AMV Error Characterization and Bias Correction by Leveraging Independent Lidar Data: a Simulation using OSSE and Optical Flow AMVs

Hai Nguyen[1], Derek Posselt[1], Igor Yanovsky[1], Longtao Wu[1], and Svetla Hristova-Veleva[1]

[1]Jet Propulsion Laboratory, California, 4800 Oak Grove Dr, Pasadena, CA

Correspondence: Hai Nguyen (hai.nguyen@jpl.nasa.gov)

Abstract.

Accurate estimation of global winds is crucial for various scientific and practical applications, such as global chemical transport modeling and numerical weather prediction. One valuable source of wind measurements is Atmospheric Motion Vectors (AMVs), which play a vital role in the global observing system and numerical weather prediction models. However, errors in AMV retrievals need to be addressed before their assimilation into data assimilation systems, as they can affect the accuracy of outputs.

An assessment of the bias and uncertainty in passive-sensor AMVs can be done by comparing them with information from independent sources such as active-sensor winds. In this paper, we examine the benefit and performance of a colocation scheme using independent and sparse lidar wind observations as a dependent variable in a supervised machine learning model. We demonstrate the feasibility and performance of this approach in an Observing System Simulation Experiment (OSSE) framework, with reference geophysical state data obtained from high resolution Weather Research and Forecasting (WRF) Model simulations of three different weather events.

Lidar wind data are typically available in only one direction, and our study demonstrates that this single component of wind in high-precision active-sensor data can be leveraged (via a machine learning algorithm to model the conditional mean) to reduce the bias in the passive-sensor winds. Further, this active-sensor wind information can be leveraged through an algorithm that models the conditional quantiles to produce stable estimates of the prediction intervals, which are helpful in design and application of error analysis such as quality filters.

1 Introduction

The accurate estimation of global winds is critical for various scientific and practical applications, including global chemical transport modeling and numerical weather prediction. One source of wind measurements is atmospheric motion vectors (AMVs), which are obtained through the tracking of cloud or water vapor features in satellite imagery. They play a crucial role in the global observing system, providing essential data for initializing numerical weather prediction (NWP) models; these AMVs are particularly valuable for constraining the wind field in remote Southern Hemisphere regions and over the world's oceans, where other wind observations are scarce. Obtaining global measurements of three-dimensional winds was emphasized as an urgent need in the NASA Weather Research Community Workshop Report (Zeng et al., 2016) and identified as a priority in the 2007 National Academy of Sciences Earth Science and Applications from Space (ESAS 2007) decadal survey, as well as in ESAS 2017. Numerous studies have demonstrated the positive impact of AMVs on the forecast accuracy of global NWP models (Bormann and Thépaut, 2004; Velden and Bedka, 2009; Gelaro et al., 2010). Further uses include studying global CO2 transport (Kawa et al., 2004), providing inputs for weather and climate reanalysis studies (Swail and Cox, 2000), and estimating present and future wind-power outputs (Staffell and Pfenninger, 2016). Major NWP centers now incorporate AMVs from various geostationary and polar-orbiting satellites, resulting in nearly global horizontal coverage, though vertical resolution is generally quite coarse.

Numerical weather prediction integrates Atmospheric Motion Vectors (AMVs) through a process called data assimilation, which involves combining observations of atmospheric variables with an a priori estimate of the atmospheric state (usually generated by a short-term forecast) to derive a posterior estimate of wind fields and other atmospheric state variables. To achieve accurate results, each input source of information is weighted using an inverse error covariance matrix meant to represent the accuracy of the data. Nguyen et al. (2019) analytically proved that inaccurate error characterizations of the inputs (i.e., a priori information) can adversely affect the bias and validity of the outputs, and similarly it is important to assess, and if possible, correct for biases in AMVs retrievals before their subsequent usage in data assimilation. Staffell and Pfenninger (2016), for instance, observed that NASA's MERRA and MERRA-2 wind product suffer significant spatial bias, overestimating wind output by 50% in northwest Europe and underestimating by 30% in the Mediterranean, and they noted that such biases can have adverse effect on the quality of data assimilation that ingests said data. Therefore, it is of paramount importance to assess and remove the biases inherent in AMV retrievals before their usage in subsequent analysis.

In practice, correcting the bias of an AMV retrieval requires an independent proxy for the 'truth', and previous studies assessing AMV uncertainty typically compared AMVs derived from Observing System Simulation Experiments (OSSE) with collocated radiosonde AMVs (Cordoba et al., 2017). Here, we propose the idea of using the independent (and sparse) lidar observations of wind as a dependent variable in a supervised machine learning model for bias correction. Following the OSSE framework of Posselt et al. (2019), we examine a proof-of-concept that demonstrates the feasibility and performance of an bias-correction scheme in an OSSE framework. We use as our reference (truth, or NatureRun) datasets output from the Weather Research and Forecasting (WRF) Model run for three different weather events (Posselt et al., 2019). The water vapor fields from these WRF model runs are processed through an Optical Flow algorithm (Yanovsky et al., 2024) to provide AMVs, and we similarly simulate lidar observations from the same WRF model data. Finally, we assess the ability of a bias-correction algorithm to model and correct biases (relative to the simulated lidar winds) that arise from the optical flow AMV retrieval.

Velden and Bedka (2009) along with Salonen et al. (2015) have highlighted the significant impact of height assignment on the uncertainty of AMVs derived from cloud movement and sequences of infrared satellite radiance images. However, this error source is intertwined with uncertainties in the water vapor profile itself, and modeling this within the OSSE framework requires extensive knowledge and parameterization of the height-assignment error process, which is beyond the scope of this paper. As such, in this paper we will focus on fixed-height errors in the AMV estimates and the bias corrections arising therefrom.

One challenge with pairing passive-sensor and active-sensor winds is that the latter typically observes only in one direction, along the instrument's line of sight. Therefore, a question one might ask is what sort of information a researcher might be able to obtain on the entire wind-vector if, for example, lidar winds are only available at sparse locations in only the line-of-sight direction. In this paper, we search for the answer to this question in an OSSE framework, and we show that passive sensor can benefit from coincident active sensor data through algorithms that model the expectation (bias reduction) or quantiles (uncertainty quantification).

We are not aware of a similar approach in the literature for leveraging lidar wind retrievals for improvement of AMV retrievals, even in an OSSE context. Perhaps the closest would be Teixeira et al. (2021), which combined random forest with Gaussian mixture models to form regime-based estimates of bias and uncertainty. While this approach in principle can be used to bias correct observations, it discretizes the bias error function into a fixed number of clusters. While this discretization is useful for understanding the geophysical regimes of the underlying atmospheric processes, it is not as efficient as a model that is purposely built for bias-minimization.

The intention of this paper is not to propose that the algorithms outlined here should replace error characterization methods for all AMVs. Instead, our primary objective is to demonstrate that residual error patterns exist in AMV retrieval algorithms, regardless of whether they involve traditional feature tracking or optical flow. Furthermore, through meticulous variable selection and algorithm refinement, it is feasible to curtail these biases. We also provide evidence that, in the four selected scenarios, the confidence intervals predicted using the Meinshausen and Ridgeway (2006) approach exhibit predominantly positive linear correspondence with the empirical validation standard error. This correlation is a notable and valuable characteristic that carries implications for devising indicators of AMV quality.

For the remainder of this paper, we will discuss the data sources, study regions, and the optical flow AMV retrieval algorithm in Section 2. In Section 3.1, we discuss the process of variable selection, and we discuss parameter optimization and bias-reduction performance in Section 3.2. We follow this treatment of bias with a discussion of modelling uncertainty via prediction intervals in Section 3.3. Finally, we end with some discussion of the merits of our approach and plans for further studies.

2 Data sources

The evaluation of the impact of bias correction on optical flow AMVs will be carried out in the context of an OSSE. All OSSEs share these key components: 1) A reference dataset, used as a basis for comparison. In our case, this is a NatureRun (NR), which is a high-fidelity simulation mimicking real-world conditions; 2) Simulators generating synthetic observations as if they were taken from the NR (this includes radiative transfer models, retrieval system simulations, and accounting for measurement errors, spatial, and temporal aspects); and 3) A quantitative methodology to evaluate information in the candidate measurements (Posselt et al., 2022). In this section, we shall discuss our choices for these components, with emphasis on the choice of study regions, the water vapor retrieval simulations, and the algorithm for computing AMV from the water vapor.

	Region	Spatial Resolution	Temporal Resolution	Beginning Time	Duration
ETC	Western Atlantic Ocean	4 km	120 sec	2006-11-22 00:02:00	12 hr
TC	Southeast Asia	3.5 km	72 sec	2008-07-10 06:00:00	4.5 hr
Harvey EDS	Gulf of Mexico	3 km	120 sec	2017-08-23 18:00:00	12 hr
Harvey LDS	Gulf of Mexico	3 km	120 sec	2017-08-24 06:00:00	12 hr

Table 1. Overview of spatial and temporal parameters for the study scenarios

2.1 Study regions

For a comprehensive view of the impact of bias correction across various atmospheric scenarios, we will examine three different systems which include an extratropical cyclone, tropical convection, and a hurricane at its early and late development stages. The details of these datasets describing the storm systems are summarized in Table 1. The extratropical cyclone (ETC) reference scenario is one that developed east of the United States in the western Atlantic Ocean in late November 2006. This cyclone showcases a diverse spectrum of wind speeds, water vapor contents, and gradients, providing an extensive assessment of the AMV algorithm's error traits across a wide array of atmospheric circumstances. This specific ETC case is selected due to its thorough examination in numerous previous observational studies (Posselt et al., 2008; Crespo and Posselt, 2016). For this case, simulations are carried out using the Advanced Research Weather Research and Forecasting (WRF) Model, version 3.8.1 (Skamarock et al. 2008). The model is configured with three nested domains (d01, d02, and d03) operating at horizontal resolutions of 20, 4, and 1.33 km, respectively, although for this paper we will focus primarily on the data at 4 km resolution and at pressure levels at 850, 500, and 300 hPa. For our analysis, we focus primarily on the 12-hour span of the storm starting on 2006-11-22 00:02:00 UTC.

For the tropical convection case, we consider a simulated water vapor dataset over the Maritime Continent from 0600 UTC to 1027 UTC on 10 July 2008. Similar to the ETC scenario, we chose pressure levels at 850, 500, and 300 hPa for analysis. Further details on this simulation can be found in Yanovsky et al. (2024).

Our third study scenario is a hurricane NatureRun, produced by initializing the Weather Research and Forecasting (WRF) Model using initial conditions from a ensemble forecast of Hurricane Harvey and described in further details in Posselt et al. (2022). It consists of a free-running simulation across four two-way nested domains using version 3.9.1 of the WRF Model. The initiation of this simulation occurs at 0000 UTC on August 23, 2017, utilizing the initial state that generated the third most powerful member within an ensemble forecast of Hurricane Harvey. The ensemble's initial conditions were established through the assimilation of a conventional set of observations and all-sky satellite brightness temperatures. The NR simulation spans 5 days, ending at 0000 UTC on August 28, 2017, while the outermost domain's boundaries are guided by analysis fields from the fifth-generation European Centre for Medium-Range Weather Forecasts (ECMWF) Reanalysis (ERA5).

This simulation is notably realistic, capturing both wind patterns and humidity levels. Posselt et al. (2022) noted that it exhibits rapid intensification; within a span of 24 hours from 1200 UTC on August 24 to 1200 UTC on August 25, the minimum sea level pressure plunges by approximately 40 hPa, and the storm's strength escalates from category 1 to category 4. To get a view of different stages of Hurricane Harvey, we shall focus on two 12-hour subsets of the storm: one between 1800 UTC on August 23 to 0600 UTC on August 24 (Early Development Stage; Harvey EDS), and one between 0600 UTC on August 24 to 1800 UTC on August 24 (Late Development Stage; Harvey LDS). Note that with the division of Harvey into early and late development stages, we have a set of four scenarios– ETC, TC, and Harvey EDS and LDS – on which we shall focus our analysis.

Traditional AMVs from cloud tracking are typically focused on high-level clouds (at around 200 hPa) and low-level clouds (at around 850 hPa). Tracking mid-level clouds poses a challenge because they are often obscured by high-level clouds. In this OSSE study, we are considering using AMVs derived from sounder-based water vapor retrievals, which are most reliable in the middle troposphere. Furthermore, lidar winds, primarily derived from the UV (Rayleigh scattering) channel, provide retrievals mainly in the middle to upper troposphere where the scattering signal is adequate for returning Doppler information, and the view is less likely to be obstructed by clouds. For these reasons, we opted to perform our OSSE error characterization experiments at the 850, 500, and 300 hPa pressure levels.

2.2 Optical flow AMVs

Optical flow methods are powerful computational techniques for analyzing the motion between two consecutive images across various fields (Horn and Schunck, 1981; Zach et al., 2007; Wedel et al., 2009). In the context of atmospheric science, these methodologies offer a sophisticated approach to extracting detailed atmospheric motion vectors (AMVs) from sequential satellite imagery, providing critical insights into wind patterns and dynamics essential for improving weather prediction models and climate research.

In Yanovsky et al. (2024), the authors employed a robust and efficient variational dense optical flow method that utilizes the conservation of pixel brightness across a pair of images with a regularization constraint. Given a pair of images $I_1(\hat{x})$ and $I_2(\hat{x})$, where $\hat{x}$ represents the pixel coordinates in the image plane, and $\hat{w} = (u, v)$ denotes the velocity vector field describing the apparent motion between the images, the functional being minimized with respect to $\hat{w}$ is conceptually defined as:

$$F(\hat{w}) \quad = \quad \lambda \int\limits_{\Omega} \big|I_2(\hat{x}+\hat{w}) - I_1(\hat{x})\big| d\hat{x} + \int\limits_{\Omega} \big(|\nabla u| + |\nabla v|\big) d\hat{x}, \tag{1}$$

where λ is the weighting parameter. The first term in $F(\hat{w})$ corresponds to the data fidelity term as an L1 norm between the first image and the warped second image, ensuring the similarity using the estimated motion field $\hat{w}$. The second term, involving the gradients of the velocity components u and v, represents the total variation (TV) regularization term that encourages smoothness in the estimated motion field.

In order to make the problem solvable, the L1 fidelity term is linearized, enabling efficient optimization. The TV regularization term is convexified to ensure that the optimization formulation is well-behaved, enabling the use of efficient numerical methods for finding the solution. To address the challenges posed by large displacements, the method employs a pyramid scheme, which processes the images at multiple scales, from coarse to fine, gradually refining the motion estimation. This

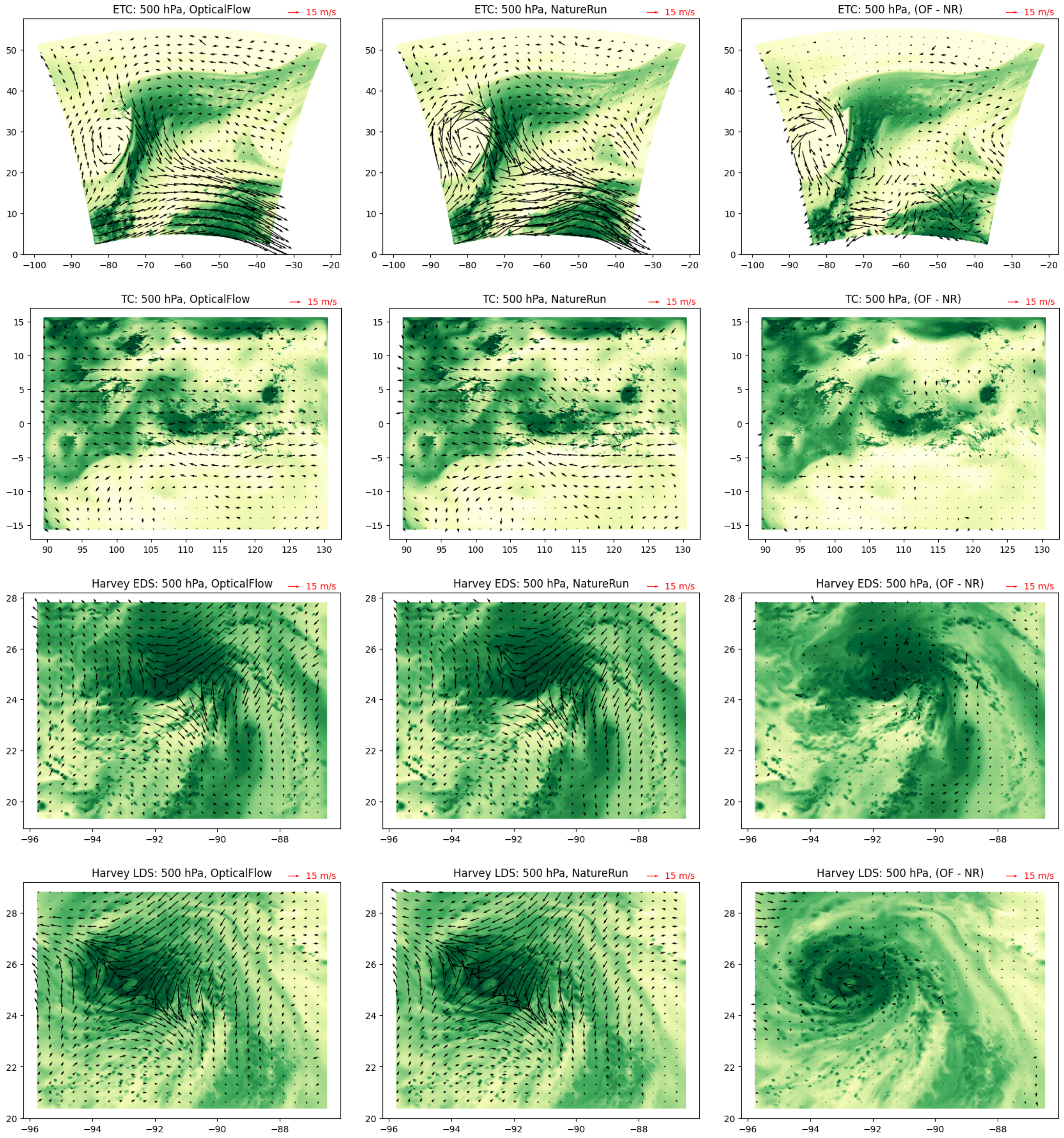

Figure 1. Quiver plot of the optical flow wind (left column), NatureRun wind (middle column) and differences (right column). The quivers are overlaid over a Yellow-Green heatmap of the water vapor, where yellow corresponds to low water vapor, and green corresponds to high water vapor. The rows correspond to ETC, TC, Harvey EDS, and Harvey LDS. These plots are selected from the middle of each study scenario, and their UTC time stamps are 2006-11-22 06:12:00 (ETC), 2008-07-10 08:18:00 (TC), 2017-08-24 00:10:00 (Harvey EDS), and 2017-08-24 12:10:00 (Harvey LDS).

multi-scale approach enhances the method's robustness to initial estimates and increases its ability to capture a wide range of motion magnitudes. The method, referred to as the TV-L1 optical flow algorithm, effectively handles discontinuities in the flow field while preserving edges in the motion, making it particularly suitable for capturing complex motion patterns in atmospheric data or other dynamic scenes. Hence, the method balances between adhering closely to the data and ensuring a physically plausible flow field.

In our effort to obtain accurate AMVs, we applied the TV-L1 dense optical flow algorithm to four distinct simulated NatureRun datasets. These datasets represented ETC, TC, and both the early and late stages of Hurricane Harvey. The analysis in Yanovsky et al. (2024) revealed that the optical flow method had a distinct advantage over the traditional feature matching technique.

For every pair of images, the optical flow algorithm generated Atmospheric Motion Vectors for every pixel. On the other hand, the feature matching algorithm had its limitations – it was unable to generate AMVs in specific areas, particularly near domain boundaries. Although the optical flow approach did not perfectly capture the strong winds around the hurricane with absolute precision, the flow fields it produced closely resembled the wind fields observed in the natural runs datasets. A notable metric, the Root Mean Square Vector Difference (RMSVD), indicated that the errors associated with the optical flow algorithm were significantly reduced compared to those obtained with the feature matching algorithm. This resulted in an average accuracy improvement of about 30% to 50% for the four datasets analyzed.

An important quality of the optical flow was its robustness. The results it produced remained relatively consistent, irrespective of the time interval. This was not the case with the feature matching method, whose results showed a significant change based on time intervals (Yanovsky et al., 2024). Given these advantages, we favor the optical flow as the preferred algorithm for retrieving AMVs in this OSSE exercise.

In Figure 1, we display the quiver plot of the wind vectors from optical flow (left column) and NatureRun (middle column). The time stamp is chosen as the middle of the model run for each study scenario. The differences between the two wind fields (optical flow and NatureRun) are displayed in the right column of Figure 1. We observe that the wind differences show a consistent pattern influenced by complex local factors, further complicated by what appears to be random variability and potential covariates such as wind rotation or water vapor gradients. Given our objective to model these error characteristics, we will approach the problem by first considering the space of predictor variables, often referred to as feature selection or variable selection.

3 Modelling Approach

3.1 Variable Selection

Before assessing the benefits of colocating passive and active wind data, we need focus on the issue of variable selection, which involves the identification of important variables or features for predicting the target quantity. In this context, our target is the bias between retrieved AMVs and actual wind values. This selection process holds significance due to its potential to trim down the input parameter space. This reduction not only speeds up the training process but also enhances the model's reliability when dealing with unfamiliar data. Additionally, it contributes to simplifying the interpretation of model parameters.

Lidar instruments typical observe only one component of winds. Aeolus, for instance, measures "[the] component of the wind vector along the instrument's line of sight [HLOS]" (Lux et al., 2020). Here, we will similarly assume that the active instrument in our OSSE study will also observe only one component of the wind vector, though we will simplify the geometry by assuming that simulated observable is the u-wind. This assumption is fairly benign, since it is simply a change of basis from the wind vector given by the HLOS wind and its (unobserved) perpendicular component to the much simpler (u,v) basis.

Since we wish to model the bias between the optical flow $\hat{u}$ and the lidar u, we define the response variable as $y = \hat{u} - u$. As for the predictor variables, Posselt et al. (2019) examined the relationship between 'tracked' and 'true' wind using an OSSE framework for the same ETC region as this study, and they noted that there is considerable heteroskedasticity (i.e., non-constant variance) in the windspeed difference (i.e., 'tracked' windspeed minus 'true' windspeed) as a function of water vapor, wind speed, water vapor gradient, and the angle between wind direction and water vapor gradient (Figure 6, Posselt et al., 2019). Therefore, we shall start our list of potential variables as these four parameters. Since wind speed is simply a magnitude of the wind vector $(\hat{u},\hat{v})$ in polar coordinates, we added the other component – wind angle – as well.

Further, we take advantage of the smooth output space of the optical flow algorithm to compute the first derivatives of $(\hat{u},\hat{v})$, giving rise to a four dimensional vector $(d\hat{u}/dx, d\hat{u}/dy, d\hat{v}/dx, d\hat{v}/dy)$. These first derivatives are computed via first-order finite differencing method. In theory, we could have also computed the second-order derivatives in this manner, but we opted otherwise due to numerical instabilities that can result from computing high-order derivatives using finite differencing. From these derivatives, we added the curl and divergence to the list of potential variables. There variables are meant to inform the model of the rotation and flux of the wind field at any particular location. Further, we also computed the angle of the gradient, which is defined as the angle made with the x-axis by a 2-dimensional vector $(d\hat{u}/dx, d\hat{u}/dy)$ and $(d\hat{v}/dx, d\hat{v}/dy)$, respectively.

To generate the data for assessing the variable importance, we start with the arrays of optical flow u-and and v-wind, along with the water vapor content. We apply the finite differencing method to water vapor, u-wind and v-wind to generate the first-order derivatives, which then provides all the precursors necessary to compute the rest of the augmented variables described above. At this point, each pixel in the domain can be represented by a 13-dimensional predictor vector (see Table 2 for a detailed description) and a scalar-valued response $y = \hat{u} - u$. We then converted this into a tabular format by randomly and uniformly sampling 1$\%$ of the available domain for each time step and append them into a training-validation dataset. These datasets, which are in tabular format, then form the basis of the following error characterization exercises.

The simulated lidar observations are created using the u-wind component of the WRF wind data (which serve as the 'truth'). To simulate lidar measurement errors, we added to the WRF u-wind data Gaussian zero-mean random errors that have pressure-dependent standard deviations: 2 m/s for 850 hPa, 3 m/s for 500 hPa, and 5 m/s for 300 hPa. These are rather conservative numbers since in practice data quality filtering on lidar wind data can typically reduce the magnitudes of the errors below what are assumed here. However, as a proof-of-concept, we chose to err on the side of having lidar measurement errors that are too large rather than too small.

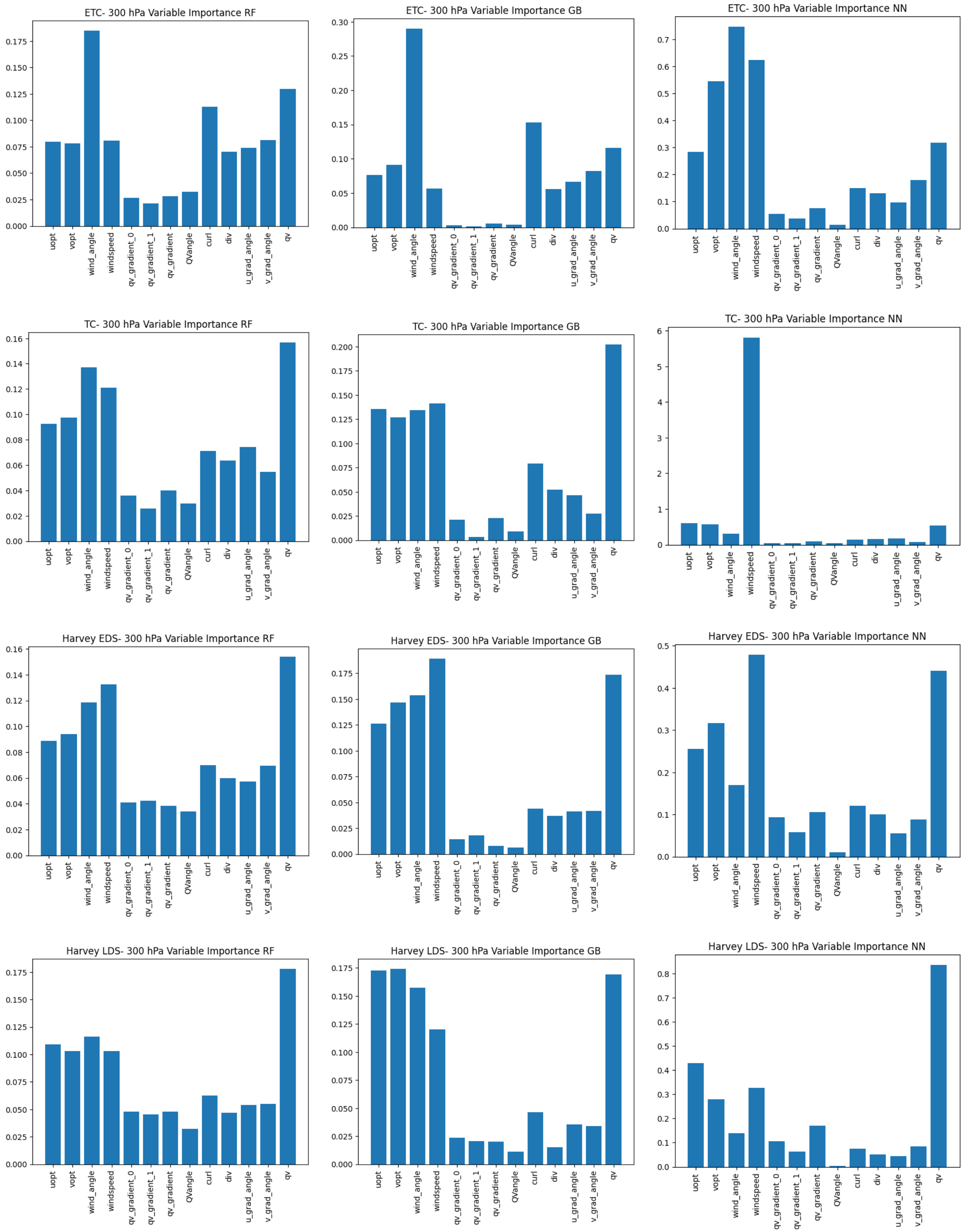

Figure 2. Variable importance plots for the 500 hPa pressure level at ETC, MCS, Harvey2318, and Harvey2406 using three different approaches: random forest (left column), gradient boosting (middle column), and permutation with neural network (right column). Higher values indicate more importance. Variable names along the x-axis are defined in Table 2.

Name	Description
uopt	u-wind from optical flow ($\hat{u}$)
vopt	v-wind from optical flow ($\hat{v}$)
wind_angle	angle of the vector $(\hat{u}, \hat{v})$ with respect to latitudinal lines
wind_speed	$\sqrt{(\hat{u}^2 + \hat{v}^2)}$
qv_gradient_1	dq/dx
qv_gradient_2	dq/dy
qv_gradient	$\sqrt{((dq/dx)^2 + (dq/dy)^2)}$
QVangle	angle between wind direction and water vapor gradient
curl	curl of $(\hat{u}, \hat{v})$
div	divergence of $(\hat{u}, \hat{v})$
u_grad_angle	arctan($d\hat{u}/dx, d\hat{u}/dy$)
v_grad_angle	arctan($d\hat{v}/dx, d\hat{v}/dy$)
qv	water vapor content (q)

Table 2. Names of variables used for variable importance analysis and their definitions.

Having constructed the datasets (i.e., the optical flow AMVs and the simulated lidar information), we consider the topic of variable importance. There is a large body of literature on the topic, particularly for regression-based methods. Examples include approaches such as genetic algorithms, jack-knifing, and forward selection (Bies et al., 2006; Lee et al., 2012; Blanchet et al., 2008). Here, due to the complexity of the functional model, we select our variable set using three different machine learning approaches that have been shown to be capable of modelling highly multivariate functional relationships: random forest (Breiman, 2001), gradient boosting regression trees (Prettenhofer and Louppe, 2014), and multi-layer perceptron (Gardner and Dorling, 1998). (Further details of parameter optimizations for these methods are discussed in Section 3.2 and Table 3).

Random forest and gradient boosting, in this case, employ decision trees (Kingsford and Salzberg, 2008), which is a popular and widely used machine learning algorithm that can be applied to both classification and regression tasks. Decision trees make predictions by mapping input features to output targets based on a series of binary decisions, and they form the basis of the two techniques considered in this section: random forest and gradient boosting trees. (For a more comprehensive overview of these machine learning methods, Chase et al. (2022) provides an excellent tutorial geared towards meteorologists). The metric for variable importance for these two methods are constructed by keeping track of the decrease in accuracy or increase in impurity (e.g., Gini impurity for classification, increase in node purity for regression) caused by a chosen specific feature (Breiman, 2001). These purity-based variable importances, where higher values indicate greater importance, are shown in the first and second columns of Fig. 2 for random forest and gradient boosting trees, respectively. The variables names on the x-axis are described in Table 2.

The results from the left and middle columns of Figure 2 indicate that the top five variables for regression are the retrieved optical flow winds ($\hat{u}$, $\hat{v}$), wind speed and angle, and water vapor (q). We note that wind speed and wind angle are the polar-coordinate transform of the rectangular coordinates ($\hat{u}$, $\hat{v}$), but their inclusion in the model significantly improves the model, since they provide an informative transformation that makes it easier for the machine learning model to model the functional form of interest.

One of the weaknesses of the purity-based variable importance plot is that high-correlation between features can inflate the importance of numerical features (Gregorutti et al., 2017; Nicodemus et al., 2010), and that the purity-based variable importance are based only on training data and can have low or no correlation with independent validation data. To address these short comings, we supplement them with another approach based on permutation, which could be applied any fitted estimator in tabular data contexts. The concept behind permutation feature importance involves quantifying the reduction in a model's score resulting from the random shuffling of a chosen variable (e.g., wind speed) while keeping all other variables the same within a fitted model (Breiman, 2001). The key insight is that if a model has significantly worse performance with a particular variable 'shuffled', then that variable must be important, and the degree of importance can be assessed by the magnitude of the performance degradation. One advantage of this technique is that it could be applied to non-linear or opaque estimators, and for this we choose to apply it in conjunction with a neural network, specifically a Multi-layer Perceptron regressor.

These permutation-based variable importance values are plotted in the right column of Figure 2. The variable importance that comes out of the permutation method has a different unit and scaling compared to purity-based variable importance but both are consistent in indicating that higher values signify greater importance. The overall patterns are the same between different approaches, indicating that for the most part, the most important variables are $\hat{u}$, $\hat{v}$), wind angle, wind speed, and water vapor content (q). Winds speed, however, is considered one of the most important predictors of bias according to the permutation method, and one additional feature that is considered somewhat important in this metric is the curl. Therefore, we shall consider the set of these six variables in the following analysis and modelling.

3.2 Algorithm comparisons

Having identified the important predictive variables, we consider the algorithms for fitting the bias functional form. The features we require of the algorithm are: able to handle complex multivariate data patterns, robust against new datasets, and computationally fast. For this reason, we have chosen four methods that are known to do well for high-dimensional problems with complex relationships: random forest, gradient boosting trees, multi-level perceptron, and nearest neighbor. Here, we touch briefly on an overview of the methods, before going into details of optimization and comparison. For readers who are not familiar with these machine learning approaches, we recommend the excellent tutorial series "A Machine Learning Tutorial for Operational Meteorology" (Chase et al., 2022, 2023).

Random Forest is a powerful ensemble learning technique used for both classification and regression tasks in machine learning. As the name suggests, it consists of an ensemble of multiple decision trees, combining these trees to create a more accurate and robust predictive model (Breiman, 2001). Random Forests are particularly popular due to their ability to handle

	Package Name	Parameter Settings
Random Forest	sklearn.ensemble.RandomForestRegressor	n_estimators ∈ (100, 300, 500), criterion ∈ (‘squared_error’, friedman_mse’), min_samples_split ∈ (2,5,10), max_features ∈ (‘sqrt’, ‘auto’)
Gradient Boosting Trees	sklearn.ensemble.GradientBoostingRegressor	learning_rate = (.01, .1, 1), n_estimators ∈ (100, 300, 500), max_depth ∈ (2, 4, 6), max_features ∈ (‘sqrt’, ‘log2’)
Multi-Layer Perceptron	sklearn.neural_network.MLPClassifier	hidden_layer_sizes ∈ ([30, 15], [20, 20], [30, 20], [20,10]), activation ∈ (‘tanh’, ‘logistic’, ‘relu’), learning_rate ∈ = (‘constant’, ‘invscaling’, ‘adaptive’), max_iter = (200, 500, 1000), learning_rate_init ∈ (1e-3, 1e-4, 1e-5)
Nearest Neighbor	sklearn.neighbors.NearestNeighbors	algorithm = ‘KDTree’, n_neighbors ∈ (3, 5, 10), p ∈ (1, 1.5, 2)

Table 3. Python package names (middle column) and the parameter settings for each of the method. Parameters not mentioned on this table are set as default in the python methods. Note that the arrays under Parameter Settings specify the option grid through which GridSearchCV is searching for the optimal choice.

complex data, reduce overfitting, and provide valuable insights into feature importance. Each tree is constructed using a random subset of the training data and a random subset of the input features. The predicted value, whether a class label or a regression value, is computed by passing the predictors to all the trees fitted within the model and aggregating the corresponding outcomes.

Gradient Boosting Trees is another powerful machine learning technique falling under the ensemble methods umbrella. Like random forests, Gradient Boosting constructs an ensemble of decision trees, with each referred to as a ’weak learner’ because it is relatively simple and typically underfits on its own. However, the trees are built sequentially, with each new tree aiming to correct the errors made by the previous ones. Similar to random forests, Gradient Boosting aims to build many trees and is widely used for both regression and classification tasks because of its capacity to create accurate predictive models capable of handling complex data patterns (Prettenhofer and Louppe, 2014).

The Multi-Layer Perceptron (MLP) is a foundational artificial neural network architecture that serves as the cornerstone for deep learning models. It is a versatile and powerful technique used for a wide range of machine learning tasks, including classification and regression. An MLP consists of interconnected layers of artificial neurons or nodes, roughly divided into three types: input layers, which typically represent the predictors; hidden layers, responsible for processing information from the previous layer and extracting relevant features; and the output layer, which produces the final result, such as a classification label or a regression value. Each neuron processes information and passes its output to the next layer, and the numeric parameters

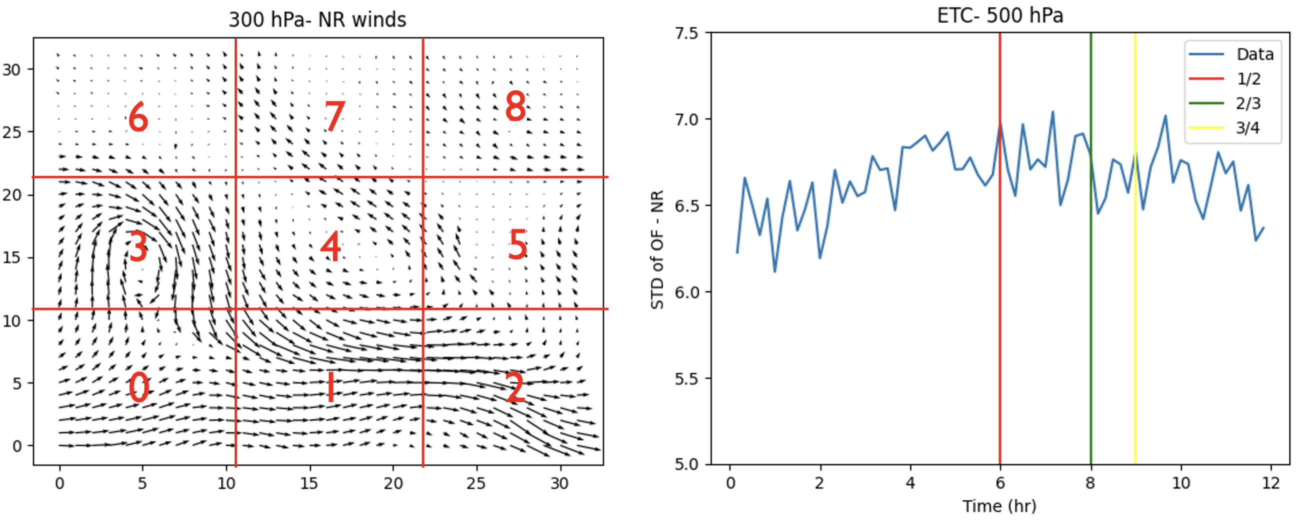

Figure 3. Left panel: spatial validation scheme where the domain is divided into a 3x3 grid and labelled from 0 to 8. Right panel: temporal validation scheme where the training data is set as the first 1/2, 2/3, and 3/4 of the full dataset, respectively.

within each node, namely the weight and bias values, are estimated from the data using backpropagation and gradient descent (Gardner and Dorling, 1998).

Nearest neighbor methods operate on the principle of identifying a fixed number of training samples that are closest in proximity to the new data point, and then predicting the label based on these identified neighbors. This number of samples can either be a user-defined constant, characteristic of k-nearest neighbor learning, or it can adapt based on the density of nearby points, as seen in radius-based neighbor learning. The measurement of distance can be achieved through various metric measures, with the standard Euclidean distance being the most commonly selected option.

We used implementations of these methods from the Python scikit-learn package (version 1.2) (Kramer and Kramer, 2016). All of these methods require tuning of algorithm parameters, such as tree leaf nodes and depth for random forest and gradient boosting, hidden layer sizes, and activation methods for the neural network, neighbor size, and distance metrics for nearest neighbors, etc. To optimize these parameters, we employed the Grid Search optimization method from the scikit-learn package (sklearn.model_selection.GridSearchCV). This method iterates through different parameter choices provided in the parameter grid and identifies the best combination of parameters that minimize the loss function, which in this case is the root mean-squared error when fitted against the training data. The parameter search space for these four methods is detailed in Table 3.

To evaluate the performance of the four different methods, we divided the tabular datasets created from the data arrays into training datasets (used for model building) and validation datasets (used for performance assessment). We employed two types of division - spatial and temporal - as illustrated in Figure 3. In the temporal division, we reserved the last 1/4, 1/3, and 1/2 of

	RandomF	GradientB	NeuralN	NearestN	Uncorrected Bias
ETC- 300 hPa	-0.521	-0.565	-0.276	-0.567	-3.062
ETC- 500 hPa	-0.091	-0.266	-0.244	-0.176	-2.116
ETC- 850 hPa	-0.027	-0.040	-0.210	-0.061	-0.437
TC- 300 hPa	-0.318	-0.296	-0.352	-0.340	0.714
TC- 500 hPa	0.067	0.083	0.303	0.125	0.452
TC- 850 hPa	-0.088	-0.096	-0.063	-0.079	0.081
Harvey EDS- 300 hPa	0.309	0.375	0.443	0.3988	-0.158
Harvey EDS- 500 hPa	-0.064	-0.050	-0.049	-0.063	-0.400
Harvey EDS- 850 hPa	0.105	0.059	0.076	0.092	0.165
Harvey LDS- 300 hPa	0.223	-0.061	-0.062	0.012	-0.777
Harvey LDS- 500 hPa	0.431	0.396	0.209	0.305	0.228
Harvey LDS- 850 hPa	-0.234	-0.136	-0.103	-0.169	-0.087

Table 4. Validation temporal bias (computed from withheld last half of data for each atmospheric regime) for Random Forest, Gradient Boosting, Neural Network, and Nearest Neighbor. Units are in m/s. Cells that are colored red indicate the best performing method, which is defined as having bias that is closest to zero. The uncorrected bias is defined as the bias of the raw optical flow data relative to the WRF data.

the data using timestamps, respectively, and utilized these withheld data to evaluate performance in terms of RMSE. For the ETC dataset, spanning 12 hours, this entailed setting aside the last 3, 4, and 6 hours of data for validation.

The results of this temporal validation for the ETC case are displayed in the right panel of Figure 4. In all pressure levels, the machine learning approach consistently exhibits smaller bias than the uncorrected optical flow data, where bias is defined as the expected value of the difference between u-wind from optical flow (both corrected and uncorrected) and the WRF simulated truth. Notably, for the 300 and 500 hPa levels, the bias magnitude is significant at 2.5-3 m/s, but it is reduced to less than 0.5 m/s, signifying a substantial improvement. Similarly, although the reduction in bias is smaller at 850 hPa due to the optical bias starting at a lower value, the same trend persists.

It is informative to assess the performance of all four algorithms across the four scenarios and three pressure levels. Therefore, we selected the case where 1/2 of the data was withheld (considered the most challenging case) and summarized the validation performance in terms of bias for all four methods in Table 4. The scenarios at different pressure levels are listed in the rows. Overall, random forest, gradient boosting, and MLP tend to exhibit comparable performance, with no clear preference among the three. Nearest neighbor, on the other hand, consistently reduces bias relative to the uncorrected optical flow but falls short of the performance achieved by the other three algorithms. This suggests that the proximity-based methodology may not be flexible enough to capture the complex dependence structure of the AMV biases. We note that there are two instances where the uncorrected optical flow has the smallest bias (Harvey EDS 300 hPa and Harvey LDS 850), but these cases share a common

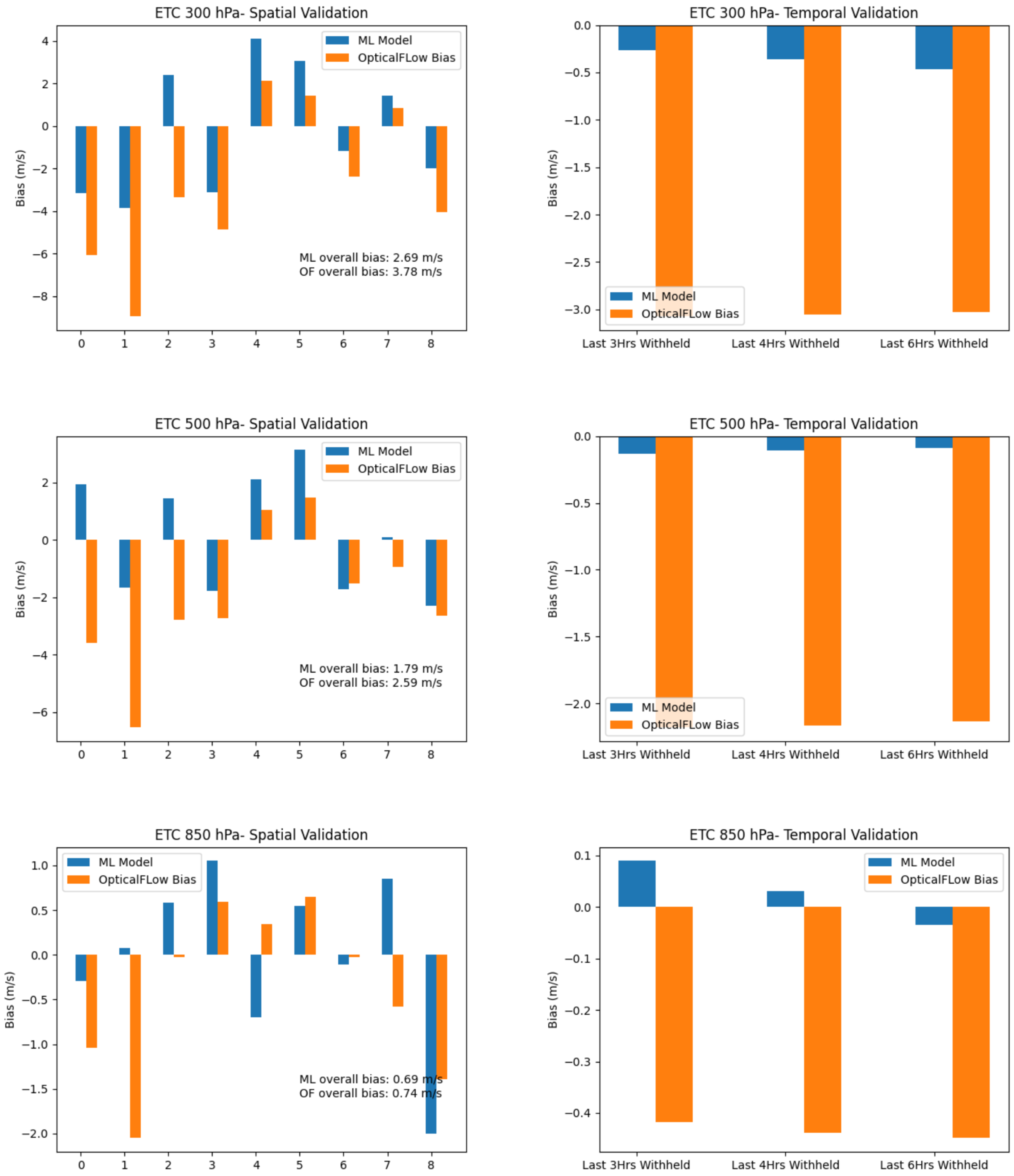

Figure 4. Sample plot of the random forest performance for ETC at three pressure levels for the spatial case (left column) and temporal case (right column).

	RandomF	GradientB	NeuralN	NearestN	Uncorrected Bias
ETC- 300 hPa	2.723	2.770	2.874	3.037	3.815
ETC- 500 hPa	1.745	1.761	1.790	2.024	2.595
ETC- 850 hPa	0.675	0.678	0.679	0.668	0.748
TC- 300 hPa	1.261	1.112	1.241	1.287	1.047
TC- 500 hPa	0.587	0.608	0.601	0.581	0.661
TC- 850 hPa	0.496	0.485	0.443	0.470	0.440
Harvey EDS- 300 hPa	0.469	0.371	0.453	0.454	0.432
Harvey EDS- 500 hPa	0.680	0.563	0.846	0.707	0.488
Harvey EDS- 850 hPa	0.444	0.336	0.389	0.509	0.401
Harvey LDS- 300 hPa	0.612	0.640	0.718	0.698	0.869
Harvey LDS- 500 hPa	0.214	0.334	0.343	0.397	0.372
Harvey LDS- 850 hPa	0.506	0.491	0.540	0.576	0.383

Table 5. Validation spatial bias (averaged over all 9 spatial regions) for Random Forest, Gradient Boosting, Neural Network, and Nearest Neighbor. Bias are computed as the mean of absolute bias within each of the spatial region. Cells that are colored red indicate the best performing method, which is defined as having bias that is closest to zero.

feature in which the original optical flow exhibits very low bias (<0.25 m/s). In such cases, the algorithms may struggle due to the limited discernable signal for modeling.

Another validation approach is a purely spatial one. In this approach, we divide the domain of each study region (as seen in Figure 1) into equal 3x3 areas and label each subregion with an index ranging from 0 to 8. In this labeling scheme, 0 represents the bottom-left cell, 2 the bottom-right cell, 4 the center cell, and 8 the top-right cell. We then withhold one of these nine regions at a time and train our models (e.g., random forest, gradient boosting, etc.) on the other eight cells. Subsequently, we apply our trained model to the withheld region. A sample of the results from these spatial validation efforts is shown in the left panel of Figure 4 for the ETC case, using the random forest algorithm. In this figure, the indices on the axes represent the region that was withheld from the training process. The overall biases, computed in the lower right of the panels, are mean absolute biases (MAB), which are defined as

$$\text{MAB(m)} = \sum_{i=1}^{9} \frac{|b_i(m)|}{9} \text{ where } b_i(m) = \sum_{j=1}^{N_i} \frac{(\hat{u}_{ij}^{m} - u_{ij})}{N_i}.$$

In this formula, m represents the methodology being evaluated (e.g., random forest, gradient boosting, optical flow, etc.), $b_i(m)$ denotes the normal bias when applying methodology m to the i-th withheld dataset, and N_i stands for the number of observations within the i-th validation dataset. The reason for calculating the mean absolute biases across these nine regions is to account for the possibility of biases having both positive and negative values. Therefore, we take the absolute value before averaging to prevent negative biases from canceling out positive biases and potentially distorting the resulting metrics.

The results from the ETC case in Figure 4 indicate that, for most of the regions, the trained model results in biases that are smaller in magnitude than those of the original optical flow u-wind. In some cases, the improvement in bias can be substantial (e.g., region 1). While in a few instances, the RF model can result in biases with increased magnitude, this adverse effect is generally small compared to the magnitude of gains observed in other regions. The MAB are displayed in the lower right corner of the left panels in Figure 4, and they suggest that random forest consistently reduces the magnitude of the bias compared to the optical flow data. Another observation is that the validation spatial biases in Table 5 tend to be bigger than the validation temporal biases in Table 4 (e.g., the typical MAB in the spatial case is around .5 m/s, while the typical bias in the temporal validation case is around .05 m/s). In both cases, the machine learning corrected values tend to be improved over the uncorrected optical flow data, indicating that the algorithm is able to capitalize on information within the training dataset for both the spatial and temporal case. However, their difference in performance in Table 4 and Table 5 indicate that functional relationship between the biases and the predictive variables in Section 3.1 may change depending on the spatial region, which makes sense intuitively since different regions of a storm system might exhibit different bias characteristics. However, this functional relationship, as demonstrated by Table 4, tends to be much more stable in terms of temporal evolution in the time scales that we examined (i.e., 3, 4, and 6 hours in advance), which allows the algorithms considered to be more accurate in predicting and correcting the biases.

In Table 5, we present the MAB of the four algorithms (as well as that of the uncorrected optical flow) across the four scenarios and pressure levels. We observe the same overall patterns as in the temporal validation shown in Table 4, noting that random forest, gradient boosting, and MLP tend to exhibit the best performance, although their dominance varies across different scenarios and pressure levels. As before, nearest neighbor does not produce notably superior results. In some cases, the uncorrected optical flow algorithm has the lowest error, but these tend to be cases where the bias initially starts off at a low level.

3.3 Uncertainty characterization of AMVs

In data assimilation, thinning high-density AMVs is often necessary. Typically, this process involves giving preference to vectors that exhibit higher accuracy. The selection procedure usually takes into consideration various indicators of error level associated with the vectors, such as the quality indicator (QI), expected error (EE), recursive filter flag (RFF), error flag (ERR), and others (Le Marshall et al., 2004). All of these metrics share the common goal of grouping AMVs with similar errors together, meaning that observations with good quality indicators should all exhibit low errors relative to the unobserved truth.

We note that pattern tracking and optical flow do not provide an intrinsic error estimate, necessitating the addition of the post-hoc error indicators mentioned above. In the existing remote sensing literature, a variant of random forest called quantile random forest has been extensively used to model uncertainty alongside the prediction of interest (e.g., digital soil mapping product, Vaysse and Lagacherie (2017); soil organic matter, Nikou and Tziachris (2022); nitrogen use efficiency, Liu et al. (2023)). In this section, we shall employ quantile forest regression to construct the prediction intervals for the u-wind and compare them with withheld validation u-wind data.

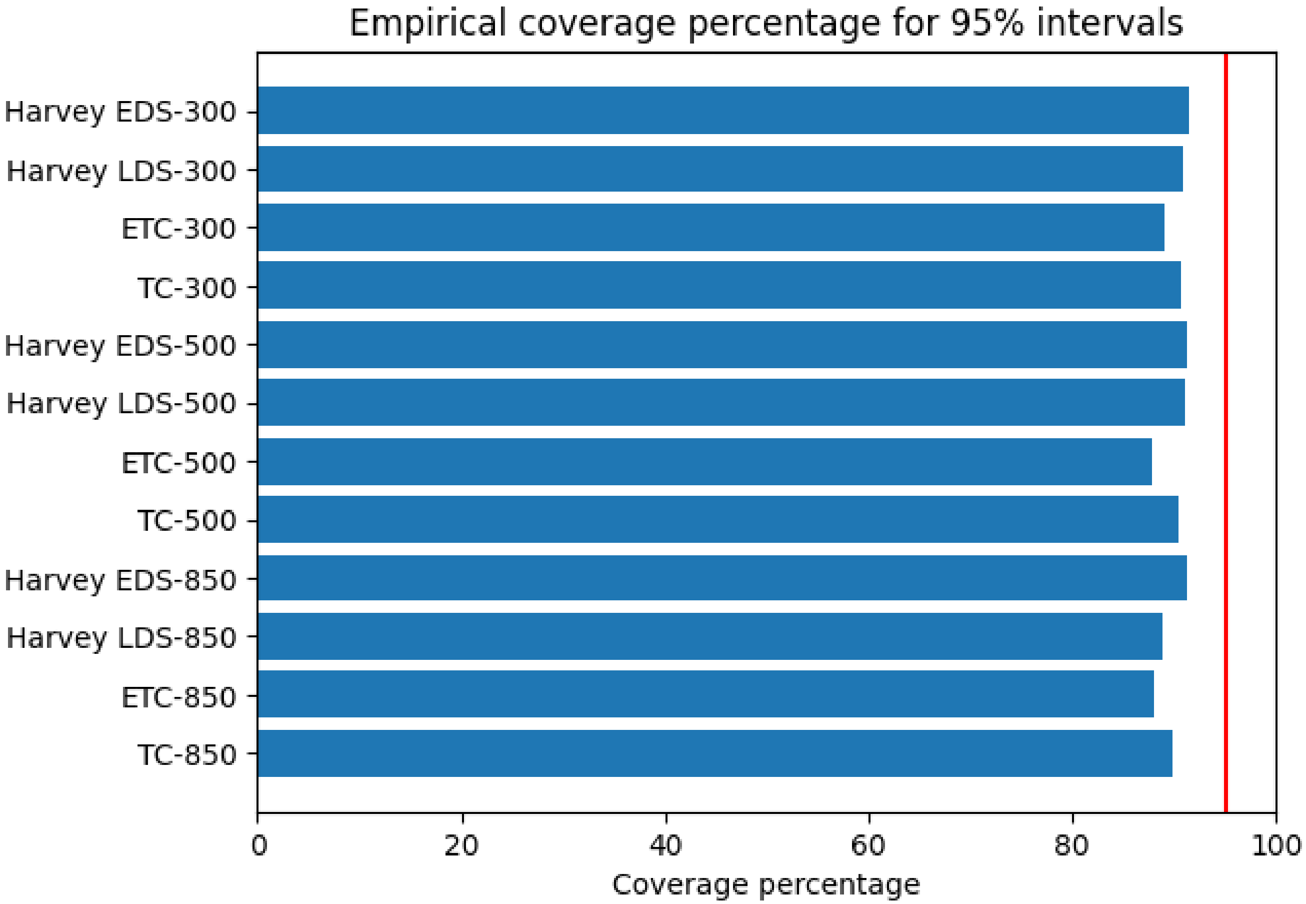

Figure 5. Actual coverage percentage of the 95% prediction intervals when evaluated against withheld validation data. The vertical red line is the ideal coverage percentage as implied by the 95% intervals.

Quantile random forest, as introduced by Meinshausen and Ridgeway (2006), is a modification to the random forest procedure that enables the estimation of prediction intervals for the intended variables. In contrast to normal random forests, which approximate the conditional mean of a response variable, quantile random forests (QRF) provide the full conditional distribution of the response variable to construct prediction intervals. (For readers who are not familiar with the random forest algorithm, Chase et al. (2022) provides an excellent meteorology-geared tutorial). The key insight that allows for this property is that, while random forests retain solely the mean of observations within each node and discard any additional information, quantile random forests preserve the values of all observations within the node, not just their mean, and use these distributions to make estimates of the quantiles of interest. In particular, Meinshausen and Ridgeway (2006) prove that the conditional quantile estimates are asymptotically consistent under specific assumptions:

1. The proportion of values in a leaf, relative to all values, vanishes as the number of observations n approaches infinity.

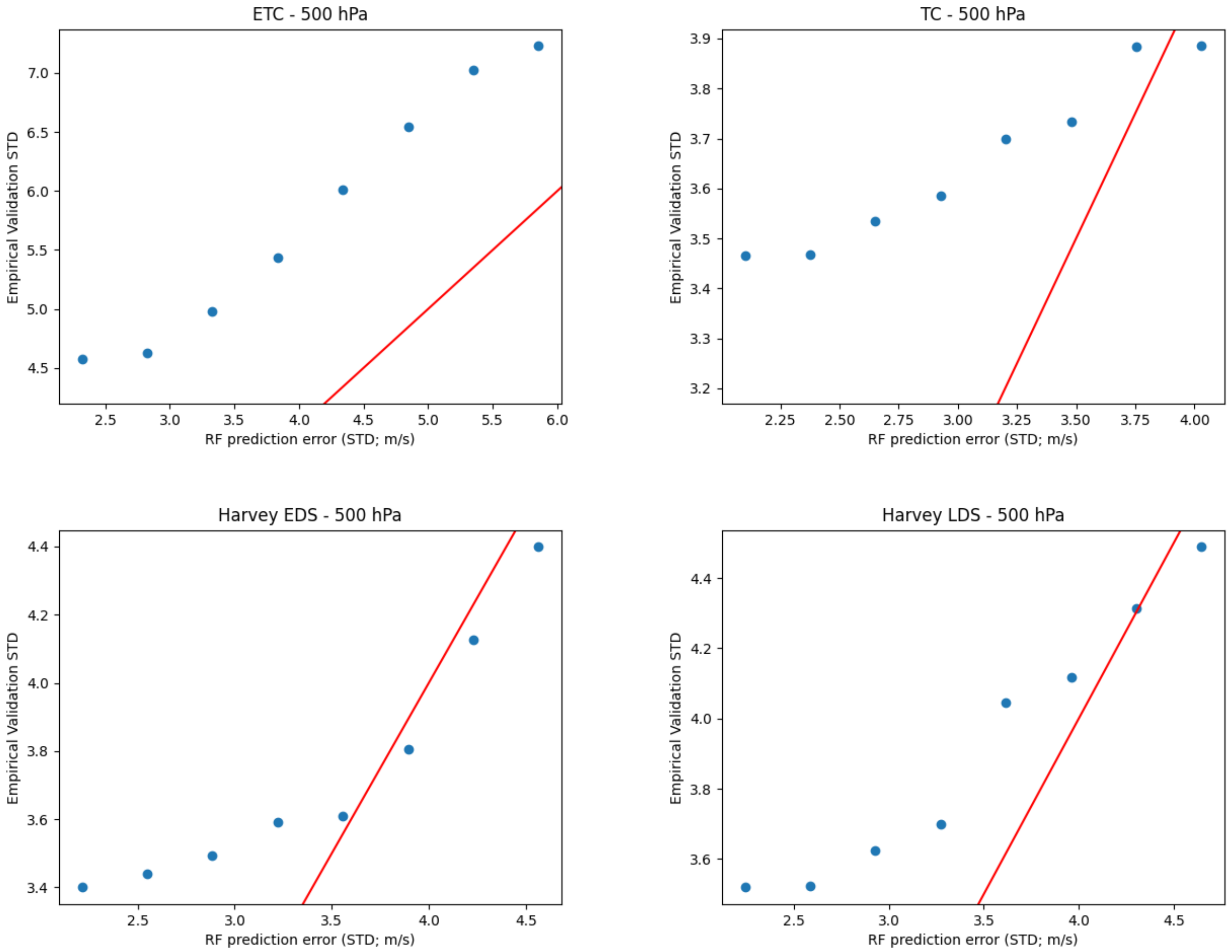

Figure 6. Plots of the estimated prediction error from random forest versus empirical validation error using quantile random forest for ETC (top left), TC (top right), Harvey EDS (bottom left), and Harvey LDS (bottom right). Red lines are the identity ($y = x$) lines.

2. The minimal number of values in a tree node grows as n approaches infinity.

3. When looking for features at a split, the probability of a feature being chosen is uniformly bounded from below.

4. There is a constant γ in the range $\gamma \in (0, 0.5)$ such that the number of values in a child node is always at least γ times the number of values in the parent node.

5. The conditional distribution function is Lipschitz continuous with positive density.

These are fairly modest assumptions, particularly with respect to the construction of the trees. However, it's worth noting that the quantiles under these assumptions is only asymptotic consistent as n approaches infinity. However, these assumptions provide some theoretical assurance that the outputs of quantile random forest should approximate the true conditional quantiles to some extent.

Here we use the python implementation of QRF provided by the Python package *quantile-forest*. We use quantile random forest to build 95% prediction intervals at the pixel level. That is, at any pixel, we compute the prediction interval as follows

$$I(\boldsymbol{x}) = [Q_{.025}(\boldsymbol{x}), Q_{.975}(\boldsymbol{x})]$$

where $Q_{.025}(\cdot)$ and $Q_{.975}(\cdot)$ are the 2.5-th percentile and 97.5-th percentile random forest estimators from *quantile-forest*, respectively, and $\boldsymbol{x}$ is the vector of predictors (e.g., optical flow winds, wind speed, angle, etc.) as discussed in Section 3.1.

We wish to assess the performance of the intervals $I(\boldsymbol{x})$ given by quantile random forest, and one approach is to compute the coverage probability of the confidence intervals when applied to withheld simulated lidar data. We repeat the exercises in the previous section, and for each of the four scenarios and three pressure levels, we use the first half of the storm for training, and the second half for validation. We then compute the coverage percentage of the prediction intervals, which is defined as the probability (expressed as percentage) that the true u-wind actually falls within the interval given by quantile random forest. That is, the coverage percentage (CP) for a given scenario and pressure level is given by

$$CP = \frac{\sum_{i=1}^{N} I\left(Q_{.025}(\boldsymbol{x}_i) \leq u_i \cap u_i \leq Q_{.975}(\boldsymbol{x}_i)\right) \cdot 100}{N}$$

where u_i is the i-th WRF wind from the withheld validation set, $I(\cdot)$ is the indicator function, and N is the size of the validation dataset. A comparison of the coverage probability for all scenarios and pressure levels is given in Figure 5. There, we see that the 95% prediction intervals from quantile random forest consistently *underestimate* the magnitude of the error variability, averaging between 85-90% coverage while the ideal number should have been 95%. This implies that the prediction interval widths given by quantile random forest in general tend to be a bit smaller than what the validation data require.

To get a clearer idea of the differences between the estimated prediction error and that of the validation data, we examine their relationship in a scatter plot. To do so, we first convert the prediction intervals $I(\boldsymbol{x})$ to their effective prediction error $\hat{\sigma}(\boldsymbol{x})$. This conversion relies on the fact that in a Gaussian distribution, the 95% confidence interval is given by a $+/-$ 2 standard deviation of the mean. Therefore, to compute the effective prediction error from our 95% confidence interval, we divide the interval width by 4. That is,

$$\hat{\sigma}(\boldsymbol{x}) = \frac{Q_{.975}(\boldsymbol{x}) - Q_{.025}(\boldsymbol{x})}{4}.$$

To compare these effective prediction error $\hat{\sigma}(\boldsymbol{x})$ against the withheld validation data, we construct the equivalent standard error using the *withheld validation data*. We use the binning approach, where the empirical validation error (EVE) for a given prediction error value σ and a given bin length d is computed by aggregating all observations where the QRF prediction error is within $+/-d$ of σ, and then we compute the RMSE on this subset. In formal terms, the EVE is given by:

$$EVE(\sigma, d) = \sqrt{\left(\sum_{i \in |\hat{\sigma}(\boldsymbol{x}_i) - \sigma| < d} \frac{(u_i^* - u_i)^2}{N} \right)} \qquad (2)$$

where u_i^* is the i-th RF-corrected u-wind in the validation data set, u_i is the i-th true WRF u-wind, and N is the size of the validation dataset.

With the formulas above, we binned the RF prediction errors into eight equally-spaced bins and computed the corresponding EVE. The results are shown for all scenarios at the 500 hPa pressure level in Figure 6. Although the overall patterns are similar at other pressure levels, this figure provides a more nuanced view, naming that while the RF prediction errors tend to under estimate the true errors, they exhibit a statistically significant *linearly increasing* relationship. This is a valuable property. It implies that while the errors are not accurate (i.e., statistically valid), their positive correlation with the true error indicates that we can use the QRF prediction error as a proxy for quality assessment.

To demonstrate the usefulness of this property, we simulate a quality indicator flag by dividing the validation data into three equal-size categories: low, mid, and high quality. The three categories are constructed by sorting the QRF prediction errors (or alternatively the prediction interval widths) from smallest-to-largest, and then classifying the smallest one-third as high-quality, the middle one-third as mid-quality, and the largest one-third as low-quality. We then compute the EVE of the optical flow wind versus the withheld 'true' wind within each bin, and we display the results in Table 6. Intuitive understanding of high-quality observations generally implies that they are more 'accurate' than low-quality observations, and indeed here Table 6 indicates that the EVE values, *for every region and every pressure level*, form an increasing pattern with high-quality observations having the smallest error, mid-quality having medium error, and low-quality having the largest error.

The differences between the high-quality and low-quality observations can be fairly small some some situations, particularly for the TC case at all pressure levels. We note that this is because of the design of the experiment. Recall that we are using withheld simulated lidar data that have added Gaussian zero-mean random errors with pressure-dependent standard deviations: 2 m/s for 850 hPa, 3 m/s for 500 hPa, and 5 m/s for 300 hPa. These random measurement errors are added to *all* quality categories, which then essentially 'dilute' the contribution of the variability coming from the bias signal. This explains why the differences between high, medium, and low-quality bins are larger when the measurement errors are relatively low.

We note that the experiment in Table 6 divided the data into three bins. In general, the results here should hold for different numbers of bins, although a high single-digit number might be unstable. A hint of this instability is seen in Figure 6, where we observe that for the top-right panel, the right-most bin has almost an identical EVE compared to the bin immediately preceding it. This may be due to the fact that there are low bin counts at the extreme edge of the domain. In general, increasing the bin count can reduce the bin counts, exacerbating these statistical artifacts. However, quality indicators in common usage tend to use a low single-digit number of bins, which works well here.

		High Quality (STD)	Mid Quality (STD)	Low Quality (STD)	Baseline STD
ETC	300 hPa	7.161	7.739	9.002	8.039
	500 hPa	4.660	5.282	6.868	5.694
	850 hPa	3.002	3.612	5.012	3.978
TC	300 hPa	5.658	5.747	5.995	5.804
	500 hPa	3.452	3.629	3.737	3.647
	850 hPa	2.502	2.604	2.751	2.623
Harvey EDS	300 hPa	5.462	5.507	5.740	5.575
	500 hPa	3.423	3.535	4.043	3.696
	850 hPa	2.469	2.554	3.060	2.722
Harvey LDS	300 hPa	5.788	5.882	6.262	5.972
	500 hPa	3.502	3.647	4.231	3.833
	850 hPa	2.678	2.823	3.476	3.038

Table 6. STD vs Quality Indicators based on RF prediction errors. Baseline STD is defined as the STD of the *entire* optical flow dataset against the WRF simulated truth. (i.e., std$(\hat{u} - u)$).

4 Conclusions

Accurately estimating global wind patterns is of paramount importance across scientific and practical domains, including applications like global chemical transport modeling and numerical weather prediction. Atmospheric Motion Vectors (AMVs) serve as crucial inputs for these applications. However, addressing errors in AMV retrievals becomes imperative before their assimilation into data assimilation systems, as these errors can significantly impact output accuracy. One noteworthy error characteristic of AMVs is bias, which varies considerably by region. These biases can lead to adverse results if the AMVs are incorporated into data assimilation systems without proper mitigation or bias removal.

In real-world applications, correcting the bias in AMV retrievals necessitates an independent benchmark or reference to establish accuracy. Independent data sources may include collocated radiosonde data or lidar AMV data, such as those available from Aeolus. In this paper, we present a proof-of-concept that demonstrates the feasibility and performance of a bias-correction scheme within an Observing System Simulation Experiment (OSSE) framework. Specifically, we examined three different storm systems in the Gulf of Mexico, North Atlantic Ocean, and Southeast Asia, and applied our bias correction and prediction error interval procedure to outputs generated by a novel AMV algorithm known as optical flow. Our results suggest that passive-sensor AMVs, which typically have high coverage but low precision, can benefit significantly from coincident high-precision active-sensor wind data. These benefits can be harnessed through algorithms that model expectations (bias reduction) or quantiles (uncertainty quantification).

In Section 3.2, we demonstrated that conventional machine learning algorithms such as random forest and gradient boosting can effectively learn the complex multivariate dependence structure of errors and correct biases in raw optical flow AMVs. It's worth noting that, despite having low bias in some cases, the standard deviation of the AMV error can be relatively large (e.g., with a standard deviation of 1-2 m/s while the error may be on the scale of <0.5 m/s). In these scenarios, the error-correction model produces biases of a similar magnitude (i.e., <0.5 m/s). Notably, we showed that, in the storm systems we considered, it is possible to estimate biases with minimal performance degradation up to six hours in advance.

One of the most valuable extensions of machine learning models in this bias-correction exercise is the ability to estimate prediction intervals. In Section 3.3, we employed the quantile random forest framework by Meinshausen and Ridgeway (2006) to generate prediction intervals for withheld validation data. We observed that, while the prediction intervals often tend to be too narrow (underestimating the variability of the true process), they generally exhibit a monotonically increasing relationship with the NatureRun wind variability. In other words, the uncertainty estimates from the quantile random forest are not statistically valid (i.e., the 95% confidence intervals may not capture the truth 95% of the time), but the algorithm does correctly rank the error magnitudes when analyzing multiple pixels. Therefore, while the prediction intervals may not be directly usable in data assimilation, they can serve as valuable components of a quality indicator. Indeed, in Section 3.3, we conducted an experiment where we categorized the optical flow retrievals into three groups — high, mid, and low quality. We demonstrated that the standard deviation within these categories relative to the validation data follows an increasing pattern, with high-quality observations having the lowest error standard deviation, mid-quality observations falling in the middle range, and low-quality observations displaying the highest error standard deviation.

These results are highly promising, particularly regarding the application of quantile random forest in quality indicators. Our future research plans involve extending this study to other global regions and various convective systems. It's worth noting that different applications or study regions may necessitate distinct sets of predictive variables, and the selection of variables employed in our feature selection process may not be universally applicable. Nevertheless, the same variable selection process presented in Section 3.1 can be adapted to determine the most relevant predictive variables. In this paper, we utilized quantile random forest for prediction interval estimation, but theoretically, other machine learning algorithms could be employed to generate quantiles (e.g., quantile neural networks, etc.), although the computational requirements may vary.

Code and data availability. The data that support the findings of this study are not publicly available due to their large volume. The data and codes are available from the corresponding author on request .

Author contributions. H. Nguyen designed the experiments, performed the computations, and wrote the paper. L. Wu generated the WRF NatureRun data, and I. Yanovsky performed optical flow retrievals. D. Posselt and S. Hristova-Veleva contributed to the design of the methodology and the validation exercise. All authors participated in providing feedback on the design and findings of the paper.

Competing interests. The authors declare that they have no conflict of interest.

References

Bies, R. R., Muldoon, M. F., Pollock, B. G., Manuck, S., Smith, G., and Sale, M. E.: A genetic algorithm-based, hybrid machine learning approach to model selection, Journal of pharmacokinetics and pharmacodynamics, 33, 195, 2006.

Blanchet, F. G., Legendre, P., and Borcard, D.: Forward selection of explanatory variables, Ecology, 89, 2623–2632, 2008.

Bormann, N. and Thépaut, J.-N.: Impact of MODIS polar winds in ECMWF's 4DVAR data assimilation system, Monthly weather review, 132, 929–940, 2004.

Breiman, L.: Random forests, Machine learning, 45, 5–32, 2001.

Chase, R. J., Harrison, D. R., Burke, A., Lackmann, G. M., and McGovern, A.: A machine learning tutorial for operational meteorology. Part I: Traditional machine learning, Weather and Forecasting, 37, 1509–1529, 2022.

Chase, R. J., Harrison, D. R., Lackmann, G. M., and McGovern, A.: A Machine Learning Tutorial for Operational Meteorology, Part II: Neural Networks and Deep Learning, Weather and Forecasting, 2023.

Cordoba, M., Dance, S. L., Kelly, G., Nichols, N. K., and Waller, J. A.: Diagnosing atmospheric motion vector observation errors for an operational high-resolution data assimilation system, Quarterly Journal of the Royal Meteorological Society, 143, 333–341, 2017.

Crespo, J. A. and Posselt, D. J.: A-Train-based case study of stratiform–convective transition within a warm conveyor belt, Monthly Weather Review, 144, 2069–2084, 2016.

Gardner, M. W. and Dorling, S.: Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences, Atmospheric environment, 32, 2627–2636, 1998.

Gelaro, R., Langland, R. H., Pellerin, S., and Todling, R.: The THORPEX observation impact intercomparison experiment, Monthly Weather Review, 138, 4009–4025, 2010.

Gregorutti, B., Michel, B., and Saint-Pierre, P.: Correlation and variable importance in random forests, Statistics and Computing, 27, 659–678, 2017.

Horn, B. K. and Schunck, B. G.: Determining optical flow, Artificial Intelligence, 17, 185–203, https://doi.org/https://doi.org/10.1016/0004-3702(81)90024-2, 1981.

Kawa, S., Erickson III, D., Pawson, S., and Zhu, Z.: Global CO2 transport simulations using meteorological data from the NASA data assimilation system, Journal of Geophysical Research: Atmospheres, 109, 2004.

Kingsford, C. and Salzberg, S. L.: What are decision trees?, Nature biotechnology, 26, 1011–1013, 2008.

Kramer, O. and Kramer, O.: Scikit-learn, Machine learning for evolution strategies, pp. 45–53, 2016.

Le Marshall, J., Rea, A., Leslie, L., Seecamp, R., and Dunn, M.: Error characterisation of atmospheric motion vectors, Australian Meteorological Magazine, 53, 2004.

Lee, H., Babu, G. J., and Rao, C. R.: A jackknife type approach to statistical model selection, Journal of statistical planning and inference, 142, 301–311, 2012.

Liu, Y., Heuvelink, G. B., Bai, Z., and He, P.: Uncertainty quantification of nitrogen use efficiency prediction in China using Monte Carlo simulation and quantile regression forests, Computers and Electronics in Agriculture, 204, 107 533, 2023.

Lux, O., Lemmerz, C., Weiler, F., Marksteiner, U., Witschas, B., Rahm, S., Geiß, A., and Reitebuch, O.: Intercomparison of wind observations from the European Space Agency's Aeolus satellite mission and the ALADIN Airborne Demonstrator, Atmospheric Measurement Techniques, 13, 2075–2097, 2020.

Meinshausen, N. and Ridgeway, G.: Quantile regression forests., Journal of machine learning research, 7, 2006.

Nguyen, H., Cressie, N., and Hobbs, J.: Sensitivity of optimal estimation satellite retrievals to misspecification of the prior mean and covariance, with application to OCO-2 retrievals, Remote Sensing, 11, 2770, 2019.

Nicodemus, K. K., Malley, J. D., Strobl, C., and Ziegler, A.: The behaviour of random forest permutation-based variable importance measures under predictor correlation, BMC bioinformatics, 11, 1–13, 2010.

Nikou, M. and Tziachris, P.: Prediction and uncertainty capabilities of quantile regression forests in estimating spatial distribution of soil organic matter, ISPRS International Journal of Geo-Information, 11, 130, 2022.

Posselt, D. J., Stephens, G. L., and Miller, M.: CloudSat: Adding a new dimension to a classical view of extratropical cyclones, Bulletin of the American Meteorological Society, 89, 599–610, 2008.

Posselt, D. J., Wu, L., Mueller, K., Huang, L., Irion, F. W., Brown, S., Su, H., Santek, D., and Velden, C. S.: Quantitative assessment of state-dependent atmospheric motion vector uncertainties, Journal of Applied Meteorology and Climatology, 58, 2479–2495, 2019.

Posselt, D. J., Wu, L., Schreier, M., Roman, J., Minamide, M., and Lambrigtsen, B.: Assessing the forecast impact of a geostationary microwave sounder using regional and global OSSEs, Monthly Weather Review, 150, 625–645, 2022.

Prettenhofer, P. and Louppe, G.: Gradient boosted regression trees in scikit-learn, in: PyData 2014, 2014.

Salonen, K., Cotton, J., Bormann, N., and Forsythe, M.: Characterizing AMV height-assignment error by comparing best-fit pressure statistics from the Met Office and ECMWF data assimilation systems, Journal of Applied Meteorology and Climatology, 54, 225–242, 2015.

Staffell, I. and Pfenninger, S.: Using bias-corrected reanalysis to simulate current and future wind power output, Energy, 114, 1224–1239, 2016.

Swail, V. R. and Cox, A. T.: On the use of NCEP–NCAR reanalysis surface marine wind fields for a long-term North Atlantic wave hindcast, Journal of Atmospheric and oceanic technology, 17, 532–545, 2000.

Teixeira, J. V., Nguyen, H., Posselt, D. J., Su, H., and Wu, L.: Using machine learning to model uncertainty for water vapor atmospheric motion vectors, Atmospheric Measurement Techniques, 14, 1941–1957, 2021.

Vaysse, K. and Lagacherie, P.: Using quantile regression forest to estimate uncertainty of digital soil mapping products, Geoderma, 291, 55–64, 2017.

Velden, C. S. and Bedka, K. M.: Identifying the uncertainty in determining satellite-derived atmospheric motion vector height attribution, Journal of Applied Meteorology and Climatology, 48, 450–463, 2009.

Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D.: An Improved Algorithm for TV-L1 Optical Flow, in: Statistical and Geometrical Approaches to Visual Motion Analysis, edited by Cremers, D., Rosenhahn, B., Yuille, A. L., and Schmidt, F. R., pp. 23–45, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

Yanovsky, I., Posselt, D., Wu, L., and Hristova-Veleva, S.: Quantifying Uncertainty in Atmospheric Winds Retrieved from Optical Flow: Dependence on Weather Regime, submitted for publication to *Journal of Applied Meteorology and Climatology*, 2024.

Zach, C., Pock, T., and Bischof, H.: A Duality Based Approach for Realtime TV-L1 Optical Flow, in: Pattern Recognition, edited by Hamprecht, F. A., Schnörr, C., and Jähne, B., pp. 214–223, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

Zeng, X., Ackerman, S., Ferraro, R. D., Lee, T. J., Murray, J. J., Pawson, S., Reynolds, C., and Teixeira, J.: Challenges and opportunities in NASA weather research, Bulletin of the American Meteorological Society, 97, ES137–ES140, 2016.