The Astrophysical Journal, 939:66 (12pp), 2022 November 10 https://doi.org/10.3847/1538-4357/ac927e

Inferring Line-of-sight Velocities and Doppler Widths from Stokes Profiles of GST/NIRIS Using Stacked Deep Neural Networks

Haodi Jiang[1,2], Qin Li[1,3], Yan Xu[1,3,4], Wynne Hsu[5,6], Kwangsu Ahn[4], Wenda Cao[1,3,4], Jason T. L. Wang[1,2], and Haimin Wang[1,3,4]

[1] Institute for Space Weather Sciences, New Jersey Institute of Technology, University Heights, Newark, NJ 07102-1982, USA; hj78@njit.edu, wangj@njit.edu, haimin.wang@njit.edu
[2] Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102-1982, USA
[3] Center for Solar-Terrestrial Research, New Jersey Institute of Technology, University Heights, Newark, NJ 07102-1982, USA
[4] Big Bear Solar Observatory, New Jersey Institute of Technology, 40386 North Shore Lane, Big Bear City, CA 92314-9672, USA
[5] Institute of Data Science, National University of Singapore, 119077, Singapore
[6] Department of Computer Science, School of Computing, National University of Singapore, 119077, Singapore
Received 2021 November 27; revised 2022 September 11; accepted 2022 September 14; published 2022 November 3

Abstract

Obtaining high-quality magnetic and velocity fields through Stokes inversion is crucial in solar physics. In this paper, we present a new deep learning method, named Stacked Deep Neural Networks (SDNN), for inferring line-of-sight (LOS) velocities and Doppler widths from Stokes profiles collected by the Near InfraRed Imaging Spectropolarimeter (NIRIS) on the 1.6 m Goode Solar Telescope (GST) at the Big Bear Solar Observatory (BBSO). The training data for SDNN are prepared by a Milne–Eddington (ME) inversion code used by BBSO. We quantitatively assess SDNN, comparing its inversion results with those obtained by the ME inversion code and related machine-learning (ML) algorithms such as multiple support vector regression, multilayer perceptrons, and a pixel-level convolutional neural network. Major findings from our experimental study are summarized as follows. First, the SDNN-inferred LOS velocities are highly correlated to the ME-calculated ones with the Pearson product–moment correlation coefficient being close to 0.9 on average. Second, SDNN is faster, while producing smoother and cleaner LOS velocity and Doppler width maps, than the ME inversion code. Third, the maps produced by SDNN are closer to ME's maps than those from the related ML algorithms, demonstrating that the learning capability of SDNN is better than those of the ML algorithms. Finally, a comparison between the inversion results of ME and SDNN based on GST/NIRIS and those from the Helioseismic and Magnetic Imager on board the Solar Dynamics Observatory in flare-prolific active region NOAA 12673 is presented. We also discuss extensions of SDNN for inferring vector magnetic fields with empirical evaluation.

Unified Astronomy Thesaurus concepts: Solar atmosphere (1477); Solar magnetic fields (1503); Convolutional neural networks (1938)

1. Introduction

Obtaining high-quality magnetic and velocity fields through Stokes inversion is crucial in solar physics (Nejezchleba 1998; Maurya & Ambastha 2010; Bobra & Couvidat 2015). Stokes inversion attempts to infer the physical conditions of the solar atmosphere based on the interpretation of observed Stokes profiles (Asensio Ramos et al. 2008; Bommier 2016). Estimates of the physical magnitudes governing the state of the solar atmosphere can be obtained through the various inversion methods that try to achieve the best fit to the observed Stokes profiles (Teng & Deng 2014). An excellent review of the status of inversions can be found in del Toro Iniesta & Ruiz Cobo (2016), in which the Milne–Eddington (ME) approximation is commonly used. The ME is an approach to the radiative transfer equation (RTE) which states that, when all the atmospheric quantities are constant with depth except for the source function that varies linearly, the RTE can be solved analytically (Unno 1956; Rachkovsky 1962, 1963; Auer et al. 1977; Landi Degl'Innocenti 1982, 1983; Landolfi & Landi Degl'Innocenti 1982; Landi Degl'Innocenti & Landolfi 2004). Implementations based on the ME approach include Helix+ (Lagg et al. 2004), MILOS (Orozco Suárez et al. 2007), MERLIN (Lites et al. 2007) and VFISV (Borrero et al. 2007, 2011). On the other hand, with the availability of high-performance computing, algorithms based on the local thermodynamic equilibrium (LTE) and non-LTE conditions, which can solve the full radiative transfer equation, have also become popular. Examples of such algorithms include SIR (Ruiz Cobo & del Toro Iniesta 1992), SPINOR (Frutiger et al. 2000), and NICOLE (Socas-Navarro et al. 2015).

Because Stokes inversion is a time-consuming task, there have been efforts to employ machine learning (ML) to accomplish this task. After an ML model is trained, one can use the trained model to perform Stokes inversion through making predictions, which reduces the inversion time significantly (Liu et al. 2020). For example, Teng (2015) developed a multiple support vector regression (MSVR) method for Stokes inversion. The MSVR method took as input Stokes I, Q, U, and V profiles scaled at six wavelengths from the Helioseismic and Magnetic Imager (HMI) on board the Solar Dynamics Observatory (SDO) and produced as output atmospheric parameters including the magnetic field strength and inclination angle. Carroll & Kopf (2008) designed three multilayer perceptrons (MLPs), which took as input Stokes I and V profiles that resulted from magnetohydrodynamic (MHD) simulations and produced as output the line-of-sight (LOS) velocity, magnetic field, and temperature, respectively.

In recent years, more powerful deep learning approaches have been developed to solve the Stokes inversion problem. For example, Asensio Ramos & Díaz Baso (2019) presented two convolutional neural networks (CNNs; LeCun et al. 2015) to perform Stokes inversion on synthetic two-dimensional (2D) maps of Stokes profiles where the authors exploited the 2D spatial coherence of the field of view to produce a three-dimensional (3D) cube of thermodynamic and magnetic properties. Liu et al. (2020) designed a pixel-level CNN, referred to as PCNN, to perform Stokes inversion on the Near InfraRed Imaging Spectropolarimeter (NIRIS) data (Cao et al. 2012) from the 1.6 m Goode Solar Telescope (GST) at the Big Bear Solar Observatory (BBSO; Cao et al. 2010; Goode et al. 2010; Goode & Cao 2012; Varsik et al. 2014). The PCNN took as input Stokes Q, U, and V profiles from GST/NIRIS and produced as output three components of the magnetic field vector (i.e., the magnetic field strength, inclination angle, and azimuth angle). Separately, Milić & Gafeira (2020) developed a similar PCNN, which took as input Stokes I and V profiles resulting from MHD simulations and produced as output the magnetic field strength, LOS velocity, microturbulent velocity, temperature, and magnetic field inclination. Later, the authors extended their approach to Stokes I, Q, U, and V profiles and used the PCNN-inverted results to accelerate the SIR inversion code (Gafeira et al. 2021).

In this paper, we present a new deep learning method, named Stacked Deep Neural Networks (SDNN), for inverting Stokes profiles of GST/NIRIS. SDNN aims to infer two kinematic parameters, namely the LOS velocity and Doppler width. In addition, SDNN can be extended to infer the magnetic field strength, inclination angle, and azimuth angle. These five atmospheric parameters, including the LOS velocity, Doppler width, magnetic field strength, inclination angle, and azimuth angle are mostly used by researchers to understand the evolution of physical properties of the solar atmosphere (Keys et al. 2021). It should be pointed out that, although both SDNN and the PCNN developed by Liu et al. (2020) perform Stokes inversion on the data from GST/NIRIS, the two tools differ in three ways. First, Liu et al.'s PCNN focuses on predicting the magnetic field strength, inclination angle, and azimuth angle. In contrast, SDNN is designed to infer the LOS velocity and Doppler width in addition to the vector magnetic field. Second, when applying Liu et al.'s PCNN to infer the LOS velocity, it fails to infer the granular patterns in the LOS velocity image of sunspot data. In contrast, SDNN can infer all the convective granulation structures in the LOS velocity image (Ortiz et al. 2014). Third, the architecture of SDNN, which is better suited for Stokes inversion, is totally different from that of Liu et al.'s PCNN. As demonstrated in our experimental study, SDNN outperforms Liu et al.'s PCNN on several different data sets.

The rest of this paper is organized as follows. Section 2 describes solar observations and GST/NIRIS data used in this study. Section 3 presents details of our SDNN model. Section 4 reports experimental results obtained by using SDNN to infer LOS velocities and Doppler widths from Stokes profiles of GST/NIRIS. We also discuss results obtained by extending SDNN to infer magnetic field strengths, inclination, and azimuth angles from the Stokes profiles of GST/NIRIS. Section 5 presents a discussion and concludes the paper.

2. Observations and Data Preparation

GST/NIRIS is a Fabry–Pérot-based imaging system that provides high-resolution Stokes parameters I, Q, U, and V of the Fe I 1560 nm line within a ± 0.25 nm spectral window (Cao et al. 2012). A typical field of view is about 85″ with an image scale of $0\farcs083$ pixel^{-1} (Wang et al. 2015; Xu et al. 2016, 2018; Liu et al. 2018). The data used in this study were obtained from three active regions (ARs), namely AR 12371, AR 12665, and AR 12673. Our first data set, denoted D1, is a 990×950 image from AR 12371 collected at 17:33:00 UT on 2015 June 22. The second data set, denoted D2, is a 720×720 image from the same AR 12371 but collected approximately three days later at 16:55:13 UT on 2015 June 25. The third data set, denoted D3, is a 720×720 image from AR 12665 collected at 16:20:12 UT on 2017 July 13. The fourth data set, denoted D4, is a 720×720 image from AR 12673 collected at 19:17:53 UT on 2017 September 6.

Because D1 has the most pixels with the largest range of Stokes component values among the four data sets, we use D1 as the training set. This data set has $990 \times 950 = 940{,}500$ pixels. Each pixel is treated as a training data sample containing Stokes component values and labeled by the LOS velocity and Doppler width calculated by an ME inversion code. The code is specifically designed for GST/NIRIS (Chae & Park 2009; Ahn & Cao 2017, 2019). Its early version was applied to the Hinode/SP data (Chae & Park 2009). The code, written with the IDL language, is based on the formulae given in Landi Degl'Innocenti (1992). After careful elimination of the crosstalk among Stokes profiles (Ahn & Cao 2019), we perform Stokes inversion under the ME atmospheric approximation (with initial parameters precalculated to resemble the observed Stokes profiles), assuming that the magnetic field and velocity are constant with height through the solar atmosphere (Borrero et al. 2014). We set the filling factor/stray light fraction parameter to unity, because magnetic structures are believed to be fully resolved in these data (Şahin et al. 2019). The code outputs nine parameters, including the Doppler shift, Doppler width, magnetic field strength, inclination angle, azimuth angle, and so on. We assign Stokes I profiles less than half the weight of Stokes Q, U, and V profiles before performing the inversion fitting process, as the Stokes I profiles intrinsically have more noise factors than the other profiles. The fitting process may return error values for the inverted nine parameters. Most of the fitting errors come from the initial guess of the longitudinal field strength calculated by the center-of-gravity method (Ahn & Cao 2018).

Thus, as in Liu et al. (2020), we use the output of the ME inversion code as the training labels. Notice that the number of spectral points scanned by GST/NIRIS is usually 60, but varied in some particular days. For instance, there were 56 spectral points for AR 12673 scanned on 2017 September 6. For consistency reasons, zeros are added so that the number of spectral points is unified and fixed at 60. There are four Stokes components, I, Q, U, and V, at each spectral position, so the length of each training data sample, corresponding to each pixel in D1, is 60×4. There are two labels, namely the LOS velocity and Doppler width produced by the ME code, associated with the pixel. Therefore, the total length of the training data sample fed to our SDNN model is $60 \times 4 + 2 = 242$.

The remaining three data sets, D2, D3, and D4, are used as test sets. The training set and test tests are disjoint. Hence, our SDNN model is tested on data that the model has never seen during training. Each test data sample in the test sets (D2, D3, and D4) has the same format as the training data samples in D1 except that the test data sample does not have the two labels.

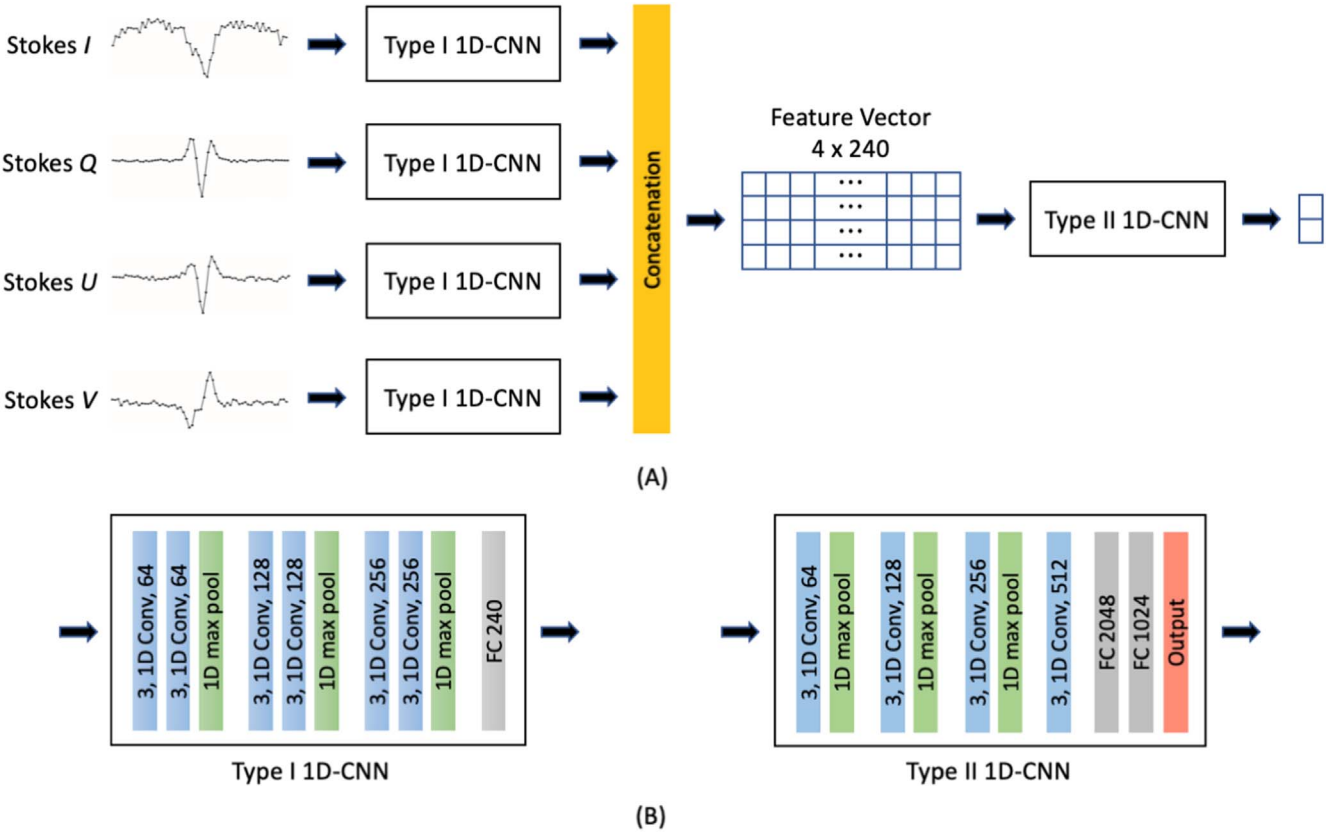

Figure 1. (A) Architecture of our SDNN model. The model contains two types of 1D convolution neural network (1D-CNN), named Type I 1D-CNN and Type II 1D-CNN, respectively. There are four Type I 1D-CNNs, which are stacked on top of a Type II 1D-CNN; hence the name SDNN (Stacked Deep Neural Networks) is used. The input of the SDNN is a sequence of four Stokes I, Q, U, and V components, where each component has 60 wavelength sampling points and the four Stokes components correspond to a pixel. Each Type I 1D-CNN takes as input a Stokes component, and encodes and produces as output a 240-dimensional feature vector. There are four Stokes components, so the four Type I 1D-CNNs output four 240-dimensional feature vectors, which are concatenated to form a 4×240 feature vector. The Type II 1D-CNN takes as input the 4×240 feature vector and produces as output two numbers representing the estimated Doppler shift and Doppler width of the input pixel. (B) Configuration details of the Type I 1D-CNN and Type II 1D-CNN used in our SDNN model.

Therefore, the length of the test data sample is $60 \times 4 = 240$. Because the values of the Stokes components vary, to facilitate machine learning (Liu et al. 2020), we normalize the Stokes Q, U, and V profiles by dividing them by 1000, as most of the Stokes measurements have values between -1000 and $+1000$. In addition, we normalize the Stokes I profile by dividing the measurements by 10,000, since the mean of the measurements is around 10,000. The Doppler shift values, which are obtained directly from the ME code and can be converted to the LOS velocities as we explain later, range from -0.5 to 0.5. We normalize the Doppler shift values by adding 0.5 to all the values, so that they range from 0 to 1. The Doppler width values calculated by the ME code already range from 0 to 1, and hence no normalization is done and the Doppler width values are used directly for model training.

3. Methodology

Figure 1 illustrates the architecture of our deep learning model (SDNN) used to infer LOS velocities and Doppler widths from Stokes profiles of GST/NIRIS. This model contains two types of 1D convolution neural network (1D-CNN; Kiranyaz et al. 2019). Type I 1D-CNN contains three convolution blocks followed by a fully connected (FC) layer with 240 neurons. Each convolution block contains two 1D convolution layers with kernels of size 3, activated by a ReLU (rectified linear unit) function, followed by a 1D max pooling layer with a kernel of size 2. Each 1D convolution layer in the first (second, and third, respectively) convolution block has 64 (128, and 256, respectively) kernels.

Type II 1D-CNN contains four convolution blocks followed by two fully connected layers activated by ReLU and having 2048 and 1024 neurons, respectively. Each of the first three convolution blocks contains a 1D convolution layer with a kernel of size 3 activated by ReLU, followed by a 1D max pooling layer with a kernel of size 2. The fourth convolution block only contains a 1D convolution layer with a kernel of size 3 activated by ReLU. The 1D convolution layer in the first (second, third, and fourth, respectively) convolution block has 64 (128, 256, and 512, respectively) kernels. The output layer has two neurons, activated by the linear function, $f(x) = ax$ (Goodfellow et al. 2016), which is suitable for predicting continuous numerical values and performs better than the Tanh function used in Liu et al. (2020).

The input of our SDNN model is a sequence of four Stokes I, Q, U, and V components where each component has 60 wavelength sampling points and the four Stokes components correspond to a pixel. There are four Type I 1D-CNNs, where each Type I 1D-CNN takes as input a Stokes component, and encodes and produces as output a 240-dimensional feature vector. There are four Stokes components, so the four Type I 1D-CNNs output four 240-dimensional feature vectors, which are concatenated to form a 4×240 feature vector. The Type II 1D-CNN takes as input the 4×240 feature vector and

produces as output two numbers representing the estimated Doppler shift and Doppler width of the input pixel.

The regression loss function used by our SDNN model is the L1 loss function (Goodfellow et al. 2016), defined as follows:

$$\text{L1 loss} = \frac{1}{N}\sum_{i=1}^{N}(|y_i^{ds} - \hat{y}_i^{ds}| + |y_i^{dw} - \hat{y}_i^{dw}|), \quad (1)$$

where N is the total number of pixels/data samples in a test set, y_i^{ds} (y_i^{dw}, respectively) is the Doppler shift (Doppler width, respectively) of the ith pixel calculated by the ME inversion code described in Section 2, and $\hat{y}_i^{ds}$ ($\hat{y}_i^{dw}$, respectively) is the Doppler shift (Doppler width, respectively) of the ith pixel predicted by our SDNN model. We use the L1 loss function here because it is more robust to outliers and hence makes the model more tolerant to noise in the training data (Willmott & Matsuura 2005).

We train the SDNN model using the Adam optimizer. The batch size is set to 1024, and the number of epochs is set to 40. During testing, the trained model takes as input the GST/NIRIS Stokes I, Q, U, and V profiles of each pixel in a test set, and predicts as output the Doppler shift and Doppler width of the pixel. The predicted values are denormalized so that they fall in the original (correct) range. We then convert the Doppler shift, denoted Δ_λ, to the LOS velocity, denoted $v_{\rm LOS}$, as follows:

$$v_{\rm LOS} = \frac{C \times \Delta_\lambda}{\lambda}, \quad (2)$$

where λ is the GST/NIRIS magnetogram wavelength, which is set to 1.56 μm, and C is the speed of light. The unit of $v_{\rm LOS}$ is km s^{-1}.

4. Results

4.1. Performance Metrics

For each test data sample, which corresponds to each pixel in a test set, we can use the proposed SDNN model to predict or infer its LOS velocity and Doppler width. In addition, we can also use the ME inversion code described in Section 2 to calculate its LOS velocity and Doppler width. We adopted four metrics to evaluate the performance of our SDNN model and compare it with related machine-learning algorithms. Specifically, we considered two quantities: LOS velocity and Doppler width. For each quantity, we compared its ME-calculated values with our SDNN-inferred values and computed the four performance metrics.

The first performance metric is the mean absolute error (MAE; Sammut & Webb 2010). MAE quantitatively assesses the difference between the ME-calculated and SDNN-inferred values for the test set (test image). The smaller the MAE is, the better performance an algorithm has. The second performance metric is the percent agreement (PA; McHugh 2012). Let y_i and $\hat{y}_i$ denote the ME-calculated and SDNN-inferred value respectively for the ith pixel in the test image. The ith pixel is an agreement pixel if $|y_i - \hat{y}_i|$ is smaller than a threshold. (The default threshold is set to 1 km s^{-1} for the LOS velocity and 0.1 Å for the Doppler width.) PA equals the number of agreement pixels divided by the total number of pixels in the test image multiplied by 100%. Thus, PA quantitatively assesses the similarity between the ME-calculated and SDNN-inferred values for the test image. The closer to 100% the PA is, the better performance an algorithm has. The third performance metric is R-squared (Moore et al. 2013; Miles 2014). R-squared is a statistical measure that determines the strength of the relationship between the ME-calculated and SDNN-inferred values for the test image. The values of R-squared range from $-\infty$ to $+1$. The closer to $+1$ R-squared is, the stronger the relationship between the ME-calculated and SDNN-inferred values is. The fourth performance metric is the Pearson product–moment correlation coefficient (PPMCC; Kirch 2008). PPMCC measures the linear correlation between the ME-calculated and SDNN-inferred values for the test image. The values of PPMCC range from -1 to $+1$. A PPMCC of $+1$ indicates a perfect positive correlation, while a PPMCC of -1 indicates a perfect negative correlation.

Table 1
Performance Metric Values of SDNN Based on Two Training Sets and Test Images from Three ARs

		LOS Velocity		Doppler Width	
Test Image	Metric	D1	D1 ∪ D5	D1	D1 ∪ D5
D2	MAE	0.251	0.260	0.048	0.047
2015 Jun 25	PA	97.8%	97.5%	89.1%	90.3%
16:55:13 UT	R-squared	0.796	0.786	0.317	0.458
(AR 12371)	PPMCC	0.915	0.912	0.780	0.787
D3	MAE	0.289	0.302	0.039	0.040
2017 Jul 13	PA	97.6%	97.0%	93.2%	93.2%
16:20:12 UT	R-squared	0.768	0.728	0.224	0.278
(AR 12665)	PPMCC	0.912	0.896	0.716	0.723
D4	MAE	0.952	0.316	0.064	0.034
2017 Sep 6	PA	51.0%	94.5%	82.3%	95.3%
19:17:53 UT	R-squared	−0.402	0.749	−1.334	0.208
(AR 12673)	PPMCC	0.804	0.883	0.516	0.711

4.2. Impact of Training Data on the Performance of the SDNN Method

Table 1 presents experimental results of using D1 as the training set to train SDNN and using D2, D3, and D4 as test sets to test SDNN as described in Section 2. SDNN works well on D2 and D3. However, the performance of SDNN degrades on D4, which contains pixels from AR 12673. We note that AR 12673 is the most flare-productive active region in solar cycle 24 (Wang et al. 2018). The training set D1 only contains pixels from a normal active region (AR 12371); the SDNN model trained by D1 does not have sufficient knowledge about the extremely flare-productive active region AR 12673. To assess and quantify the impact of training data on the performance of SDNN, we additionally selected a 720×720 image with 518,400 pixels (data samples) from AR 12673 collected at 16:18:41 UT on 2017 September 6. We referred to this additional data set as D5. Thus, the image of D5 was taken approximately 3 hr before the image of D4. We then combined the 940,500 pixels in D1 and the 518,400 pixels in D5 to get a new training set, denoted D1 ∪ D5. D1 ∪ D5 contains 1,458,900 training data samples (pixels) in total. The results obtained by using D1 ∪ D5 as the training set to train SDNN and using D2, D3, and D4 as test sets to test SDNN are also presented in Table 1. Notice that, due to the time difference,

D1 ∪ D5 and D4 are disjoint even though the data samples in D5 and D4 are from the same AR 12673.

We see from Table 1 that the SDNN model trained by D1 ∪ D5 outperforms the SDNN model trained by D1 when the two models are tested on D4. This happens because the SDNN model trained by D1 ∪ D5 acquires more knowledge concerning D4 than the SDNN model trained by D1, due to the fact that D4 and D5 are from the same AR 12673 as indicated above. On the other hand, the two models have similar performance when tested on D2 and D3. Notice that D1, D2, and D3 are from normal active regions (AR 12371 and AR 12665) while D4 contains special pixels from an extremely flare-productive active region (AR 12673). The special pixels (data samples) in D4 do not occur in D1, D2, and D3. Thus, the model trained by D1 lacks knowledge of the special pixels, and hence does not work well on D4. On the other hand, like D4, D5 also contains special pixels from the extremely flare-productive AR 12673. As a consequence, the model trained by D1 ∪ D5 performs well on D4. These results indicate that, when dealing with normal ARs, the training data samples in D1 are sufficient to produce good results. On the other hand, when dealing with special ARs such as AR 12673, the training set must be expanded to include data samples from the special ARs so that our SDNN model can acquire sufficient knowledge about the special ARs to produce good results. In view of the experimental results, we used D1 ∪ D5 as the training set in subsequent experiments.

4.3. Comparison between the SDNN and ME Methods

Here, we compare the LOS velocity maps and Doppler width maps produced by the SDNN and ME methods. Figures 2 (3, and 4, respectively) present the LOS velocity maps (top panels) and Doppler width maps (bottom panels) produced by the two methods based on the test sets/images D2 (D3, and D4, respectively) with data samples/pixels from AR 12371 (AR 12665, and AR 12673, respectively) collected on 2015 June 25 16:55:13 UT (2017 July 13 16:20:12 UT, and 2017 September 6 19:17:53 UT, respectively). The first columns in the figures show scatter plots, the second columns show the maps produced by the ME inversion code described in Section 2, and the third columns show the maps produced by the SDNN model. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values. We see from Figures 2, 3, and 4 that the maps produced by the ME code and SDNN model are highly correlated. The high correlation can be seen particularly from the scatter plots of the LOS velocity maps, and the corresponding PPMCC values of ∼0.9 as shown in Table 1. Since the training labels for the SDNN model are produced by the ME inversion code described in Section 2, these results demonstrate the good learning capability of SDNN. The Doppler width inferences, as seen from the figures, show larger scatter. This happens because of the degeneracy of the Doppler width with the rest of the ME thermodynamic parameters as described in Orozco Suárez & del Toro Iniesta (2007). Notice also that the maps produced by SDNN are smoother and cleaner than those produced by the ME code. We see many salt-and-pepper noise pixels in the maps of the ME code. The many noise pixels from the ME code are also reflected in the scatter plots in Figures 2, 3, and 4. For example, refer to the scatter plot of the LOS velocity maps in Figure 3, where there is a vertical line on which pixels have an LOS velocity of zero. Many of the pixels on the vertical line are noisy ones in the LOS velocity map produced by the ME code.

It is worth noting that the LOS velocity maps in Figure 3 contain granular patterns and a portion of a sunspot penumbra. For example, there are granular patterns located in the region whose E–W coordinates are between −460″ and −450″ and N–S coordinates are between 190″ and 200″. A partial sunspot penumbra is located in the region whose E–W coordinates are between −450″ and −440″ and N–S coordinates are between 150″ and 160″. Figure 5 presents an enlarged view of these two regions. Both ME and SDNN produce granular patterns with a PPMCC of 0.982 and a partial sunspot penumbra with a PPMCC of 0.893. The similarity between the local maps produced by the two methods is also reflected in the scatter plots in Figure 5. In addition, we see from Figure 5 that the local maps produced by the SDNN model are smoother and cleaner than those produced by the ME code. There are many salt-and-pepper noise pixels in the maps of the ME code, in particular in the region of the partial sunspot penumbra.

4.4. Comparison with Related Machine-learning Methods

Here, we compare SDNN with three related ML methods, including MSVR (multiple support vector regression; Rees et al. 2004), MLP (multilayer perceptrons; Carroll & Kopf 2008), and PCNN (pixel-level convolutional neural network; Liu et al. 2020), all of which have previously been used in Stokes inversion as surveyed in Section 1. The MSVR method uses the radial basis function kernel where the regularization parameter is set to 1 and the epsilon is set to 0.2. The MLP model consists of an input layer, an output layer, and two simple hidden layers, each of which has 1024 neurons. The PCNN model, which was originally designed for inferring vector magnetic fields, is modified to output the LOS velocity and Doppler width, though the same model architecture and hyperparameter setting are used here.

Figure 6 compares the MAE, PA, R-squared, and PPMCC values of the four ML methods based on the test image/data set D2 (D3, and D4, respectively) from AR 12371 (AR 12665, and AR 12673, respectively) collected on 2015 June 25 16:55:13 UT (2017 July 13 16:20:12 UT, and 2017 September 6 19:17:53 UT, respectively) where training data were taken from D1 ∪ D5. It can be seen from Figure 6 that SDNN outperforms the other three ML methods in terms of all the four performance metrics. When compared to the most closely related PCNN method, SDNN achieves an MAE of 0.260 (0.302, and 0.316, respectively) while PCNN achieves an MAE of 0.455 (0.537, and 0.542, respectively), showing an improvement of 42.9% (43.8%, and 41.7%, respectively) by SDNN, on D2 (D3, and D4, respectively) in producing the LOS velocity maps for the three data sets. Furthermore, in producing these LOS velocity maps, SDNN achieves a PPMCC of 0.912 (0.896, and 0.883, respectively) while PCNN achieves a PPMCC of 0.849 (0.676, and 0.832, respectively) on D2 (D3, and D4, respectively); SDNN beats PCNN by 7.4%, 32.5%, and 6%, respectively, on the three data sets. These results indicate that our SDNN model is more robust and has a better generalization and inference capability than the closely related PCNN model. It is worth pointing out that the granular patterns in the LOS velocity map of the SDNN method shown in Figure 5, which exhibit important kinematic information on the photospheric surface (Jaeggli & Norton 2016), are missing from the LOS velocity maps produced by the other three ML methods, another sign indicating the superiority of the proposed SDNN model.

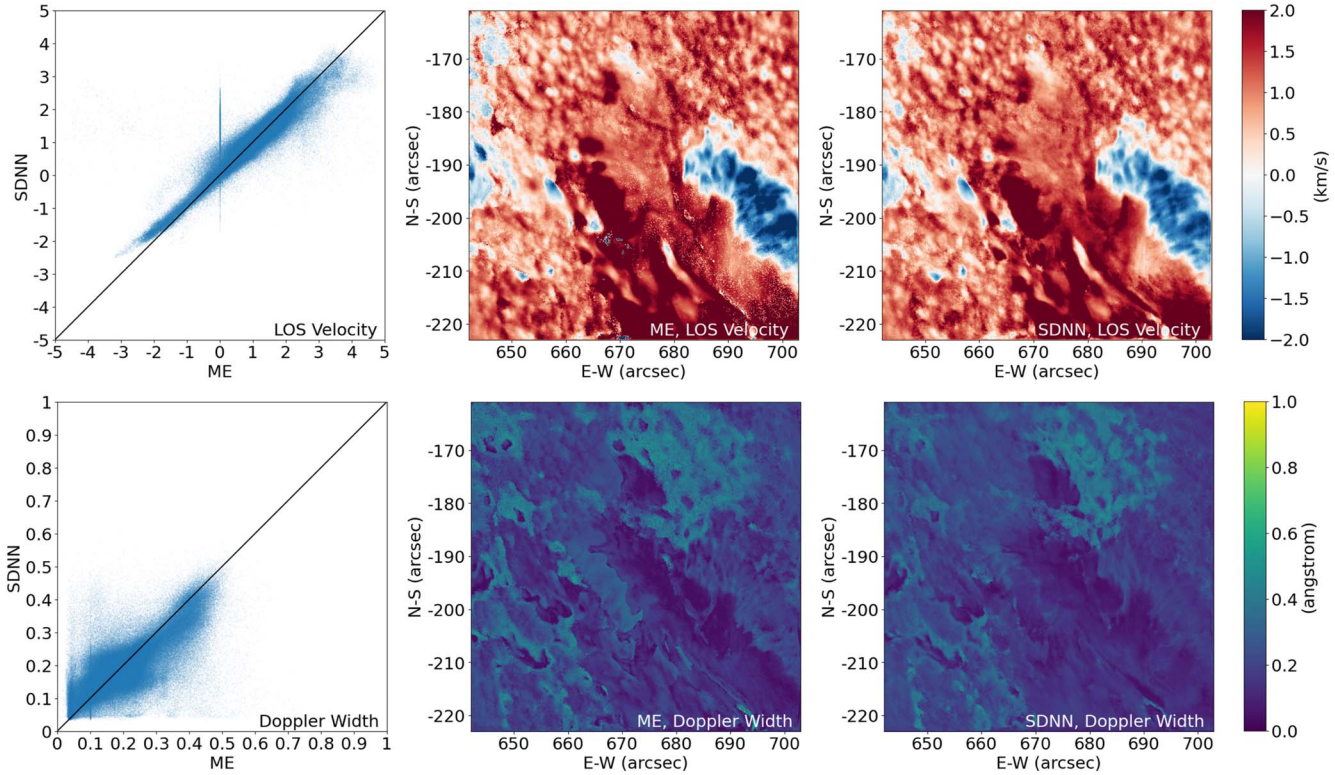

Figure 2. Comparison between the ME inversion code and SDNN for producing the LOS velocity maps (top panels) and Doppler width maps (bottom panels) based on the test image/data set D2 from AR 12371 collected on 2015 June 25 16:55:13 UT, where training data were taken from D1 ∪ D5. The first column shows scatter plots where the *X*-axis and *Y*-axis represent the values obtained by the ME code and SDNN, respectively. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values. The second and third columns show the maps produced by the ME code and SDNN, respectively.

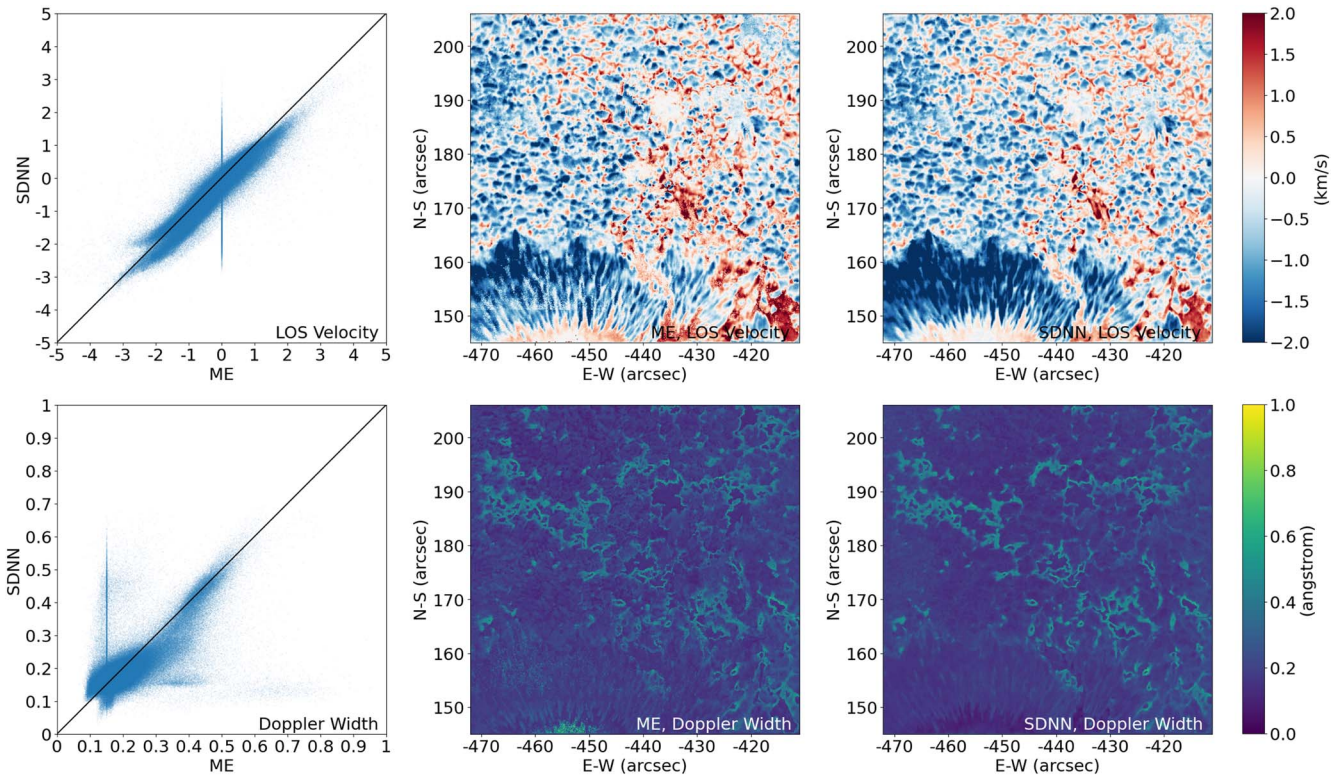

Figure 3. Comparison between the ME inversion code and SDNN for producing the LOS velocity maps (top panels) and Doppler width maps (bottom panels) based on the test image/data set D3 from AR 12665 collected on 2017 July 13 16:20:12 UT, where training data were taken from D1 ∪ D5. The first column shows scatter plots where the *X*-axis and *Y*-axis represent the values obtained by the ME code and SDNN, respectively. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values. The second and third columns show the maps produced by the ME code and SDNN, respectively.

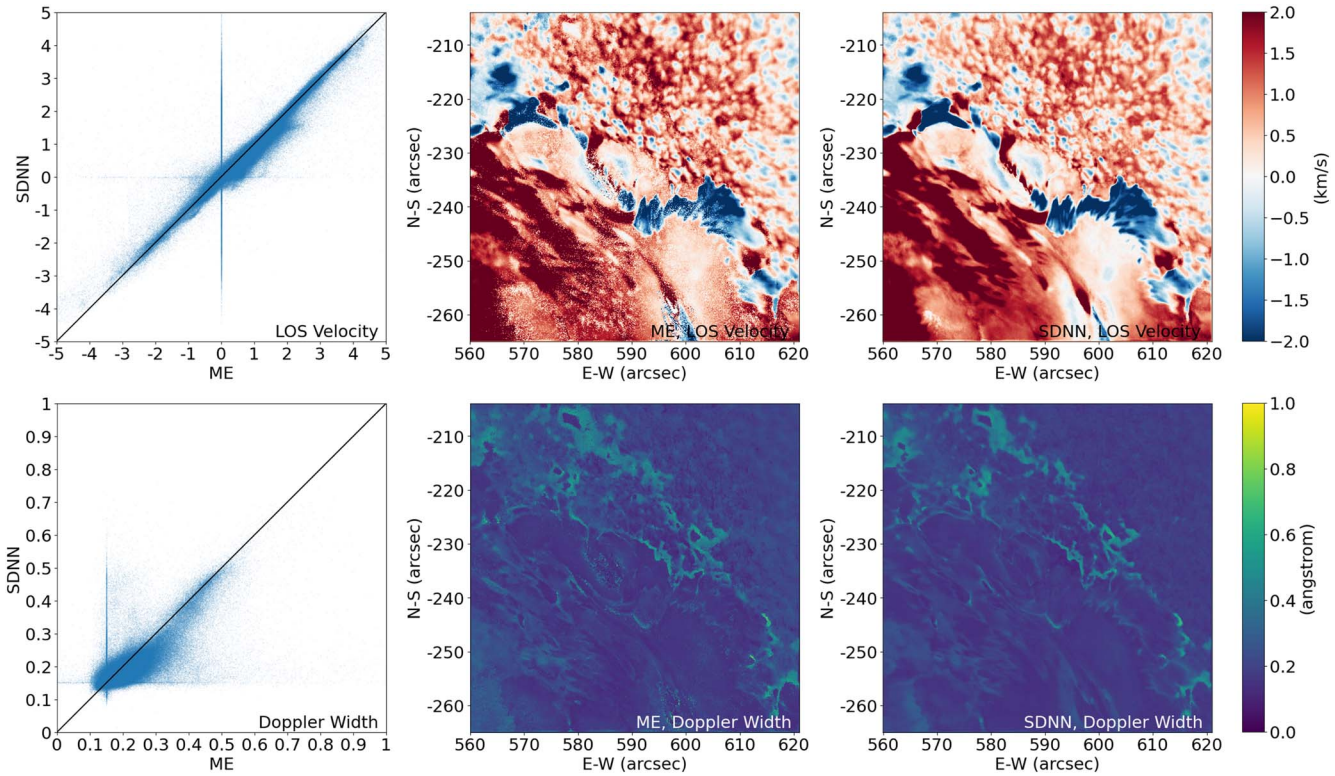

Figure 4. Comparison between the ME inversion code and SDNN for producing the LOS velocity maps (top panels) and Doppler width maps (bottom panels) based on the test image/data set D4 from AR 12673 collected on 2017 September 6 19:17:53 UT, where training data were taken from D1 ∪ D5. The first column shows scatter plots where the *X*-axis and *Y*-axis represent the values obtained by the ME code and SDNN, respectively. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values. The second and third columns show the maps produced by the ME code and SDNN, respectively.

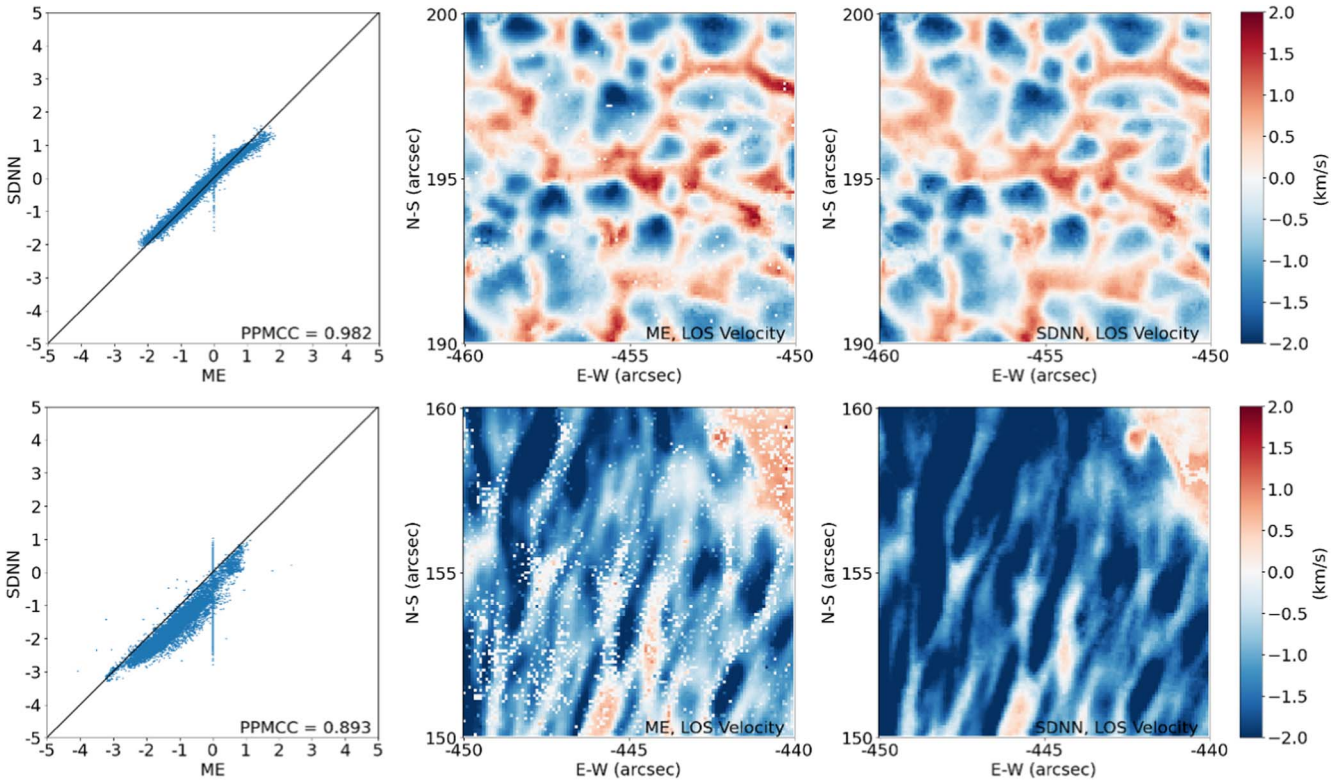

Figure 5. Comparison between the ME inversion code and SDNN for producing the local LOS velocity maps containing granular patterns (top panels) and a portion of a sunspot penumbra (bottom panels) based on the test image/data set D3 from AR 12665 collected on 2017 July 13 16:20:12 UT, where training data were taken from D1 ∪ D5. The first column shows scatter plots where the *X*-axis and *Y*-axis represent the values obtained by the ME code and SDNN, respectively. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values. The second and third columns show the local LOS velocity maps produced by the ME code and SDNN, respectively.

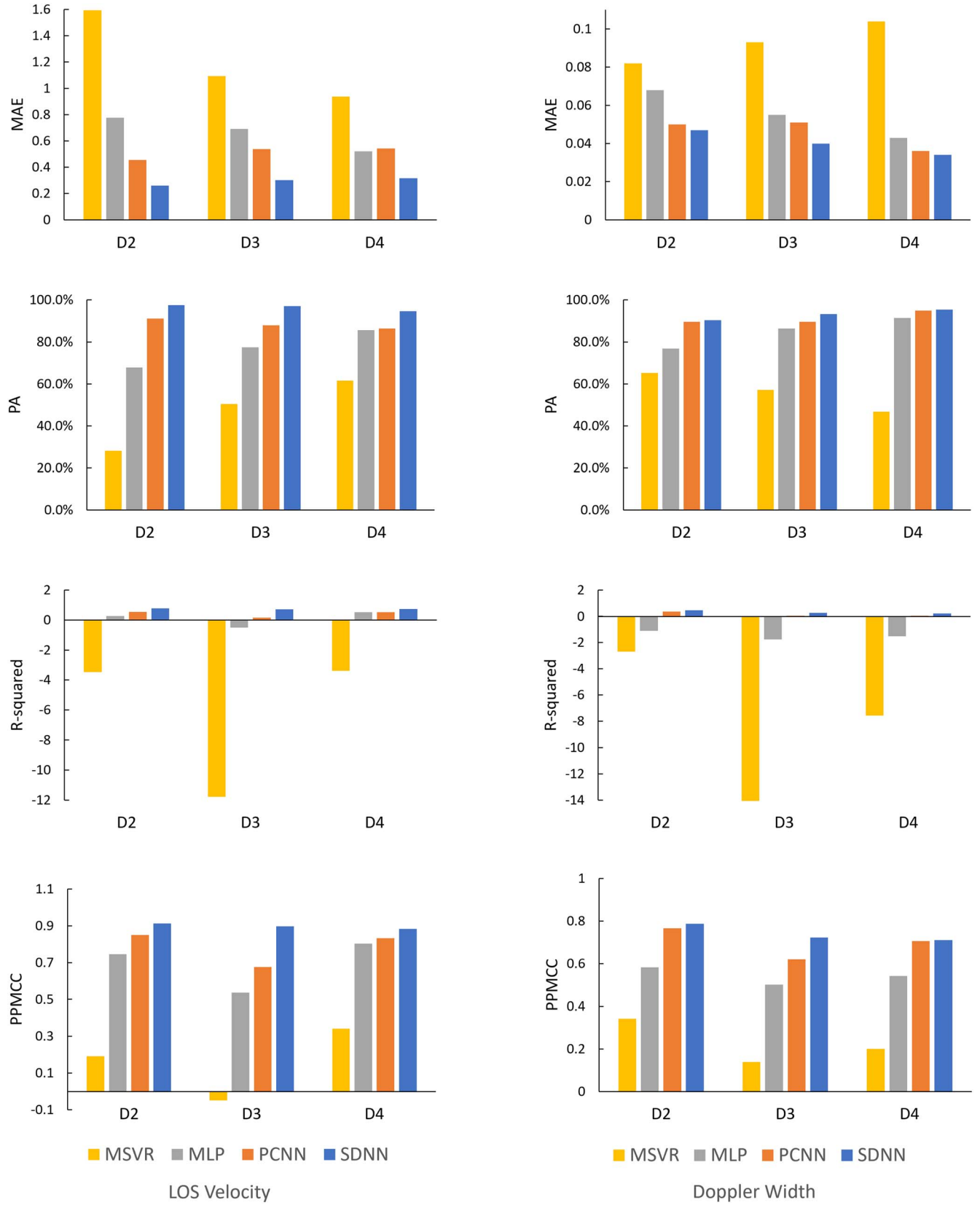

Figure 6. Comparison of MSVR, MLP, PCNN, and SDNN based on the test image/data set D2 (D3, and D4, respectively) from AR 12371 (AR 12665, and AR 12673, respectively) collected on 2015 June 25 16:55:13 UT (2017 July 13 16:20:12 UT, and 2017 September 6 19:17:53 UT, respectively) where training data were taken from D1 ∪ D5. Left column: performance metric values, displayed by bar charts, that are obtained by the four machine learning (ML) methods in predicting LOS velocities. Right column: performance metric values obtained by the four ML methods in predicting Doppler widths.

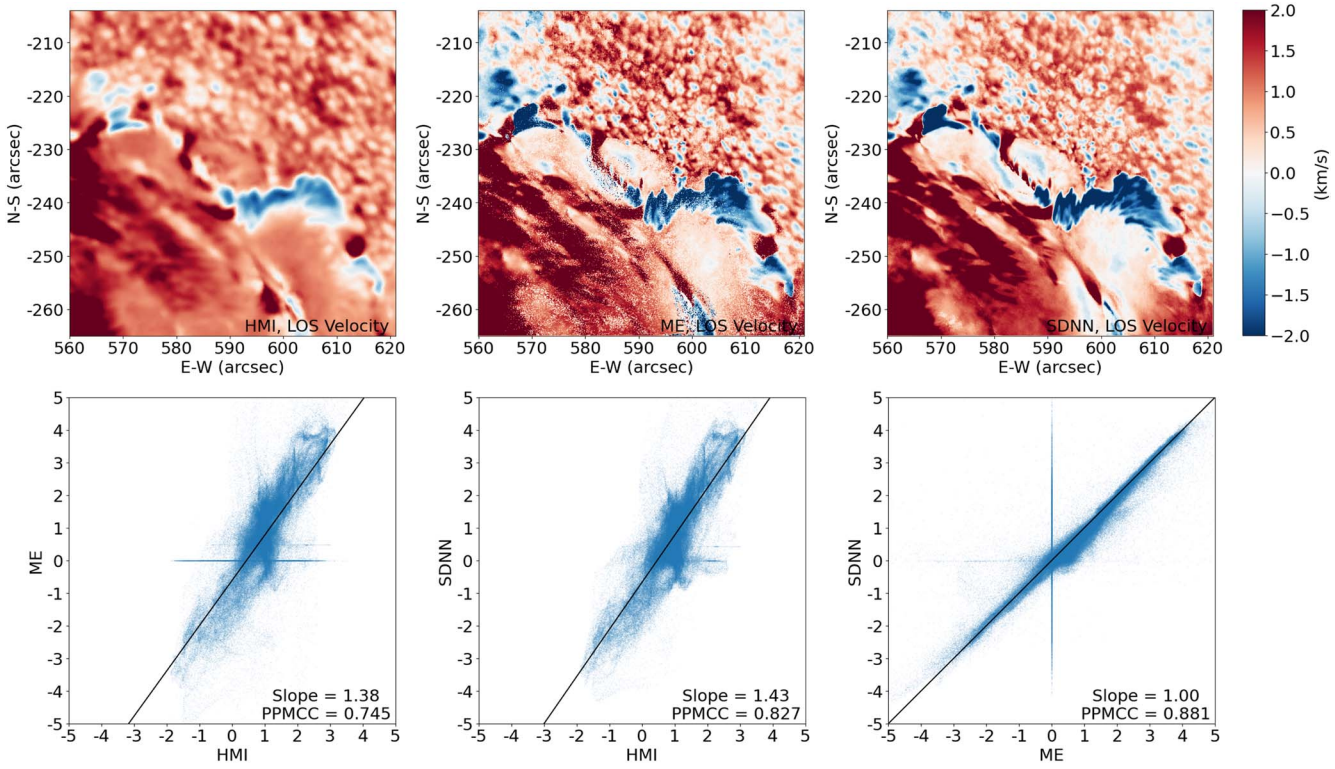

Figure 7. Comparison between the inversion results of SDO/HMI and GST/NIRIS in AR 12673. Top panels (from left to right): the LOS velocity map from the HMI Dopplergram collected on 2017 September 6 19:00:00 UT, the LOS velocity map produced by the ME inversion code, and the LOS velocity map produced by SDNN with training data from D1 ∪ D5, where the ME code and SDNN were applied to the test image/data set D6 collected by GST/NIRIS on 2017 September 6 19:01:48 UT. Bottom panels (from left to right): the scatter plot between ME and HMI, the scatter plot between SDNN and HMI, and the scatter plot between SDNN and ME.

4.5. Comparison between the Inversion Results of SDO/HMI and GST/NIRIS

Thus far, we have focused on Stokes inversion for GST/NIRIS. There are inversion results from the Helioseismic and Magnetic Imager (HMI; Scherrer et al. 2012) on board the Solar Dynamics Observatory (SDO; Pesnell et al. 2012). Here, we compare the inversion results of the spaceborne observatory (SDO/HMI) and ground-based observatory (BBSO/GST). We selected an HMI Dopplergram from AR 12673 collected on 2017 September 6 19:00:00 UT, and the temporally closest test image/data set, denoted D6, from the same AR 12673 collected by GST/NIRIS on 2017 September 6 19:01:48 UT. We aligned the HMI and GST/NIRIS images, and respectively applied our SDNN model trained by the data set D1 ∪ D5 and the ME inversion code described in Section 2 to D6.

Figure 7 presents the LOS velocity map from the HMI Dopplergram and the LOS velocity maps produced by ME and SDNN (top panels), and shows the corresponding scatter plots (bottom panels). It can be seen from the top panels that the maps of GST/NIRIS obtained by ME and SDNN are much clearer than the map from HMI. This happens due to the higher-resolution imaging data offered by GST/NIRIS. On the other hand, the maps from HMI and SDNN are smoother than the map of ME, which contains salt-and-pepper noise pixels. Furthermore, we see from the left and middle scatter plots in Figure 7 that the map produced by SDNN is closer to the map from HMI, with a PPMCC of 0.827, than the map produced by ME, which has a PPMCC of 0.745. The black regression lines in the scatter plots have a slope of ∼1.4, possibly caused by the offset in calibration, which occurs due to the intrinsic difference between the two instruments (i.e., HMI and GST/NIRIS). These results further demonstrate that machine learning can complement the ME approach. Specifically, our SDNN model is trained with the labels (i.e., LOS velocities and Doppler widths) produced by the ME inversion code described in Section 2. With its generalization and inference capabilities, the ML-based SDNN model can produce better inversion results than the calculation-based ME code.

4.6. Extending the SDNN Method to Infer Vector Magnetic Fields

Finally, we extend our SDNN model to infer vector magnetic fields from Stokes profiles of GST/NIRIS. We normalized the total magnetic field strength ($B_{\rm total}$), inclination angle, and azimuth angle as done in Liu et al. (2020). The SDNN model was trained by the data in D1 ∪ D5. The training labels (i.e., total magnetic field strengths, inclination angles, and azimuth angles) were produced by the ME inversion code described in Section 2. Figure 8 presents inversion results obtained from the ME code and SDNN, respectively, based on the test set (image) D3 with data samples (pixels) from AR 12665 collected on 2017 July 13 16:20:12 UT. (The inversion results for the test sets D2 and D4 are similar and omitted here.) The first column in Figure 8 shows scatter plots, the second column shows the magnetic field maps produced by the ME code, and the third column shows the magnetic field maps produced by SDNN. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values.

It can be seen from Figure 8 that the magnetic field maps produced by the ME code and SDNN are highly correlated, with PPMCC being 0.980 (0.867, and 0.866, respectively) for the total magnetic field strength (inclination angle, and

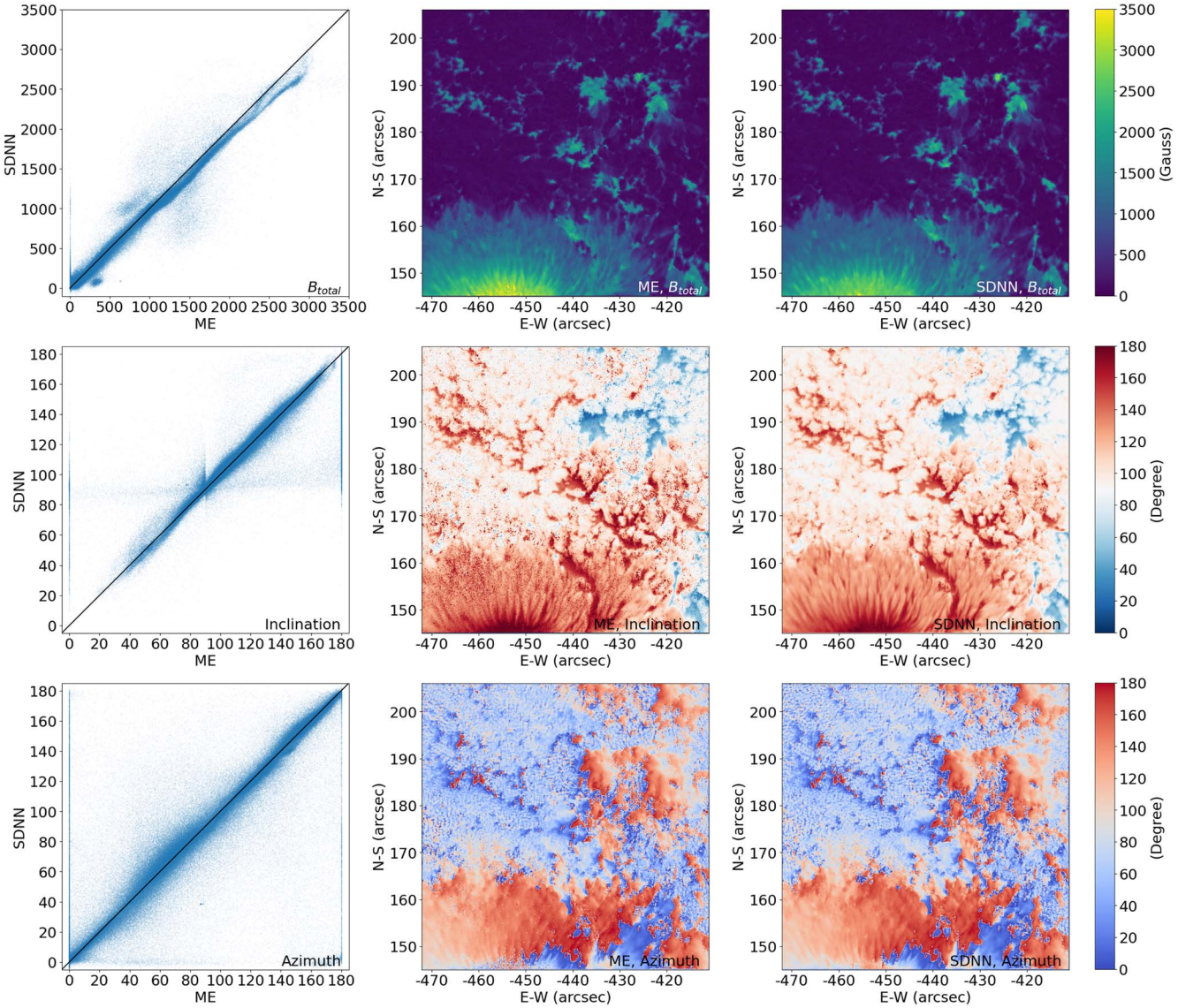

Figure 8. Comparison between the ME inversion code and SDNN for producing the total magnetic field strengths (top panels), inclination angles (middle panels), and azimuth angles (bottom panels) of pixels based on the test image D3 from AR 12665 collected on 2017 July 13 16:20:12 UT, where training data were taken from D1 ∪ D5. The first column shows scatter plots where the *X*-axis and *Y*-axis represent the values obtained by the ME code and SDNN, respectively. Pixels on the diagonal line in a scatter plot have identical ME-calculated and SDNN-inferred values. The second and third columns show the magnetic field maps produced by the ME code and SDNN, respectively.

azimuth angle, respectively). Like LOS velocity maps and Doppler width maps, the magnetic field maps produced by SDNN are smoother and cleaner than those produced by the ME code. We see many salt-and-pepper noise pixels in the magnetic field maps produced by the ME code. When compared to the closely related PCNN method developed by Liu et al. (2020), SDNN achieves an MAE of 63.715 (5.721, and 11.402, respectively) while PCNN achieves an MAE of 79.089 (6.270, and 12.785, respectively), showing an improvement of 19.4% (8.8%, and 10.8%, respectively) by SDNN, for the total magnetic field strength (inclination angle, and azimuth angle, respectively). Furthermore, SDNN outperforms PCNN in PPMCC, with SDNN's PPMCC being 0.980 (0.867, and 0.866, respectively) compared to PCNN's PPMCC of 0.976 (0.860, and 0.848, respectively) for the total magnetic field strength (inclination angle, and azimuth angle, respectively).

5. Discussion and Conclusions

We develop a deep learning model (SDNN) to infer LOS velocities and Doppler widths from Stokes profiles of GST/NIRIS at BBSO. The labels for training SDNN are prepared by a Milne–Eddington (ME) inversion code used by BBSO. We compare the LOS velocity and Doppler width maps produced by SDNN with those from the ME inversion code and related ML algorithms, including MSVR, MLP, and a PCNN. We next compare the inversion results of the ME code and SDNN based on GST/NIRIS with those from SDO/HMI in flare-prolific AR 12673. Finally, we extend SDNN to infer vector magnetic fields from Stokes profiles of GST/NIRIS.

Our main results are summarized as follows.

1. For the test sets from GST/NIRIS considered in the paper, SDNN produces smoother and cleaner LOS velocity and Doppler width maps than the ME inversion

code. The same conclusion is obtained when comparing the inversion results of GST/NIRIS and SDO/HMI. Furthermore, SDNN performs Stokes inversion through making predictions, and hence is faster than the computation-based ME code. It takes ∼75 s for SDNN to process a test image here, which is approximately five times faster than the ME code.

2. The SDNN-inferred LOS velocities are highly correlated to the ME-calculated ones, with PPMCC being close to 0.9 on average. The LOS velocity and Doppler width maps produced by SDNN are closer to the ME maps than those from the related ML algorithms (MSVR, MLP, and PCNN). Furthermore, both ME and SDNN are able to infer granular patterns in LOS velocity maps. These patterns exhibit important kinematic information on the photospheric surface, which are missing from the LOS velocity maps produced by the related ML algorithms. These results demonstrate that the learning capability of SDNN is better than those of the related ML algorithms.
3. Training data has a significant impact on the performance of SDNN. When SDNN is trained by data from a normal active region (AR 12371), it performs well on normal active regions such as AR 12665, but suffers on AR 12673, which is the most flare-productive active region in solar cycle 24 (Wang et al. 2018). To acquire knowledge concerning the special pixels in the extremely flare-productive active region AR 12673, we have to include some of the special pixels/data samples in the training set. This finding is consistent with the guidelines suggested in the ML literature (Bishop 2006), where an iterative training process with increasing training data is often used to improve the performance of an ML model.
4. Our work focuses on LOS velocity and Doppler width inference. The input of our SDNN method consists of Stokes I, Q, U, and V profiles of GST/NIRIS. One wonders whether using Stokes I suffices to infer the LOS velocity and Doppler width. We have conducted additional experiments in which SDNN is merely trained by Stokes I of pixels in D1 ∪ D5. Our experimental results showed that using the four Stokes I, Q, U, and V profiles performs better than using Stokes I only. Specifically, with only Stokes I as the input, SDNN obtains an MAE of 0.774 (0.395, and 0.534, respectively) and PPMCC of 0.412 (0.782, and 0.787, respectively) for LOS velocity inference, compared to the MAE of 0.260 (0.302, and 0.316, respectively) and PPMCC of 0.912 (0.896, and 0.883, respectively) obtained by the four Stokes profiles, showing a deterioration of 197.7% (30.8%, and 69.0%, respectively) on MAE and 54.8% (12.7%, and 10.9%, respectively) on PPMCC in D2 (D3, and D4, respectively). Similarly, with only Stokes I as the input, SDNN obtains an MAE of 0.093 (0.059, and 0.047, respectively) and PPMCC of 0.010 (0.343, and 0.441, respectively) for Doppler width inference, compared to the MAE of 0.047 (0.040, and 0.034, respectively) and PPMCC of 0.787 (0.723, and 0.711, respectively) obtained by the four Stokes profiles, showing a deterioration of 97.9% (47.5%, and 38.2%, respectively) on MAE and 98.7% (52.6%, and 38.0%, respectively) on PPMCC in D2 (D3, and D4, respectively). These results indicate that one should use all four of the Stokes profiles of GST/NIRIS for LOS velocity and Doppler width inference.
5. SDNN performs equally well for vector magnetic field inference. For example, with the four Stokes I, Q, U, and V as the input, D1 ∪ D5 as the training set, and D3 as the test set, SDNN achieves an MAE of 63.715 (5.721, and 11.402, respectively) while the most closely related PCNN method developed by Liu et al. (2020) achieves an MAE of 79.089 (6.270, and 12.785, respectively), showing an improvement of 19.4% (8.8%, and 10.8%, respectively) by SDNN, for the total magnetic field strength (inclination angle, and azimuth angle, respectively). Furthermore, SDNN's PPMCC values are 0.980 (0.867, and 0.866, respectively), which are better than PCNN's PPMCC values of 0.976 (0.860, and 0.848, respectively) for the total magnetic field strength (inclination angle, and azimuth angle, respectively). Similar results are obtained for test sets D2 and D4.

Based on these results, we conclude that SDNN is an effective and alternative method for inferring LOS velocities, Doppler widths, and vector magnetic fields from Stokes profiles of GST/NIRIS. It is hoped that SDNN will be a useful tool in producing the high-quality velocity and magnetic fields that are crucial for understanding the evolution of physical properties of the solar atmosphere.

The authors gratefully acknowledge the use of data from the Goode Solar Telescope (GST) of the Big Bear Solar Observatory (BBSO). The BBSO operation is supported by NJIT and U.S. NSF grant AGS-1821294. The GST operation is partly supported by the Korea Astronomy and Space Science Institute and Seoul National University. This work was supported by U.S. NSF grants AGS-1927578, AGS-1954737, and AGS-2228996. K.A. and W.C. acknowledge the support of NASA under grant 80NSSC20K0025. Q.L., Y.X., and H.W. acknowledge the support of NASA under grants 80NSSC 18K1705, 80NSSC19K0068, and 80NSSC20K1282. We thank the scientific editor for his guidance and an anonymous referee for the thoughtful comments that have helped us improve the presentation and content of this paper.

Facilities: Big Bear Solar Observatory, Solar Dynamics Observatory.

ORCID iDs

Haodi Jiang https://orcid.org/0000-0001-6460-408X
Wenda Cao https://orcid.org/0000-0003-2427-6047
Jason T. L. Wang https://orcid.org/0000-0002-2486-1097
Haimin Wang https://orcid.org/0000-0002-5233-565X

References

Ahn, K., & Cao, W. 2017, AAS Meeting, 48, 115.04
Ahn, K., & Cao, W. 2018, in 2018 Triennial Earth-Sun Summit (TESS), ed. D. Brousseau, 308.118A
Ahn, K., & Cao, W. 2019, in ASP Conf. Ser. 526, Solar Polariation Workshop 8, ed. L. Belluzzi et al. (San Francisco, CA: ASP), 317
Asensio Ramos, A., & Díaz Baso, C. J. 2019, A&A, 626, A102
Asensio Ramos, A., Trujillo Bueno, J., & Landi Degl'Innocenti, E. 2008, ApJ, 683, 542
Auer, L. H., Heasley, J. N., & House, L. L. 1977, SoPh, 55, 47
Bishop, C. M. 2006, Pattern Recognition and Machine Learning (Information Science and Statistics) (Berlin: Springer-Verlag), https://dl.acm.org/doi/book/10.5555/1162264
Bobra, M. G., & Couvidat, S. 2015, ApJ, 798, 135
Bommier, V. 2016, JGRA, 121, 5025

Borrero, J. M., Lites, B. W., Lagg, A., Rezaei, R., & Rempel, M. 2014, A&A, 572, A54
Borrero, J. M., Tomczyk, S., Kubo, M., et al. 2011, SoPh, 273, 267
Borrero, J. M., Tomczyk, S., Norton, A., et al. 2007, SoPh, 240, 177
Cao, W., Goode, P. R., Ahn, K., et al. 2012, in ASP Conf. Ser. 463, Second ATST-EAST Meeting: Magnetic Fields from the Photosphere to the Corona, ed. T. R. Rimmele et al. (San Francisco, CA: ASP), 291
Cao, W., Gorceix, N., Coulter, R., et al. 2010, AN, 331, 636
Carroll, T. A., & Kopf, M. 2008, A&A, 481, L37
Chae, J., & Park, S. 2009, in ASP Conf. Ser. 415, The Second Hinode Science Meeting: Beyond Discovery-Toward Understanding, ed. B. Lites et al., 101
Şahin, S., Yurchyshyn, V., Kumar, P., et al. 2019, ApJ, 873, 75
del Toro Iniesta, J. C., & Ruiz Cobo, B. 2016, LRSP, 13, 4
Frutiger, C., Solanki, S. K., Fligge, M., & Bruls, J. H. M. J. 2000, A&A, 358, 1109
Gafeira, R., Orozco Suárez, D., Milić, I., et al. 2021, A&A, 651, A31
Goode, P. R., & Cao, W. 2012, Proc. SPIE, 8444, 844403
Goode, P. R., Yurchyshyn, V., Cao, W., et al. 2010, ApJL, 714, L31
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (Cambridge, MA: MIT Press), http://www.deeplearningbook.org
Jaeggli, S. A., & Norton, A. A. 2016, ApJ, 820, L11
Keys, P. H., Steiner, O., & Vigeesh, G. 2021, RSPTA, 379, 20200182
Kiranyaz, S., Ince, T., Abdeljaber, O., Avci, O., & Gabbouj, M. 2019, in ICASSP 2019-2019, IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP) (Piscataway, NJ: IEEE), 8360
Kirch, W. (ed.) 2008, Pearson's Correlation Coefficient (Dordrecht: Springer), 1090
Lagg, A., Woch, J., Krupp, N., & Solanki, S. K. 2004, A&A, 414, 1109
Landi Degl'Innocenti, E. 1982, MmSAI, 53, 841
Landi Degl'Innocenti, E. 1983, SoPh, 85, 3
Landi Degl'Innocenti, E. 1992, in Solar Observations: Techniques and Interpretation, ed. F. Sanchez, M. Collados, & M. Vazquez (Cambridge: Cambridge Univ. Press), 71
Landi Degl'Innocenti, E., & Landolfi, M. 2004, Polarization in Spectral Lines, Vol. 307 (Dordrecht: Springer)
Landolfi, M., & Landi Degl'Innocenti, E. 1982, SoPh, 78, 355
LeCun, Y., Bengio, Y., & Hinton, G. 2015, Natur, 521, 436
Lites, B., Casini, R., Garcia, J., & Socas-Navarro, H. 2007, MmSAI, 78, 148
Liu, C., Cao, W., Chae, J., et al. 2018, ApJ, 869, 21
Liu, H., Xu, Y., Wang, J., et al. 2020, ApJ, 894, 70
Maurya, R. A., & Ambastha, A. 2010, in ASSP Conf. Proc. 19, Magnetic Coupling between the Interior and Atmosphere of the Sun, ed. S. S. Hasan & R. J. Rutten (Berlin: Springer), 517
McHugh, M. 2012, Biochem. Med., 22, 276
Milić, I., & Gafeira, R. 2020, A&A, 644, A129
Miles, J. 2014, R Squared, Adjusted R Squared (New York: Wiley)
Moore, D., Notz, W., & Fligner, M. 2013, The Basic Practice of Statistics (New York: W.H. Freeman and Company), https://www.worldcat.org/title/basic-practice-of-statistics/oclc/733238099
Nejezchleba, T. 1998, A&AS, 127, 607
Orozco Suárez, D., Bellot Rubio, L. R., del Toro Iniesta, J. C., et al. 2007, ApJL, 670, L61
Orozco Suárez, D., & del Toro Iniesta, J. C. 2007, A&A, 462, 1137
Ortiz, A., Rubio, L. R. B., Hansteen, V. H., de la Cruz Rodríguez, J., & van der Voort, L. R. 2014, ApJ, 781, 126
Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. 2012, SoPh, 275, 3
Rachkovsky, D. N. 1962, IzKry, 28, 259
Rachkovsky, D. N. 1963, IzKry, 30, 267
Rees, D., Guo, Y., Ariste, A. L., & Graham, J. 2004, 8th Int. Conf. 3215, Knowledge-Based Intelligent Information and Engineering Systems (Berlin: Springer), 388
Ruiz Cobo, B., & del Toro Iniesta, J. C. 1992, ApJ, 398, 375
Sammut, C., & Webb, G. I. (ed.) 2010, Encyclopedia of Machine Learning (Boston, MA: Springer), 652
Scherrer, P. H., Schou, J., Bush, R. I., et al. 2012, SoPh, 275, 207
Socas-Navarro, H., de la Cruz Rodríguez, J., Asensio Ramos, A., Trujillo Bueno, J., & Ruiz Cobo, B. 2015, A&A, 577, A7
Teng, F. 2015, SoPh, 290, 2693
Teng, F., & Deng, Y.-Y. 2014, RAA, 14, 1469
Unno, W. 1956, PASJ, 8, 108
Varsik, J., Plymate, C., Goode, P., et al. 2014, Proc. SPIE, 9147, 91475D
Wang, H., Cao, W., Liu, C., et al. 2015, NatCo, 6, 7008
Wang, H., Yurchyshyn, V., Liu, C., et al. 2018, RNAAS, 2, 8
Willmott, C. J., & Matsuura, K. 2005, ClRes, 30, 79
Xu, Y., Cao, W., Ahn, K., et al. 2018, NatCo, 9, 46
Xu, Y., Cao, W., Ding, M., et al. 2016, ApJ, 819, 89