Two-Stage Artificial Intelligence Algorithm for Calculating Moisture-Tracking Atmospheric Motion Vectors

AMIR OUYED,[a] XUBIN ZENG,[a] LONGTAO WU,[b] DEREK POSSELT,[b] AND HUI SU[b]

[a] *Department of Hydrology and Atmospheric Sciences, The University of Arizona, Tucson, Arizona*
[b] *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California*

(Manuscript received 15 April 2021, in final form 20 October 2021)

ABSTRACT: Much of the errors of atmospheric motion vectors (AMV) may be a consequence of algorithms not incorporating dynamical information. A physics-informed, artificial intelligence algorithm was developed that corrects errors of moisture tracking AMV (from the movement of water vapor) using numerical weather prediction (NWP) fields. The University of Arizona (UA) algorithm uses a variational method as a first step (fsUA); the second step then filters the first-stage AMVs using a random forest model that learns the error correction from NWP fields. The UA algorithm is compared with a traditional image feature tracking algorithm (JPL) using a global nature run as the "ground truth." Experiments use global all-sky humidity fields at 500 and 850 hPa for 1–3 January 2006 and 1–3 July 2006. UA outputs AMVs with root-mean-square vector differences (RMSVDs) of 2 m s^{-1} for the tropics and ~2–3 m s^{-1} for midlatitudes and the poles, whereas JPL outputs much higher RMSVDs of ~3 m s^{-1} for the tropics and ~3–9 m s^{-1} for the midlatitudes and poles. Although the algorithm fsUA produces approximately the same global RMSVDs as the JPL algorithm, fsUA has a higher resolution since it outputs an AMV per pixel, whereas the JPL algorithm uses a target box that effectively smooths the vectors. Furthermore, UA's RMSVDs are lower than the intrinsic error (calculated from the differences between two reanalysis datasets). Even for error-prone regions with low moisture gradients and where winds are oriented along moisture isolines, UA's absolute speed difference with "truth" stays within ~3 m s^{-1}.

KEYWORDS: Cloud tracking/cloud motion winds; Artificial intelligence; Machine learning

1. Introduction

Horizontal wind measurements are widely recognized to be essential for numerical weather prediction (NWP) and a variety of other applications. Much of these observations come from radiosondes that are released from weather stations at most twice per day with limited spatial coverage (e.g., missing large areas such as over oceans; Baker et al. 2014). More recently, the Aeolus satellite has returned measurements of wind velocity by firing a laser pulse toward the atmosphere and detecting the Doppler shift of backscattered radiation (Reitebuch et al. 2009). However, only the wind in the line of sight of the satellite instrument is measured, measurements must be averaged over a horizontal length scale of 10–90 km along-track (Šavli et al. 2019), and the data have a relatively low temporal resolution of twice per day crossings at the equator.

Horizontal wind can also be estimated from feature tracking of remotely sensed cloud top or moisture fields across a specific time interval. These feature-tracking wind products, usually called atmospheric motion vectors (AMVs) are available from various meteorological organizations around the world (Santek et al. 2019). The AMV products can roughly be divided by the type of satellite (or instrument) used to retrieve them: geostationary (GEO) satellites, low-Earth-orbiting (LEO) satellites, mixed, and stereoscopic (Cotton et al. 2020). All these types have their strengths and weaknesses.

Most AMVs come from GEO satellites, such as the series from the Geostationary Operational Environmental Satellite (GOES), Meteosat, and Himawari (Cotton et al. 2020). GOES-R's Advanced Baseline Imager (ABI) instrument only has 16 bands and only a few that are relevant to water vapor. They have high horizontal resolution and frequent temporal coverage (Schmit et al. 2017), but the imaging instruments in these satellites tend to have few bands. This means that the AMVs produced are very sparse vertically (Santek et al. 2019; Lean and Bormann 2018), either located at a single level (cloud top, in the case of cloud-tracked winds) or at most at very few vertical levels in moisture tracking retrieval. Finally, GEO satellites have no coverage in the high latitudes (poleward of ± 60°), due to unfavorable geometry of Earth's disc (Lazzara et al. 2014).

In contrast to GEO satellites, LEO satellites can produce AMVs in the higher latitudes due to the instrument swath overlaps (Martins et al. 2019). For instance, the Moderate Resolution Imaging Spectroradiometer (MODIS), which is aboard the *Terra* and *Aqua* satellites, can only view the same location 4 times per day at 50° latitude, whereas at the poles it views the same location 14 times per day (Key et al. 2003). Other LEO satellites that produce AMVs include the MetOp series, *NOAA-15/18/19/20* series, and *Suomi-NPP* (Cotton et al. 2020).

The previous techniques for capturing cloud-tracking AMVs rely on error-prone indirect means to calculate AMV

Supplemental information related to this paper is available at the Journals Online website: https://doi.org/10.1175/JAMC-D-21-0070.1.

Corresponding author: Amir Ouyed, amirouyed@email.arizona.edu

DOI: 10.1175/JAMC-D-21-0070.1

height, which make them the largest source of retrieval uncertainties. In contrast, the Multiangle Imaging Spectroradiometer (MISR) on *Terra* is a stereoscopic instrument that can calculate AMVs with a more accurate height assignment through a geometrical algorithm that aggregates snapshots at various angles (Carr et al. 2018). However, MISR only has four spectral bands and suffers the same temporal resolution problems as the other LEO instruments.

Finally, GEO and LEO satellites cannot usually produce AMVs in the region between 50° and 70° latitudes, which contains the polar jet streams. Mixed AMV techniques can be used for this region: using two LEO satellites orbiting slightly out of phase such as in Dual-MetOp (Hautecoeur et al. 2014), or compositing LEO and GEO imaging to calculate AMVs (LEO-GEO) (Lazzara et al. 2014). The Dual-MetOp configuration can also be used to generate global AMV fields.

However, these AMV products share a basic problem: their computation only takes information from cloud and moisture images and the assumptions of the tracking algorithm. Beyond the noise inherent to the measuring instruments, there are sources of missing information related to physical and dynamical processes. For example, a limiting assumption is that the features tracked are passive and do not experience any distortion/destruction. The actual features can be distorted by processes that retrieval algorithms do not take into account. Clouds can grow, disappear, and split, and also clouds can remain immobile in standing waves even if there are nonzero wind speeds (Carr et al. 2018). Finally, the algorithms cannot capture motion even in nonzero wind speeds when processing low contrast images, such as in the case of water vapor fields with low gradients or low moisture, or when winds blow along water vapor isolines (Posselt et al. 2019). No matter how sophisticated the feature tracking algorithms are, they cannot extract information beyond what is contained in the imaging bands and the algorithm's assumptions, and therefore the algorithm will miss important features, creating errors in the AMVs. All these factors lead to cloud-tracking AMVs with errors as large as 5.9–9.0 m s^{-1} (Santek et al. 2019).

A related approach to deriving wind fields is through the assimilation of satellite radiances through four-dimensional variational data assimilation (4D-Var). In this case, the forecasting model itself constrains the wind fields based on the calculated horizontal displacement of the assimilated mass fields. Studies have shown that the assimilation of infrared, clear-sky water vapor radiances can aid the region between 850 and 500 hPa, since there are not as many clouds to be tracked at those levels. However, radiances require thinning and averaging, leading to an effective resolution of about 125 km (Forsythe et al. 2014). Furthermore, 4D-Var calculates winds from horizontal mass advection, whereas water vapor changes are due to vertical displacement as well, especially in dynamically interesting zones, such as in convective systems (Stoffelen et al. 2020). Another interesting application of 4D-Var assimilation is from microwave humidity profiles, since they are "all sky," meaning that microwave sounders can measure humidity profiles below clouds. Studies have shown that all-sky assimilation of microwave-derived humidity profiles from the LEOs *MetOp-A/B* and *NOAA-18/19* can add as much as 1.5 h in forecast skill (Geer et al. 2014). We should expect in the future that all-sky humidity profiles derived from microwave sounders deployed in GEOs would add even more skill through assimilation, due to the enhanced temporal and global coverage (Lambrigtsen et al. 2018).

Recently, Posselt et al. (2019) argued that, for the case of moisture-tracked AMVs, the error of the tracking algorithm is a function of the underlying local state variables, such as moisture, wind speed, and wind direction. Therefore, Posselt et al. imply that for certain regions in observed space, such as in regions of low moisture, or where the wind vector is perpendicular to the moisture gradient, the AMV algorithm lacks sufficient information to generate accurate wind speeds. Regardless of the sophistication of the algorithm, ultimately tracking cannot produce more information than what is already given by the images and the algorithm's assumptions. Furthermore, these errors caused by missing information of the underlying dynamics lead to the rejection of many AMVs through quality control before data assimilation. For example, sometimes more than 50% of cloud-tracking AMVs can be rejected by operational algorithms (Santek et al. 2019). These are rejected data that could otherwise be assimilated or used.

The question is then, how can we substantially improve the retrieval of AMVs (e.g., as shown in Posselt et al. 2019), including over regions of low moisture or where the wind vector is perpendicular to the moisture gradient? The goal of this study is to do so by incorporating dynamical information into the moisture-tracking algorithm in a two-stage process. In the first stage, we use a feature-tracking algorithm that tracks the motion of moisture based on a variational method that calculates an AMV per pixel. In the second stage, a random forest (RF) model learns from NWP short-term forecast fields and conducts error corrections for the first stage AMVs. The advantage of this approach is that since the NWP model is based on the primitive equations of the atmosphere, the algorithm can extract dynamical information from the NWP model and use them to correct the "dynamics blind" algorithm of the first step. We will refer to this two-stage algorithm as the University of Arizona (UA) algorithm.

To directly compare the UA algorithm with the Jet Propulsion Laboratory (JPL) algorithm as used in Posselt et al. (2019) (as a representative of traditional image feature tracking methods used in prior studies), we use the same moisture field for our retrieval and the same "ground truth" for validation. Both are from the Goddard Earth Observing System Model, version 5 nature run (G5NR; Putman et al. 2014), rather than from real observations. Just as with Posselt et al. (2019), our use of G5NR moisture fields will not be specific to cloudy or clear skies, and therefore cloudy pixels will not be masked. This is because microwave sounders (Lambrigtsen et al. 2018) can potentially retrieve moisture fields below clouds, even if infrared sounders can only deal with clear-sky pixels. Furthermore, hyperspectral infrared sounding can be combined with microwave sounding data for all-sky moisture retrieval.

Another motivation for this work is the recognition that hyperspectral infrared instruments can retrieve high-vertical-resolution water vapor fields (Pagano et al. 2016).

Hyperspectral moisture fields would also avoid the height assignment problem experienced with the low vertical resolution of GOES's ABI imager and other similar instruments. Furthermore, 3D AMVs derived from hyperspectral moisture fields could capture many AMVs in the midtroposphere, in contrast to the paucity of cloud tracking AMVs in this layer. The UA algorithm developed here is intended for the 3D AMVs that in the future will be obtained using a geostationary hyperspectral infrared sounder, a constellation of infrared sounders (e.g., via cubesats; McCarty et al. 2021), or regions with overlapped swaths from two satellites within a time interval [e.g., via the hyperspectral infrared instrument CrIS (Cross-track Infrared Sounder) on the *JPSS-1* and *Suomi-NNP* satellites] (Zou et al. 2020). In this study, we use simulated pressure-level fields, mimicking the pressure-level retrievals from hyperspectral instruments, to circumvent the issues inherent in simulating broad sensitivity of *GOES-16* channels.

The rest of the paper is divided in the following way: section 2 describes the two-stage algorithm of UA, the more traditional JPL algorithm, and the statistical methods used for the comparison. Section 3 explains the behavior of the error distribution for the various AMV algorithms, and their robustness to error-prone regions, such as points where advection is parallel to water vapor isolines. Section 4 discusses how UA avoids common problems encountered in operational constraints and naive machine learning (ML) models, including overfitting. Finally, in section 5 we conclude and describe the significance of the UA algorithm in the operational and scientific context.

2. Methods

a. The UA algorithm

The University of Arizona algorithm is based on two stages. For water vapor fields $\mathbf{q}$ at given pressure level (e.g., 850 hPa), at times $t - \Delta t$, t, and $t + \Delta t$, the first stage consists of a variational method (Brox et al. 2004; Weinzaepfel et al. 2013) that estimates the atmospheric motion vector field $\mathbf{V}^*$ at time t, as

$$\mathbf{V}^*(t) = f_{\text{variational}}\left[\mathbf{q}(t-\Delta t), \mathbf{q}(t), \mathbf{q}(t+\Delta t)\right], \tag{1}$$

where Δt is the time-step size. We refer to this stage as first stage UA or fsUA. The second stage uses an RF model (James et al. 2013) and NWP short-term forecast fields:

$$\mathbf{V}(t) = f_{\text{RF}}\left[\mathbf{V}^*(t), \mathbf{V}_{\text{NWP}}(t)\right], \tag{2}$$

where $\mathbf{V}_{\text{NWP}}(t)$ is the horizontal wind field from NWP, and $\mathbf{V}(t)$ is the corrected AMV field. Fields here denote matrices that have a value for each spatial grid point. For example $\mathbf{V}(t) = \mathbf{V}(\mathbf{x}, t)$, where $\mathbf{x}$ are the spatial coordinates of the vectors. Below, we will describe the different parts of Eqs. (1) and (2), including the variational algorithm, the RF model, and the approach used to approximate the error correction of first stage AMVs.

The variational method in Eq. (1) calculates an approximation of $\mathbf{V}(t + \Delta t/2)$ based on a hypothetical displacement from $\mathbf{q}(\mathbf{x}, t)$ to $\mathbf{q}(\mathbf{x} + \Delta t\mathbf{V}, t + \Delta t)$. This variational method is based on the minimization of a function E that penalizes deviations from gradient constancy and moisture constancy across time steps. A sample function that fits the above conditions is

$$E = \int \psi(g, m)dA, \tag{3}$$

$$m = \mathbf{q}\left[\mathbf{x} + \Delta t\mathbf{V}^*(t+\Delta t/2), t+\Delta t\right] - \mathbf{q}(\mathbf{x}, t), \tag{4}$$

$$g = \nabla\mathbf{q}\left[\mathbf{x} + \Delta t\mathbf{V}^*(t+\Delta t/2), t+\Delta t\right] - \nabla\mathbf{q}(\mathbf{x}, t), \tag{5}$$

where ψ is some cost *function, A is the area of the computational domain, g is the constant moisture gradient penalty term, m is the constant moisture penalty term, and $\mathbf{V}^*$ is the horizontal velocity field that minimizes E. The parameters g and m penalize against a change of density in a Lagrangian parcel of water vapor, enforcing mass conservation. We use the "deepflow" implementation of OpenCV (Weinzaepfel et al. 2013; Itseez 2015) which solves a closely related equation to Eq. (3). The complete equation has also a smoothness term that penalizes large gradients in velocity. This implementation also has a multiscale approach, where the minimization of E is first approximated at a coarser resolution, and then subsequently refined at a higher resolution.

To increase the accuracy of $\mathbf{V}^*(t + \Delta t/2)$, we iterate the algorithm of Eq. (3) two times. After solving Eq. (3) one time, we warp $\mathbf{q}(t)$ with the computed velocity field from Eq. (3), leading to a shifted water vapor field $\mathbf{q}^*(t)$. We then solve Eq. (3) again with $\mathbf{q}^*(t)$, generating another velocity field. We then finally add the velocity fields from both steps, generating a more accurate $\mathbf{V}^*(t + \Delta t/2)$. While solving Eq. (3), we interpolate the missing data from the water vapor fields with an image processing algorithm inspired by the Navier–Stokes equations (Bertalmio et al. 2001).

Finally, we calculate the first stage velocity by averaging

$$\mathbf{V}^*(t) = \frac{\mathbf{V}^*(t+\Delta t/2) + \mathbf{V}^*(t-\Delta t/2)}{2}. \tag{6}$$

We will refer to this stage as fsUA.

We will compare the UA method with a feature tracking method provided by JPL (Posselt et al. 2019), based on a sum of absolute differences (SAD) algorithm, where motion vectors are calculated from the distance of correlated areas between two images.

For the second stage in Eq. (2), we use the sci-kit learn (Pedregosa et al. 2011) implementation of RF in the following way: after the first stage algorithm processes humidity fields $\mathbf{q}(t - \Delta t)$, $\mathbf{q}(t)$, $\mathbf{q}(t + \Delta t)$ and produces first stage $\mathbf{V}^*(t)$ at a pressure level P, we use $\mathbf{V}^*(t)$ as an input for the RF function that will be fitted. The $\mathbf{V}_{\text{NWP}}$ fields become the output of the fitted function. Therefore, the RF function is fitted in such a way that

$$\mathbf{V}_{\text{NWP}}(t) = h\{\mathbf{I}[\mathbf{V}^*(t)]\}, \tag{7}$$

where h is the fitting function and $\mathbf{I}$ is an input matrix that includes $\mathbf{V}^*(t)$ and other features. We randomly sample 5%

of the available NWP data to fit Eq. (7). The final fitted function h should output a "good" approximation of $\mathbf{V}_{\mathrm{NWP}}(t)$, which will be our final, second stage AMV field $\mathbf{V}(t)$.

The complete input matrix $\mathbf{I}$ is

$$\mathbf{I} = (\phi, \lambda, \mathbf{c}, \mathbf{V}^*), \tag{8}$$

where ϕ is the latitude field, λ is the longitude field, and $\mathbf{c}$ computes whether a pixel is on land or ocean.

As mentioned above, we divide the data ($\mathbf{I}$, $\mathbf{V}_{\mathrm{NWP}}$) between a training dataset ($\mathbf{I}_{\mathrm{train}}$, $\mathbf{V}_{\mathrm{NWP\ train}}$) (representing 5% of grid boxes) and a testing dataset ($\mathbf{I}_{\mathrm{test}}$, $\mathbf{V}_{\mathrm{NWP\ test}}$) (representing the remaining 95% of grid boxes). Separating training from testing data makes sure that the algorithm is tested with data that were not fitted.

Given that $\mathbf{V}_{\mathrm{NWP}}$ has inherent uncertainty, we must take care that meaningful structures are captured by h and are not overfitted to error, and therefore the final second stage $\mathbf{V}$ field should be a filtered version of $\mathbf{V}_{\mathrm{NWP}}$.

To avoid overfitting, we train h with noisy data. The noise, however, must account for the structure of the systematic error inherent to the NWP fields. Here we model the spatial structure of the variance as a function of the differences between two reanalyses.

We create the noisy training dataset by taking the G5NR fields (ϕ_{truth}, λ_{truth}, $\mathbf{V}_{\mathrm{truth}}$,) as ground truth and adding noise:

$$x_{\mathrm{train}}(i) = x_{\mathrm{truth}}(i) + \epsilon_x(i) \tag{9}$$

$$\epsilon_x(i) \sim \mathcal{N}\left[0, \sigma_x^2(i)\right], \tag{10}$$

where $x = \phi$, λ, u, v. Here i indicates a grid point iterated across both meridional and zonal directions, since the noise is calculated for each individual grid point. The (ϕ_{truth}, λ_{truth}) fields are part of a regular grid with $\Delta x = 0.0625°$ resolution (equivalent to the G5NR nature run horizontal grid spacing). Here u and v are zonal and meridional components of the horizontal wind velocity, respectively, and ϵ_i is the noise. We use Gaussian distributed noise with $\mathcal{N}\left[\mu, \sigma_x^2(i)\right]$, where μ is the mean, and $\sigma_x(i)$ is the standard deviation for x at grid point i. To avoid extreme values, we put an upper bound on the magnitude of errors at two standard deviations, so $|\epsilon_x(i)| \le 2\sigma_x(i)$. To be more precise, if the magnitude of the value sampled from $\mathcal{N}\left[0, \sigma_x^2(i)\right]$ exceeds the upper bound, it is automatically replaced by either $2\sigma_x(i)$ or $-2\sigma_x(i)$, depending on the sign of the sampled value.

The standard deviations for the position of the AMVs are constant: $\sigma_\lambda(i) = \sigma_\lambda = 1.5°$ and $\sigma_\phi(i) = \sigma_\phi = 0.15°$. These constants are adjusted by trial and error so that the final AMVs are not overfitted. By overfitting here we mean that the algorithm is depending on the NWP fields alone for information, rather than also extracting a signal from fsUA.

The parameters σ_λ and σ_ϕ, although introduced for overfitting, can by physically interpreted as the uncertainty of collocating an AMV on its ground truth coordinates, since tracked AMVs, like most meteorological observations, will have an irreducible spatial representativeness error (Nappo et al. 1982). We discuss these parameters more thoroughly in section 4.

As mentioned before, these NWP fields have an inherent error structure, since they are not equivalent to ground truth velocity fields. To simulate the uncertainty of NWP fields $\mathbf{V}_{\mathrm{NWP}}$, we can approximate a function for the (σ_u, σ_v) fields that already includes assimilated radiosonde data by calculating the difference between the velocity fields of two different reanalysis products in which radiosonde data are assimilated. For real-time applications in the future, the differences from two NWP short-range forecasts (in which radiosonde data are assimilated) would be used. We choose for the reanalysis products ERA-5 (Hersbach et al. 2020) and Climate Forecast System (CFSR) (Saha et al. 2010). Therefore, the standard deviations of the errors for both $\mathbf{u}_{\mathrm{train}}$ and $\mathbf{v}_{\mathrm{train}}$ are

$$\sigma_y(i) = |y_{\mathrm{ERA5}}(i) - y_{\mathrm{CFSR}}(i)|, \tag{11}$$

where $y = u$, v. In other words, our approximation assumes that $\mathbf{V}_{\mathrm{NWP}} = \mathbf{V}_{\mathrm{train}} = \mathbf{V}_{\mathrm{truth}} + \boldsymbol{\epsilon}$.

Equation (11) ensures $\mathbf{V}_{\mathrm{NWPtrain}}$ has less error in the regions where radiosonde data are available, such as in the United States, China, and western Europe (Fig. 1), making the signal stronger in regions that are rich in observations (used in NWP data assimilation). Since reanalysis fields from ERA5 and CFSR have different resolutions than the G5NR fields used in this study, we use two-dimensional linear interpolation to compute $\sigma_y(i)$ by interpolating $y_{\mathrm{ERA5}}(i)$ and $y_{\mathrm{CFSR}}(i)$ for every pixel i.

We must emphasize that $\mathbf{V}_{\mathrm{train}}$ inputted into Eq. (8) must be of the same time t as $\mathbf{V}^*$. This is because the spatial structures learned by the RF algorithm are very time dependent, and therefore one cannot use $\mathbf{V}_{\mathrm{NWP}}$ from a previous t to correct $\mathbf{V}^*$. In this study, we therefore recalculate the reanalysis differences on an NWP cycle by cycle basis, e.g., for 0000, 0600, 1200, and 1800 UTC.

We run our algorithm on data from G5NR (Putman et al. 2014) for our comparison between UA and JPL algorithms. There are eight numerical experiments, and the parameters that define each experiment are described in Table 1. All experiments use $\Delta x = 0.0625°$ for the spatial resolution. For each date, we analyze four triplets of datasets spaced in 6-h intervals. For example, a particular day has triplets centered on hours 0000, 0600, 1200, 1800 UTC. If $\Delta t = 30$ min, then the 0600 UTC triplet is computed from 0530, 0600, and 0630 UTC. Although we explore eight experiments, we will not discuss all of them in the paper. The results of the remainder experiments will be included in the online supplemental material and they are similar to the results discussed in the main text.

b. JPL algorithm

The JPL algorithm is described in Posselt et al. (2019). It is a traditional feature tracking algorithm that works by first finding a feature that has a two-dimensional gradient that exceeds some threshold. Then a search area is established around the feature. At each pixel of the search box, the sum of absolute differences (SAD) between the feature and that

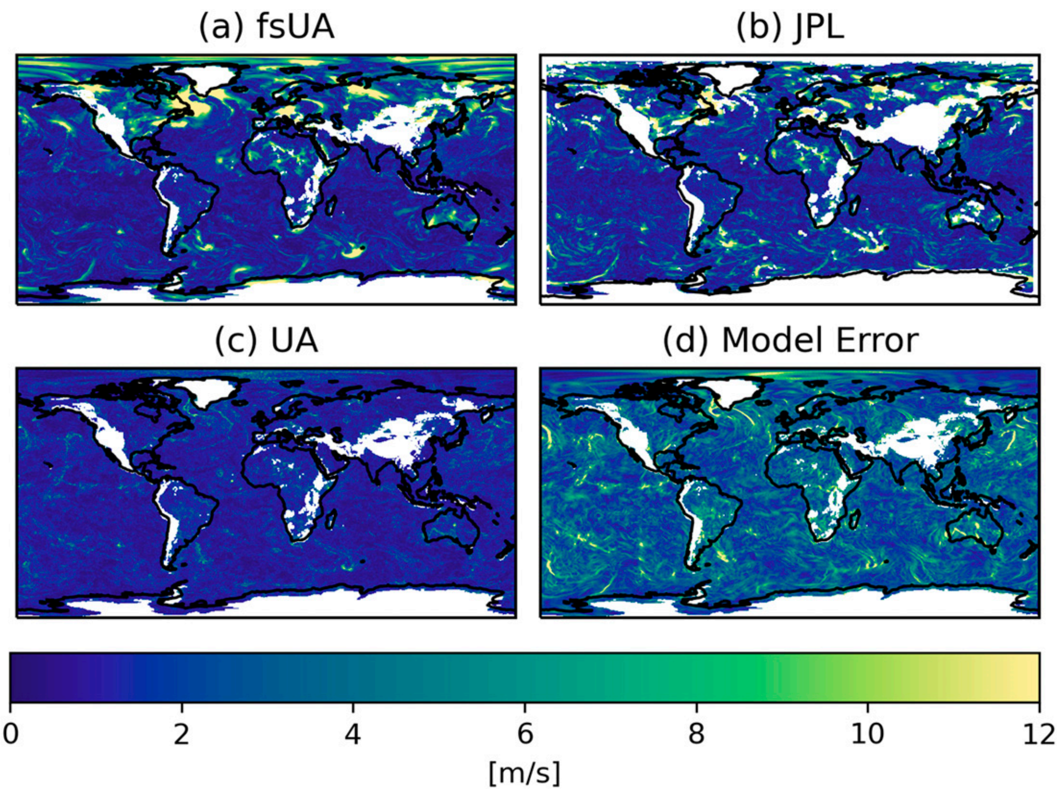

FIG. 1. Maps of magnitudes of velocity vector differences between ground truth and AMVs produced by the different algorithms. The panels represent (a) the first-stage variational algorithm (fsUA), (b) the Jet Propulsion Laboratory algorithm (JPL), (c) the two-stage University of Arizona (UA) algorithm, and (d) the forecast error extracted from reanalysis differences. Calculated for 0000 UTC 1 Jan 2006, $\Delta t = 60$ min, and $P = 850$ hPa. Regions with no data are rendered white.

pixel is computed. The SAD is computed through a 44×44 target box that is centered at each of the search area's pixels. The pixel with the minimum SAD is the displaced feature. A quadratic cost function is used in addition to achieve subpixel accuracy in displacement.

As described in Posselt et al. (2019), the JPL algorithm assumes that the height assignment error is small enough to retrieve specific pressure levels. This approximation is reasonable for hyperspectral infrared sounders for clear-sky pixels and hyperspectral infrared sounders in combination with microwave sounders for all-sky pixels. However, there are not operational hyperspectral, moisture-tracking AMVs yet. Operational moisture tracking products usually use just a few bands, for example GOES-R's ABI instrument only has three bands that can be used for AMVs. This generates a large height assignment error.

TABLE 1. Parameters for numerical experiments. Each set of dates covers the first three days of the chosen month. All experiments have spatial resolution of $\Delta x = 0.0625°$.

Experiment	Δt (min)	Dates	Level (hPa)
1	30	1–3 Jan 2006	850
2	60	1–3 Jan 2006	850
3	30	1–3 Jan 2006	500
4	60	1–3 Jan 2006	500
5	30	1–3 Jul 2006	850
6	60	1–3 Jul 2006	850
7	30	1–3 Jul 2006	500
8	60	1–3 Jul 2006	500

c. *Statistical methods*

For the UA algorithm to outperform the JPL algorithm, its errors must be smaller. However, for either the UA or JPL algorithm to be skillful, its errors need to be smaller than the above forecast errors (due to reanalysis differences and collocation errors).

We will calculate root-mean-square vector difference (RMSVD) across different regions of latitude, where RMVSD is

$$\delta(\mathbf{V}) = \sqrt{\frac{\sum_i \left\{\left[u(i) - u_{\text{truth}}(i)\right]^2 + \left[v(i) - v_{\text{truth}}(i)\right]^2\right\}\cos\phi(i)}{\sum_i \cos\phi(i)}}, \tag{12}$$

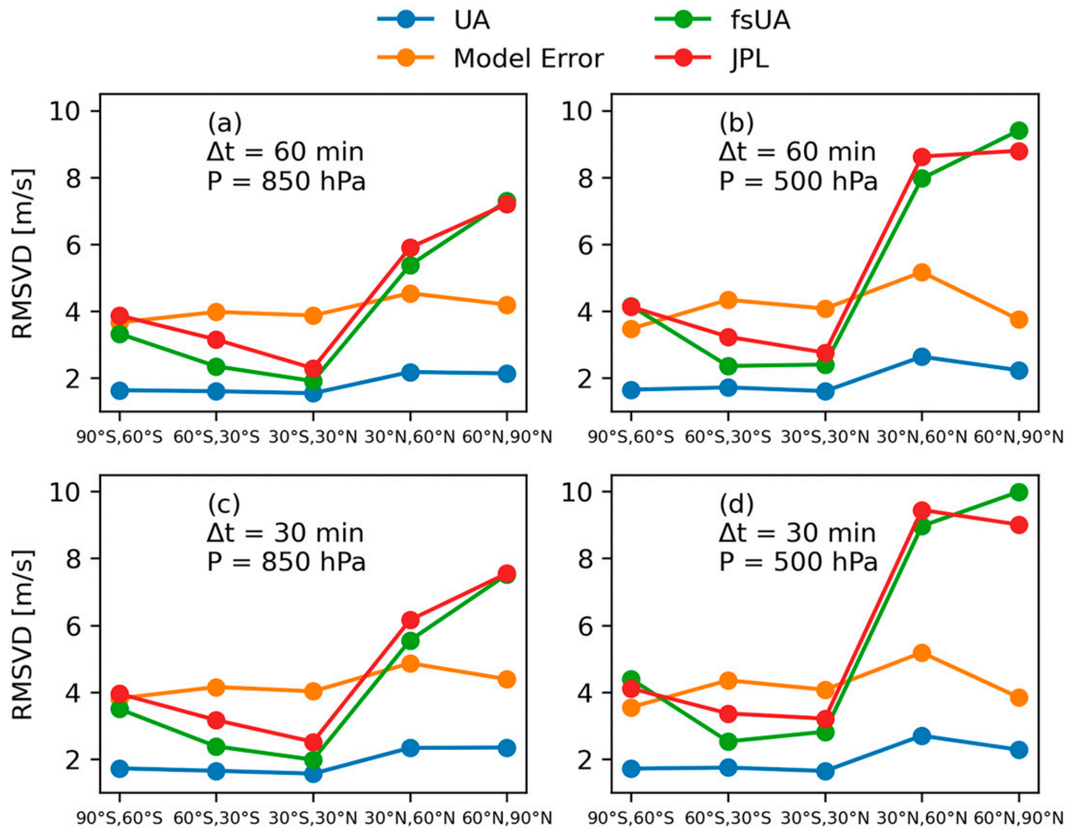

FIG. 2. Plots showing root-mean-square vector differences (RMVSDs) of AMVs vs latitudinal regions for 1–3 Jan 2006. The parameters are on the panels, where dt is the time step and P is the pressure level.

where $\mathbf{V} = (\mathbf{u}, \mathbf{v})$. We also calculate the speed differences Δs:

$$\Delta s(i) = \sqrt{u^2(i) + v^2(i)} - \sqrt{u_{\rm truth}{}^2(i) + v_{\rm truth}{}^2(i)}. \quad (13)$$

Finally, we will calculate the "forecast error," which incorporates both the reanalysis error (ϵ_u, ϵ_v) and the collocation error (ϵ_ϕ, ϵ_λ). This calculation is done by first generating (ϵ_u, ϵ_v) and (ϵ_ϕ, ϵ_λ) through Eqs. (9)–(11), and then collocating $\mathbf{V}_{\rm NWP}(\lambda_{\rm truth} + \epsilon_\lambda, \phi_{\rm truth} + \epsilon_\phi)$, which is in an irregular grid due to random error, into the regular $\Delta x = 0.0625°$ grid of ($\lambda_{\rm truth}$, $\phi_{\rm truth}$), generating $\mathbf{V}_{\rm forecast}(\lambda_{\rm truth}, \phi_{\rm truth})$. We use nearest neighbor interpolation for the collocation. Finally we calculate the RMSVD by computing $\delta(\mathbf{V}_{\rm forecast})$ through Eq. (12).

3. Results

We begin by looking at the spatial distribution of the errors of both the UA and JPL algorithms, and then we discuss the state variables, such as moisture, that underlie these errors. We then conduct a quantitative analysis on how UA produces an error distribution that is less biased, narrower, and more symmetric about zero than the other algorithms.

Figure 1 shows the spatial distribution of the errors, with a pronounced latitudinal anisotropy for JPL and fsUA, since the error is larger in the polar regions. This polar error corresponds to the dryness of the poles. The lower moisture leads to less contrast, and therefore motion becomes more difficult to detect (Posselt et al. 2019). In contrast, UA's errors are not only smaller, but more spatially even than for the other algorithms. Figure 1 also shows that the reanalysis error does not suffer from the polar error anisotropy. However, UA's errors are still smaller than the errors derived from reanalysis differences.

Figure 2 quantifies the above spatial anisotropy of the error for 1–3 January 2006, for both P = 850 hPa and P = 500 hPa. This figure shows that the anisotropic error inherent to both fsUA and JPL is considerably dampened by UA. In fact, UA's error over the tropics is ~2 m s^{-1} and ~2–3 m s^{-1} over the midlatitudes and poles. This error is considerably smaller than for both fsUA and JPL, sometimes smaller by ~70 percent in the poles. Another striking fact is that JPL and fsUA have very similar spatial error structures since their RMSVD plot lines run almost parallel. This is likely because both algorithms lack dynamical information and just limit themselves to pixel changes. However, even if both algorithms have similar global RMSVDs,

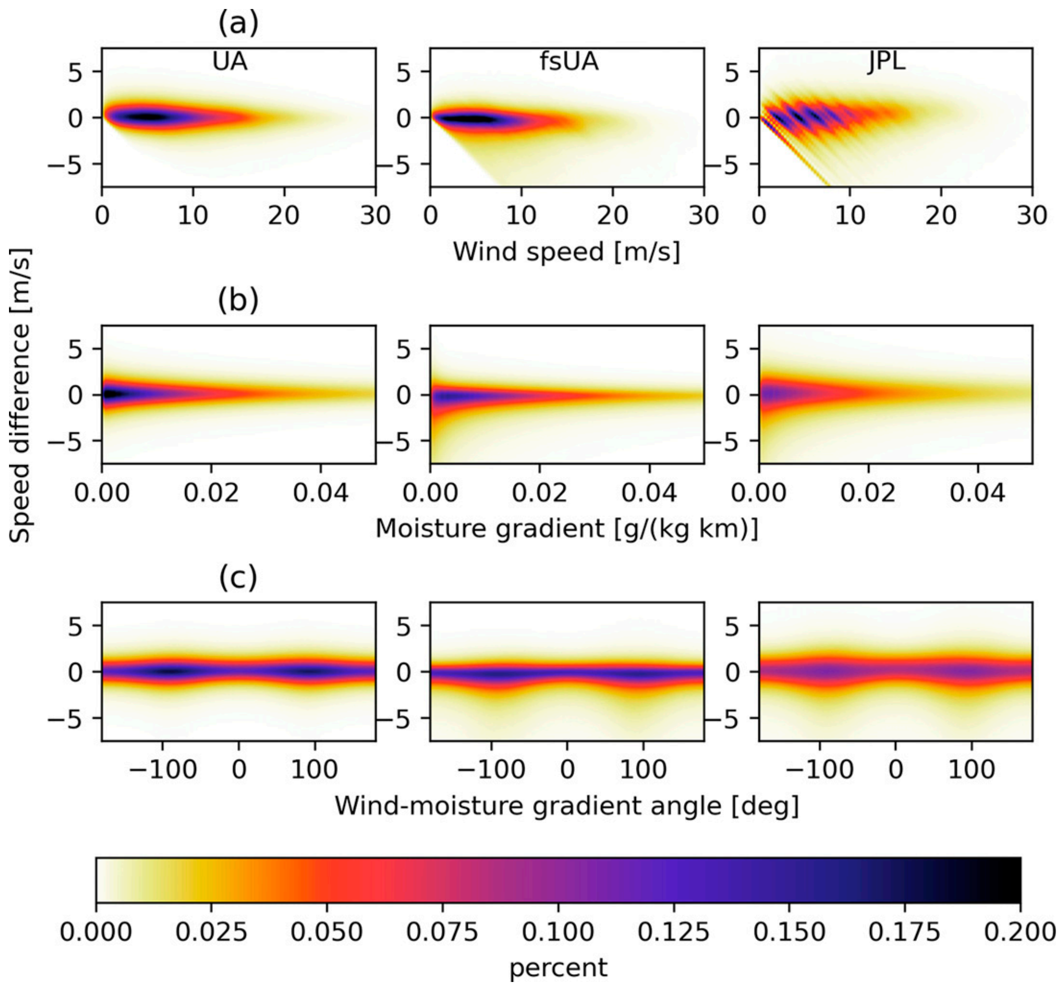

FIG. 3. Histograms showing the speed difference distribution for January 2006, $\Delta t = 60$ min and $P = 850$ hPa for (a) horizontal wind speed, (b) the magnitude of the moisture gradient, and (c) the angle between the moisture gradient and the horizontal velocity vector. (left) The two-stage UA algorithm (UA), (center) fsUA, and (right) the JPL algorithm. The color palette for (c) is scaled by half from what is depicted in the color bar in order to increase the contrast.

fsUA has the advantage of higher resolution, and we will explore this advantage in section 4. The experiments for July are plotted in Fig. S1 of the supplemental material, which confirm this distribution of errors, giving very similar values to Fig. 2.

These spatial error structures are ultimately a function of the underlying state variables, as shown in Fig. 3. Although Fig. 3 refers to $P = 850$ hPa and $dt = 60$ min, it is representative of the other seven experiments (see supplemental material). We already mentioned the moisture gradient, but another two important state variables are horizontal wind speed and angle between moisture gradient and horizontal velocity. We mimic the approach of Posselt et al. (2019) by calculating the speed differences using Eq. (13) between tracked AMVs and ground truth velocity vectors, in order to visualize the speed error bias. We also use cosine weights for the histograms in a similar way to Eq. (12). In the case of the moisture gradient, Fig. 3 confirms that speed difference increases as the moisture gradient decreases. The error is also a function of the angle between water vapor gradient and horizontal velocity. Figure 3 shows that right angles generate the largest speed differences. This relation is expected since winds parallel to the water vapor isolines would leave the gradient unchanged, which inhibits the ability of the feature tracking to capture motion (Posselt et al. 2019). Another important state variable is horizontal wind speed. Figure 3 shows that fsUA and the JPL algorithms have a slow speed bias, since the speed difference distribution is biased toward negative values. This bias is corrected by UA, since the error distribution becomes narrower and more symmetric.

To understand quantitatively how the UA algorithm narrows the distribution of error around lower speed differences and also how it reduces the slow speed bias, we show some summary statistics in Table 2. We calculated various statistical quantities associated with speed difference for each of the four

TABLE 2. Global miscellaneous statistics for speed difference calculated for the dates of 1–3 Jan 2006, for the two-stage University of Arizona algorithm (UA), the first stage variational algorithm (fsUA), and the Jet Propulsion Laboratory algorithm (JPL).

Algorithm	Skewness	Standard deviation (m s^{-1})	Mean (m s^{-1})
	Δt = 60 min, P = 850 hPa		
JPL	−2.60	3.15	−0.62
UA	−1.13	1.29	−0.03
fsUA	−3.64	2.77	−1.12
	Δt = 60 min, P = 500 hPa		
JPL	−3.47	4.18	−0.53
UA	−0.81	1.42	−0.03
fsUA	−3.59	3.79	−1.39
	Δt = 30 min, P = 850 hPa		
JPL	−2.22	3.37	−0.87
UA	−1.01	1.35	−0.04
fsUA	−3.09	2.86	−1.21
	Δt = 30 min, P = 500 hPa		
JPL	−3.49	4.71	−0.94
UA	−0.86	1.46	−0.04
fsUA	−3.28	4.24	−1.64

TABLE 3. Quantile statistics for absolute speed difference calculated for the dates of 1–3 Jan 2006. These statistics are only for horizontal velocity vectors within right angles, or 90° ± 1° and 270° ± 1° for the moisture gradient vector for UA, fsUA, and JPL.

Algorithm	50% (m s^{-1})	68% (m s^{-1})	95% (m s^{-1})
	Δt = 60 min, P = 850 hPa		
JPL	1.08	1.78	7.56
UA	0.61	0.97	2.67
fsUA	0.86	1.49	7.34
	Δt = 60 min, P = 500 hPa		
JPL	1.39	2.36	10.63
UA	0.68	1.08	3.05
fsUA	1.04	2.01	12.48
	Δt = 30 min, P = 850 hPa		
JPL	1.36	2.18	8.34
UA	0.64	1.01	2.81
fsUA	0.96	1.66	7.63
	Δt = 30 min, P = 500 hPa		
JPL	1.70	2.83	12.82
UA	0.70	1.12	3.16
fsUA	1.25	2.41	14.03

experiments. We see that UA filters the noise of fsUA by reducing the magnitude of both the mean and skewness. Furthermore, this denoising means the error distribution becomes more symmetric and narrower, and therefore less biased.

For example, the upper part of Table 2, which corresponds to parameters Δt = 60 min and P = 850 hPa, shows that the mean speed difference for UA is −0.03 m s^{-1} while for fsUA is −1.12 m s^{-1}, indicating that the speed difference distribution shifted to almost zero by the second stage. In contrast, JPL has a slow speed difference mean of −0.62 m s^{-1}. The larger slow speed bias of fsUA to JPL is probably related to the global nature of the fsUA algorithm, since the variational function is minimized globally and therefore it is influenced by large-scale, slow motions as well. In contrast JPL's algorithm calculates AMVs through a local search box, so large-scale motions do not influence it as much.

The standard deviation of fsUA's distribution (2.77 m s^{-1}) is more than halved by UA (1.29 m s^{-1}), narrowing the distribution around the ground truth. Finally, UA's distribution is more symmetric since its skewness is −1.13 which is ~56% smaller than JPL's (−2.60), making the distribution less anisotropic toward slow values and therefore more symmetric and less biased.

Finally, the behavior of the speed differences also shows how UA denoises even the highest error regions in the distribution. Table 3 lists the quantiles for the absolute speed differences for velocity vectors within 1° of right angle with the moisture gradient (e.g., 90° ± 1° and 270° ± 1°). We choose this region in error space since Fig. 3 indicates that the AMVs significantly degrade for velocity vectors within that angle. For example, the second part of Table 3 (1–3 January 2006, Δt = 60 min, P = 500 hPa) shows that the 95th quantile for JPL and fsUA has absolute speed differences of >10 m s^{-1}, making these AMVs almost useless for tracking or data assimilation. In contrast, even for the 95th quantile, UA's speed difference is ~3 m s^{-1}. In other words, Table 3 shows that UA can learn the structure behind the largest errors of fsUA, correcting AMVs that would not otherwise pass operational quality thresholds. UA therefore increases the volume of passable AMVs.

4. Discussion

Not only does the UA algorithm offers a low global RMSVD of ~2–3 m s^{-1} and serves as a filter for both fsUA and model data (due to reanalysis differences), but it has the potential to output a comparatively large data yield of higher resolution, since the volume of operational AMVs is constrained by lower-resolution output and quality control (Santek et al. 2019). For example, the approach used by the JPL algorithm computes the sum of absolute differences within a multipixel target box (in our case, with area of 44 × 44 pixels) for every pixel, which effectively coarsens the resolution.

To compare the resolution of JPL and UA, we plot AMV vectors from a select region in Fig. 4. The JPL vector field is much more uniform than both the ground truth and UA vector fields, showing that JPL's resolution is coarser than UA and the ground truth. Figure 4 also shows qualitatively how UA successfully filters the noise of $\mathbf{V}_{forecast}$, leading to a field similar to the ground truth vectors. Here, the noisy vector fields simulate the inherent error of NWP fields, since NWP fields are not equal to the ground truth (i.e., they are not error-free). Finally, the JPL vector field has more discontinuous horizontal gradients than either UA or fsUA, which generate the jagged features seen in the wind speed histogram of Fig. 3. This is probably related to the fact that JPL tracks only specific features, which are discontinuous from each other, whereas fsUA generates a global continuous field of AMVs through a variational approach.

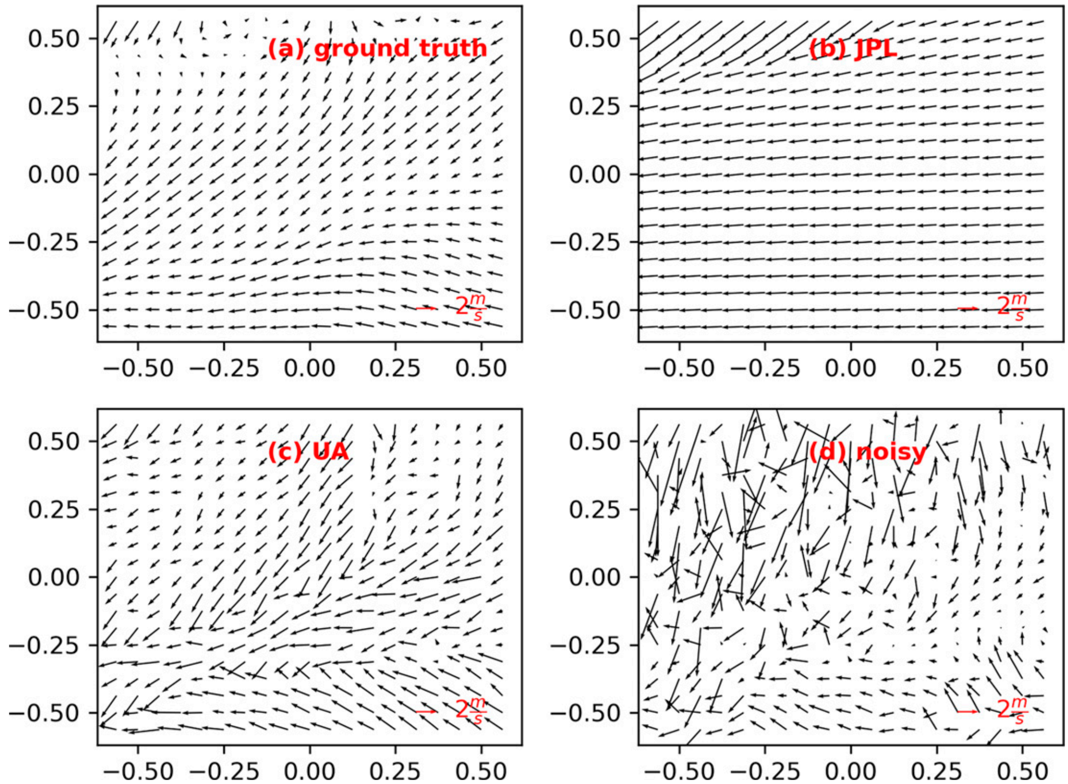

FIG. 4. Comparison of AMV vectors with ground truth and $\mathbf{V}_{NWP}$ retrieved from a G5NR moisture field. The panels represent the (a) ground truth wind velocities, (b) JPL algorithm, (c) UA algorithm, and (d) noisy velocity fields $\mathbf{V}_{forecast}$. The noisy vector fields simulate the inherent error of the NWP fields. Calculated for 0000 UTC 1 Jan 2006, $\Delta t = 60$ min, and $P = 850$ hPa. The region is a patch within borders 0.6°W–0.6°E and 0.6°S–0.6°N. The horizontal axis is longitude, whereas the vertical axis is latitude.

The algorithm fsUA also has a higher resolution than JPL. This is because fsUA produces an AMV per pixel simultaneously through the calculation of a global minimum of the function in Eq. (1), whereas JPL's algorithm uses a local, multipixel target box to calculate feature displacement. Although Fig. 2 shows that the average RMSVD of fsUA is very similar to JPL's RMSVD, JPL's fields are much smoother and therefore do not capture pixel level structures. Therefore, fsUA has the advantage of higher resolution over JPL.

Given the high resolution and low RMSVDs, UA betters the odds that more AMVs could break through the operational quality thresholds, since not only AMV produces AMVs with higher resolution, outputting an AMV per pixel, but also denoises the AMVs. In Fig. 2 the errors of ~8 m s^{-1} or higher in the north pole for fsUA and JPL make these AMVs almost useless. In fact operational, cloud-tracking AMVs are more useful than ordinary feature tracking algorithms when dealing with the poles, since they have a lower error due to the availability of clouds. In contrast, for the moisture tracking algorithms of JPL and fsUA, the dryness in the poles adds less image contrast and therefore makes tracking more difficult.

However, the UA algorithm can decrease the errors in the poles down to ~3 m s^{-1}; this implies that machine learning can be used to compensate for the dryness of the poles. These results hint that many AMVs that would be otherwise thrown away through quality indicators may remain to be useful (e.g., for data assimilation in NWP). Furthermore, UA also performs quite well in error-prone regions that cause other algorithms to output unusable AMVs, such as in low moisture gradients or where velocity is perpendicular to water vapor gradients.

Because satellite infrared channels cannot penetrate clouds, there would be more high-quality water vapor retrievals for clear pixels in the upper troposphere (e.g., 300 hPa) than in the middle (e.g., 500 hPa) or lower (e.g., 850 hPa) troposphere. Therefore, besides the moisture data at 850 and 500 hPa as used in Posselt et al. (2019) and above, we have also tested the UA algorithm using the moisture data at 300 hPa. In general, AMV retrievals in the upper troposphere are more challenging since the moisture content is lower and the wind speeds higher, making them prone to error. Figure 5 shows the RMSVDs for $P = 300$ hPa, $\Delta t = 60$ min, 1–3 July 2006. The overall AMVs of UA stay within a RMSVD of ~3 m s^{-1}, even when the error of both JPL and fsUA exceeds 10 m s^{-1}. This demonstrates the robustness of the UA algorithm for AMV retrievals in the lower, middle, and upper troposphere.

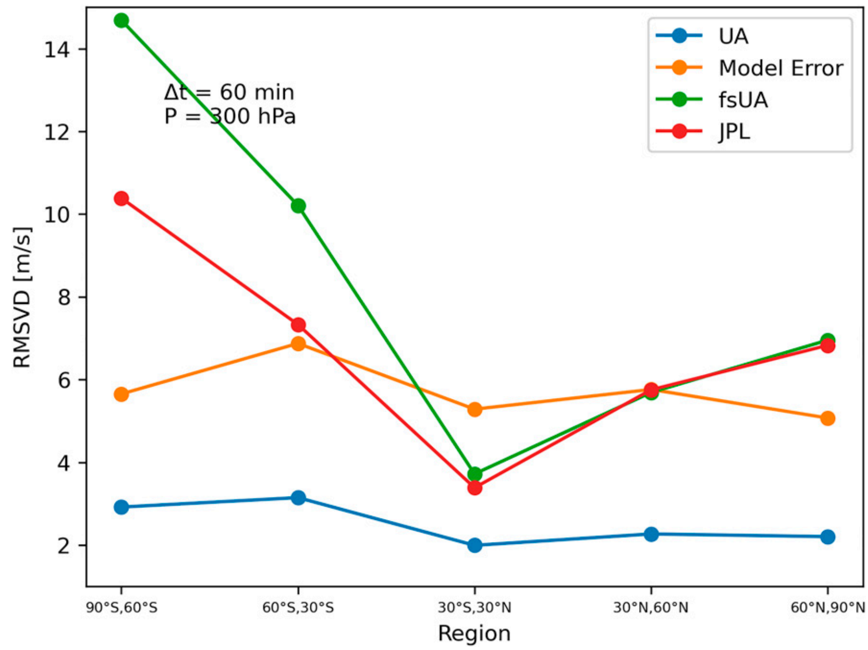

FIG. 5. Plot showing RMSVDs of AMVs vs latitudinal regions for 1–3 Jul 2006 for $\Delta t = 60$ min, and $P = 300$ hPa.

The above results are based on moisture fields from a nature run (i.e., without moisture retrieval uncertainties). In operational settings, the retrieved moisture fields from satellite remote sensing have inherent noise. To test the sensitivity of our algorithm to this source of error, we retrieve moisture fields from simulated radiances that assume the Cubesat Infrared Atmospheric Sounder (CIRAS) instrument as a source (Pagano et al. 2016). To derive the noisy moisture fields from the radiances, we use a linear regression model that fitted the principal components from the hyperspectral radiances to known observations. In contrast to the nature run, the IR-derived moisture fields are vertically smoothed, since individual IR wavelengths are emitted from extended water vapor layers, which adds error in height assignment. In Fig. 6 we depict the AMV error maps for these radiance derived moisture fields, calculated for 1700 UTC 4 November 2009, $\Delta t = 15$ min, and $P = 800$ hPa. UA's RMSVD in this case is 2.68 m s^{-1}, which is smaller than the forecast error of 4.92 m s^{-1} (primarily due to reanalysis differences) and is much smaller than that for fsUA (9.84 m s^{-1}). The region is a patch within borders 150°–100°W and 0°–20°N.

An important point of infrared retrievals such as the ones from CIRAS alone is that clouds affect yield by lowering the number of tractable features, and also by adding error to the AMVs. Assuming 50%–70% of cloud cover as fiducial values, we should expect a yield of about 50%–30%. Furthermore, according to Posselt et al. (2019), the RMSVD added by cloud cover to IR retrieved AMVs is about 1.1 m s^{-1}. This yield and error act as an upper bound for the cloud cover effects in UA when applied to infrared sounding alone, since we expect the random forest algorithm to correct some of the systematic errors brought by cloud cover.

Another possible source of error is the variability inherent to the random sampling of training data. We conducted an ensemble of training experiments that randomly sampled 5% of ($\mathbf{I}$, $\mathbf{V}_{\mathrm{NWP}}$) for training the UA algorithm. We found a standard deviation of less than 0.2 m s^{-1} in the global RMSVD, implying that the algorithm is robust to the random sampling.

Another possible source of noise is that that satellite moisture fields have coarser vertical resolution than the model fields used in this present study, since the former has a ~1.5-km vertical resolution. However, Posselt et al. (2019) found that the vertical resolution does not have a significant effect on the errors of retrieved moisture-tracked AMVs.

Beyond noise, another important source of error is overfitting. Overfitting is defined here when the UA algorithm uses only information from NWP fields rather than also capturing a signal for fsUA fields derived from feature tracking. Overfitted AMVs are therefore too similar to the NWP fields. To prevent this problem, we seeded noise into the latitude and longitude coordinates, creating an effective collocation error, so that UA does not overfit the AMV locations. Instead, the UA algorithm is forced to use the information of the first-stage AMVs, since the signal within the coordinates is reduced. To test whether the collocation error safeguards against overfitting, we ran the complete UA algorithm which includes the first step (UA) and also ran the UA algorithm without fsUA (UA-nofs). We then varied the collocation error in the latitude and longitude coordinates. Table 4 lists some of the results. For the case where the collocation error is

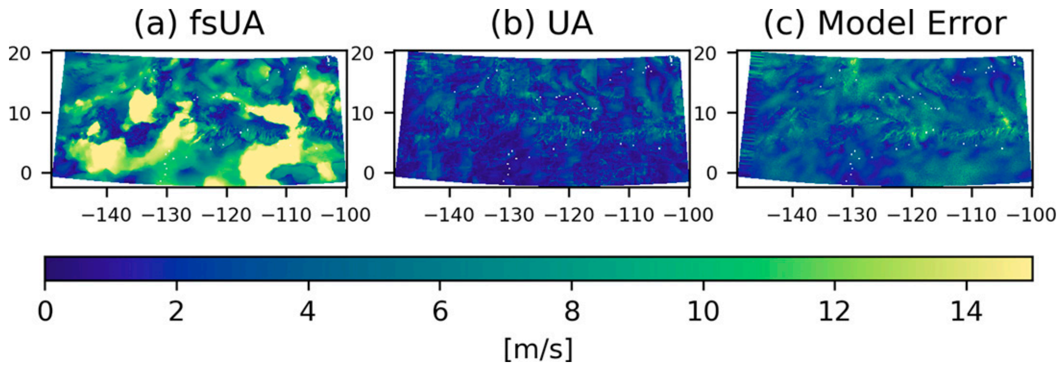

FIG. 6. Maps of magnitudes of velocity vector differences between ground truth and AMVs produced by the different algorithms. The panels represent (a) fsUA, (b) the two-stage UA algorithm, and (c) the forecast error extracted from reanalysis differences. Calculated for 1700 UT 4 Nov 2009, $\Delta t = 15$ min, and $P = 800$ hPa. The region is a patch within borders 150°–100°W and 0°–20°N. The horizontal axis is longitude, whereas the vertical axis is latitude. Regions with no data are rendered white.

zero in both latitude and longitude, UA overfits, since the RMSVD of 1.7 m s^{-1} is basically the same for UA-nofs. This implies that UA with zero collocation error is producing AMVs that are too similar to NWP velocity fields, without capturing the fsUA signal. Once we add collocation error, UA-nofs fares much worse than the complete UA algorithm, even for the collocation errors in the lower end (e.g., $\sigma_\lambda = 0.50°$), implying that UA is forced to capture information from fsUA since the coordinates are not as informative anymore. The collocation error therefore forces UA to use

TABLE 4. Sensitivity tests where the collocation error (σ_ϕ, σ_λ) is varied. The upper part of the table shows the RMSVDs for UA using the full two-stage algorithm. The middle part of the table is the same as the upper part, but it does not use fsUA values as features. The lower part of the table is benchmark values shown for comparison, which include the RMSVDs for JPL, fsUA and the reanalysis error [calculated from Eq. (11)]. Values were computed for 0000 UTC 1 Jan 2006, $\Delta t = 60$ min, and $P = 850$ hPa.

σ_ϕ (°)	σ_λ (°)	RMSVD (m s^{-1})
	UA with fsUA	
0	0	1.71
0.50	0.50	1.83
0.30	1.00	1.94
0.15	1.50	2.07
	UA without fsUA	
0	0	1.73
0.50	0.50	2.71
0.30	1.00	3.48
0.15	1.50	4.23

Benchmarks	
Algorithm	RMSVD (m s^{-1})
JPL	3.88
Reanalysis	2.99
fsUA	3.77

information from fsUA AMVs, guaranteeing that not all information comes from NWP fields. Given that UA extracts a signal from fsUA in addition to the NWP fields, then the algorithm does not overfit. These results suggest that UA is essentially filtering the inherent errors of NWP, implying that the algorithm may work regardless of the NWP model choice (e.g., U.S.-based models or European ones), since the fsUA stage gives information that should aid in filtering model-specific errors.

The structure of the error distribution also raises some questions about overfitting. In this study, we derived an error distribution, under the assumption of an available ground truth $\mathbf{V}_{\text{truth}}$. In an operational setting, we would not calculate the error $\boldsymbol{\epsilon} = \mathbf{V}_{\text{NWP}} - \mathbf{V}_{\text{truth}}$, since we will use the NWP field itself, e.g., $\mathbf{V}_{\text{NWP}} = \mathbf{V}_{\text{truth}} + \boldsymbol{\epsilon}$, rather than constructing an artificial $\mathbf{V}_{\text{NWP}}$ from $\mathbf{V}_{\text{truth}}$ (from the nature run) and $\boldsymbol{\epsilon}$ (primarily from the reanalysis differences) as we did in this study. One concern is that UA could overfit $\mathbf{V}_{\text{NWP}}$, generating AMVs that are too similar to NWP wind fields. As mentioned earlier, the key to prevent overfitting are the σ_λ and σ_ϕ parameters. Increasing σ_λ and σ_ϕ is a way to attenuate the information carried in the coordinates, and therefore increase the weight of the first stage AMVs, balancing the NWP signal with the fsUA signal. Therefore, future research on the operational application of the UA algorithm should explore the parameters σ_λ and σ_ϕ.

Another concern that UA must address is the related to statistical learning and nonlinear dynamics. The humidity fields are a function of nonlinear fluid dynamical equations that govern atmospheric motion. The naive ML approach of learning a universal function for the correction of fsUA fields may require processing an impossibly high volume of data. To side-step this data bottleneck problem, we choose to "retrain" the algorithm for every time step using NWP short-range forecast fields. Instead of fitting a universal error function, our model is recalibrated for every time step.

In this study, reanalysis differences are used to generate the error structure. For operational use, the use of short-term (e.g., ~6 h) forecast fields (with assimilation of data available from earlier period) is more adequate than using reanalysis fields. Even if the short-term forecast fields are not the same as the reanalysis fields, their broad error structures should be similar, since for both cases, regions dense with sounding data should have less error than observation-sparse regions like the oceans. Our derived AMVs can then be used for data assimilation in NWP models for the current period, and the short-range forecasts with these data assimilations will then be used to help our AMV retrieval at the next time period.

5. Conclusions

Following Posselt et al.'s (2019) finding that traditional algorithms underperform when images do not capture enough dynamical information, such as in low moisture gradients, we developed a novel, two-stage AI algorithm for calculating AMVs that extracts dynamical information from NWP short-range forecast fields (mimicked by using nature run "ground truth" plus error structures as a function of reanalysis differences), which we refer to as the UA algorithm.

We also compared this algorithm with a traditional, feature tracking algorithm (JPL). The UA algorithm performs very well, producing AMVs with RMSVDs of ~2 m s^{-1} in the tropics and ~2–3 m s^{-1} for the midlatitudes and poles. In contrast, the JPL algorithm yields higher RMSVDs of ~3 m s^{-1} for the tropics and up to ~9 m s^{-1} for the midlatitudes and poles. Furthermore, UA produces one AMV per pixel, whereas JPL's AMVs are smoothed by a target box matching algorithm. The value of the UA algorithm is further enhanced by its smaller errors than the NWP errors (due to reanalysis differences and collocation errors).

The first stage fsUA is a feature-tracking algorithm that detects the motion of features by processing changes in pixel values. In particular, we use variational method, which assumes that each "pixel particle" conserves its value and gradient. We show in this study that fsUA has a higher resolution than JPL, since the former produces an AMV per pixel, whereas the latter produces smoothed fields. Although both have similar global RMSVDs, fsUA can capture finer structure. Therefore, fsUA could be more effective than more traditional algorithms when dealing with high-resolution features.

In the second stage, an RF algorithm extracts information from NWP fields in order to correct the first stage values. Since the NWP model follows the primitive equations of the atmosphere, the algorithm can extract physical signals from the NWP fields and use it to correct the "dynamics blind" algorithm of the first step. We approximated the error distribution of the NWP fields using reanalysis differences plus collocation errors, since the true forecast error of the velocity fields is unknown.

The UA algorithm is robust to error-prone regions such as low moisture gradients or velocity fields perpendicular to water vapor gradients. The physics blindness of tracking algorithms such as JPL's algorithm makes them underperform when there is insufficient contrast in the images to capture motion. In contrast, UA extracts physical structure that can ameliorate the image's lack of signal, outputting better quality AMVs than physics-blind algorithms like JPL.

UA is advantageous beyond its low RMSVD and its robustness to error-prone regions. For example, for operational products, sometimes more than ~50% of the cloud-tracking AMVs can be removed due to large errors (Santek et al. 2019). However, the UA algorithm can potentially lead to large yields by ensuring more AMVs stay within a certain quality threshold. Furthermore, the fact that UA outputs high resolution fields at each pixel (rather than lower resolution fields through some spatial smoothing, as in JPL) would also potentially increase the volume of AMVs passing quality thresholds. Finally the UA algorithm might aid in the assimilation of more moisture-tracking AMVs, since most AMVs assimilated in operational settings are cloud tracking.

Although for this study we are using moisture fields derived from models, UA is developed for moisture fields retrieved from hyperspectral instruments. Existing hyperspectral moisture fields are retrieved through a radiative transfer model that uses a least squares residual minimization (Gambacorta et al. 2012). One issue is that this method creates error correlations between the retrieved moisture fields and forecast fields, since the forward model requires model moisture fields as guesses. Furthermore, real satellite images would likely increase the noise. Finally, clouds can corrupt retrieved moisture fields, so they pose a challenge to the AMV algorithms.

Given the potential issues arising from retrieved moisture fields, for future work, we will apply the UA algorithm to actual satellite measurements. The current results act as a lower bound of the error, since simulation data were used to calculate the AMVs. Future tests should collocate the UA output with other observations (e.g., radiosonde and Aeolus lidar measurements) to test the assumptions behind the error approximation. Using NWP fields for training data may also produce biases in the AMVs, since the NWP fields themselves suffer from systematic errors. For future work, we will compare the UA AMVs with unbiased sources such as radiosondes to find, and hopefully correct, biases. We are currently working in deriving hyperspectral AMVs from the swath overlap of *JPSS-1* and *Suomi-NPP*, which are satellites separated by a 50-min interval. The use of authentic infrared retrievals will also make it possible to test the effects of cloud cover as well. Given that these satellites use infrared sounders, our results act as an upper bound, since we expect clouds to deteriorate the quality and yield of AMVs, as discussed in Posselt et al. (2019).

Note that clear-sky hyperspectral radiances at different times can be assimilated by 4D-Var to derive wind fields (e.g., in the ERA-5 reanalysis). Here we show the UA algorithm yields smaller errors than the NWP errors (due to reanalysis differences and collocation errors). For the future, we will also compare our retrieved AMVs from actual satellite hyperspectral measurements versus those from 4D-Var (e.g., ERA-5) against radiosonde and dropsonde measurements.

Acknowledgments. This work is funded by NASA Grant 80NSSC19K0442 in support of ACTIVATE, which is a NASA Earth Venture Suborbital-3 (EVS-3) investigation funded by NASA's Earth Science Division and managed through the Earth System Science Pathfinder Program Office, and by the Arizona Space Institute. The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). We thank the reviewers for their constructive comments and suggestions for our significant revision of the original manuscript.

Data availability statement. All the data analyzed are available publicly. The ERA5 reanalysis data can be downloaded from the Copernicus data store: https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-pressure-levels?tab=overview. The CFSR pressure level reanalysis data are located at https://www.ncei.noaa.gov/data/climate-forecast-system/access/reanalysis/6-hourly-by-pressure-level/. Finally, the GEOS-5 nature run datasets are located at https://gmao.gsfc.nasa.gov/global_mesoscale/7km-G5NR/.

REFERENCES

Baker, W. E., and Coauthors, 2014: Lidar-measured wind profiles: The missing link in the global observing system. *Bull. Amer. Meteor. Soc.*, **95**, 543–564, https://doi.org/10.1175/BAMS-D-12-00164.1.

Bertalmio, M., A. L. Bertozzi, and G. Sapiro, 2001: Navier-stokes, fluid dynamics, and image and video inpainting. *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, HI, Institute of Electrical and Electronics Engineers, https://doi.org/10.1109/CVPR.2001.990497.

Brox, T., A. Bruhn, N. Papenberg, and J. Weickert, 2004: High accuracy optical flow estimation based on a theory for warping. *Computer Vision - ECCV 2004*, T. Pajdla and J. Matas, Eds., Lecture Notes in Computer Science, Vol. 3024, Springer, 25–36, https://link.springer.com/chapter/10.1007/978-3-540-24673-2_3.

Carr, J., D. L. Wu, M. A. Kelly, and J. Gong, 2018: MISR-GOES 3D Winds: Implications for future LEO-GEO and LEO-LEO winds. *Remote Sens.*, **10**, 1885, https://doi.org/10.3390/rs10121885.

Cotton, J., A. Doherty, K. Lean, M. Forsythe, and A. Cress, 2020: NWP SAF AMV monitoring: The 9th analysis report (AR9). NWPSAF-MO-TR-039, 48 pp., http://nwp-saf.eumetsat.int/monitoring/amv/nwpsaf_mo_tr_039.pdf.

Forsythe, M., C. Peubey, C. Lupu, and J. Cotton, 2014: Assimilation of wind information from radiances: AMVs and 4D-Var tracing. ECMWF Annual Seminar, 15 pp., https://www.ecmwf.int/sites/default/files/elibrary/2015/9444-assimilation-wind-information-radiances-amvs-and-4d-var-tracing.pdf.

Gambacorta, A., and Coauthors, 2012: The NOAA unique CrIS/ATMS processing system (NUCAPS): First light retrieval results. *Proc. ITWG Meeting, ITWG*, Toulouse, France, NOAA/NESDIS/STAR, 9 pp., https://cimss.ssec.wisc.edu/itwg/itsc/itsc18/program/files/links/1.03_Gambacorta_pa.pdf.

Geer, A. J., B. Fabrizio, N. Bormann, and S. English, 2014: All-sky assimilation of microwave humidity sounders. Tech. Memo. 741, European Centre for Medium-Range Weather Forecasts, 57 pp., https://www.ecmwf.int/sites/default/files/elibrary/2014/9507-all-sky-assimilation-microwave-humidity-sounders.pdf.

Hautecoeur, O., R. Borde, M. Doutriaux-Boucher, and M. Carranza, 2014: EUMETSAT operational dual-Metop wind products. *Proc. 12th Int. Winds Workshop*, Copenhagen, Denmark, EUMETSAT, 8 pp., https://www-cdn.eumetsat.int/files/2020-04/pdf_conf_p61_s2_04_hautecoe_v.pdf.

Hersbach, H., and Coauthors, 2020: The ERA5 global reanalysis. *Quart. J. Roy. Meteor. Soc.*, **146**, 1999–2049, https://doi.org/10.1002/qj.3803.

Itseez, 2015: Open source computer vision library. GitHub, https://github.com/itseez/opencv.

James, G., D. Witten, T. Hastie, and R. Tibshirani, 2013: *An Introduction to Statistical Learning*. Springer Texts in Statistics, Vol. 103, Springer, 426 pp.

Key, J. R., D. Santek, C. S. Velden, N. Bormann, J. Thepaut, L. P. Riishojgaard, Y. Zhu, and W. P. Menzel, 2003: Cloud-drift and water vapor winds in the polar regions from MODISIR. *IEEE Trans. Geosci. Remote Sens.*, **41**, 482–492, https://doi.org/10.1109/TGRS.2002.808238.

Lambrigtsen, B., H. Van Dang, F. J. Turk, S. M. Hristova-Veleva, H. Su, and Y. Wen, 2018: All-weather tropospheric 3-D wind from microwave sounders. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **11**, 1949–1956, https://doi.org/10.1109/JSTARS.2018.2814540.

Lazzara, M. A., R. Dworak, D. A. Santek, B. T. Hoover, C. S. Velden, and J. R. Key, 2014: High-latitude atmospheric motion vectors from composite satellite data. *J. Appl. Meteor. Climatol.*, **53**, 534–547, https://doi.org/10.1175/JAMC-D-13-0160.1.

Lean, K., and N. Bormann, 2018: GOES-16 AMV data evaluation and algorithm assessment. *Proc. 14th International Winds Workshop*, Jeju Island, South Korea, ECMWF, 20 pp., http://cimss.ssec.wisc.edu/iwwg/iww14/talks/02_Tuesday/1630_IWW14_GOESR_ECMWF_Lean.pdf.

Martins, V. S., A. Lyapustin, Y. Wang, D. M. Giles, A. Smirnov, I. Slutsker, and S. Korkin, 2019: Global validation of columnar water vapor derived from EOS MODIS-MAIAC algorithm against the ground-based AERONET observations. *Atmos. Res.*, **225**, 181–192, https://doi.org/10.1016/j.atmosres.2019.04.005.

McCarty, W., D. Carvalho, I. Moradi, and N. C. Privé, 2021: Observing system simulation experiments investigating atmospheric motion vectors and radiances from a constellation of 4–5-μm infrared sounders. *J. Atmos. Oceanic Technol.*, **38**, 331–347, https://doi.org/10.1175/JTECH-D-20-0109.1.

Nappo, C., and Coauthors, 1982: The workshop on the representativeness of meteorological observations, June 1981, Boulder, Colo. *Bull. Amer. Meteor. Soc.*, **63**, 761–764, https://doi.org/10.1175/1520-0477-63.7.761.

Pagano, T. S., D. Rider, M. Rud, D. Ting, and K. Yee, 2016: Measurement approach and design of the CubeSat Infrared Atmospheric Sounder (CIRAS). *Proc. SPIE*, **9978**, 997806, https://doi.org/10.1117/12.2235986.

Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Posselt, D. J., and Coauthors, 2019: Quantitative assessment of state-dependent atmospheric motion vector uncertainties. *J. Appl. Meteor. Climatol.*, **58**, 2479–2495, https://doi.org/10.1175/JAMC-D-19-0166.1.

Putman, W. M., A. M. Da Silva, L. E. Ott, and A. Darmenov, 2014: Model configuration for the 7-km GEOS-5 nature run, Ganymed release. GMAO Office Note 5, NASA, 18 pp., https://gmao.gsfc.nasa.gov/pubs/docs/Putman727.pdf.

Reitebuch, O., C. Lemmerz, E. Nagel, U. Paffrath, Y. Durand, M. Endemann, F. Fabre, and M. Chaloupy, 2009: The airborne demonstrator for the direct-detection Doppler wind lidar ALADIN on ADM-Aeolus. Part I: Instrument design and comparison to satellite instrument. *J. Atmos. Oceanic Technol.*, **26**, 2501–2515, https://doi.org/10.1175/2009JTECHA1309.1.

Saha, S., and Coauthors, 2010: The NCEP Climate Forecast System Reanalysis. *Bull. Amer. Meteor. Soc.*, **91**, 1015–1058, https://doi.org/10.1175/2010BAMS3001.1.

Santek, D., and Coauthors, 2019: 2018 Atmospheric Motion Vector (AMV) intercomparison study. *Remote Sens.*, **11**, 2240, https://doi.org/10.3390/rs11192240.

Šavli, M., J. de Kloe, G.-J. Marseille, M. Rennie, N. Žagar, and N. Wedi, 2019: The prospects for increasing the horizontal resolution of the Aeolus horizontal line-of-sight wind profiles. *Quart. J. Roy. Meteor. Soc.*, **145**, 3499–3515, https://doi.org/10.1002/qj.3634.

Schmit, T. J., P. Griffith, M. M. Gunshor, J. M. Daniels, S. J. Goodman, and W. J. Lebair, 2017: A closer look at the ABI on the GOES-R series. *Bull. Amer. Meteor. Soc.*, **98**, 681–698, https://doi.org/10.1175/BAMS-D-15-00230.1.

Stoffelen, A., and Coauthors, 2020: Wind profile satellite observation requirements and capabilities. *Bull. Amer. Meteor. Soc.*, **101**, E2005–E2021, https://doi.org/10.1175/BAMS-D-18-0202.1.

Weinzaepfel, P., J. Revaud, Z. Harchaoui, and C. Schmid, 2013: DeepFlow: Large displacement optical flow with deep matching. *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, Sydney, NSW, Australia, Institute of Electrical and Electronics Engineers, 1385–1392, https://doi.org/10.1109/ICCV.2013.175.

Zou, C.-Z., and Coauthors, 2020: The reprocessed Suomi NPP satellite observations. *Remote Sens.*, **12**, 2891, https://doi.org/10.3390/rs12182891.