the MMAE for identifying latent component dependencies on temperature or water vapor content.

Latent space dependencies are identified by measuring the absolute change in latent component values when the MMAE input is modified with either temperature or water vapor profile deviations. Two MMAE models are considered here: a model utilizing KL divergence loss with $\beta = 0.2$ and a second model that does not use KL divergence loss, $\beta = 0.0$. Modifying temperature and water vapor density profiles results in the latent component deviations, Δz_i, shown in Figure 72 where $\beta = 0.2$ creates a more disentangled representation. Without using KL divergence, multiple components must be modified at

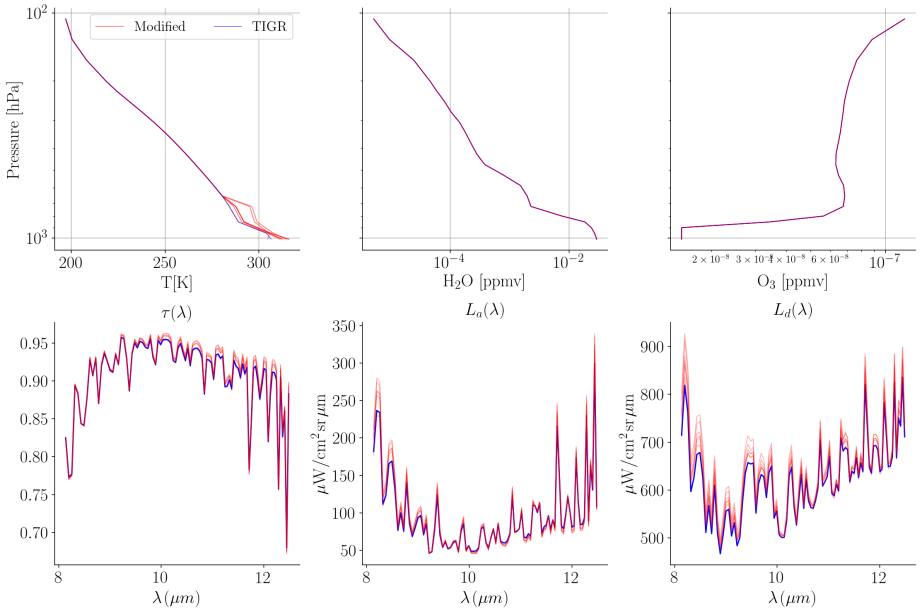


Figure 70. Generated temperature profiles and the corresponding TUD vectors based on low level temperature modifications.

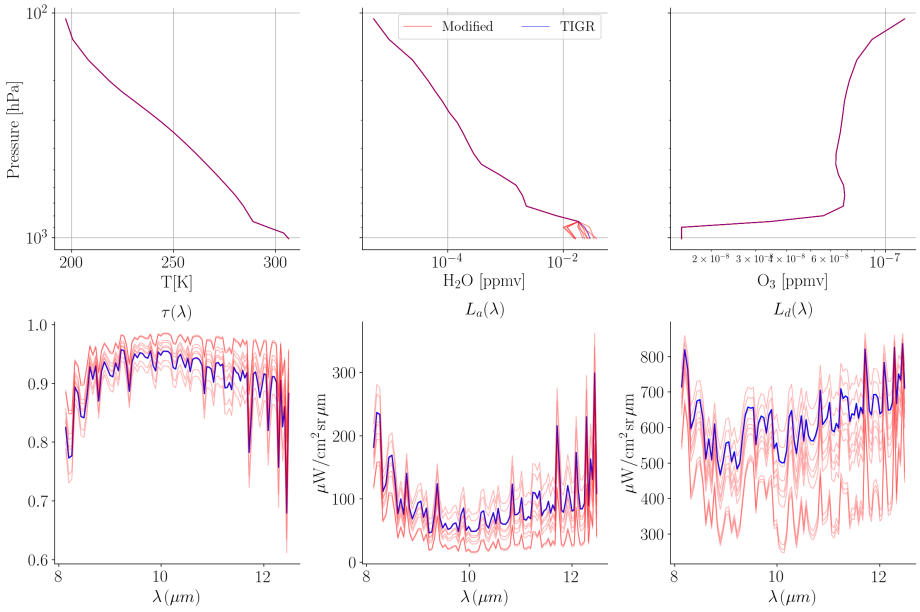


Figure 71. Generated water vapor density profiles and the corresponding TUD vectors showing the significant influence water vapor content has on LWIR spectral features.

once to reflect changes in the underlying generative process. For $\beta = 0.2$, component 2 is predominantly responsible for changes in water vapor content, however, component 6 also plays a role. Neither model has a definitive component for changes in atmospheric temper-

ature, but as noted in Figure 70 the changes applied to atmospheric temperature resulted in small deviations to the TUD vector.

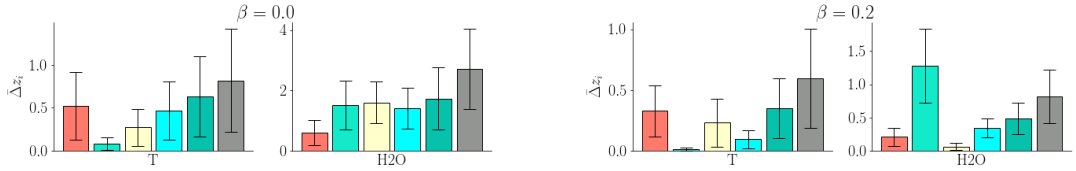


Figure 72. Latent component changes as the underlying generative process is modified. Using KL divergence creates a more disentangled latent representation as highlighted by the larger deviations for a single component, rather than small changes in multiple components at once.

The information contained in Figure 72 is useful for sampling the latent space to create atmospheric state vectors and TUD vectors with specific properties. Modifying component 2 using $\beta = 0.2$ leads to atmospheric state vectors with varying water vapor profiles and corresponding changes to the TUD vectors as shown in Figure 73. This model can now be used to generate more training to support this research, leading to semi-supervised algorithm development. The results shown in Figure 73 are not dependent on MODTRAN and allow for thousands of possible measurements and TUD vectors to be generated in seconds.

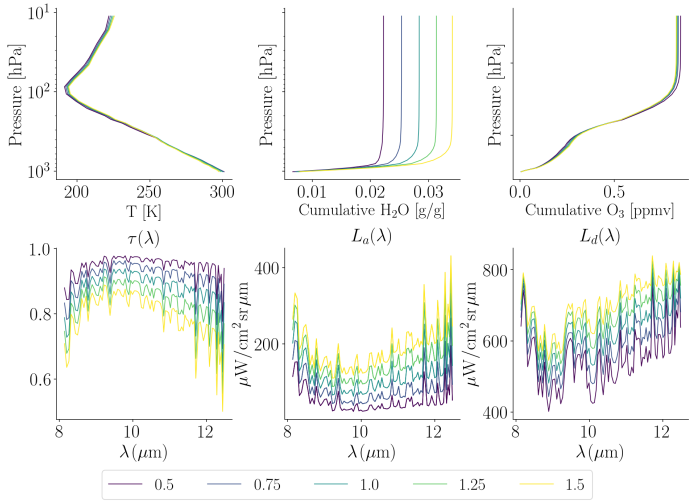


Figure 73. Modifying component 2 in the MMAE model using $\beta = 0.2$ results in a generative model with a disentangled component sensitive to changes in water vapor content. The lines plotted represent the different values of component 2 listed in the legend.

Bibliography

[1] J. R. Schott, *Remote Sensing: The Image Chain Approach*, 2nd ed. Oxford University Press, 2007.

[2] M. T. Eismann, *Hyperspectral Remote Sensing*. Bellingham, WA USA: SPIE, 2012.

[3] J. B. Campbell and R. H. Wynne, *Introduction to Remote Sensing*, 4th ed. Guilford Press, 2011.

[4] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote Sensing and Image Interpretation*, 5th ed. John Wiley & Sons, 2003.

[5] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 12–16, 2002.

[6] D. Manolakis, D. Marden, and G. A. Shaw, "Hyperspectral image processing for automatic target detection applications," *Lincoln Laboratory Journal*, vol. 14, no. 1, pp. 79–116, 2003.

[7] F. Vagni, "Survey of hyperspectral and multispectral imaging technologies," NATO Research and Technology Organization, Neuilly-Sur-Seine, France, Tech. Rep., 2007.

[8] "NATO/multinational joint intelligence, surveillance and reconnaissance–a feasibility study," Joint Air Power Competence Center, Tech. Rep., 2015. [Online]. Available: https://www.japcc.org/wp-content/uploads/JAPCC_MJISRU_web.pdf

[9] M. Shimoni, R. Haelterman, and C. Perneel, "Hypersectral imaging for military and security applications: Combining myriad processing and sensing techniques," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 101–117, 2019.

[10] "Space science and technology strategy - report," U.S. Department of Defense, Washington D.C., Tech. Rep. DF-2015-00066, 2015.

[11] Y. Zhong, X. Wang, Y. Xu, S. Wang, T. Jia, X. Hu, J. Zhao, L. Wei, and L. Zhang, "Mini UAV-borne hyperspectral remote sensing," *IEEE Geoscience and Remote Sensing Magazine*, vol. 6, no. 4, pp. 46–62, 2018.

[12] A. Filis, Z. B. Haim, N. Pundak, and R. Broyde, "Microminiature rotary stirling cryocooler for compact, lightweight, and low-power thermal imaging systems," in *Infrared Technology and Applications XXXV*, vol. 7298. International Society for Optics and Photonics, 2009, p. 729818.

[13] B.-C. Gao, M. J. Montes, C. O. Davis, and A. F. Goetz, "Atmospheric correction algorithms for hyperspectral remote sensing data of land and ocean," *Remote Sensing of Environment*, vol. 113, pp. S17–S24, 2009.

[14] D. Manolakis, M. Pieper, E. Truslow, R. Lockwood, A. Weisner, J. Jacobson, and T. Cooley, "Longwave infrared hyperspectral imaging: Principles, progress, and challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 72–100, 2019.

[15] S. J. Young, B. R. Johnson, and J. A. Hackwell, "An in-scene method for atmospheric compensation of thermal hyperspectral data," *Journal of Geophysical Research: Atmospheres*, vol. 107, no. D24, 2002.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[18] D. S. O'Keefe, "Oblique longwave infrared atmospheric compensation," Air Force Institute of Technology Wright-Patterson AFB United States, Tech. Rep., 2017.

[19] N. M. Westing, B. J. Borghetti, and K. C. Gross, "Analysis of LWIR hyperspectral classification performance across changing scene illumination," in *Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*. International Society for Optics and Photonics, 2019, p. To appear.

[20] N. Westing, K. C. Gross, B. J. Borghetti, J. Martin, and J. Meola, "Learning set representations for LWIR in-scene atmospheric compensation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1438–1449, 2020.

[21] D. G. Manolakis, R. B. Lockwood, and T. W. Cooley, *Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms*. Cambridge University Press, 2016.

[22] A. F. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing," *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.

[23] G. Camps-Valls, J. Munoz-Mari, L. Gomez-Chova, L. Guanter, and X. Calbet, "Nonlinear statistical retrieval of atmospheric profiles from MetOp-IASI and MTG-IRS infrared sounding data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 5, pp. 1759–1769, 2011.

[24] A. Berk, P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. van den Bosch, "MODTRAN6: a major upgrade of the MODTRAN radiative transfer code," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX*, vol. 9088. International Society for Optics and Photonics, 2014, p. 90880H.

[25] M. W. Matthew, S. M. Adler-Golden, A. Berk, G. Felde, G. P. Anderson, D. Gorodetzky, S. Paswaters, and M. Shippert, "Atmospheric correction of spectral imagery: evaluation of the FLAASH algorithm with AVIRIS data," in *Applied Imagery Pattern Recognition Workshop, 2002. Proceedings. 31st*. IEEE, 2002, pp. 157–163.

[26] S. Adler-Golden, P. Conforti, M. Gagnon, P. Tremblay, and M. Chamberland, "Long-wave infrared surface reflectance spectra retrieved from telops hyper-cam imagery," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX*, vol. 9088. International Society for Optics and Photonics, 2014, p. 90880U.

[27] C. Borel, "Error analysis for a temperature and emissivity retrieval algorithm for hyperspectral imaging data," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII*, vol. 6565. International Society for Optics and Photonics, 2007, p. 65651Q.

[28] P. S. Kealy and S. J. Hook, "Separating temperature and emissivity in thermal infrared multispectral scanner data: Implications for recovering land surface temperatures," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 31, no. 6, pp. 1155–1164, 1993.

[29] M. Diani, M. Moscadelli, and G. Corsini, "Improved alpha residuals for target detection in thermal hyperspectral imaging," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 779–783, 2018.

[30] F. Chevallier, F. Chéruy, N. Scott, and A. Chédin, "A neural network approach for a fast and accurate computation of a longwave radiative budget," *Journal of Applied Meteorology*, vol. 37, no. 11, pp. 1385–1397, 1998.

[31] A. Chedin, N. Scott, C. Wahiche, and P. Moulinier, "The improved initialization inversion method: A high resolution physical method for temperature retrievals from satellites of the tiros-n series," *Journal of Climate and Applied Meteorology*, vol. 24, no. 2, pp. 128–143, 1985.

[32] S. A. Clough, M. J. Iacono, and J.-L. Moncet, "Line-by-line calculations of atmospheric fluxes and cooling rates: Application to water vapor," *Journal of Geophysical Research: Atmospheres*, vol. 97, no. D14, pp. 15 761–15 785, 1992.

[33] S. Clough, M. Shephard, E. Mlawer, J. Delamere, M. Iacono, K. Cady-Pereira, S. Boukabara, and P. Brown, "Atmospheric radiative transfer modeling: a summary of the aer codes," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 91, no. 2, pp. 233–244, 2005.

[34] "U.S. standard atmosphere, 1976," National Aeronautics and Space Administration, Tech. Rep. NASA-TM-X-74335, September 1976.

[35] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[36] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[37] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, "Grandmaster level in starcraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[38] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[39] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[41] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 315–323.

[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[43] F. Chollet, *Deep learning with python*. Manning Publications Co., 2017.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[45] B. Chen, T. Medini, J. Farwell, S. Gobriel, C. Tai, and A. Shrivastava, "Slide: In defense of smart algorithms over hardware acceleration for large-scale deep learning systems," *arXiv preprint arXiv:1903.03129*, 2019.

[46] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, NY, USA, 2001, vol. 1.

[47] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[48] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[49] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Omnipress, 2011, pp. 833–840.

[50] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[51] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on International Conference on Machine Learning*, vol. 32, 2014, pp. 1278—1286.

[52] D. P. Kingma and M. Welling, "Stochastic gradient VB and the variational auto-encoder," in *Second International Conference on Learning Representations, ICLR*, 2014.

[53] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial autoencoders," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1511.05644

[54] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, p. 689–696.

[55] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Journal of Machine Learning Research*, vol. 15, no. 1, p. 2949–2980, 2014.

[56] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 609–617.

[57] L.-P. Morency, R. Mihalcea, and P. Doshi, "Towards multimodal sentiment analysis: Harvesting opinions from the web," in *Proceedings of the 13th international conference on multimodal interfaces*, 2011, pp. 169–176.

[58] S. Poria, E. Cambria, N. Howard, G.-B. Huang, and A. Hussain, "Fusing audio, visual and textual clues for sentiment analysis from multimodal content," *Neurocomputing*, vol. 174, pp. 50 – 59, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231215011297

[59] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8M: A large-scale video classification benchmark," *CoRR*, 2016. [Online]. Available: http://arxiv.org/abs/1609.08675

[60] A. S. M. Alharbi and E. de Doncker, "Twitter sentiment analysis with a deep neural network: An enhanced approach using user behavioral information," *Cognitive Systems Research*, vol. 54, pp. 50–61, 2019.

[61] L. Liu, J. Shen, M. Zhang, Z. Wang, and J. Tang, "Learning the joint representation of heterogeneous temporal events for clinical endpoint prediction," in *AAAI*, 2018.

[62] A. A. Pourzanjani, T. B. Wu, R. M. Jiang, M. J. Cohen, and L. R. Petzold, "Understanding coagulopathy using multi-view data in the presence of sub-cohorts: A hierarchical subspace approach," in *Proceedings of the 2nd Machine Learning for Healthcare Conference*, vol. 68, 18–19 Aug 2017, pp. 338–351.

[63] A. Bagher Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, Jul. 2018, pp. 2236–2246.

[64] M. Hou, J. Tang, J. Zhang, W. Kong, and Q. Zhao, "Deep multimodal multilinear fusion with high-order polynomial pooling," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., 2019, pp. 12 136–12 145.

[65] Y. S. Zhun Liu, "Efficient low-rank multimodal fusion with modality-specific factors," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 2018.

[66] G. Sahu and O. Vechtomova, "Dynamic fusion for multimodal data," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1911.03821

[67] J.-M. Pérez-Rúa, V. Vielzeuf, S. Pateux, M. Baccouche, and F. Jurie, "Mfas: Multimodal fusion architecture search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6966–6975.

[68] M. Suzuki, K. Nakayama, and Y. Matsuo, "Joint multimodal learning with deep generative models," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017. [Online]. Available: https://openreview.net/forum?id=BkL7bONFe

[69] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 3744–3753.

[70] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[71] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 918–927.

[72] S. Ravanbakhsh, J. G. Schneider, and B. Póczos, "Deep learning with sets and point clouds," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=Bkj2v6XYl

[73] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4967–4976.

[74] I. Korshunova, J. Degrave, F. Huszár, Y. Gal, A. Gretton, and J. Dambre, "Bruno: A deep recurrent model for exchangeable data," in *Advances in Neural Information Processing Systems*, 2018, pp. 7190–7198.

[75] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 3391–3401. [Online]. Available: http://papers.nips.cc/paper/6931-deep-sets.pdf

[76] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[77] E. Wagstaff, F. Fuchs, M. Engelcke, I. Posner, and M. A. Osborne, "On the limitations of representing functions on sets," in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, vol. 97, pp. 6487–6494. [Online]. Available: http://proceedings.mlr.press/v97/wagstaff19a.html

[78] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 652–660.

[79] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[80] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "ERNIE 2.0: A continual pre-training framework for language understanding," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 8968–8975. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6428

[81] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 10–15 Jul 2018, pp. 2127–2136.

[82] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition." *Neural Networks*, vol. 1, no. 2, pp. 119–130, 1988.

[83] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[84] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. D. Shet, "Multi-digit number recognition from street view imagery using deep convolutional neural networks," in *2nd International Conference on Learning Representations, ICLR, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. [Online]. Available: http://arxiv.org/abs/1312.6082

[85] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2011, pp. 2809–2813.

[86] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *2013 IEEE international conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8614–8618.

[87] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *Journal of Sensors*, 2015.

[88] J. A. Benediktsson and P. Ghamisi, *Spectral-Spatial Classification of Hyperspectral Remote Sensing Images*. Artech House, 2015.

[89] L. O. Jimenez and D. A. Landgrebe, "Supervised classification in high-dimensional space: geometrical, statistical, and asymptotical properties of multivariate data," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 1, pp. 39–54, 1998.

[90] C. Rodarmel and J. Shan, "Principal component analysis for hyperspectral image classification," *Surveying and Land Information Science*, vol. 62, no. 2, pp. 115–122, 2002.

[91] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[92] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[93] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[94] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, 2015, pp. 4959–4962.

[95] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep learning classification of land cover and crop types using remote sensing data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[96] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[97] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, no. 99, pp. 1–18, 2018.

[98] L. Windrim, R. Ramakrishnan, A. Melkumyan, and R. J. Murphy, "A physics-based deep learning approach to shadow invariant representations of hyperspectral images," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 665–677, 2018.

[99] N. M. Nasrabadi, "Hyperspectral target detection: An overview of current and future challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 34–44, 2014.

[100] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[101] I. S. Reed and X. Yu, "Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, 1990.

[102] M. T. Eismann, J. Meola, A. D. Stocker, S. G. Beaven, and A. P. Schaum, "Airborne hyperspectral detection of small changes," *Applied Optics*, vol. 47, no. 28, pp. F27–F45, 2008.

[103] A. P. Schaum and A. Stocker, "Hyperspectral change detection and supervised matched filtering based on covariance equalization," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery X*, vol. 5425. International Society for Optics and Photonics, 2004, pp. 77–90.

[104] M. Eismann, J. Martin, J. Meola, K. Gross, and N. Westing, "Hyperspectral data exploitation: Progress from statistical methods toward machine learning," in *2019 IEEE Research and Applications of Photonics in Defense Conference (RAPID)*, 2019.

[105] D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 1, pp. 208–216, 1992.

[106] S. Kraut, L. L. Scharf, and L. T. McWhorter, "Adaptive subspace detectors," *IEEE Transactions on Signal Processing*, vol. 49, no. 1, pp. 1–16, 2001.

[107] B. M. Rankin, J. Meola, and M. T. Eismann, "Spectral radiance modeling and bayesian model averaging for longwave infrared hyperspectral imagery and subpixel target identification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 12, pp. 6726–6735, 2017.

[108] N. P. Wurst, S. H. An, and J. Meola, "Comparison of longwave infrared hyperspectral target detection methods," in *Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*, vol. 10986. International Society for Optics and Photonics, 2019, p. 1098617.

[109] B. Schölkopf, A. J. Smola, F. Bach *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[110] H. Kwon and N. M. Nasrabadi, "A comparative analysis of kernel subspace target detectors for hyperspectral imagery," *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 193–193, 2007.

[111] G. Healey and D. Slater, "Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 6, pp. 2706–2717, 1999.

[112] J. A. Martin, "Target detection using artificial neural networks on LWIR hyperspectral imagery," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIV*, vol. 10644. International Society for Optics and Photonics, 2018, p. 1064402.

[113] Y. Cai, K. Guan, J. Peng, S. Wang, C. Seifert, B. Wardlow, and Z. Li, "A high-performance and in-season classification system of field-level crop types using time-series landsat data and a machine learning approach," *Remote Sensing of Environment*, vol. 210, pp. 35–47, 2018.

[114] J. S. Pearlman, P. S. Barry, C. C. Segal, J. Shepanski, D. Beiso, and S. L. Carman, "Hyperion, a space-based imaging spectrometer," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 6, pp. 1160–1173, 2003.

[115] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep&dense convolutional neural network for hyperspectral image classification," *Remote Sensing*, vol. 10, no. 9, p. 1454, 2018.

[116] D. Chutia, D. Bhattacharyya, K. K. Sarma, R. Kalita, and S. Sudhakar, "Hyperspectral remote sensing classifications: a perspective survey," *Transactions in GIS*, vol. 20, no. 4, pp. 463–490, 2016.

[117] D. G. Manolakis, "Taxonomy of detection algorithms for hyperspectral imaging applications," *Optical Engineering*, vol. 44, no. 6, p. 066403, 2005.

[118] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[119] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Evaluation of kernels for multiclass classification of hyperspectral remote sensing data," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.*, vol. 2. IEEE, 2006, pp. 813–816.

[120] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[121] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller, "Striving for simplicity: The all convolutional net," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6806

[122] J. L. Hall, R. H. Boucher, K. N. Buckland, D. J. Gutierrez, E. R. Keim, D. M. Tratt, and D. W. Warren, "Mako airborne thermal infrared imaging spectrometer: performance update," in *Imaging Spectrometry XXI*, vol. 9976. International Society for Optics and Photonics, 2016, p. 997604.

[123] X. Liu, W. L. Smith, D. K. Zhou, and A. Larar, "Principal component-based radiative transfer model for hyperspectral sensors: theoretical concept," *Applied Optics*, vol. 45, no. 1, pp. 201–209, 2006.

[124] P. Kopparla, V. Natraj, R. Spurr, R.-L. Shia, D. Crisp, and Y. L. Yung, "A fast and accurate pca based radiative transfer model: Extension to the broadband shortwave region," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 173, pp. 65–71, 2016.

[125] R. Goody, R. West, L. Chen, and D. Crisp, "The correlated-k method for radiation calculations in nonhomogeneous atmospheres," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 42, no. 6, pp. 539–550, 1989.

[126] V. Natraj, X. Jiang, R.-l. Shia, X. Huang, J. S. Margolis, and Y. L. Yung, "Application of principal component analysis to high spectral resolution radiative transfer: A case study of the o2 a band," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 95, no. 4, pp. 539–556, 2005.

[127] M. Matricardi, "A principal component based version of the RTTOV fast radiative transfer model," *Quarterly Journal of the Royal Meteorological Society*, vol. 136, no. 652, pp. 1823–1835, 2010.

[128] A. Aguila, D. S. Efremenko, V. Molina Garcia, and J. Xu, "Analysis of two dimensionality reduction techniques for fast simulation of the spectral radiances in the hartley-huggins band," *Atmosphere*, vol. 10, 2019.

[129] D. Gu, A. R. Gillespie, A. B. Kahle, and F. D. Palluconi, "Autonomous atmospheric compensation (AAC) of high resolution hyperspectral thermal infrared remote-sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 6, pp. 2557–2570, 2000.

[130] C. C. Borel, "ARTEMISS–an algorithm to retrieve temperature and emissivity from hyper-spectral thermal image data," in *28th Annual GOMACTech Conference, Hyperspectral Imaging Session*, vol. 31. Citeseer, 2003, pp. 1–4.

[131] N. Acito, M. Diani, and G. Corsini, "Coupled subspace-based atmospheric compensation of LWIR hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 8, pp. 5224–5238, 2019.

[132] D. Warren, R. Boucher, D. Gutierrez, E. Keim, and M. Sivjee, "Mako: a high-performance, airborne imaging spectrometer for the long-wave infrared," in *Imaging Spectrometry XV*, vol. 7812. International Society for Optics and Photonics, 2010, p. 78120N.

[133] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[134] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[135] M. T. Eismann, A. D. Stocker, and N. M. Nasrabadi, "Automated hyperspectral cueing for civilian search and rescue," *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1031–1055, 2009.

[136] J. A. Sobrino, R. Oltra-Carrió, J. C. Jiménez-Muñoz, Y. Julien, G. Soria, B. Franch, and C. Mattar, "Emissivity mapping over urban areas using a classification-based approach: Application to the dual-use european security ir experiment (desirex)," *International Journal of Applied Earth Observation and Geoinformation*, vol. 18, pp. 141–147, 2012.

[137] D. Zhou, J. Xiao, S. Bonafoni, C. Berger, K. Deilami, Y. Zhou, S. Frolking, R. Yao, Z. Qiao, and J. Sobrino, "Satellite remote sensing of surface urban heat islands: progress, challenges, and perspectives," *Remote Sensing*, vol. 11, no. 1, p. 48, 2019.

[138] N. Acito, M. Diani, and G. Corsini, "Subspace-based temperature and emissivity separation algorithms in LWIR hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1523–1537, 2018.

[139] B. D. Bue, D. R. Thompson, M. L. Eastwood, R. O. Green, B.-C. Gao, D. Keymeulen, C. M. Sarture, A. S. Mazer, and H. H. Luong, "Real-time atmospheric correction of AVIRIS-NG imagery." *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 12, pp. 6419–6428, 2015.

[140] C. T. Lane, "In-scene atmospheric compensation of thermal hyperspectral imaging with applications to simultaneous shortwave data collection," Air Force Institute of Technology, WPAFB OH, Tech. Rep., 2017.

[141] N. Westing, B. Borghetti, and K. C. Gross, "Fast and effective techniques for LWIR radiative transfer modeling: A dimension-reduction approach," *Remote Sensing*, vol. 11, no. 16, p. 1866, Aug 2019. [Online]. Available: http://dx.doi.org/10.3390/rs11161866

[142] J. A. Hackwell, D. W. Warren, R. P. Bongiovi, S. J. Hansel, T. L. Hayhurst, D. J. Mabry, M. G. Sivjee, and J. W. Skinner, "LWIR/MWIR imaging hyperspectral sensor for airborne and ground-based remote sensing," vol. 2819. International Society for Optics and Photonics, 1996, pp. 102–107. [Online]. Available: https://doi.org/10.1117/12.258057

[143] H. Edwards and A. J. Storkey, "Towards a neural statistician," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=HJDBUF5le

[144] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, ser. JMLR Workshop and Conference Proceedings, vol. 37. JMLR.org, 2015, pp. 448–456. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[145] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[146] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A research platform for distributed model selection and training," *arXiv preprint arXiv:1807.05118*, 2018.

[147] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[148] M. L. Pieper, D. Manolakis, E. Truslow, T. W. Cooley, M. Brueggeman, J. Jacobson, and A. Weisner, "Performance limitations of temperature-emissivity separation techniques in long-wave infrared hyperspectral imaging applications," *Optical Engineering*, vol. 56, no. 8, pp. 1–11, 2017.

[149] A. Gillespie, "A new approach for temperature and emissivity separation," *International Journal of Remote Sensing*, vol. 21, no. 10, pp. 2127–2132, 2000.

[150] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework." *ICLR*, vol. 2, no. 5, p. 6, 2017.

[151] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017, p. 933–941.

[152] M. E. Muller, "A note on a method for generating points uniformly on n-dimensional spheres," *Communications of the ACM*, vol. 2, no. 4, p. 19–20, Apr. 1959. [Online]. Available: https://doi.org/10.1145/377939.377946

[153] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[154] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, 20–22 Jun 2016, pp. 1558–1566.

[155] M. R. Smith, A. R. Gillespie, H. Mizzon, L. K. Balick, J. C. Jiménez-Muñoz, and J. A. Sobrino, "In-scene atmospheric correction of hyperspectral thermal infrared images with nadir, horizontal, and oblique view angles," *International Journal of Remote Sensing*, vol. 34, no. 9-10, pp. 3164–3176, 2013.

[156] H. D. Kabir, A. Khosravi, M. A. Hosen, and S. Nahavandi, "Neural network-based uncertainty quantification: A survey of methodologies and applications," *IEEE Access*, vol. 6, pp. 36 218–36 234, 2018.

[157] E. Zio, "A study of the bootstrap method for estimating the accuracy of artificial neural networks in predicting nuclear transient processes," *IEEE Transactions on Nuclear Science*, vol. 53, no. 3, pp. 1460–1478, 2006.

[158] R. Tibshirani, "A comparison of some error estimates for neural network models," *Neural Computation*, vol. 8, no. 1, pp. 152–163, 1996.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
09-07-2020	Doctoral Dissertation	Sept 2017 – Sept 2020

4. TITLE AND SUBTITLE

Physics-Constrained Hyperspectral Data Exploitation Across Diverse Atmospheric Scenarios

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Nicholas M. Westing

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-DS-20-S-021

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Mr. Steven Zech
National Air and Space Intelligence Center
4180 Watson Way
Wright-Patterson AFB, OH 45433
Steven.Zech@us.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

NASIC/GSP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Hyperspectral target detection promises new operational advantages, with increasing instrument spectral resolution and robust material discrimination. Resolving surface materials requires a fast and accurate accounting of atmospheric effects to increase detection accuracy while minimizing false alarms. This dissertation investigates deep learning methods constrained by the processes governing radiative transfer to efficiently perform atmospheric compensation on data collected by long-wave infrared (LWIR) hyperspectral sensors. These compensation methods depend on generative modeling techniques and permutation-invariant neural network architectures to predict LWIR spectral radiometric quantities. The compensation algorithms developed in this work were examined from the perspective of target detection performance using collected data. These deep learning-based compensation algorithms resulted in comparable detection performance to established methods while accelerating the image processing chain by 8X.

15. SUBJECT TERMS

Deep Learning; Hyperspectral Imaging; Atmospheric Compensation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Brett Borghetti, AFIT/ENG
U	U	U	UU	225	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636 x4612; brett.borghetti@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18