This is a preprint of the paper published in IEEE Transactions on Geoscience and Remote Sensing (Volume: 63)
https://doi.org/10.1109/TGRS.2025.3640712

Physics-Constrained Bayesian Neural Networks for Aerosol Retrieval from Hyperspectral Satellite Measurements with Integrated Uncertainty Quantification

Lanlan Rao, Dmitry Efremenko, Adrian Doicu, Chong Shi, Shuai Yin, Husi Letu, *Senior Member, IEEE*, Jian Xu, *Senior Member, IEEE*

***Abstract*—This study introduces an innovative operational Bayesian neural network framework for high-precision joint retrieval of aerosol optical depth (AOD) and layer height (ALH) with physically-consistent uncertainty decomposition from TROPOMI hyperspectral measurements. Unlike conventional approaches, three different full-physics Bayesian neural network architectures (implemented via Bayes-by-Backprop, Dropout, and Batch Norm techniques) are developed to simultaneously estimate target parameters and their heteroscedastic aleatoric uncertainties while preserving radiative transfer constraints. Epistemic uncertainties are quantified via Monte Carlo sampling of stochastic forward propagation, enabling systematic separation of data-driven vs. model-driven uncertainties. A comprehensive validation demonstrates: (1) Synthetic experiments show epistemic uncertainties strongly correlate with retrieval errors, particularly for observing geometries outside the training data distribution, outperforming aleatoric estimates; (2) Analyses using TROPOMI measurements demonstrate that the framework delivers comparable accuracy to operational products while providing unique uncertainty diagnostics. The framework's computational efficiency combined with its probabilistic outputs establishes a new paradigm for characterizing aerosol properties from satellite measurements, particularly valuable for climate and air quality applications.**

***Index Terms*—Aerosol retrieval, Uncertainty quantification, Bayesian Neural Network, TROPOMI**

I. Introduction

Aerosols significantly influence Earth's climate by affecting solar radiation and interacting with clouds [1]–[3]. They also degrade air quality by reducing visibility and contributing to respiratory issues [4], [5]. Accurate quantification of aerosol properties, such as optical depth and vertical distribution, is essential for understanding and modeling their effects on the climate system and air quality.

Satellite remote sensing techniques harness data from sensors on both polar-orbiting and geostationary satellites to provide a comprehensive view of aerosol properties on global and regional scales. These techniques, often leveraging information across various spectral bands such as visible, near-infrared, and shortwave infrared, enable detailed analysis of atmospheric particles. The conventional retrieval algorithm are grounded in sophisticated radiative transfer models and non-linear optimization methods, built upon rigorous physical and mathematical principles [6], [7]. However, the accurate retrieval of aerosol parameters remains a challenging task due to the inherent complexities, such as surface properties, instrument limitations and atmospheric variability. Robust uncertainty quantification methods appears to be essential to assess the reliability and limitations of aerosol retrievals and further to support processes related to application of satellite aerosol products. Moreover, conventional retrieval algorithms require multiple radiative transfer calculations, especially when processing hyperspectral remote sensing data.

Machine learning enables algorithms to learn patterns from data and make prediction/decision without explicit programming. Among its techniques, neural networks stand out for their ability to extract information from large data, making them highly effective for dealing with satellite retrievals. Neural networks have been widely applied in remote sensing for tasks such as classification and object detection, and their use has expanded to atmospheric remote sensing due to their computational strength and advanced data mining in capabilities.

Neural networks in this domain are typically employed through two main approaches: data-driven and physics-based. Data-driven neural networks integrate data from diverse datasets from satellite sensors, ground-based instruments, and climate models to maximize data mining potential [8]–[13]. Physics-based neural networks based on radiative transfer calculations and inverse techniques, focus on deriving specific information from satellite spectra. These neural networks have been trained either to model the forward process [14]–[21] or

Manuscript received xxxx, 2025; revised xxx, 2025. This work was supported by the National Key R & D Program of China (Grant No. 2023YFB3907500), the National Natural Science Foundation of China (Grant No. 42305154), the Open Fund of the State Key Laboratory of Remote Sensing Science (Grant No. OFSLRSS202422), the China Postdoctoral Science Foundation (Grant No. 2024M753243), the Postdoctoral Fellowship Program of CPSF (Grant No. GZC20232694). *(Corresponding author: Jian Xu.)*
Lanlan Rao is with the National Space Science Center, Chinese Academy of Sciences and the State Key Laboratory of Remote Sensing and Digital Earth, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China (e-mail: raolanlan@nssc.ac.cn).
Jian Xu is with the National Space Science Center, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xujian@nssc.ac.cn).
Chong Shi, Shuai Yin, and Husi Letu are with the State Key Laboratory of Remote Sensing and Digital Earth, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100101, China (e-mail: shichong@aircas.ac.cn; yinshuai@aircas.ac.cn; huslt@aircas.ac.cn).
Dmitry Efremenko and Adrian Doicu are with the Remote Sensing Technology Institute, German Aerospace Center (DLR), Oberpfaffenhofen 82234, Germany (e-mail: dmitry.efremenko@dlr.de; adrian.doicu@dlr.de).

to learn the inverse mapping [22]–[36].

The former of these requires the use of supplementary nonlinear optimization techniques, such as Tikhonov regularization or Bayesian inference theory [37], [38], to stabilize the solution-finding process and obtain extra meaningful results. These techniques, while robust, can be computationally intensive and may require considerate tuning to achieve optimal results. On the other hand, the networks which focuses on learning the direct mapping from observations to desired outputs, can be inherently faster. This method delivers point estimates almost instantaneously, often within milliseconds, making it highly suitable for real-time applications. However, the speed comes at the potential cost of accuracy and interpretability.

Neural networks have recently gained widespread application in the operational processing of satellite remote sensing data, demonstrating their ability to handle complex, high-dimensional inputs. Despite their success, conventional neural networks act as "black boxes" and are often deterministic, making it challenging to conduct a theoretical uncertainty analysis. Uncertainty quantification is critical as it not only helps in identifying potential error sources but also enhances confidence in the model's predictions by providing an assessment of their reliability.

There are two main types of uncertainty that should be addressed: aleatoric and epistemic uncertainties. Aleatoric uncertainty refers to the intrinsic variability or noise present in the data, which cannot be reduced, no matter how much additional data is collected. This type of uncertainty arises from factors like measurement errors or natural variability in the system being observed. In contrast, epistemic uncertainty associates with the limitations of the model itself—specifically, the uncertainty stemming from the lack of knowledge or assumptions during the model development. Unlike aleatoric, epistemic uncertainty can potentially be reduced by incorporating additional information or refining the model structure. Disentangling aleatoric and epistemic uncertainty poses a significant challenge, as these two sources of uncertainty often interact and influence each other.

Bayesian theory provides a probabilistic framework for inversion, allowing both the estimation of the parameters of interest and the computation of their posterior probability distributions. The stochastic nature of Bayesian methods introduces variability into the model, which enables the assessment of epistemic uncertainty—uncertainty due to limited knowledge or data. Consequently, the model can distinguish between epistemic uncertainty and aleatoric uncertainty, which arises from inherent randomness in the data. By incorporating Bayesian probability theory into neural networks, it becomes possible to represent and quantify uncertainties in both the model and the data simultaneously. The use of the Dropout or Batch Norm method in neural networks is an approximation of Bayesian NNs (BNNs).

This study proposes a novel hybrid framework that synergizes radiative transfer calculations with Bayesian neural networks to carry out aerosol properties retrieval with inherent uncertainty quantification. To overcome the limitations of conventional deterministic approaches, we have developed three distinct full-physics Bayesian neural network architectures (implemented through Bayesian by Backprop, Dropout, and Batch Norm techniques) to retrieve aerosol optical depth and aerosol layer height from TROPOMI hyperspectral measurements. Section II provides a theoretical description of the employed Bayesian neural networks. Section III describes the procedure of model training, whereas Sections IV and V discuss the retrieval performance based on synthetic and real data. Through comprehensive validation using both synthetic and real satellite observations, our work provides an operational-ready retrieval framework that maintains physical interpretability while delivering probabilistic uncertainty estimates critical for climate modeling and air quality monitoring applications.

II. Bayesian Neural network

A. Inverse-operator neural network

The inverse-operator retrieval algorithm can be expressed by the following equation:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}) + \boldsymbol{\delta}_{\mathbf{y}}, \tag{1}$$

where $\mathbf{y}$ represents the parameters of interest, $\mathbf{f}$ denotes the neural network with weight parameters $\boldsymbol{\omega}$, and $\mathbf{x}$ is the input, including both the forward model parameters and measured radiance. The term $\boldsymbol{\delta}_{\mathbf{y}}$ represents the random error in $\mathbf{y}$, assumed to be Gaussian noise with zero mean and covariance $\mathbf{C}_{\mathbf{y}}^{\delta}$, expressed as $\boldsymbol{\delta}_{\mathbf{y}} \sim \mathcal{N}(0, \mathbf{C}_{\mathbf{y}}^{\delta})$, and therefore we have $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}), \mathbf{C}_{\mathbf{y}}^{\delta})$. Nevertheless, if the actual error in layer height shows noticeable skewness, the Gaussian assumption may introduce bias in the retrieved mean and lead to underestimated uncertainties. Such effects can be diagnosed through residual analysis (e.g., skewness statistics or Q–Q plots), and more flexible error models such as skew-normal distributions may be considered in future work.

Given a dataset $D = \{(\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\}_{n=1}^{N}$, the neural network is trained by finding the parameter set $\hat{\boldsymbol{\omega}}$ that maximizes the posterior probability $p(\boldsymbol{\omega}|D)$. According to the Bayesian theorem, $p(\boldsymbol{\omega}|D)$ is computed as:

$$p(\boldsymbol{\omega}|D) = \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{p(D)} \propto p(D|\boldsymbol{\omega})p(\boldsymbol{\omega}). \tag{2}$$

The corresponding loss function can be expressed as:

$$E(\boldsymbol{\omega}) = E_{\mathrm{D}}(\boldsymbol{\omega}) + E_{\mathrm{R}}(\boldsymbol{\omega}) \propto -\ln p(\boldsymbol{\omega}|D), \tag{3}$$

where E_{D} is the contribution from the likelihood $p(D|\boldsymbol{\omega})$:

$$\begin{aligned} E_{\mathrm{D}}(\boldsymbol{\omega}) = \frac{1}{2}\sum_{n=1}^{N}[\mathbf{y}^{(n)} - \mathbf{f}(\mathbf{x}^{(n)}, \boldsymbol{\omega})]^T[\mathbf{C}_{\mathbf{y}}^{\delta}(\mathbf{x}^{(n)}, \boldsymbol{\omega})]^{-1} \\ [\mathbf{y}^{(n)} - \mathbf{f}(\mathbf{x}^{(n)}, \boldsymbol{\omega})] \\ \propto -\ln p(D|\boldsymbol{\omega}), \end{aligned} \tag{4}$$

and E_{R} represents the contribution of the prior $p(\boldsymbol{\omega})$:

$$E_{\mathrm{R}}(\boldsymbol{\omega}) = \frac{1}{2}\boldsymbol{\omega}^T\mathbf{C}_{\boldsymbol{\omega}}^{-1}\boldsymbol{\omega} \propto \ln p(\boldsymbol{\omega}), \tag{5}$$

where $p(\boldsymbol{\omega})$ is assumed to be normally distributed: $p(\boldsymbol{\omega}) = \mathcal{N}(0, \mathbf{C}_{\boldsymbol{\omega}})$.

The optimization of the model parameters $\boldsymbol{\omega}$ are obtained as:

$$\hat{\boldsymbol{\omega}} = \boldsymbol{\omega}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\omega}} \ln p(\boldsymbol{\omega}|D) = \arg\min_{\boldsymbol{\omega}} E(\boldsymbol{\omega}) \quad (6)$$

The point estimate of output is the modeled value calculated by $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{x}, \hat{\boldsymbol{\omega}})$.

B. Uncertainty

Bayesian Neural Networks (BNNs) aim to represent uncertainty by calculating the posterior probability distribution of the output, $p(\mathbf{y}|\mathbf{x}, D)$, rather than simply providing single-point estimates, where $p(\mathbf{y}|\mathbf{x}, D)$ is the posterior probability distribution of the output $\mathbf{y}$ given the input $\mathbf{x}$ and the dataset D, computed as:

$$p(\mathbf{y}|\mathbf{x}, D) = \int p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|D) d\boldsymbol{\omega}. \quad (7)$$

The variance or standard deviation of the output $\mathbf{y}$ within this distribution is used to quantify the uncertainty in predictions.

The covariance $\mathrm{Cov}(\mathbf{y})$ of the output $\mathbf{y}$, given the dataset D and input $\mathbf{x}$, is computed and then approximated as follows:

$$\begin{aligned}\mathrm{Cov}(\mathbf{y}) &= \int \mathbf{y}\mathbf{y}^T p(\mathbf{y}|\mathbf{x}, D) d\mathbf{y} - \mathbb{E}(\mathbf{y})\mathbb{E}(\mathbf{y})^T \\ &= \int \int \mathbf{y}\mathbf{y}^T p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) d\mathbf{y} p(\boldsymbol{\omega}\ D) d\boldsymbol{\omega} - \mathbb{E}(\mathbf{y})\mathbb{E}(\mathbf{y})^T \\ &= \int (\mathbf{C}_{\mathbf{y}}^{\delta} + \mathbf{f}(\mathbf{x}, \boldsymbol{\omega})\mathbf{f}(\mathbf{x}, \boldsymbol{\omega})^T) p(\boldsymbol{\omega}|D) d\boldsymbol{\omega} - \mathbb{E}(\mathbf{y})\mathbb{E}(\mathbf{y})^T\end{aligned} \quad (8)$$

where $\mathbf{C}_{\mathbf{y}}^{\delta}$ is the covariance of the noise in $\mathbf{y}$, representing aleatoric uncertainty, or the coherent noise in the output and the rest terms represent epistemic uncertainty. However, the calculation of $p(\boldsymbol{\omega}|D)$ is an intractable problem. By using a Monte Carlo sampling method to repeatedly sample $\boldsymbol{\omega}$ from the posterior distribution $p(\boldsymbol{\omega}|D)$, BNNs can approximate this integration and therefore calculate this covariance.

The expected value $\mathbb{E}(\mathbf{y})$ of the output $\mathbf{y}$ given an input $\mathbf{x}$ and Dataset D is computed by integrating over the posterior distribution of the weights:

$$\begin{aligned}\mathbb{E}(\mathbf{y}) &= \int \mathbf{y} p(\mathbf{y}|\mathbf{x}, D) d\mathbf{y} \\ &= \int \int \mathbf{y} p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) d\mathbf{y} p(\boldsymbol{\omega}|D) d\boldsymbol{\omega} \\ &= \int \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}) p(\boldsymbol{\omega}|D) d\boldsymbol{\omega}.\end{aligned} \quad (9)$$

The BNNs generate weights for T times from the posterior distribution $p(\boldsymbol{\omega}|D)$ using the Monte Carlo sampling, this integration can be approximated by averaging the outputs over T samples of the network's weights:

$$\mathbb{E}(\mathbf{y}) \approx \frac{1}{T} \sum_{t=1}^{T} \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t), \quad (10)$$

where $\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)$ is the output of the network for the t_{th} sample of the weights. $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega})$ follows a normal distribution $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\omega}) = \mathcal{N}(\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}), \mathbf{C}_{\mathbf{y}}^{\delta})$. The covariance $\mathrm{Cov}(\mathbf{y})$ of the output $\mathbf{y}$, given the dataset D and input $\mathbf{x}$, is computed and then approximated as follows:

$$\begin{aligned}\mathrm{Cov}(\mathbf{y}) \approx \frac{1}{T} \sum_{t=1}^{T} \left(\left(\mathbf{C}_{\mathbf{y}}^{\delta}\right)_t + \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)^T \right) - \\ \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t) \right) \left(\frac{1}{T} \sum_{t=1}^{T} \mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t) \right)^T\end{aligned} \quad (11)$$

where $\left(\mathbf{C}_{\mathbf{y}}^{\delta}\right)_t$ represents aleatoric uncertainty for the t_{th} sample while $\frac{1}{T}\sum_{t=1}^{T}\left(\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)^T\right) - \left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)\right)\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{f}(\mathbf{x}, \boldsymbol{\omega}_t)\right)^T$ represents the epistemic uncertainty, calculated as the covariance of the outputs over T samples.

1) Aleatoric uncertainty: To estimate aleatoric uncertainty, the covariance matrix of the output $\mathbf{y}$ is assumed to be a diagonal matrix with heteroscedastic noise. The heteroscedastic covariance matrix can be expressed as: $\mathbf{C}_{\mathbf{y}}^{\delta}(\mathbf{x}^{(n)}, \boldsymbol{\omega}) = \mathrm{diag}[\sigma_j^{(n)}]_{j=1}^{N_{\mathbf{y}}}$ where $N_{\mathbf{y}}$ is the dimension of the output. The term $E_D(\boldsymbol{\omega})$ in the loss function is calculated as:

$$E_D(\boldsymbol{\omega}) = \frac{1}{2} \sum_{n=1}^{N} \sum_{j=1}^{N_{\mathbf{y}}} \left(\left(\frac{y_j^{(n)} - \mu_j^{(n)}}{\sigma_j^{(n)}} \right)^2 + \ln \left(\sigma_j^{(n)} \right)^2 \right) \quad (12)$$

where μ_j and $\sigma_j (j = 1, 2, ..., N_y)$ are the mean and variance in the estimates, modeled by the neural network, with the variance representing the heteroscedastic aleatoric uncertainty. This means that $[\mu_1^{(n)}, ..., \mu_{N_{\mathbf{y}}}^{(n)}, \sigma_1^{(n)}, ..., \sigma_{N_{\mathbf{y}}}^{(n)}]$ are $2N_{\mathbf{y}}$ units in the output layer of the neural network.

2) Epistemic uncertainty: Epistemic uncertainty, or model uncertainty, arises from an imperfect model or insufficient data. BNNs allows them to estimate epistemic uncertainty in predictions through stochastic sampling of the weights from the posterior probability $p(\boldsymbol{\omega}|D)$. Therefore determination of $p(\boldsymbol{\omega}|D)$ is the primary problem that needs to be solved.

Since $p(\boldsymbol{\omega}|D)$ is intractable, various approximation methods such as Markov Chain Monte Carlo (MCMC), Laplace approximation, and Variational Inference (VI) can be employed. MCMC generates samples directly from the posterior distribution, providing accurate but computationally intensive estimates. The Laplace method approximates the distribution as a second-order Taylor expansion around $\hat{\boldsymbol{\omega}}$ employing a Hessian matrix. Computing the full Hessian matrix is computationally expensive, especially for deep neural network models.

The Variational Inference method uses the Kullback-Leibler (KL) divergence to find an approximated distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ to approach $p(\boldsymbol{\omega}|D)$, where $\boldsymbol{\theta}$ represents the variational parameters, which are adjusted to minimize the gap between the approximate the target distributions. The KL divergence quantifies the difference between the approximate and the

target probability densities, and is computed as:

$$\begin{aligned}\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega}|D)) &= \int q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln \frac{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})}{p(\boldsymbol{\omega}|D)} d\boldsymbol{\omega}, \\ &= -\int q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} d\boldsymbol{\omega} \\ &\quad + \int q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln p(D) d\boldsymbol{\omega} \\ &= -\int q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} d\boldsymbol{\omega} + \ln p(D) \\ &= -E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} + \ln p(D)\end{aligned} \tag{13}$$

Since KL divergence is always positive, we have $\ln p(D) > E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})}$. The term $E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})}$ is called the Evidence Lower Bound (ELBO), representing the lower bound of the log model evidence $\ln p(D)$.

The best approximation $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ is achieved when the KL divergence is minimized. Since $p(D)$ is independent on $\boldsymbol{\omega}$, minimizing KL divergence is equivalent to minimizing the negative ELBO:

$$\hat{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}} -\mathrm{ELBO} = \arg\min_{\boldsymbol{\omega}} -E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \tag{14}$$

Given $p(\boldsymbol{\omega}|D) \approx q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$, the weights are sampled from $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ and the epistemic uncertainty is estimated by calculating the variance of the modeled output (as shown in Equation (11)).

C. Methods

1) Bayes-by-Backprop: Bayes-by-Backprop is a variational inference technique for training BNNs. The weight parameters are treated as random variables with a posterior probability distribution. The aim is to define a simpler distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ to approximate the posterior distribution $p(\boldsymbol{\omega}|D)$ and use a reparameterization trick to transfer the each weight parameters to deterministic variational parameters and a random variable. The variational distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\omega})$ is typically specified as a normal distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\sigma}^2_{\boldsymbol{\omega}})$. The goal is to learn the variational parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\sigma}_{\boldsymbol{\omega}})$, which define the approximate posterior distribution over the weight parameters. To ensure that $\boldsymbol{\sigma}_{\boldsymbol{\omega}} > 0$ during training, the variational parameters to be learned are actually $\boldsymbol{\theta} = (\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\rho}_{\boldsymbol{\omega}})$ with $\boldsymbol{\sigma}_{\boldsymbol{\omega}} = \exp(\boldsymbol{\rho}_{\boldsymbol{\omega}}/2)$.

To optimize the Evidence Lower Bound (ELBO) efficiently, a reparameterization trick is used to transform the sampling process as:

$$\boldsymbol{\omega} = \boldsymbol{\mu}_{\boldsymbol{\omega}} + \boldsymbol{\sigma}_{\boldsymbol{\omega}} \circ \boldsymbol{\epsilon}_{\boldsymbol{\omega}}, \boldsymbol{\epsilon}_{\boldsymbol{\omega}} \sim \mathcal{N}(0, \mathbf{I}), \tag{15}$$

where $\boldsymbol{\epsilon}$ is sampled from the standard normal distribution and $\circ$ denotes the element-wise product.

During training, we found that if we sample and keep the sampling for each epoch, the loss becomes highly unstable between consecutive epochs. The weight parameters $\boldsymbol{\omega}$ consist of weights and biases. Given an input matrix $\mathbf{X}_l \in \mathbb{R}^{(N \times N^l_{\mathbf{x}})}$ to layer l and weight matrix $\mathbf{W}_l \in \mathbb{R}^{(N^l_{\mathbf{x}} \times N_l)}$, calculated as $[\boldsymbol{\mu}_{\mathbf{w}}]_l + [\boldsymbol{\sigma}_{\mathbf{w}}]_l \circ [\boldsymbol{\epsilon}_{\mathbf{w}}]_l$, and bias $\mathbf{b}_l \in \mathbb{R}^{(1 \times N_l)}$, calculated as $[\boldsymbol{\mu}_{\mathbf{b}}]_l + [\boldsymbol{\sigma}_{\mathbf{b}}]_l \circ [\boldsymbol{\epsilon}_{\mathbf{b}}]_l$ for layer l, where N is the number of the samples, and N_l is the number of units in layer l, the output matrix $\mathbf{Y}_l$ is calculated as:

$$\mathbf{Y}_l = \mathbf{X}_l \mathbf{W}_l + \mathbf{b}_l = \mathbf{X}_l([\boldsymbol{\mu}_{\mathbf{w}}]_l + [\boldsymbol{\sigma}_{\mathbf{w}}]_l \circ [\boldsymbol{\epsilon}_{\mathbf{w}}]_l) + [\boldsymbol{\mu}_{\mathbf{b}}]_l + [\boldsymbol{\sigma}_{\mathbf{b}}]_l \circ [\boldsymbol{\epsilon}_{\mathbf{b}}]_l, \tag{16}$$

where $[\boldsymbol{\epsilon}_{\mathbf{w}}]_l \in \mathbb{R}^{(N^l_{\mathbf{x}} \times N_l)}$ and $[\boldsymbol{\epsilon}_{\mathbf{b}}]_l \in \mathbb{R}^{(1 \times N_l)}$ have the same dimension as the weight and bias respectively.

To address this instability, we impose the sampling and vary $\boldsymbol{\epsilon}$ for each input. The output $\mathbf{y}_l$ is then calculated as:

$$\mathbf{Y}_l = \mathbf{X}_l[\boldsymbol{\mu}_{\mathbf{w}}]_l + [\boldsymbol{\mu}_{\mathbf{b}}]_l + (\mathbf{X}_l \circ [\boldsymbol{\epsilon}_{\mathbf{w}}]_l)(\boldsymbol{\sigma}_{\mathbf{w}})_l + [\boldsymbol{\epsilon}_{\mathbf{b}}]_l[\boldsymbol{\sigma}_{\mathbf{b}}]_l, \tag{17}$$

where $[\boldsymbol{\epsilon}_{\mathbf{w}}]_l \in \mathbb{R}^{(N \times N^l_{\mathbf{x}})}$ and $[\boldsymbol{\epsilon}_{\mathbf{b}}]_l \in \mathbb{R}^{(N \times 1)}$.

The cost function is the negative ELBO:

$$\begin{aligned}-\mathrm{ELBO} &= -E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln \frac{p(D|\boldsymbol{\omega})p(\boldsymbol{\omega})}{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} d\boldsymbol{\omega} \\ &= -E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln p(D|\boldsymbol{\omega}) + \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega})).\end{aligned} \tag{18}$$

The first part $p(D|\boldsymbol{\omega})$ can be approximated by sampling weights $\boldsymbol{\omega}^{(s)}$ for S times from the posterior distribution $q_{\boldsymbol{\theta}}(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\sigma}_{\boldsymbol{\omega}})$ and calculating as:

$$E_{q_{\boldsymbol{\theta}}(\boldsymbol{\omega})} \ln p(D|\boldsymbol{\omega}) \approx \frac{1}{S} \sum_{s=1}^{S} \ln p(D|\boldsymbol{\omega}^{(s)}), \tag{19}$$

where $\ln p(D|\boldsymbol{\omega}^{(s)})$ is the log-likelihood term calculated as $E_D(\boldsymbol{\omega}^{(s)})$ using Equation (12). Assuming that the prior distribution of $\boldsymbol{\omega}$ follow the standard normal distribution $\mathcal{N}(0, \mathbf{I})$, the second part can be calculated analytically to a simpler form:

$$\begin{aligned}\mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega})) &= \mathrm{KL}(\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\omega}}, \boldsymbol{\sigma}_{\boldsymbol{\omega}})||\mathcal{N}(0, \mathbf{I})) \\ &= \frac{1}{2} \sum_{j=1}^{W} \left([\boldsymbol{\mu}_{\boldsymbol{\omega}}]^2_j + [\boldsymbol{\sigma}_{\boldsymbol{\omega}}]^2_j - \ln [\boldsymbol{\sigma}_{\boldsymbol{\omega}}]^2_j \right),\end{aligned} \tag{20}$$

where W is the dimensionality of $\boldsymbol{\omega}$, and the KL term can now be computed directly without requiring sampling.

2) Dropout: Dropout randomly sets weights to zero with probability $1 - p$, while keeping them with probability p. We define the input layer as layer 0 and the output layer as layer L, with the number of hidden layers being $L - 1$. The weights $\boldsymbol{\omega}$ at layer l ($l = 1, 2, ..., L$) are represented as the weight matrix $\mathbf{W}_l = [\mathbf{w}_{k,l}]^{N_l}_{k=1}$ and bias $\mathbf{b}_l$, for a neural network with $L - 1$ hidden layers and N_l units in layer l. The dropout process is expressed as:

$$\mathbf{h}_l = (\mathbf{x}_l \mathbf{W}_l) \circ \mathbf{z}_l + \mathbf{b}_l, \tag{21}$$

where $\mathbf{h}_l$ is the pre-active input and each element of $\mathbf{z}_l$ follows a Bernoulli distribution $\mathrm{Bernoulli}(p_l)$, with $1 - p_l$ representing the zero-out rate in layer l. In this experiment, dropout is applied to the units in layer 2 through $L - 1$ with the same zero-out rate p, such that $p_2 = ... = p_{L-1} = p$, $p_1 = p_L = 1$.

Adding a regularization term, the cost function of Dropout can be expressed as:

$$\begin{aligned}E(\boldsymbol{\omega}) = &-\frac{1}{N} \sum_{i=1}^{N} \ln p(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\omega}^{(i)}) \\ &+ \sum_{l=1}^{L} \sum_{k=1}^{N_l} \lambda_{k,l} ||\mathbf{w}_{k,l}||^2_2 + \sum_{l=1}^{L} \lambda_l ||\mathbf{b}_l||^2_2,\end{aligned} \tag{22}$$

where $\boldsymbol{\omega}^{(i)}$ represents a sampled set of weights obtained by applying a Dropout mask to the weights $\hat{\boldsymbol{\omega}}$ for each sample and $\lambda_{k,l}$ represents the weight decay.

This stochastic mechanism allows dropout to be viewed as an approximate to variational inference in Bayesian Neural Networks (BNNs). By treating the estimated weight parameters as variational parameters and reparameterizing the weights as samples from the product of Bernoulli and normal distributions, the $-$ELBO of Dropout can be expressed in a form similar to Equation (22). Therefore, Dropout could be viewed as an approximation of a Bayesian neural network. The detailed proof can be found in [39].

3) Batch Norm: With input $\mathbf{x}_l$ into layer l, Batch Norm (BN) applies a normalization over the mini-batch to the pre-activation values $\mathbf{h}_l$ before the activation function. The pre-activation input to the layer is defined as $\mathbf{h}_l = \mathbf{x}_l \mathbf{W}_l$, where $\mathbf{h}_l = [h_l^1, h_l^2, ..., h_l^{N_l}]$, and N_l is the total number of the units in layer l.

During training, for a given unit u and a mini-batch dataset $B = \{(\mathbf{x}^{(m)}, \mathbf{y}^{(m)})\}_{m=1}^{M}$, the mini-batch mean μ_{B}^u and variance $(\sigma_{\mathrm{B}}^u)^2$ are calculated as follows:

$$\mu_{\mathrm{B}}^u = \frac{1}{M}\sum_{m=1}^{M} h_{l,m}^u, \ (\sigma_{\mathrm{B}}^u)^2 = \frac{1}{M}\sum_{m=1}^{M}(h_{l,m}^u - \mu_{\mathrm{B}}^u)^2. \quad (23)$$

During training, the moving mean μ_{moving}^u and moving variance $(\sigma_{\mathrm{moving}}^u)^2$are updated for each forward pass:

$$\begin{aligned}\mu_{\mathrm{moving}}^u &= (1-\mathrm{Momentum}) \times \mu_{\mathrm{moving}}^u + \mu_{\mathrm{B}}^u,\\ (\sigma_{\mathrm{moving}}^u)^2 &= (1-\mathrm{Momentum}) \times (\sigma_{\mathrm{moving}}^u)^2 + (\sigma_{\mathrm{B}}^u)^2,\end{aligned} \quad (24)$$

where $\mathrm{Momentum} = 0.1$ is the averaging factor. The moving mean and variance are used for the inference stage to calculate the normalized input, while the statistics μ_{B} and $(\sigma_{\mathrm{B}}^u)^2$ are used to calculate the normalized input $\hat{h}_{l,m}^u$ for each mini-batch example:

$$\hat{h}_{l,m}^u = \frac{h_{l,m}^u - \mu_{\mathrm{B}}^u}{\sqrt{(\sigma_{\mathrm{B}}^u)^2 + \epsilon}}, \quad (25)$$

where ϵ is a small constant added to avoid division by zero.

A scale parameter γ_l^u and a shift parameter β_l^u , which are learned along with the weight matrix $\mathbf{W}_l$ in the neural network, are applied to the normalized input to allow for identity transformations. The output of the BN layer is:

$$y_{l,m}^u = \gamma_l^u \hat{h}_{l,m}^u + \beta_l^u. \quad (26)$$

The optimization of BN for a mini-batch can be expressed as the minimization of the loss function:

$$E(\boldsymbol{\omega}) = -\frac{1}{M}\sum_{m=1}^{M} \ln p(\mathbf{y}^{(m)}|\mathbf{x}^{(m)}, \boldsymbol{\omega}^{(m)}) + \sum_{l=1}^{L}\lambda_l ||\boldsymbol{\theta}_l||_2^2), \quad (27)$$

where $\boldsymbol{\theta}_l$ is the learnable parameters in layer l, including $\mathbf{W}_l$, $\{\gamma_l^u\}_{u=1}^{N_l}$ and $\{\beta_l^u\}_{u=1}^{N_l}$ and $\boldsymbol{\omega}$ is the stochastic parameters comprising $\{\mu_{\mathrm{B}}^{1:N_l}\}_{l=1}^{L}$ and $\{\sigma_{\mathrm{B}}^{1:N_l}\}_{l=1}^{L}$. The second term can be treated as a weight decay regularization. Since the weights for each sample in the mini-batch are different, the weights can be viewed as being sampled for each individual sample.

To illustrate why optimization for BN can be approximated as variational approximation, we first represent the $-$ELBO, which is aimed to be minimized for the mini-batch, as follows:

$$\begin{aligned}-\mathrm{ELBO} = -\frac{N}{M}\sum_{m=1}^{M} \ln p(\mathbf{y}^{(m)}|\mathbf{x}^{(m)}, \boldsymbol{\omega}^{(m)})\\ + \mathrm{KL}(q_{\boldsymbol{\theta}}(\boldsymbol{\omega})||p(\boldsymbol{\omega})).\end{aligned} \quad (28)$$

Assume μ_{B}^u and σ_{B}^u follow a normal distribution centered around the population value $\mu_{\mathrm{true}}^u = \frac{1}{N}\sum_{n=1}^{N} h_{l,n}^u$ and $\sigma_{\mathrm{true}}^u = \frac{1}{N}\sum_{n=1}^{N}(h_{l,n}^u - \mu_{\mathrm{true}}^u)^2$ respectively. With large batch M and dataset size N, and with $\gamma_l^u = 1$ and $\beta_l^u = 1$, the optimization of the derivative of $E(\boldsymbol{\omega})$ and $-ELBO$ with respect to $\boldsymbol{\omega}$ could be equivalent, provided that the prior distribution are set as $p(\mu_{\mathrm{B}}^u) = \mathcal{N}(0, \infty)$ and $p(\sigma_{\mathrm{B}}^u) = \mathcal{N}(0, \frac{1}{2N\lambda_l})$. The detailed prove can be found in [40].

III. Training

The training dataset is generated by sampling the forward model parameters $[\tau, H, \theta_0, \theta, \Delta\varphi, H_{\mathrm{s}}, A_{\mathrm{s}}]$ using a smart technique based on Halton sequences [41], as described in [42]. The Halton sequence is used because it generates low-discrepancy points over the parameter space. Compared with simple random sampling, it reduces clustering and gaps, resulting in faster convergence of network optimization and more representative coverage of the input–output space in the training dataset. The variation intervals for these parameters are provided in Table I. The neural networks are trained with the moderately absorbing aerosol model from the MODIS DT algorithm. The aerosol layer is modeled as homogeneous, with a constant thickness of 0.5 km, distributed evenly between $H - 0.25$ km to $H + 0.25$ km.

TABLE I
INTERVALS OF VARIATION OF THE OPTICAL AND GEOMETRICAL PARAMETERS FOR GENERATING THE DATA SET.

Parameter	Description	Interval of variation
τ	Aerosol Optical Depth	$0.05 - 5$
H	Aerosol Layer Height	$0.1 - 15.75\,\mathrm{km}$
θ_0	Solar Zenith Angle	$0 - 75°$
θ	Viewing Zenith Angle	$0 - 70°$
$\Delta\varphi$	Relative Azimuth Angle	$0 - 180°$
H_{s}	Surface Height	$0 - 2.61\,\mathrm{km}$
A_{s}	Surface Albedo	$0 - 0.4$

The input to the BNNs, denoted as $\mathbf{z}$, consists of the biased radiances on a measurement wavelength grid, along with the forward model parameters. A heteroscedastic aleatoric covariance is incorporated into the output of the BNNs. The maximum posterior estimate of the weights is given by $\hat{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega}} E(\boldsymbol{\omega})$, where the loss function $E(\boldsymbol{\omega})$ calculated using Equations (18), (22) and (27) for Bayes-by-Backprop, Dropout and Batch Norm, respectively. The output vector is defined as $\mathbf{y} = [\tau, H]^T$, where τ is the aerosol optical depth and H is the aerosol layer height. The output layer of neural networks produces estimates of the aerosol optical depth μ_τ, aerosol layer height μ_H, and parameters that representing their associated heteroscedastic aleatoric uncertainties (σ_τ, σ_H).This

is based on the assumptions that $p(\tau|\hat{\boldsymbol{\omega}}, \mathbf{z}) = \mathcal{N}(\mu_\tau, \sigma_\tau^2)$ and $p(H|\hat{\boldsymbol{\omega}}, \mathbf{z}) = \mathcal{N}(\mu_H, \sigma_H^2)$. To ensure that the uncertainty values are positive, the actual output terms representing the uncertainty are denoted as ρ_τ and ρ_H, with the following relations: $\sigma_\tau = \exp(\frac{\rho_\tau}{2}), \sigma_H = \exp(\frac{\rho_H}{2})$. The input and the complete output information are as follows:

$$\begin{aligned} \textbf{Input} = \mathbf{x} = & \begin{bmatrix} [I(\lambda_{\mathrm{m}k}^r) + \delta_{\mathrm{m}k}]_{k=1}^{N_{\mathrm{m}\lambda}} \\ [\theta_0, \theta, \Delta\varphi, H_{\mathrm{s}}, A_{\mathrm{s}}]^T \end{bmatrix} \\ \longmapsto \textbf{Output} = & [\mu_\tau, \mu_H, \rho_\tau, \rho_H]^T. \end{aligned} \tag{29}$$

Here $\delta_{\mathrm{m}k}$ is a Gaussian noise added to the input radiances. $N_{\mathrm{m}\lambda} = 131$ is the total number of the measurement wavelengths in the grid and $r = 1, 2, ..., 448$ denotes the index of the swath row. The input vector has a dimension of $N_{\mathrm{x}} = N_{\mathrm{m}\lambda} + 5$ and the output has a dimension of $2N_{\mathrm{y}} = 2 \times 2 = 4$.

To handle a set of measurement wavelength grids and reduce the dataset size, a jitter approach is employed to randomly select a wavelength grid for forward simulation. Detailed description of the method can be found in [43]. The training dataset consists of $404\,901$ samples, 10% of which are used for validation to optimize the architecture of the neural network. In the validation stage, holdout cross-validation is used alongside a grid search. The grid search explores different combinations of hyperparameters, specifically the number of hidden layers, which is selected from $\{2, 3, 4\}$, and the number of units per layer, which is selected from $\{20, 40, 60, 80, 100, 120, 140, 160, 180, 200\}$. A linear rectification activation function is employed; mini-batch gradient descent with Adaptive Moment Estimation (ADAM) [44] is utilized, and a total of 3,000 epochs is used for training these neural networks.

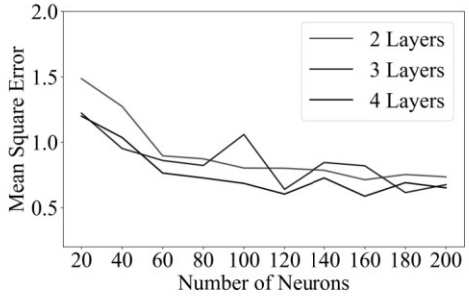
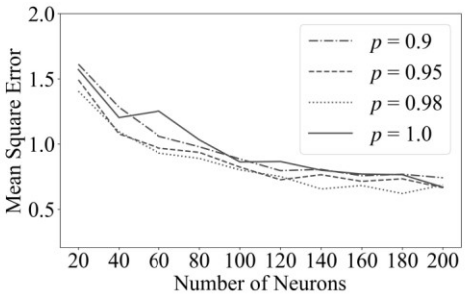
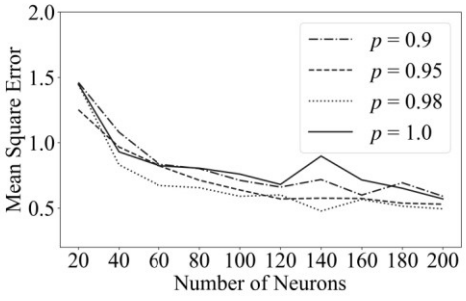


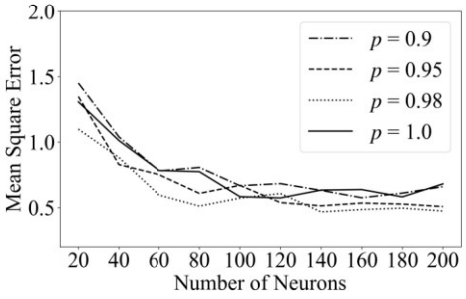
Fig. 1. Mean squared error for the validation dataset across different numbers of neurons and layers using the Bayes-by-Backprop method.



(a) 2 Layers



(b) 3 Layers



(c) 4 Layers

Fig. 2. Mean squared error for the validation dataset across different p, and numbers of neurons and layers using the Dropout method.

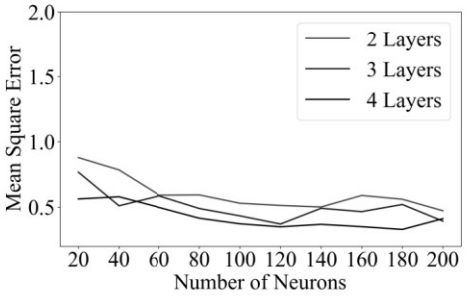


Fig. 3. Mean squared error for the validation dataset different numbers of neurons and layers using the Batch Norm method.

These neural network are evaluated by using the quality assessment parameter, i.e. the mean squared error of the output $\mathbf{y}$. Figure 1 illustrates the mean squared errors obtained using the Bayes-by-Backprop method with these different combinations. A more complex neural network structure, with additional layers and more units per layer, typically exhibits

better performance. The neural network comprising 4 hidden layers and 160 units in each layer produces the lowest mean squared error. The probability $1-p$ of zeroing out weights has a significant effect on model performance. Therefore, p, ranging from $\{0.9, 0.95, 0.98, 1\}$ is also included the grid search process. The neural network with $p = 1$ represents a standard neural network. The evaluation of these neural networks is shown in Fig. 2. Implementing Dropout in the neural network can reduces errors in the retrieval; however, model performance will decrease with a high zeroing-out probability. The highest performance is achieved by the neural network using the Dropout method with $p = 0.98$, 4 hidden layers and 140 units per layer. The mean squared error for neural networks using the Batch Norm method is shown in Fig. 3. A mini-batch size of 1000 is used. With four hidden layers and 180 units per layer, the network produces the most plausible retrievals. The optimal architectures differ slightly among the three methods. A consistent trend is observed: as the number of hidden layers and neurons per layer increases, the loss decreases, suggesting that more complex architectures generally yield better performance. Across all methods, four hidden layers provide the best results. However, when the number of neurons per layer exceeds 140, the improvement in loss becomes marginal. To balance computational efficiency and model generalization, and to ensure a fair comparison among the three methods by isolating the effect of network architecture, we use the entire dataset and adopt the same architecture, consisting of four hidden layers with 140 units per layer, to train all three Bayesian neural networks.

IV. Retrieval Using Synthetic Data

To analyze both epistemic and aleatoric uncertainty, we generate testing datasets and test them with the three BNNs. As described in Section II-B and assuming a diagonal covariance matrix for the output $\mathbf{y}$, the uncertainty are represented by the variance of the output. For each set of input, the calculation is conducted for $T = 100$ times. Aleatoric uncertainties are assumed to be heteroscedastic variance and modeled directly together with retrieved estimates of aerosol optical depth as described in Section III.

Taken τ as example, the total variance of τ is calculated as:

$$\mathrm{Var}(\tau) \approx \frac{1}{T}\sum_{t=1}^{T}(\sigma_\tau^2)_t + \frac{1}{T}\sum_{t=1}^{T}((\mu_\tau)_t)^2 - \left(\frac{1}{T}\sum_{t=1}^{T}(\mu_\tau)_t\right)^2, \tag{30}$$

where $(\mu_\tau)_t$ and σ_τ are the estimate and aleatoric uncertainty of τ for t_{th} sample. The first term is the mean aleatoric uncertainty over T samples. The second term, $\frac{1}{T}\sum_{t=1}^{T}((\mu_\tau)_t)^2 - \left(\frac{1}{T}\sum_{t=1}^{T}(\mu_\tau)_t\right)^2 = \mathrm{Var}(\mu_\tau)$, represents the epistemic uncertainty, calculated as the variance of μ_τ over T samples.

To estimate the uncertainty, the Bayes-By-Backprop method employs sampling weights, while the Dropout method randomly zeroes out weights. In the Batch Norm method, the mini-batch mean and variance will vary as the input to the neural network changes. However, if the input variance significantly exceeds that of the training dataset, the output may become unstable. To mitigate this, we randomly select 1000 samples (the size of the mini-batch used in the training process) from the training dataset to calculate the mean and variance, which are then used for normalizing the inputs of the testing dataset.

Dataset A consists of 10000 radiances simulated by the radiative transfer model. These radiances are computed using the input parameters randomly generated within the variation specified in Table I.

The retrievals of τ and H from dataset A are calculated using the three neural networks. The mean absolute error and the mean uncertainties of the retrievals are listed in Table II.

The Batch Norm method produces the smallest errors and the smallest aleatoric uncertainty for both τ and H, while the Bayes-by-Backprop method produces the smallest epistemic uncertainty and the biggest errors for both τ and H. It is worth noting that the epistemic uncertainty for the three BNNs is of a similar magnitude.

We separate the samples into 100 bins and calculate the mean absolute error, mean aleatoric uncertainty and mean epistemic uncertainty for each bin. The relationship between the input parameters and the errors/uncertainty is plotted in Figs. 4, 5 and 6. As the epistemic uncertainty is much smaller than the aleatoric uncertainty, three times the epistemic uncertainty is shown in these plots. The analysis identifies several consistent patterns: (1) Strong correlations exist between aleatoric uncertainty, epistemic uncertainty, and absolute errors across all parameters; (2) Uncertainties in H decrease with increasing SZA and VZA, while uncertainties in τ increase with increasing SZA and VZA; (3) Both retrieval errors and uncertainties for H and τ are higher under low aerosol loading conditions; (4) Surface albedo positively correlates with retrieval errors and uncertainties for both parameters; (5) Parameter-specific trends emerge, with epistemic uncertainty in H increasing with H and epistemic uncertainty in τ increasing with τ.

The aleatoric uncertainty in τ for the Batch Norm method is smaller than that of the other two neural networks. The epistemic uncertainty for the Batch Norm method depends heavily on the calculation of moving mean and variance. Figure 7 shows the errors and uncertainties when using the Batch Norm method with 100 randomly selected samples from the training dataset to calculate the moving statistics, compared to using 1000 samples. The epistemic uncertainty increases substantially when using 1000 samples compared to 100 samples, while the overall aleatoric uncertainty remains relatively unchanged. For practical applications, it is recommended to use the training mini-batch size of 1000, while ensuring that samples are randomly selected from the training dataset.

The aleatoric and epistemic uncertainties show strong correlation. To study their differences, we simulate Datasets B, C and D using the radiative transfer model, each containing 2000 radiance spectra. These spectra are computed using the same viewing geometry parameters ($\mathrm{SZA} = 0, \mathrm{VZA} = 0, \mathrm{RA} = 0$)) and a surface height of 0 km. Dataset B is generated with $\tau = 1.5, H = 3$ km, and surface albedo randomly selected from $[0, 1]$ to study uncertainties when surface albedo exceeds the range of the training dataset. Datasets C and D are generated with a fixed surface albedo of $S_a = 0.02$. Dataset

TABLE II
MEAN ERROR AND UNCERTAINTIES OF τ AND H.

	Δ_τ	$\sigma_\tau^{\text{aleatoric}}$	$\sigma_\tau^{\text{epistemic}}$	Δ_H	$\sigma_H^{\text{aleatoric}}$	$\sigma_H^{\text{epistemic}}$
Bayes-by-Backprop	0.23	0.34	0.09	0.54	1.20	0.24
Dropout	0.17	0.28	0.10	0.42	0.91	0.28
Batch Norm	0.12	0.17	0.10	0.38	0.75	0.27

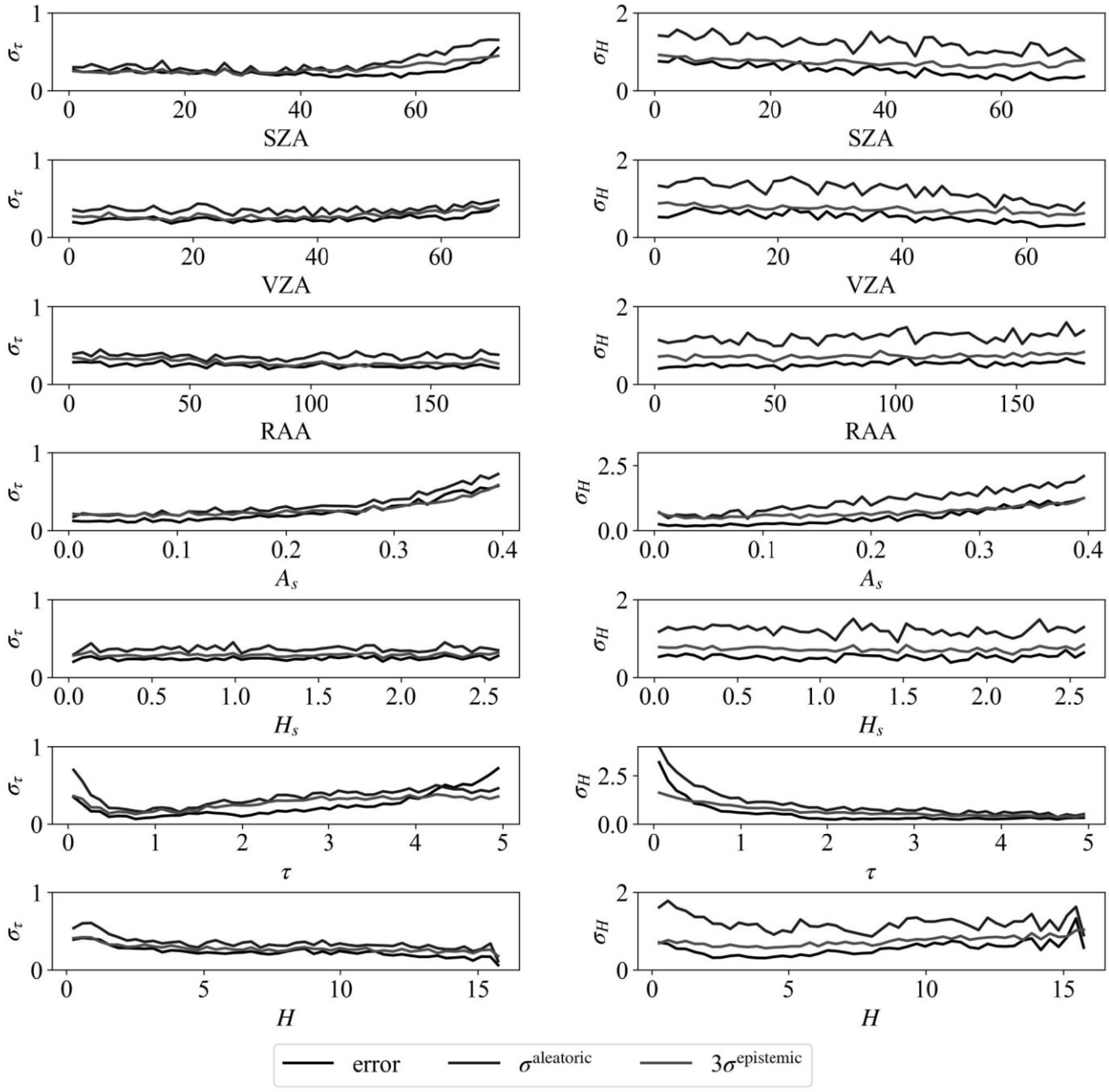


Fig. 4. Error, aleatoric uncertainty, and epistemic uncertainty versus input parameters for Bayes-by-Backprop.

C uses H within the training range and τ within the range $[0, 10]$, while Dataset D uses τ within the training range and H within the range $[0, 19.75]$ km. The relationships between retrieval errors and uncertainties for the three neural networks are shown in Figs. 8, 9, and 10, respectively.

Aleatoric uncertainty correlates with errors when the input

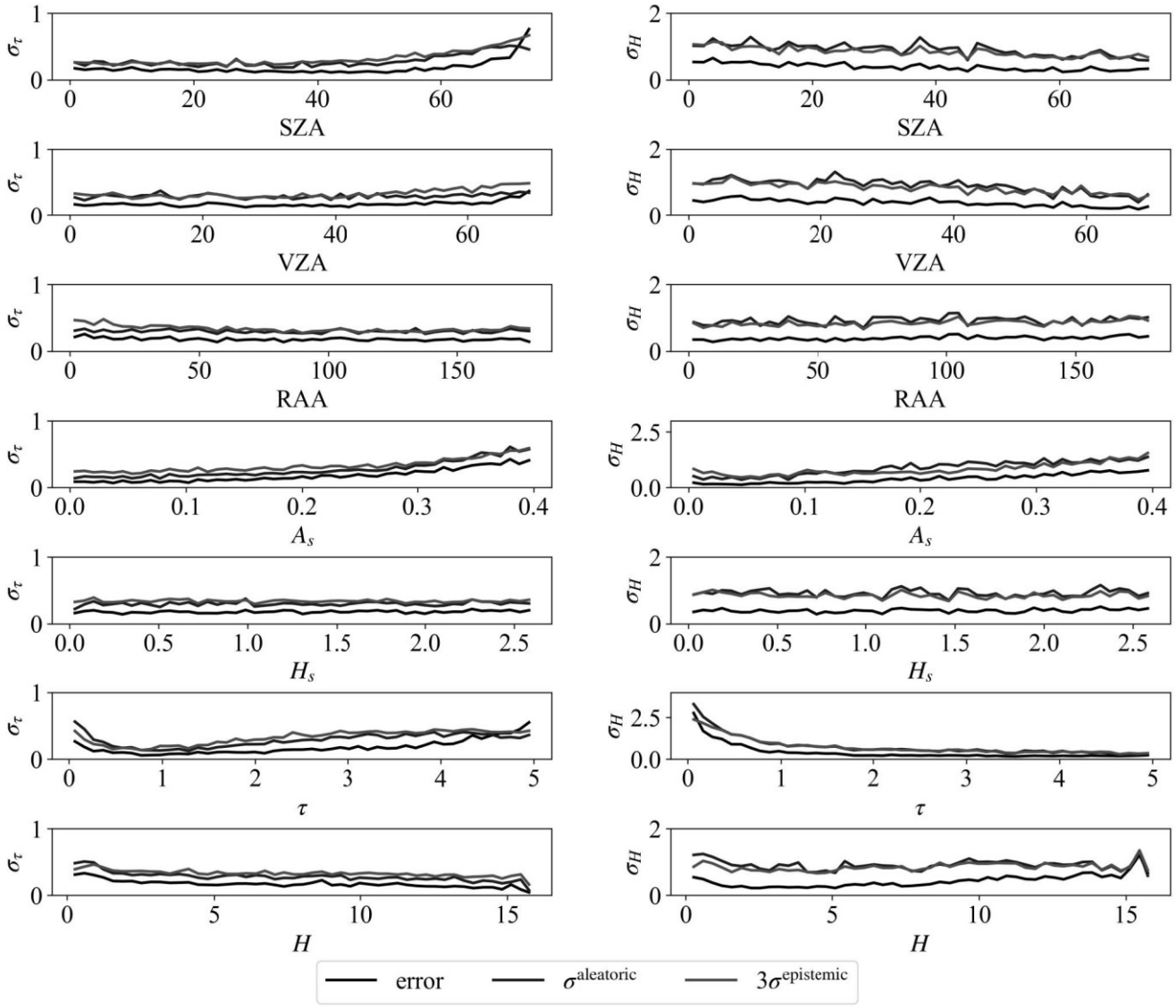


Fig. 5. Error, aleatoric uncertainty, and epistemic uncertainty versus input parameters for Dropout.

variance falls within the training dataset. However, for out-of-distribution scenarios, aleatoric uncertainty can remain low despite significant retrieval errors, particularly evident in: (1) τ and H uncertainties under extremely high surface albedo, and (2) τ uncertainty under extremely high aerosol loading. Notably, the Batch Norm method shows an increasing trend in the aleatoric uncertainty of τ with increasing τ, contrasting with the behaviors of Bayes-by-Backprop and Dropout methods. Overall, aleatoric uncertainty originates from data characteristics, appearing to be pronounced under unfavorable retrieval conditions (e.g., bright surface or low aerosol loading). However, it may underestimate actual uncertainty when viewing conditions deviate substantially from the training data distribution.

Epistemic uncertainty represents the uncertainty arising from model limitations. The epistemic uncertainty across all cases exhibits a strong correlation with the absolute errors, indicating its effectiveness in characterizing retrieval uncertainty compared to the aleatoric uncertainty. This correlation suggests that higher epistemic uncertainty typically corresponds to larger retrieval errors. The increased uncertainty mainly results from inadequate representation of the retrieval process, either due to its limitation in performance under unfavorable viewing conditions or insufficient training data to fully capture the system variability.

V. Retrieval Using Real Data

The three neural networks are evaluated using TROPOMI measurements for two different aerosol events: a dust storm case over China and a wildfire case over California. Surface albedo data are obtained from the TROPOMI Surface LER & DLER database [45], with the snow/ice-covered flag from

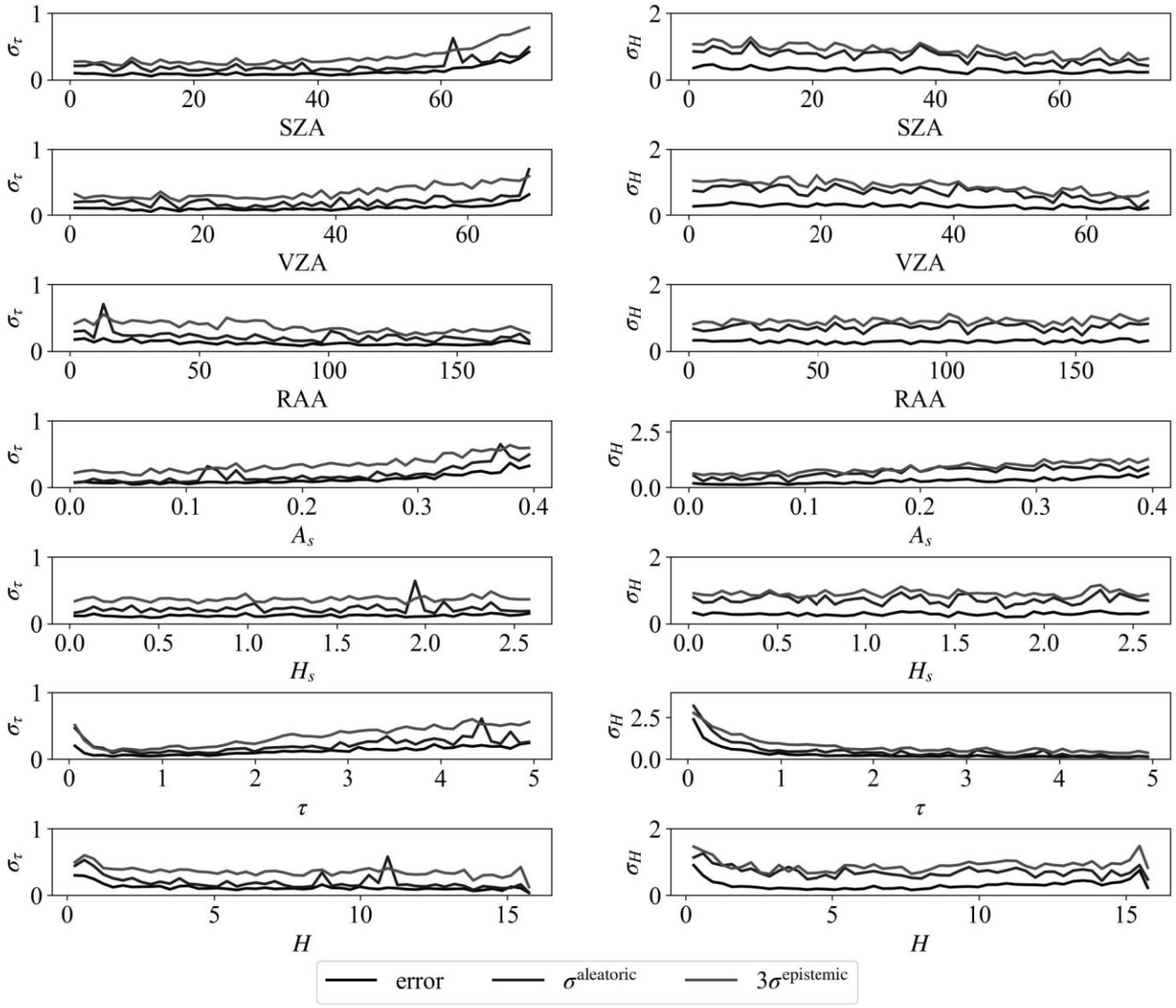


Fig. 6. Error, aleatoric uncertainty, and epistemic uncertainty versus input parameters for Batch Norm.

the official TROPOMI L2 product determining the selection between 'clear' (snow/ice free) and 'snice' (snow/ice-covered) albedo fields in the retrieval process.

The retrieved τ and H for the dust case and the wildfire case together with the logarithm of their epistemic and aleatoric uncertainties are shown in Figs. 11a, 11b, 11c and 13a, 13b, 13c. To provide better insight into the results, Figs. 11d and 13d show ancillary parameters including cloud fraction, aerosol index from the official TROPOMI L2 product, and surface albedo used in the retrievals. The retrievals are also compared with the τ and H results from the official TROPOMI L2 product as shown in Figs. 12a, 12b, 12c and 14a. 14b, 14c Based on the synthetic data analysis where $\sigma^{\text{aleatoric}}$ and $3\sigma^{\text{epistemic}}$ were found comparable to the actual retrieval errors, we implement quality filtering using thresholds of: $\ln \sigma_\tau^{\text{aleatoric}} > -1.6$, $\ln \sigma_\tau^{\text{epistemic}} > -2.7$, $\ln \sigma_H^{\text{aleatoric}} > -0.7$ and $\ln \sigma_H^{\text{epistemic}} > -1.8$ corresponding to the maximum permissible errors of 0.2 for τ and 0.5 km for H.

Higher uncertainties are observed under conditions of high surface albedo or low τ, suggesting suboptimal viewing conditions. In contrast, uncertainties decrease in cases with pronounced aerosol loading, characterized by high values of aerosol index and optical depth. A strong correlation exists between aleatoric and epistemic uncertainties, especially for the Batch Norm method. However, the Bayes-by-Backprop and the Dropout methods show divergence between these uncertaint measures when dealing with viewing conditions absent from the training dataset (e.g., very high cloud fractions). Under such circumstances, although the aleatoric uncertainty remains low, the consistently high epistemic uncertainty for all three neural networks demonstrates its superior reliability as an

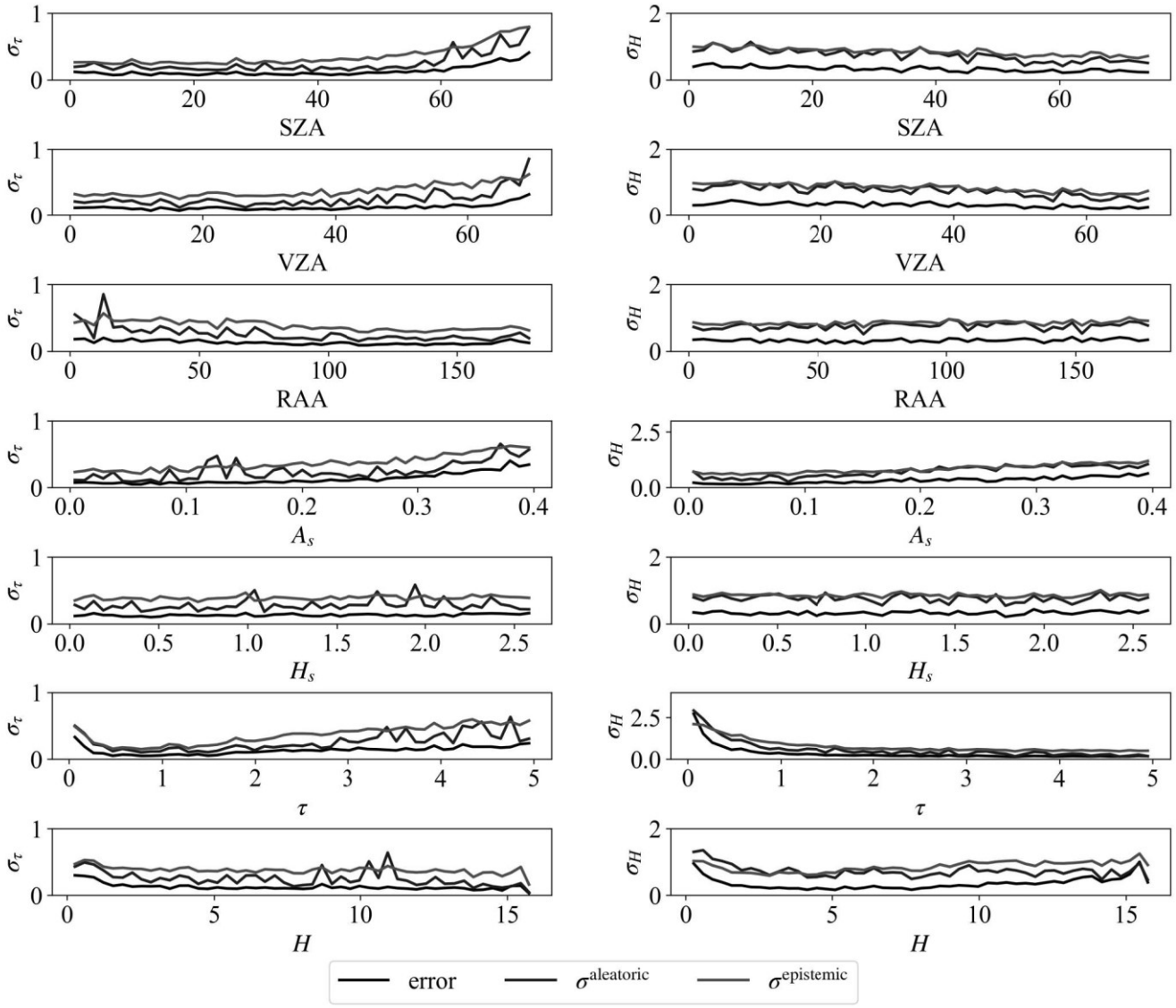


Fig. 7. Error, aleatoric uncertainty, and epistemic uncertainty versus input parameters for Batch Norm, with 100 samples to calculate the moving mean and variance.

uncertainty indicator. All three algorithms face challenges in separating aerosol and cloud signals when their magnitudes are comparable. Notably, even at moderately high cloud fractions, the epistemic uncertainty may remain low, indicating continued reliance on aerosol index and cloud fraction parameters for aerosol-cloud separation.

The comparison with the official TROPOMI aerosol product reveals several important insights. The retrieved τ and H demonstrate strong correlation with the official product, though minor discrepancies are expected given the different aerosol models employed in our retrieval algorithms. The Batch Norm method yields systematically elevated estimates of H in the upper-right panel of Fig. 13c, a feature requiring further investigation. Most significantly, the magnitude of H uncertainties shows quantitative correspondence with retrieval accuracy, confirming their practical utility as reliability indicators in operational applications.

VI. Conclusions and Outlook

This study describes three Bayesian neural networks implemented with the Bayes-by-Backprop, Dropout and Batch Norm method for joint retrieval of aerosol optical depth and layer height with integrated uncertainty quantification. The framework quantifies the total uncertainty through variance estimation based on the Bayesian theory, where:

- Aleatoric uncertainty represents the uncertainty arising from data noise in both input and output domains. By adopting a heteroscedastic formulation and incorporating output variance into the loss function, the neural network can directly estimate this uncertainty alongside target parameters.

- Epistemic uncertainty is evaluated via Monte Carlo sampling of the Bayesian network's stochastic forward propagation. The output variance across multiple samplings is calculated to represent the epistemic (model) uncertainty.

The performance of the three neural networks has been investigated using both synthetic and real measurements. According to the synthetic analysis, errors in retrievals show a good correlation with epistemic uncertainty, while aleatoric uncertainty may not align well with errors if the viewing conditions are beyond the variation range of the training dataset. Therefore, epistemic uncertainty can explain the uncertainty in retrievals more effectively. The experiment on real measurements suggests the same conclusion. The estimated aerosol optical depth and layer height from all three neural networks are comparable to the results from the official TROPOMI product.

The three neural networks have demonstrated significant potential for aerosol retrieval. Retrieving all 4172×448 pixels in a single orbit with uncertainty quantification (100 Monte Carlo samplings) requires approximately 650, 300, and 180 seconds for Bayes-by-Backprop, Dropout, and Batch Norm, respectively, on a workstation equipped with 64 GB of RAM and a 32-core Intel Core i9-13900K processor. In contrast, physical retrieval algorithms require comparable processing time to retrieve only 1–4 pixels.

Among the three Beyesian Neural Networks, Bayes-by-Backprop offers the most theoretically rigorous implementation, while Dropout and Batch Norm serve as practical approximations. Bayes-by-Backprop explicitly learns probability distributions for all neural network parameters by training both the mean and variance of each weight and bias, providing an interpretable representation of parameter uncertainty. However, its implementation is complex and computationally demanding due to repeated sampling of every parameter during both training and inference. In comparison, Dropout turns out to be the most straightforward and stable method, randomly zeroing neuron weights during training and yielding consistent performance across training, validation, and inference. Batch Norm shows superior optimization efficiency, typically achieving faster convergence and lower training loss, which makes it particularly suitable for complex and large-scale problems. Its main limitation lies in the requirement for representative mini-batches to compute reliable normalization statistics, as poor batch selection may degrade performance.

This study provides a foundation for improving Bayesian neural networks in aerosol retrieval. Future work should enhance the physical constraints in the neural networks, particularly for optically thick aerosol conditions. Expanding the training datasets with more diverse observation scenarios would improve robustness. Operational applications would benefit from near-real-time assimilation of ground measurements to refine the retrievals. Further development should focus on better uncertainty quantification under challenging conditions like high cloud cover.

REFERENCES

[1] X. Li, F. Wagner, W. Peng, J. Yang, and D. L. Mauzerall, "Reduction of solar photovoltaic resources due to air pollution in china," *Proceedings of the National Academy of Sciences*, vol. 114, no. 45, pp. 11 867–11 872, 2017.

[2] J. Li, B. E. Carlson, Y. L. Yung, D. Lv, J. Hansen, J. E. Penner, H. Liao, V. Ramaswamy, R. A. Kahn, P. Zhang, O. Dubovik, A. Ding, A. A. Lacis, L. Zhang, and Y. Dong, "Scattering and absorbing aerosols in the climate system," *Nature Reviews Earth & Environment*, vol. 3, no. 6, pp. 363–379, 2022.

[3] Q. Zhang, J. Quan, X. Tie, M. Huang, and X. Ma, "Impact of aerosol particles on cloud formation: Aircraft measurements in china," *Atmos. Environ.*, vol. 45, no. 3, pp. 665–672, 2011.

[4] Z. Li, J. Guo, A. Ding, H. Liao, J. Liu, Y. Sun, T. Wang, H. Xue, H. Zhang, and B. Zhu, "Aerosol and boundary-layer interactions and impact on air quality," *National Science Review*, vol. 4, no. 6, pp. 810–833, 2017.

[5] M. L. Pöhlker, C. Pöhlker, O. O. Krüger, J.-D. Förster, T. Berkemeier, W. Elbert, J. Fröhlich-Nowoisky, U. Pöschl, G. Bagheri, E. Bodenschatz, J. A. Huffman, S. Scheithauer, and E. Mikhailov, "Respiratory aerosols and droplets in the transmission of infectious diseases," *Rev. Mod. Phys.*, vol. 95, p. 045001, 2023.

[6] D. Efremenko and A. Kokhanovsky, *Foundations of Atmospheric Remote Sensing*. Springer International Publishing, 2021.

[7] I. Chuprov, D. Konstantinov, D. Efremenko, V. Zemlyakov, and J. Gao, "Solution of the radiative transfer equation for vertically inhomogeneous media by numerical integration solvers: Comparative analysis," *Light & Engineering*, vol. 30, pp. 21–30, 2022.

[8] S. Zhu, J. Xu, C. Yu, Y. Wang, D. S. Efremenko, X. Li, and Z. Sui, "Decsolnet: A noise resistant missing information recovery framework for daily satellite no2 columns," *Atmos. Environ.*, vol. 246, p. 118143, 2021.

[9] L. David, F.-M. Bréon, and F. Chevallier, "Xco_2 estimates from the oco-2 measurements using a neural network approach," *Atmos. Meas. Tech.*, vol. 14, no. 1, pp. 117–132, 2021.

[10] S. Chen, V. Natraj, Z.-C. Zeng, and Y. L. Yung, "Machine learning-based aerosol characterization using OCO-2 O2 A-band observations," *J. Quant. Spectrosc. Radiat. Transf.*, vol. 279, p. 108049, 2022.

[11] S. Zhu, J. Xu, J. Zeng, P. He, Y. Wang, S. Bao, J. Ma, and J. Shi, "UFLUX-GPP: A Cost-Effective Framework for Quantifying Daily Terrestrial Ecosystem Carbon Uptake Using Satellite Data," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–17, 2024.

[12] W. Wang, J. Xu, H. Letu, L. Zhang, Z. Wang, and J. Shi, "A New Deep-Learning-Based Framework for Ice Water Path Retrieval From Microwave Humidity Sounder-II Aboard FengYun-3D Satellite," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–14, 2024.

[13] Z. Wang, X. Su, L. Wang, Q. Lang, Y. Lu, and L. Wang, "A physics-guided neural network model to estimate all-sky diffuse solar radiation using Himawari-8 data," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–18, 2025.

[14] Y. Fan, W. Li, C. K. Gatebe, C. Jamet, G. Zibordi, T. Schroeder, and K. Stamnes, "Atmospheric correction over coastal waters using multilayer neural networks," *Remote Sensing of Environment*, vol. 199, pp. 218–240, 2017.

[15] C. Fan, G. Fu, A. Di Noia, M. Smit, J. HH Rietjens, R. A Ferrare, S. Burton, Z. Li, and O. P Hasekamp, "Use of a neural network-based ocean body radiative transfer model for aerosol retrievals from multi-angle polarimetric measurements," *Remote Sens.*, vol. 11, no. 23, p. 2877, 2019.

[16] S. Nanda, M. de Graaf, J. P. Veefkind, M. ter Linden, M. Sneep, J. de Haan, and P. F. Levelt, "A neural network radiative transfer model approach applied to the TROPOspheric Monitoring Instrument aerosol height algorithm," *Atmospheric Measurement Techniques*, vol. 12, no. 12, pp. 6619–6634, 2019.

[17] C. Shi, M. Hashimoto, K. Shiomi, and T. Nakajima, "Development of an algorithm to retrieve aerosol optical properties over water using an artificial neural network radiative transfer scheme: First result from GOSAT-2/CAI-2," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[18] M. Gao, B. A. Franz, K. Knobelspiesse, P.-W. Zhai, V. Martins, S. Burton, B. Cairns, R. Ferrare, J. Gales, O. Hasekamp *et al.*, "Efficient multi-angle polarimetric inversion of aerosols and ocean color powered by a deep neural network forward model," *Atmos. Meas. Tech.*, vol. 14, no. 6, pp. 4083–4110, 2021.

[19] D. G. Loyola R, "Applications of neural network methods to the processing of earth observation satellite data," *Neural networks*, vol. 19, no. 2, pp. 168–177, 2006.

[20] D. S. Efremenko, "Discrete ordinate radiative transfer model with the neural network based eigenvalue solver: Proof of concept," *Light & Engineering*, vol. 01, pp. 56–62, 2021.

[21] J. Xu, Z. Zhang, L. Rao, Y. Wang, H. Letu, C. Shi, G. Tana, W. Wang, S. Zhu, S. Liu, E. Shi, Y. Wang, L. Chen, X. Dong, and J. Shi, "Remote sensing of tropospheric ozone from space: Progress and challenges," *J. Remote Sens.*, vol. 4, p. 0178, 2024.

[22] G. Holl, S. Eliasson, J. Mendrok, and S. Buehler, "SPARE-ICE: Synergistic ice water path from passive operational sensors," *Journal of Geophysical Research: Atmospheres*, vol. 119, no. 3, pp. 1504–1523, 2014.

[23] J. Strandgren, L. Bugliaro, F. Sehnke, and L. Schröder, "Cirrus cloud retrieval with MSG/SEVIRIs using artificial neural networks," *Atmospheric Measurement Techniques*, vol. 10, no. 9, pp. 3547–3573, 2017.

[24] D. Wang, C. Prigent, F. Aires, and C. Jimenez, "A statistical retrieval of cloud parameters for the millimeter wave Ice Cloud Imager on board MetOp-SGs," *IEEE Access*, vol. 5, pp. 4057–4076, 2017s.

[25] M. Brath, S. Fox, P. Eriksson, R. C. Harlow, M. Burgdorf, and S. A. Buehler, "Retrieval of an ice water path over the ocean from ISMAR and MARSS millimeter and submillimeter brightness temperatures," *Atmospheric Measurement Techniques*, vol. 11, no. 1, pp. 611–632, 2018.

[26] N. Håkansson, C. Adok, A. Thoss, R. Scheirer, and S. Hörnquist, "Neural network cloud top pressure and height for MODIS," *Atmospheric Measurement Techniques*, vol. 11, no. 5, pp. 3177–3196, 2018.

[27] D. S. Efremenko, D. G. Loyola R., P. Hedelt, and R. J. D. Spurr, "Volcanic SO_2 plume height retrieval from UV sensors using a full-physics inverse learning machine algorithm," *Int. J. Remote Sensing*, vol. 38, no. sup1, pp. 1–27, 2017.

[28] J. Xu, O. Schüssler, D. Loyola R, F. Romahn, and A. Doicu, "A novel ozone profile shape retrieval using Full-Physics Inverse Learning Machine (FP-ILM)," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 12, pp. 5442–5457, 2017.

[29] D. G. Loyola, J. Xu, K.-P. Heue, and W. Zimmer, "Applying FP_ILM to the retrieval of Geometry-dependent Effective Lambertian Equivalent Reflectivity (GE_LER) daily maps from UVN satellite measurements," *Atmos. Meas. Tech.*, vol. 13, no. 2, pp. 985–999, 2020.

[30] A. Di Noia, O. Hasekamp, G. Van Harten, J. Rietjens, J. Smit, F. Snik, J. Henzing, J. De Boer, C. Keller, and H. Volten, "Use of neural networks in ground-based aerosol retrievals from multi-angle spectropolarimetric observations," *Atmospheric Measurement Techniques*, vol. 8, no. 1, p. 281, 2015.

[31] A. D. Noia, O. P. Hasekamp, L. Wu, B. v. Diedenhoven, B. Cairns, and J. E. Yorks, "Combined neural network/Phillips–Tikhonov approach to aerosol retrievals over land from the NASA Research Scanning Polarimeter," *Atmospheric Measurement Techniques*, vol. 10, no. 11, pp. 4235–4252, 2017.

[32] D. Efremenko, H. Jain, and J. Xu, "Two machine learning based schemes for solving direct and inverse problems of radiative transfer theory," *Proceedings of the 30th International Conference on Computer Graphics and Machine Vision (GraphiCon 2020). Part 2*, pp. paper45–1–paper45–12, 2020.

[33] W. Wang, H. Letu, H. Shang, J. Xu, H. Yan, L. Gao, C. Yu, J. Gu, J. Tao, N. Xu, L. Chen, and L. Chen, "A novel physics-based cloud retrieval algorithm based on neural networks (CRANN) from hyperspectral measurements in the O2-O2 band," *Remote Sens. Environ.*, vol. 311, p. 114267, 2024.

[34] C. Li, X. Xu, X. Liu, J. Wang, K. Sun, J. van Geffen, Q. Zhu, J. Ma, J. Jin, K. Qin, Q. He, P. Xie, B. Ren, and R. C. Cohen, "Direct retrieval of NO_2 vertical columns from UV-Vis (390-495 nm) spectral radiances using a neural network," *J. Remote Sens.*, vol. 2022, 2022.

[35] J. Xu, Y. Wang, L. Chen, D. Efremenko, L. Rao, G. Tana, S. Liu, Q. Wang, J. Mao, Y. Wang, L. Sun, H. Yan, N. Xu, X. Hu, H. Letu, and J. Shi, "First total ozone column observations from the Ozone Monitoring Suite-Nadir (OMS-N) onboard China's FengYun-3F satellite," *Sci. China Earth Sci.*, vol. 68, 2025.

[36] W. Chen, T. Ren, C. Zhao, Y. Wen, Y. Gu, M. Zhou, and P. Wang, "Transformer-based fast mole fraction of CO_2 retrievals from satellite-measured spectra," *J. Remote Sens.*, vol. 5, p. 0470, 2025.

[37] S. Sasi, V. Natraj, V. Molina García, D. S. Efremenko, D. Loyola, and A. Doicu, "Model selection in atmospheric remote sensing with an application to aerosol retrieval from DSCOVR/EPIC, Part 1: Theory," *Remote Sens.*, vol. 12, no. 22, 2020, 3724.

[38] ——, "Model selection in atmospheric remote sensing with application to aerosol retrieval from DSCOVR/EPIC. Part 2: Numerical analysis," *Remote Sens.*, vol. 12, no. 21, 2020, 3656.

[39] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation," *arXiv preprint arXiv:1506.02157*, 2015.

[40] M. Teye, H. Azizpour, and K. Smith, "Bayesian uncertainty estimation for batch normalized deep networks," 2018. [Online]. Available: https://arxiv.org/abs/1802.06455

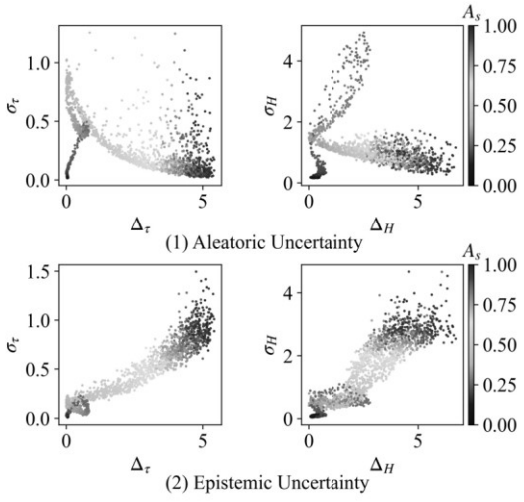
[41] J. H. Halton, "Algorithm 247: Radical-inverse quasi-random point sequence," *Communications of the ACM*, vol. 7, no. 12, pp. 701–702, 1964.

[42] D. G. Loyola, M. Pedergnana, and S. Gimeno Garcia, "Smart sampling and incremental function learning for very large high dimensional data," *Neural Networks*, vol. 78, pp. 75–87, 2016.

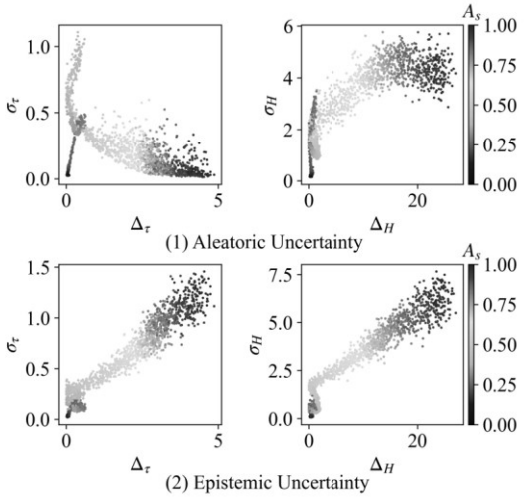
[43] L. Rao, J. Xu, D. S. Efremenko, D. G. Loyola, and A. Doicu, "Aerosol parameters retrieval from tropomi/s5p using physics-based neural networks," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 6473–6484, 2022.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

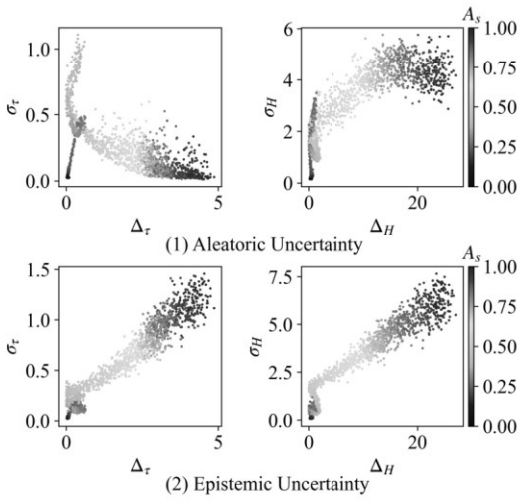
[45] L. G. Tilstra, M. de Graaf, V. J. H. Trees, P. Litvinov, O. Dubovik, and P. Stammes, "A directional surface reflectance climatology determined from tropomi observations," *Atmospheric Measurement Techniques*, vol. 17, no. 7, pp. 2235–2256, 2024. [Online]. Available: https://amt.copernicus.org/articles/17/2235/2024/



(a) Bayes-by-Backprop

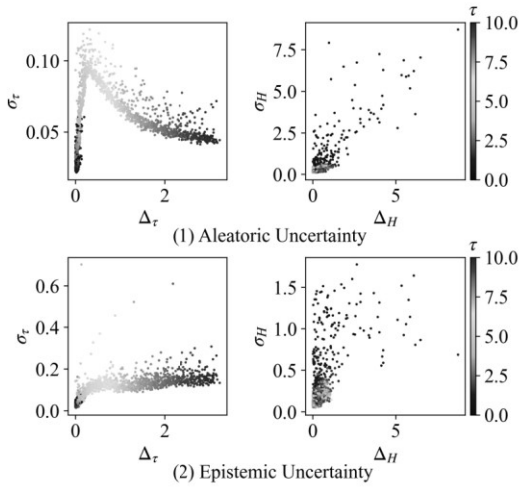


(b) Dropout

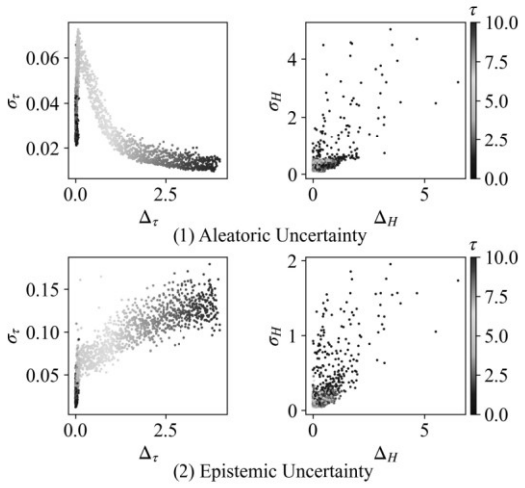


(c) Batch Norm

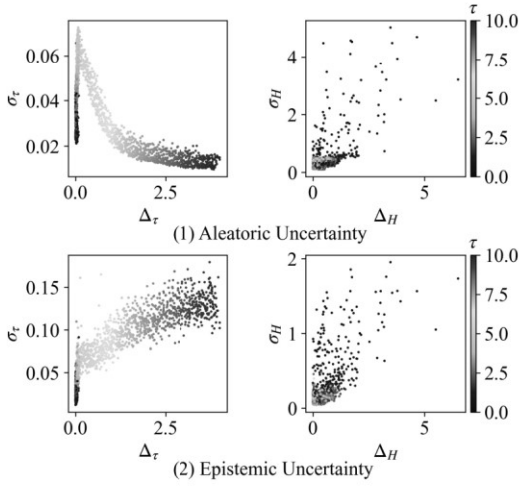
Fig. 8. Error versus aleatoric and epistemic uncertainty for surface albedo between 0 and 1.



(a) Bayes-by-Backprop



(b) Dropout



(c) Batch Norm

Fig. 9. Error versus aleatoric and epistemic uncertainty for aerosol optical depth between 0 and 10.

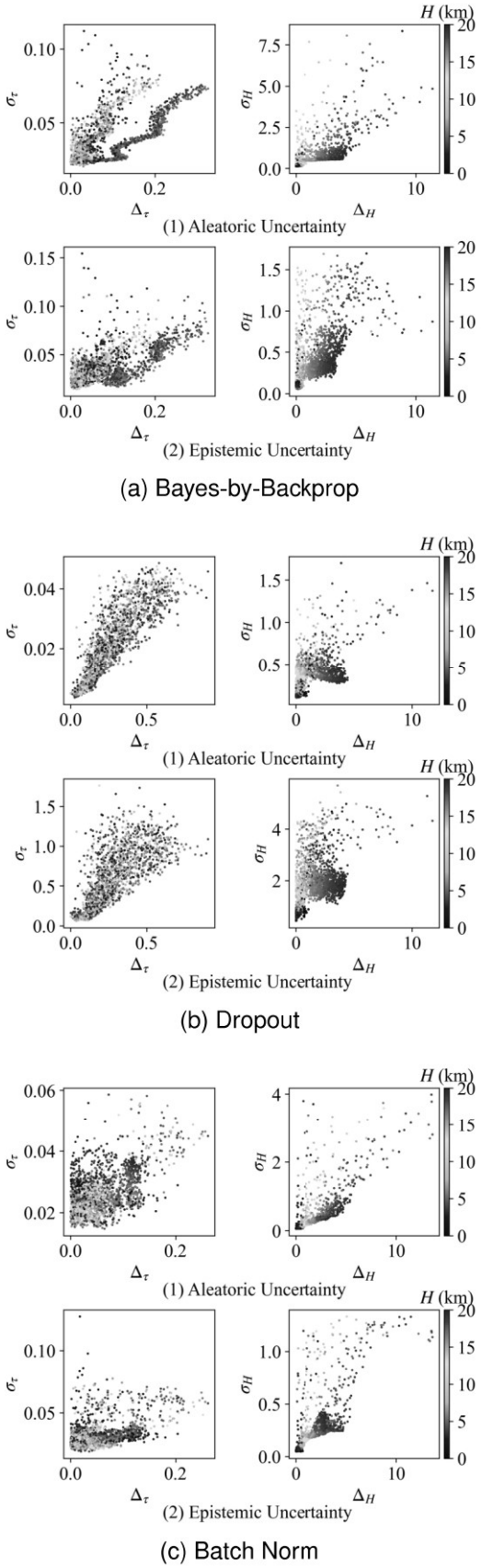


Fig. 10. Error versus aleatoric and epistemic uncertainty for aerosol layer height between 0 and 20 km.

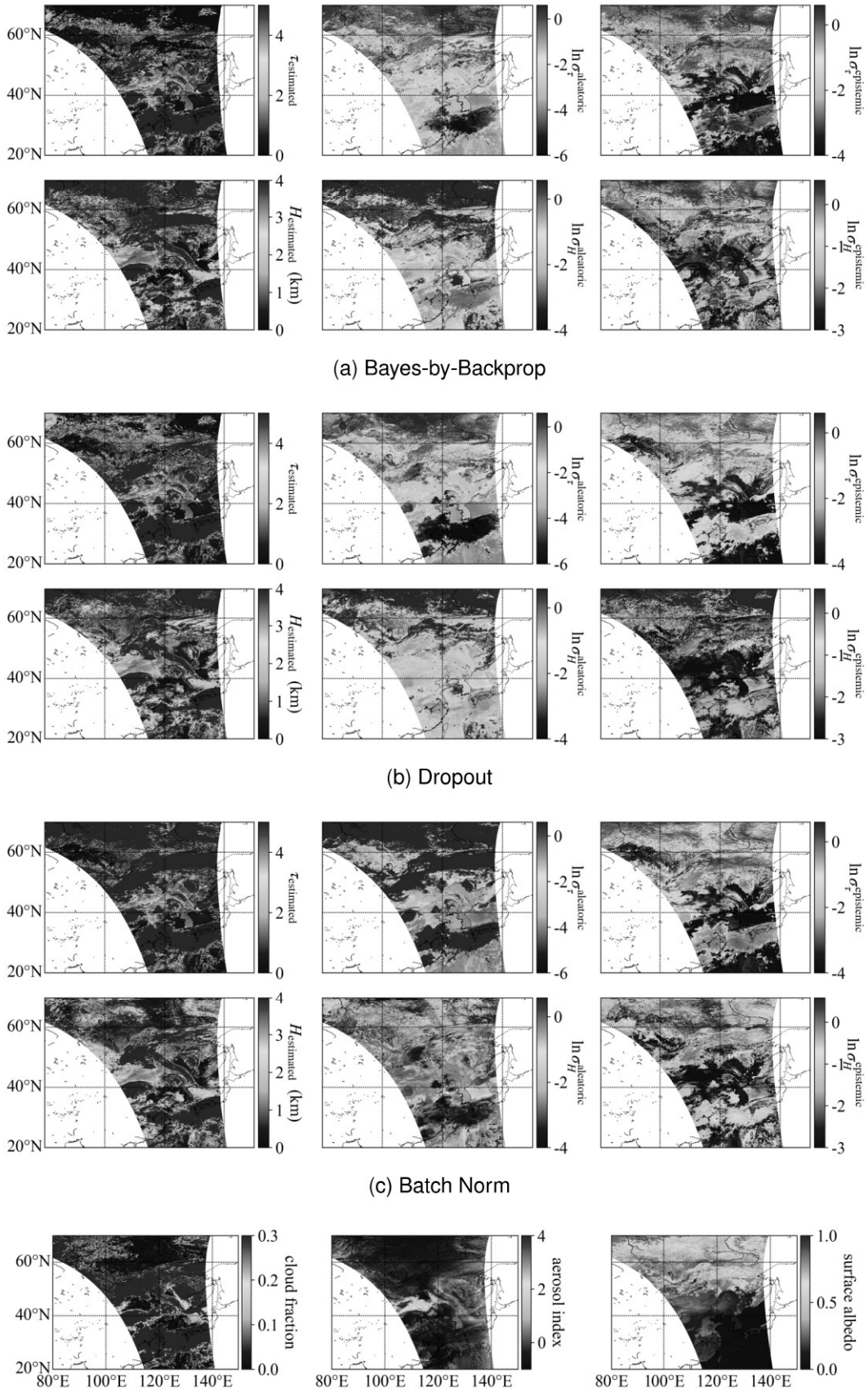


Fig. 11. Retrieval of aerosol optical depth and aerosol layer height for a dust storm case over China on March 21, 2023

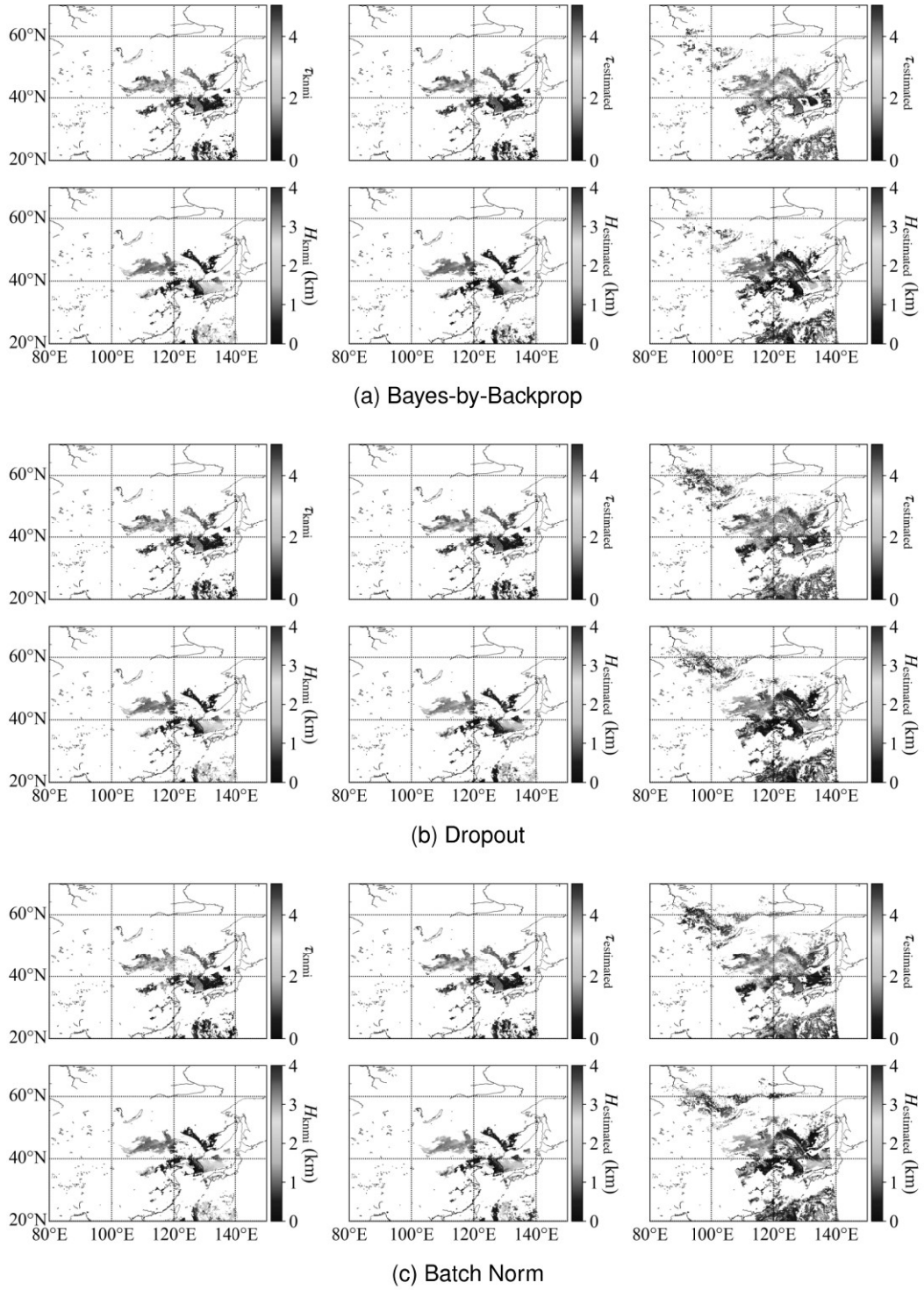


Fig. 12. Comparison with TROPOMI Aerosol Layer Height Product for a dust storm case over China on March 21, 2023.

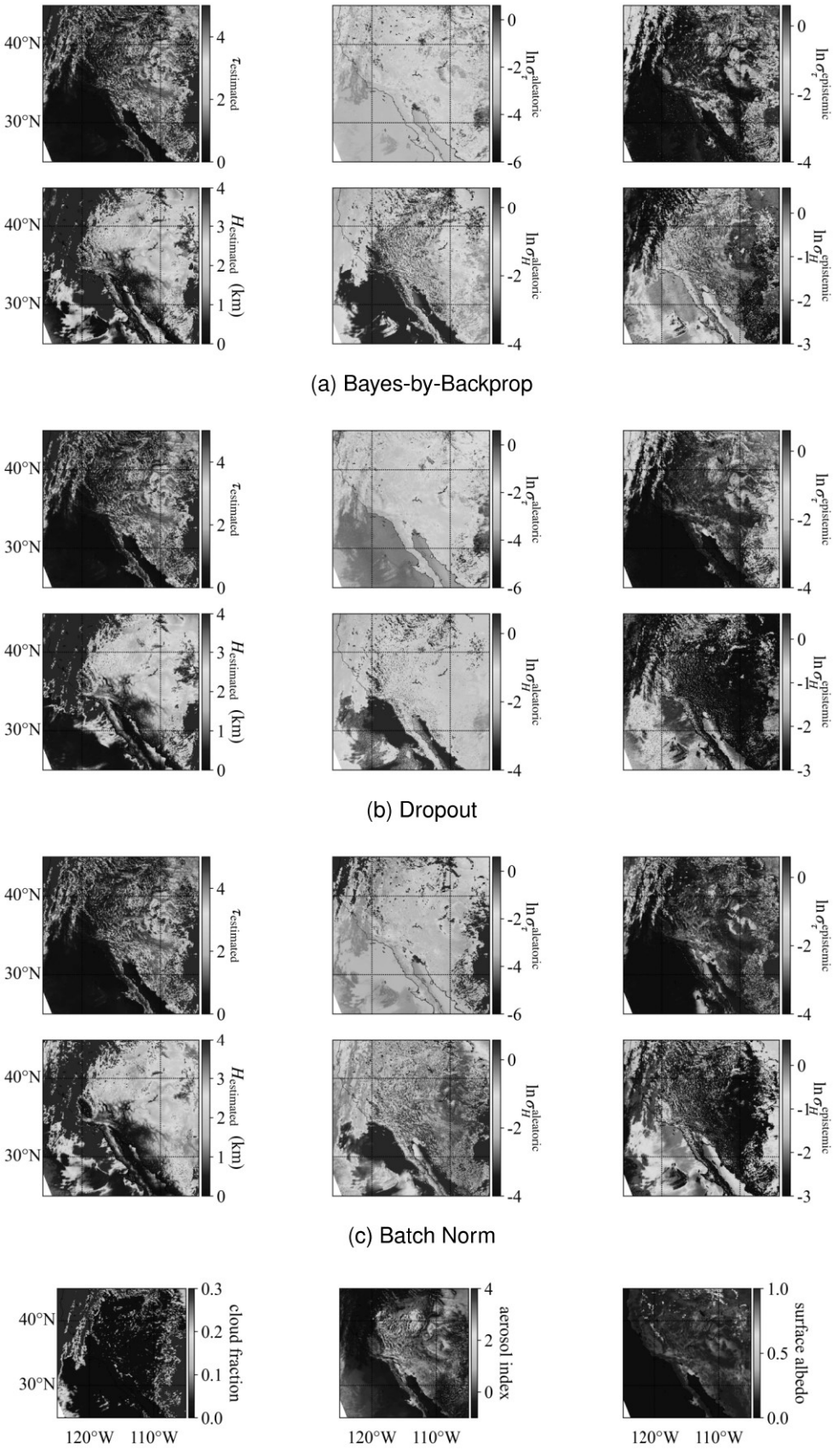


Fig. 13. Retrieval of aerosol optical depth and aerosol layer height for a wildfire case over California on November 9, 2023

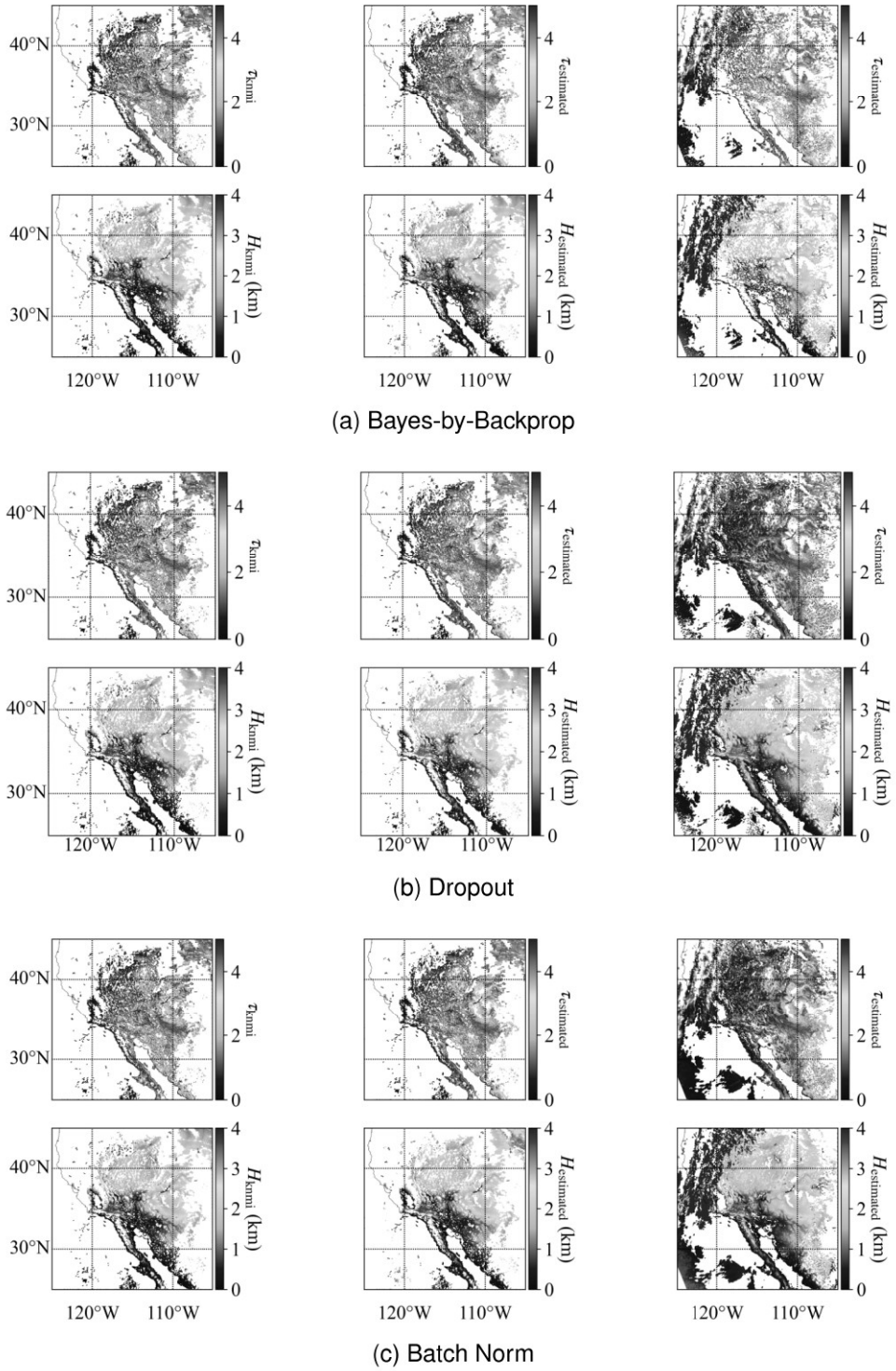


Fig. 14. Comparison with TROPOMI Aerosol Layer Height Product for the wildfire case over California on November 9, 2023.