A&A, 688, A126 (2024)
https://doi.org/10.1051/0004-6361/202449756

Astronomy & Astrophysics

FLAME: Fitting Lyα absorption lines using machine learning

P. Jalan[1], V. Khaire[2,3], M. Vivek[4], and P. Gaikwad[5]

[1] Center for Theoretical Physics of the Polish Academy of Sciences, Al. Lotników 32/46, 02-668 Warsaw, Poland
e-mail: priyajalan14@gmail.com
[2] Indian Institute of Space Science and Technology, Thiruvananthapuram, Kerala 695547, India
[3] Physics Department, Broida Hall, University of California Santa Barbara, Santa Barbara, CA 93106-9530, USA
[4] Indian Institute of Astrophysics, Koramangala, Bengaluru, Karnataka 560034, India
[5] Max-Planck-Institut für Astronomie, Königstuhl 17, 69117 Heidelberg, Germany

Received 27 February 2024 / Accepted 6 May 2024

ABSTRACT

We introduce FLAME, a machine-learning algorithm designed to fit Voigt profiles to H I Lyman-alpha (Lyα) absorption lines using deep convolutional neural networks. FLAME integrates two algorithms: the first determines the number of components required to fit Lyα absorption lines, and the second calculates the Doppler parameter b, the H I column density $N_{\rm HI}$, and the velocity separation of individual components. For the current version of FLAME, we trained it on low-redshift Lyα forests observed with the far-ultraviolet gratings of the Cosmic Origin Spectrograph (COS) on board the *Hubble* Space Telescope (HST). Using these data, we trained FLAME on $\sim 10^6$ simulated Voigt profiles – which we forward-modeled to mimic Lyα absorption lines observed with HST-COS – in order to classify lines as either single or double components and then determine Voigt profile-fitting parameters. FLAME shows impressive accuracy on the simulated data, identifying more than 98% (90%) of single (double) component lines. It determines b values within $\approx \pm 8$ (15) km s^{-1} and log $N_{\rm HI}$/cm^2 values within $\approx \pm 0.3$ (0.8) for 90% of the single (double) component lines. However, when applied to real data, FLAME's component classification accuracy drops by $\sim$10%. Nevertheless, there is reasonable agreement between the b and $N_{\rm HI}$ distributions obtained from traditional Voigt profile-fitting methods and FLAME's predictions. Our mock HST-COS data analysis, designed to emulate real data parameters, demonstrates that FLAME is able to achieve consistent accuracy comparable to its performance with simulated data. This finding suggests that the drop in FLAME's accuracy when used on real data primarily arises from the difficulty in replicating the full complexity of real data in the training sample. In any case, FLAME's performance validates the use of machine learning for Voigt profile fitting, underscoring the significant potential of machine learning for detailed analysis of absorption lines.

Key words. line: profiles – methods: data analysis – intergalactic medium

1. Introduction

The gas that permeates the space between galaxies is called the intergalactic medium (IGM). One of the best ways to explore the IGM is to study the large range of Lyα absorption lines present in quasar spectra known as the Lyα forest (Rauch 1998; Meiksin 2009). These absorption lines result from the quasar's continuum being absorbed by the redshifted Lyα (1215.67 Å) resonance line of the neutral hydrogen gas. The Lyα forest has been shown to be an exceptional tool for studying the thermal state of the IGM (e.g., Schaye 2001; Bolton et al. 2008; Lidz et al. 2011; Hiss et al. 2018; Gaikwad et al. 2021; Hu et al. 2023), the intensity of the ionizing ultraviolet background (e.g., Bolton & Haehnelt 2007; Becker et al. 2013; Gaikwad et al. 2017b; Khaire et al. 2019; Hu et al. 2023), and a wide range of cosmological parameters, including the mass of neutrinos (McDonald et al. 2006; Baur et al. 2017; Yèche et al. 2017) and dark matter properties (Busca et al. 2013; Viel et al. 2013; Iršič et al. 2017; Alam et al. 2021).

The Lyα forest at high redshift shows remarkable consistency with theoretical expectations for the IGM, such as the expected Gunn-Peterson troughs in $z \sim 6$ quasar spectra (Fan et al. 2006; Bosman et al. 2018; Eilers et al. 2018) and the peak in temperature around the epoch of He II reionization (e.g., Walther et al. 2019; Gaikwad et al. 2021), which is believed to conclude around $z \sim 3$ (McQuinn et al. 2009; Shull et al. 2010; Worseck et al. 2011; Khaire 2017).

In contrast, the low-redshift ($z < 1$) Lyα forest has yielded several unexpected results, prompting new investigations. For instance, the distribution of line widths in the Lyα forest at $z < 0.5$ is broader than that reproduced by simulations of the IGM (Viel et al. 2017; Gaikwad et al. 2017a) and there is evidence of a higher-than-expected temperature at $z \sim 1$ (Hu et al. 2023). Furthermore, the epoch $z < 1$ is critically important for galaxy formation, as it is during this period that feedback from galaxy formation (Springel 2005; Hopkins et al. 2008; Bolton et al. 2017; Weinberger et al. 2017; Davé et al. 2019) is believed to have a significant impact on the galaxies in order to explain the observed properties of galaxies and the sharp decline in star formation rate (Madau & Dickinson 2014; Khaire & Srianand 2015). Additionally, this is the epoch where more than 30% of baryons are still not accounted for (Shull et al. 2012) by observations (however see de Graaff et al. 2019; Tanimura et al. 2019; Macquart et al. 2020). Moreover, the degree to which galaxy formation feedback impacts the low-z IGM remains unclear (Khaire et al. 2023; Tillman et al. 2023), and simulations, even with extreme feedback, are still unable to reproduce the line-width distribution of the low-z Lyα forest (Gurvich et al. 2017; Bolton et al. 2022; Khaire et al. 2023). Given these challenges, studying the low-z Lyα forest becomes particularly interesting

and is crucial in order to understand these discrepancies. The present work therefore focuses mostly on the low-z Lyα forest.

Despite its potential, effectively extracting information from the Lyα forest has proven to be challenging, especially when it is done via fitting Voigt profiles to the swath of absorption lines. Overlapping lines, varying signal-to-noise ratios, instrumental line-spread functions, and other systematic uncertainties can lead to parameter degeneracies, making it difficult to extract accurate physical information from the data. Usually, for dealing with a swath of Lyα lines, semi- or fully-automated codes are used to fit Voigt profiles, such as "VoIgt profile Parameter Estimation Routine" (VIPER; Gaikwad et al. 2017a, hereafter G17), BayesVP (Liang & Kravtsov 2017), GVPFIT (Bainbridge & Webb 2017), and VoigtFit (Krogager 2018); however, these are still computationally expensive for large samples.

Moreover, efficient and automated analysis techniques are required to handle the increasing volume of data from modern surveys, both ongoing and upcoming, such as the Sloan Digital Sky Survey (SDSS; Bolton et al. 2012), Dark Energy Spectroscopic Instrument (DESI, Flaugher & Bebek 2014), *William Herschel* Telescope Enhanced Area Velocity Explorer (WEAVE, Pieri et al. 2016), and 4-metre Multi-Object Spectroscopic Telescope (4MOST, de Jong et al. 2012), which are providing an unprecedented wealth of absorption spectra for analysis. In this regard, machine learning (ML) techniques offer a promising solution.

ML algorithms, particularly those under the umbrella of deep learning (LeCun et al. 2015; Goodfellow et al. 2016), have demonstrated remarkable competency in pattern recognition, noise handling, and parameter estimation, making them well-suited to the challenges posed by the Lyα forest. ML models can learn the relationships between input data and desired outputs by training on large datasets of simulated or observed Lyα forest. Furthermore, ML techniques can improve the efficiency of the fitting process. Automating the analysis using ML models reduces human intervention and subjective bias while enabling the rapid analysis of large datasets.

In recent years, various applications of ML techniques have been developed to deal with a multitude of cosmological problems (Akhazhanov et al. 2022; Lee & Shin 2021; Vattis et al. 2021; Liu et al. 2021; de Dios Rojas Olvera et al. 2022). Deep learning or convolutional neural networks (CNNs) have been shown to be particularly powerful tools for cosmological data analysis. For example, Parks et al. (2018) used a CNN to predict the HI column density of the damped Lyα. Huang et al. (2021) used a neural network to estimate the Lyα optical depth values from noisy and saturated transmitted flux data in quasar spectra. Veiga et al. (2021) used a deep neural network to infer the matter density power spectrum from the quasar spectra. Cheng et al. (2022) also used CNN to identify the column density and Doppler widths of the Lyα lines at high redshifts. Recently, Stemock et al. (2024) also used deep learning to identify these parameters for Mg II doublet absorption lines.

Motivated by these studies and with the aim of combining the power of ML and the rich information contained within Lyα forest spectra, we developed FLAME (Fitting Lyα Absorption lines using machine learning), which is the combination of a two-part algorithm that identifies the number of components in each Lyα absorption system and a fitting of the Voigt profiles to each of them. To train these ML models, we generated multiple simulated absorption lines with properties similar to low-z Lyα absorption lines observed with the Cosmic Origin Spectrograph (COS) on board the *Hubble* Space Telescope (HST). In addition to the reasons mentioned above, we focus our models exclusively on low-z data to avoid complexities, because the Lyα forest is less dense and shows minimal blending compared to high-redshift Lyα forest regions. This allows easier isolation and fitting of each absorption line system. Nonetheless, even within low-z data, both single and multiple components are present. Therefore, our two-part algorithm FLAME first determines the number of components present and then fits these identified components accordingly.

To assess the robustness of our networks, we evaluated their performance on simulated, real observed (Danforth et al. 2016), and mock datasets. For the real observed dataset, we compared the model parameters with those derived using VPFIT[1] (Carswell & Webb 2014) and VIPER. Our findings reveal that the neural networks demonstrate comparable performance while requiring significantly fewer computational resources.

The paper is organized as follows. We explain the terminology related to the ML algorithms in Sect. 2. In Sect. 3 we discuss the creation and preprocessing of the simulated data. We present the model and performances of the two ML algorithms in Sects. 4 and 5. In Sect. 6, we compare the accuracy of our ML algorithm to that of traditional algorithms on the observed data. In Sect. 7, we discuss our key findings when using ML for Voigt profile fitting, and summarize our conclusions in Sect. 8.

2. Machine learning

Machine learning is a branch of artificial intelligence that focuses on developing algorithms that learn from data and then make predictions without explicit programming. ML involves designing statistical and mathematical frameworks to uncover patterns, correlations, and trends within datasets autonomously.

ML can be broadly classified as supervised or unsupervised. Unsupervised ML involves discovering patterns and structures in data without predefined labels. One of its applications is grouping similar data points based on certain features or characteristics. However, the ML model that finds a relation between the features in the measurements (training data) and its defining variables (labels) is known as the supervised ML. This trained model then predicts the label for any given set of measurements; the accuracy of this can be measured using validation data. In this paper, we only discuss supervised ML.

After defining the problem at hand, supervised ML can be summarized in six steps:

- Preparing the training data that includes collecting and preprocessing the data (Sect. 3) with their corresponding labels.
- Splitting the data into training and test/validation datasets. It is important that this splitting is random and that the training and testing cover a similar range of parameters to avoid possible extrapolation problems.
- Generating an ML model and training it on the training dataset and predicting the labels for the testing dataset (Sects. 4.1 and 5.1).
- Assessing the model's performance using the test or validation data by comparing the true and predicted labels.
- If the model's performance is unsatisfactory, adjusting hyperparameters, using different feature sets, or modifying the model architecture,
- Selecting the best-performing model, evaluating its performance (see Sects. 4.2 and 5.2).

These supervised ML algorithms commonly employ neural networks used across multiple applications.

[1] https://www.ast.cam.ac.uk/~rfc/vpfit.html

2.1. Neural networks and activation functions

Neural networks work by sending information through layers of connected nodes or neurons, each contributing to the ability of the network to learn and make predictions. In these networks, every neuron is connected to all neurons in the next layer, and these connections have weights. These weights help determine how important each input is. At each jth neuron, the inputs are combined into a weighted sum,

$$z_j = \sum_{i=1}^{n} (w_{ij} \cdot x_i) + b_j, \tag{1}$$

where x_i is the input to the neuron from the previous layer, w_{ij} as the weight of the connection between the ith neuron in the previous layer and the jth neuron in the current layer, b_j as the bias term for the jth neuron in the current layer, z_j as the weighted sum of inputs to the jth neuron in the current layer, and n is the number of neurons in the previous layer. This output is then processed by the activation function $f(z)$, which leads the output of the jth neuron in the current layer to be: $a_j = f(z_j)$. This function $f(z)$ allows each neuron to introduce nonlinear transformations to its input, enabling the network to capture complex patterns in the data. Without activation functions, the network would be limited to performing linear operations, resulting in a model incapable of capturing complex patterns in the data. Activation functions let the network handle a wide range of tasks and data patterns, making it capable of doing everything from simple classification to solving intricate problems across different areas. The combination of these weights, inputs, and activation functions is what enables neural networks to learn from various data and perform a broad spectrum of tasks, as discussed in Parhi & Nowak (2019).

We used two activation functions in this study: (a) Leaky Rectified Linear Unit (Leaky ReLU): $f(z) = \max(0.01z, z)$, which returns z if it receives any positive weighted sum input, but returns a small value of $0.01z$ if it receives any negative value of z. Therefore, it gives a positive output for negative values as well; and (b) Sigmoid: $f(z) = 1/(1 + e^{-z})$. The value of this function exists between 0 to 1. This activation function is beneficial for models that predict the output as a probability, as the probability of anything exists between 0 and 1. We use sigmoid at the final layer of the classification model (Sect. 4).

The hierarchical arrangement of layers enables the neural network to learn increasingly "abstract representations" as information progresses through the network. A network containing multiple fully connected layers is known as a "deep" neural network. However, to study the structure and patterns present in complex data, CNNs are designed. In the following subsection, we describe the CNNs.

2.2. Convolutional neural network

CNNs are specialized neural networks designed for processing data arranged in a grid-like structure, such as images or time series. They are particularly effective at detecting spatial relationships within the input data through a sequence of interconnected layers. In this study, we used CNNs to identify the number of blended Lyα absorption lines and fit Voigt profiles to them.

At the heart of CNNs lie "convolutional layers", which utilize "filters" to apply convolutions to the input. A filter consists of multiple "kernels", with each kernel dedicated to a specific channel of the input. As these filters move across the input, they perform element-wise multiplications and summations, generating feature maps in the process. These feature maps are crucial for extracting various data characteristics, such as edges, textures, and patterns. A key hyper-parameter, "stride", governs the step size of the filter as it scans across the input. Padding, another important hyper-parameter, ensures comprehensive coverage of the input's edges, preserving the size of the input through the convolution process. This study employs the 'SAME' padding technique, ensuring that the dimensions of the output post-convolution remain consistent with the input dimensions.

Following the convolutional layers, "Pooling layers" are used to reduce the spatial dimensions of the feature maps, thereby simplifying the information while retaining the most relevant features (Gholamalinezhad & Khosravi 2020). In this study, we use max pooling (Matoba et al. 2022), a technique that identifies and retains the maximum value within a specific region defined by the kernel's coverage. This approach effectively captures the most prominent features within each region, enhancing the network's ability to understand spatial hierarchies and relationships.

In the following subsection, we explore how neural networks are trained and discuss the importance of loss functions in improving their accuracy.

2.3. Neural network training and optimization

After designing the neural network, we optimize the network's hyper-parameters by training the neural network to enable accurate predictions. The training procedure involves two main steps: forward propagation and backpropagation.

During forward propagation, input "training data" flows through the network, with each neuron applying an activation function to the received signals, producing outputs. The computed outcomes are then compared to the true labels using a loss function, such as mean squared error (MSE) or cross-entropy, quantifying the error in predictions. For example, in this study, one of the loss functions used is binary cross-entropy (BCE), which has a functional form of:

$$H_p(q) = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(p[y_i]) + (1 - y_i) \log(1 - p[y_i]), \tag{2}$$

$H_p(q)$ signifies the entropy between the predicted probability distribution p and the true distribution q. N represents the dataset's total number of samples. y_i is the "true label" of the ith sample in the dataset. It can be either 0 or 1 in a binary classification scenario. $p(y_i)$ is the predicted probability that the ith sample belongs to class 1 (or has label 1). The other loss function used in this study is the mean squared error (MSE),

$$\mathrm{MSE} = -\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2, \tag{3}$$

where y_i and $\hat{y}_i$ represent the true and predicted labels for the ith sample, respectively.

Back-propagation involves propagating the error backward through the network to compute gradients of the loss function with respect to the network's hyper-parameters. These gradients provide valuable information on adjusting the parameters to minimize the error. Popular optimization algorithms, like Adagrad, RMSprop, Stochastic Gradient Descent (SGD), and "Adam" (Kingma & Ba 2014), update the parameters iteratively to minimize these gradients. One of the crucial hyperparameters in ML algorithms is the learning rate. It determines the step size at

which the model parameters are updated during the optimization process. This study uses the Adam optimizer, which is a more efficient alternative to the other methods, to adjust the model weights.

During training, the network updates its parameters by utilizing smaller subsets or batches of the dataset. This study uses smaller "batch sizes" that allow more frequent updates to the network's parameters, leading to faster convergence and mitigating memory limitations (You et al. 2017). Another hyperparameter is the number of "epochs determining how often the network will iterate over the training dataset. We define the epochs for this study by implementing an "early stopping technique". This approach halts the training process when the validation performance no longer improves, helps prevent overfitting, and conserves computational resources.

Neural network training relies on a labeled dataset, careful selection of architecture, regularization techniques to prevent overfitting, and hyperparameter tuning to achieve optimal results. We iterate the hyperparameters mentioned above after carefully selecting the data and model. Then, we choose the hyperparameters that produce satisfactory results, measured using an evaluation metric. Below, we discuss the evaluation metrics used in this study.

2.4. Evaluation metrics

After training the network and finding the best hyperparameters, we validate the model using a test or validation dataset. The evaluation metrics of this test dataset ensure the model is unbiased and test its performance. The labels of the testing dataset are called the "true" labels, and the predictions made by the model are known as the "predicted" labels. In this study, we use two models as described in Sects. 4.1 and 5.1; (i) Binary classification algorithm – to classify the number of absorptions into single and double lines, and (ii) Regression algorithm – to identify the Voigt profile parameters of the absorption lines. Therefore, we require two evaluation metrics.

In the binary classification algorithm, there are two classes: positive and negative. To test the performance of the binary classification algorithm (Hossin & Sulaiman 2015), we calculate the accuracy,

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}}, \tag{4}$$

where TP is true positive, TN is true negative, FP is false positive, FN is false negative. The true positives and negatives imply that the model's outcome correctly predicts the positive and negative classes. The false positive means the model's outcome incorrectly predicts a positive class for an actual negative class. The false negative means the model's outcome incorrectly predicts a negative class for an actual positive class.

Other parameters to identify the robustness of the classification algorithm are as follows:

- Sensitivity, also known as recall, assesses a model's ability to correctly identify positive instances, focusing on minimizing false negatives.
- Specificity, also known as true negative rate, gauges a model's capacity to recognize negative instances, aiming to minimize false positives accurately.
- Precision quantifies the proportion of correctly predicted positive cases among all instances predicted as positive, emphasizing the minimization of false positives.
- Negative predictive value evaluates the proportion of accurately predicted negative cases among all instances predicted as negative, giving insight into the model's ability to avoid false negatives.

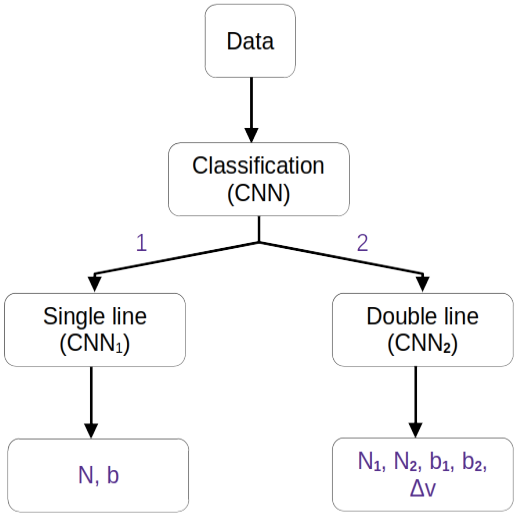

Fig. 1. The flowchart outlines the sequence of three neural networks comprising FLAME. The input training data to the classification algorithm is the normalized flux with 301 array size labeled as 1 or 2 absorption lines. The same input data is then fed to different regression algorithms based on the number of absorption lines. For this algorithm, the labels with the input dataset are the physical parameters like column density and Doppler width.

The precision and sensitivity lead to the F1-Score. The F1-score is the harmonic mean of precision and recall as given by

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

The F1-score is a single metric that balances both precision and recall and is especially useful when the class distribution is imbalanced. The F1-score ranges between 0 and 1, with higher values indicating better performance. An F1-score of 1 indicates perfect precision and recall. In practical terms, this indicates that the test has achieved the highest possible accuracy, with no false positives or false negatives. However, an F1-score of zero arises when either precision or recall is zero, indicating that the test's accuracy is at its lowest. During the regression analysis (as described in Sect. 5), we predict values from the model and compare them to the true labels of the dataset. To test the accuracy of the regression analysis, we use the MSE. We also calculate the 90 and 68 percentile values of the absolute differences in the true and predicted values to understand the data distribution and its concentration. We use percentiles because they are less sensitive to extreme values or outliers since they're based on rank order rather than actual values.

Figure 1 shows the flowchart sequence of the two models used in this study. The first algorithm is a CNN that classifies the number of absorption lines into either single or double. The choice of only two states is dictated by the low-z dataset we used (see Sect. 3). Once the classification is performed, we use the second algorithm, which consists of two CNNs, to estimate the features/parameters, one for single lines and the other for double lines. These networks are created using the TensorFlow[2] interface (Abadi et al. 2015). We construct a simulated dataset of low-z Lyα lines and use it to train the networks created here. We outline the details of the dataset in the next section.

[2] https://www.tensorflow.org/api_docs/python/tf/keras

3. Training and testing dataset

The training dataset is the initial input to the network during the training process and plays a crucial role in shaping the network's ability to learn and make predictions. Therefore, we require a well-constructed training dataset that covers a range of parameter space and represents the real-observed data. However, due to the limited sample available of the low-z Lyα absorption lines, we generate our training dataset by simulating Voigt profiles. In order to model realistic simulated Lyα lines, we use the general properties of the observed low-z Lyα data. Below, we explain the properties of the low-z data and how it was used in simulating the lines for training.

3.1. Observational data description

We aim to apply ML techniques to fit low-redshift Lyα lines, drawing on the extensive survey of the low-redshift IGM at $z < 0.48$ conducted by Danforth et al. (2016, henceforth referred to as D16). This survey uses 82 high signal-to-noise ratio (S/N) quasar spectra observed with the HST/COS in the far-ultraviolet (FUV) band using medium-resolution gratings G130M and G160M ($R \sim 18\,000$, $\Delta v \sim 18\,\mathrm{km\,s^{-1}}$) across different lifetime positions to cover a wavelength range of 1030 Å–1800 Å. D16 combined data from both gratings whenever available, applied manual continuum fitting to each spectrum, and cataloged all absorption lines, including those intervening, associated, and arising from the Milky Way's interstellar medium.

D16 determined locations of absorption lines using a crude significance level vector $\mathrm{SL}(\lambda) = W(\lambda)\bar{\sigma}(\lambda) > 3$, where $W(\lambda)$ represents the equivalent width vector and $\bar{\sigma}(\lambda)$ denotes the error vector in regions without lines. Following this localization, a standard procedure was employed to identify lines by using coincident higher-order lines or lines from different ions. They identified a total of 2611 intervening Lyα lines and modeled them with Voigt profiles. The line list tables from D16, publicly available in the high-level science product at the Mikulski Archive for Space Telescopes[3], include details of these fits, such as the Doppler width (b), redshift ($z_{\rm abs}$), and neutral hydrogen column density ($N_{\rm HI}$), along with their associated errors.

For the present study, we selected 1917 Lyα lines that are not blended with metal lines or higher-order lines from the D16 catalog. According to D16's fits, 81.2% (1557) of these lines are single lines, 14.9% (286) are doublet structures, and the remaining 3.8% (74) consist of three or more lines. We used the parameters of these lines to generate the simulated training dataset described in the following section.

3.2. Simulated data

The simulated Lyα absorption lines are generated with randomly selected (from a uniform distribution) column density ($N_{\rm HI}$/[cm^{-2}]) and Doppler width (b/[km s^{-1}]) and combined it with instrumental properties of HST/COS data. The range for each parameter includes the minimum and maximum values (for 98.5% data avoiding the outliers in each parameter) of the HST data, as mentioned above. Since the real observed dataset (Sect. 3.1) has <4% lines with >2 components, our simulated dataset is limited to single and double lines. The parameter ranges are as follows:

- Doppler width: b (km s^{-1}) = 5–100,
- Column density: log $N_{\rm HI}$ (cm^{-2}) = 12–17,
- Signal-to-noise ratio: S/N = 5–100,
- Central wavelength: C_λ (Å) = 1220–1800.

Given these parameters space, we used the following steps to create the dataset for single Voigt profiles:

1. Generate the Voigt profiles using the 'Faddeeva function', for a randomly selected value of b and $N_{\rm HI}$ from the above range, resulting in optical depth (τ) over a velocity resolution of 0.6 km s^{-1}.
2. Convolve the simulated flux ($F = e^{-\tau}$) with the tabulated line spread function (LSF) of the HST COS spectrograph[4]. The Line Spread Function (LSF) varies with each grating and is wavelength-dependent. Thus, the convolution depends on both the observed wavelength ($z_{\rm abs}$) and the grating used.
3. We randomly select a central wavelength value (C_λ) from above range. For $C_\lambda > 1450$ Å, a value of z_{abs} or z is selected randomly from a uniform sample between 0.2 to 0.47, and the grating is G160M. However, for central wavelength ≤1450 Å, $z_{\rm abs}$ are randomly selected from a uniform sample between 0.005 to 0.2, and grating is G130M. The convolution is performed in the observed frame.
4. Resample the convolved data to a similar wavelength scale (Δ_λ = 0.0299 Å i.e., $\Delta v = 6\,\mathrm{km\,s^{-1}}$) with which D16 resampled their combined spectra.
5. Add Gaussian random noise to the simulated Voigt profile with a signal-to-noise ratio (S/N) selected from 5–100.
6. Convert the rest-wavelength to the Δv. We align the absorber at the center by selecting a chunk of 301 pixels and pad the rest of the chunk with continuum flux = 1.

After creating a sample of single absorption lines, we used the following procedure to generate a sample representing double lines, which simulate blends of two components within an absorption system. Initially, we randomly selected two simulated Voigt profiles with optical depths τ_1 and τ_2 from the first step. Each absorption line was then shifted by $\pm\Delta v/2$, where Δv is randomly chosen from the range of 5–350 km s^{-1}. This ensures that the center of the absorption line does not align closely with the spectral edges. Subsequently, the combined optical depth ($\tau = \tau_1 + \tau_2$) was converted into flux ($\mathrm{F} = e^{-\tau}$), serving as the input for the second step. The procedures outlined above are similarly applied throughout.

Figure 2 illustrates two examples of our simulated absorption lines. The left panel of Fig. 2 shows a single Voigt profile for a given b, $N_{\rm HI}$ and $z_{\rm abs}$; the right panel shows an overlapping double Voigt profile structure with velocity separation (Δv) for a given pair of b and $N_{\rm HI}$. A solid brown line in Fig. 2 shows the simulated line generated in step (i). After convolving the line with COS LSF corresponding to step (ii) is shown in the green line. The red dashed line shows the final absorption lines after rebinning and adding the Gaussian noise.

The above-generated dataset is divided into 80% training and 20% testing datasets, covering the same parameter ranges. The ML models use the training dataset as input and evaluate their accuracy on new, unseen data with the testing dataset.

3.3. Mock data

Ideally, the accuracy of the ML algorithm should be consistent with simulated and real testing datasets. In the case of any disagreement, generating a mock dataset can effectively resolve any issues with the algorithm's performance or the creation of the simulated testing dataset. Therefore, complementing our study

[3] https://archive.stsci.edu/prepds/igm/

[4] https://spacetelescope.github.io/COS-Notebooks/LSF.html

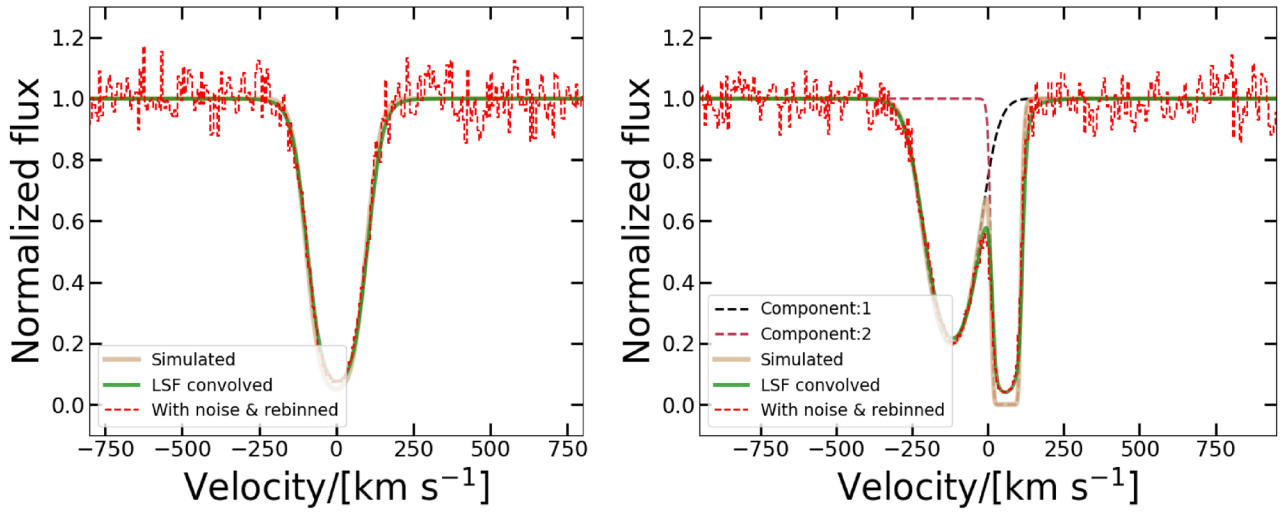

Fig. 2. Simulated Voigt profiles for single and double absorption lines in brown color. The green lines show the absorption lines after convolving with the HST's tabulated LSF. The red dashed line shows the absorption line after adding the Gaussian noise and rebinning it to a similar velocity frame to the HST data. This red dashed line represents the typical absorption line used as the training dataset in this study. *Left panel*: brown line shows the simulated Voigt profiles for a single absorption line with $N_{\rm HI} = 10^{14}\,{\rm cm}^{-2}$, $b = 80\,{\rm km\,s}^{-1}$ and $S/N = 25$. *Right panel*: same as the left panel but a simulated double absorption line. The dashed lines show two simulated single absorption lines ($N_{\rm HI} = 10^{14.29}$ and $10^{15.06}\,{\rm cm}^{-2}$, $b = 92.67$ and $25.43\,{\rm km\,s}^{-1}$) that are shifted by $\pm\Delta v/2 \sim 75\,{\rm km\,s}^{-1}$ value and the combined profile is shown in brown color. The training dataset for double absorption lines is shown in red dashed lines.

of the "real" data, we also generated a set of "mock" testing data. These mock Lyα lines mimic the real dataset generated by the procedure discussed in Sect. 3.2. The mock absorption lines have exactly the same physical parameters (b, log $N_{\rm HI}$, S/N, z, and C_λ) as the real dataset. In subsequent sections, we also present an assessment of the ML algorithm's performance on this realistic mock data and demonstrate its reliability.

4. Classifying the number of absorption lines

4.1. CNN architecture

This section introduces an ML algorithm based on CNN architecture that is specifically designed for binary classification to identify the number of absorption lines. The input data consists of a training dataset generated in Sect. 3, comprising 301 pixels representing simulated absorption lines. The model's output is a single value ranging between 0 and 1, where values < a threshold value indicate a single absorption line ($N_l = 1$) and ≥ threshold value corresponds to a double absorption line ($N_l = 2$). We tested model outputs with various threshold values and found that 0.5 gives an unbiased result.

The CNN architecture shown in Fig. 3 includes an input layer with 301 neurons, two convolutional layers, two max-pooling layers, and four fully connected dense layers with a decreasing number of neurons. The activation function employed throughout the model is the Leaky ReLU, except the sigmoid function in the last layer. We use the binary cross-entropy loss function (Eq. (2)) and the Adam optimizer with a learning rate of 10^{-3} for optimization.

The CNN is trained on 1.6 million absorber samples and tested on 4×10^5 samples (see Sect. 3), with an equal number of single and double absorption lines. We apply batch propagation with batch size 100 and implement early stopping criteria with a patience of 20 epochs. To ensure robustness and accurate identification of the number of Voigt profiles in a chunk of 301 pixels, we carefully fine-tune the parameters of the binary classification algorithm (Fig. 3). Our objective is to achieve an accuracy of over 90% (see Eq. (4)) and F1-score greater than 0.8 (see Eq. (5)).

We also tested with other ML algorithms, like random forest classifiers (Breiman 2001) and support vector machines (Cortes & Vapnik 1995). However, we found that the F1-score for all other classifiers was less than 0.8.

4.2. Classification of the number of Lyα absorption lines

Figure 4 shows the performance of the binary classification algorithm computed for the simulated test dataset. In this study, the true positives and true negatives are "true-single" and "true-double" lines, respectively. The simulated test sample has an impressive TP rate of 97.5%, accurately identifying $N_l = 1$ absorption lines. The FP rate was 10.1%, reflecting a moderate number of misclassifications of $N_l = 2$ absorption lines as $N_l = 1$. However, the TN rate was high at 89.9%, indicating a highly reliable identification of $N_l = 2$ absorption lines. We also noted a small FN rate of 2.5%, indicating a low number of misclassifications of $N_l = 1$ absorption lines as $N_l = 2$. The normalized confusion matrix (Fig. 4) demonstrates the binary classification algorithm's ability to distinguish between the two absorption line categories. The caption mentions the values of sensitivity, specificity, precision, and negative predictive values. This leads to an F1-score (see Eq. (5)) of 0.93, suggesting the model accurately identifies single and double absorption lines.

Figure 5 shows two representative examples from each category. Even by visual inspection, it becomes evident that the misclassified absorption lines also appear visually ambiguous. We evaluated the model's performance across different parameters, including S/N, b-parameter, and $N_{\rm HI}$ as shown in Fig. 6. As expected, accuracies are notably lower for small values of S/N. This effect is also evident from the examples FP and FN in Fig. 5. However, excluding cases with $S/N < 20$ in the lower percentile consistently yields accuracies above 98% and 91% for single and double lines, which is promising. We also find that the accuracy for single lines is consistently better than the accuracy for double lines. We also find that the accuracy decreases slightly

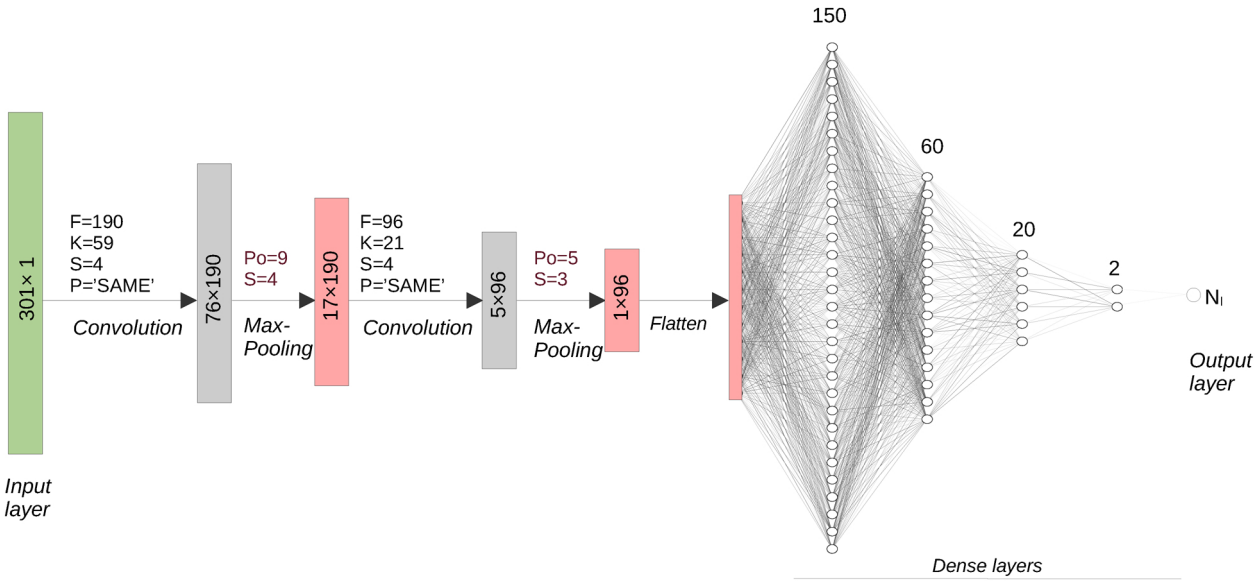

Fig. 3. Schematic diagram of the CNN architecture to classify the number of absorption lines. The notation used here is F – filter size, K – kernel size, S – strides, P – Padding, and Po – Pooling size. The number of neurons in the dense layers is written at the top. The last layer has one output varying between 0 and 1. The values <0.5 are assigned $N_l = 1$, and values ≥0.5 are assigned $N_l = 2$.

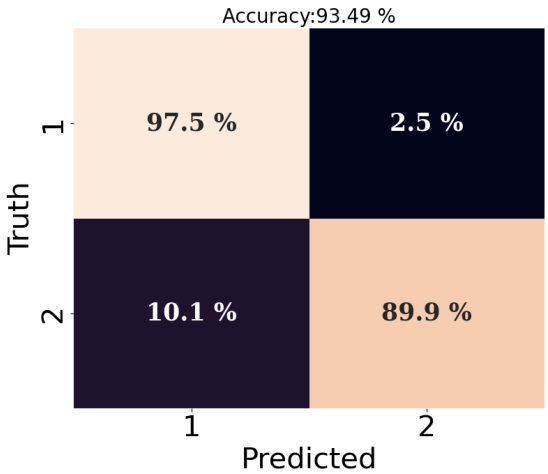

Fig. 4. Confusion matrix for the predictions for the number of absorption lines using the CNN (as shown in Fig. 3). The CNN was trained on 1.6 million lines and tested on 400 K samples, with an equal number of single and double lines. We find the Sensitivity = 97.47%, Specificity = 89.92%, Precision = 89.64% and Negative Predictive Value=97.55%.

with increasing Doppler width. For broad absorption features, it is reasonable to expect that the CNN may encounter greater difficulty in discerning whether it originates from a single line or double lines. This pattern is similar for column density for single lines but not for double lines. The accuracy for lower column densities is influenced by S/N, as absorbers with low column densities may be more susceptible to being hidden within noise. On the other hand, for higher column densities, the saturation of absorption lines can limit the accuracy of classification. Even if one of the lines in double lines is saturated, regardless of the column density of the other line, classification becomes challenging. The evaluation of the simulated test sample and the visual examination of classification examples highlight the CNN-based algorithm's effectiveness and reliability in absorption line classification tasks.

5. Regression analysis

5.1. Algorithm

In this section, we outline the regression analysis ML algorithm designed to determine the physical properties of absorption lines. We train this network on the simulated dataset generated in Sect. 3, similar to the previous model. The primary output predictions are the parameters characterizing the absorption lines. For single lines, these parameters encompass log $N_{\rm HI}$ (or log N) and b, while for double lines, they extend to log N_1, b_1, log N_2, b_2 (the subscripts represent components), and Δv.

To build the regression analysis, we initially experimented with fully connected deep neural networks with varying numbers of hidden layers and other units to develop the ML algorithm. However, we found that a large input dataset was required for the algorithm to learn effectively, which was computationally expensive. Therefore, to optimize the model's learning efficiency, we use CNNs for parameter estimation. The architecture, illustrated in Fig. 7, comprises of three convolutional layers, each followed by a max pooling layer, with decreasing size of hyper-parameters, connected to three dense layers. The output layer results in the estimate of b and $N_{\rm HI}$.

The single absorption lines dataset consists of 2.5 million samples randomly divided into 80% as training and 20% as testing datasets. The data size was selected to minimize computational time and achieve higher accuracy. All the convolutional and dense layers are activated using the LeakyReLU activation function. We used the Adam optimizer with a 10^{-4} learning rate. The model undergoes training in batches of ten instances, employing early stopping with the patience of 100 epochs. Training halts after 128 epochs based on the specified early stopping criteria. The weights are updated based on the MSE loss function (Eq. (3)). The hyper-parameters were varied in order to select those hyper-parameters that resulted in the minimum discrepancy between the true and predicted values.

For double absorption lines, the architecture was similar to the above CNN (Fig. 7) with minor modifications:

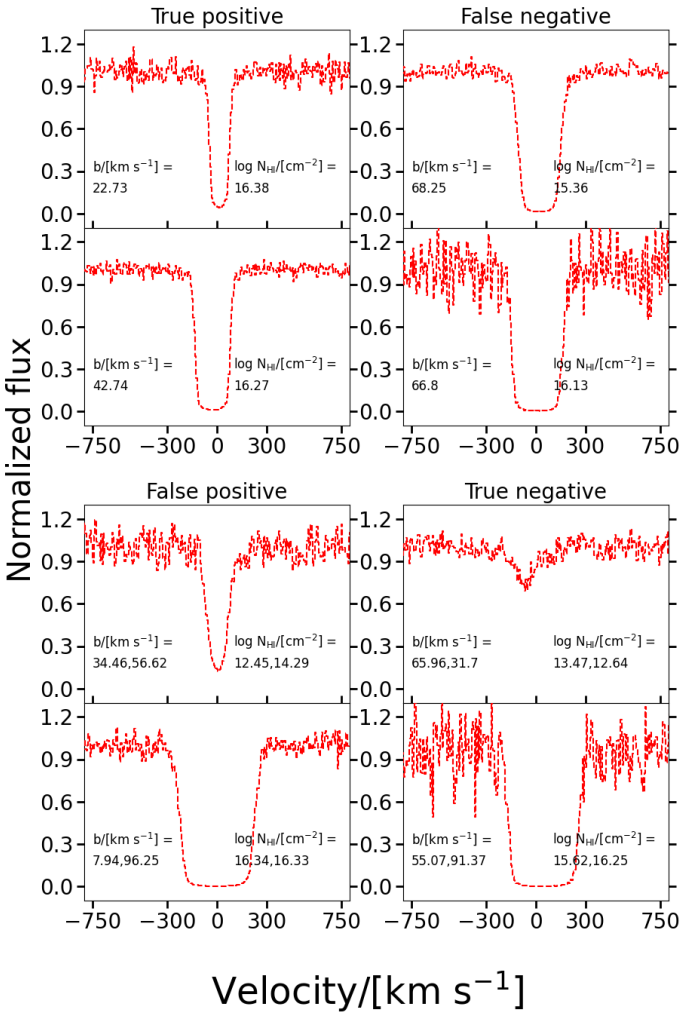

Fig. 5. Each panel shows two examples from the four classes of Fig. 4. The top left and bottom right panels show the correctly predicted single and double absorption lines. The top right and bottom left show the misclassified examples.

- To obtain similar outcomes, the input data consisted of a sample size of 4 million, split into two sets: 80% for training and 20% for testing.
- The output layer has five values (b_1, b_2 , log N_1, log N_2 and Δv).
- Using early stopping criteria, we trained the algorithm for 430 epochs.

5.2. Parameter estimation of the Lyα absorption lines

Figure 8 shows the performance of the CNN model predicting the physical properties of the simulated test dataset of a single Lyα absorption line. The upper panels of Fig. 8 show the comparison between the true (horizontal axis) and predicted (vertical axis) values of Doppler width and column density for the test-simulated data. We find a tight correlation between the intrinsic and predicted parameters, indicating that our CNN model accurately predicts the physical properties of a single absorption line. The black line is the one-to-one line marking a perfect prediction. The figure shows a nominal scatter in the algorithm's prediction of b and log $N_{\rm HI}$. We found negligible bias for the selected range of the parameters of b and log $N_{\rm HI}$.

The middle panels of Fig. 8 show the histogram of the difference between the true and predicted values of b and log $N_{\rm HI}$. We find that 68% [90%] of the predictions for the Doppler width (σ_{68b} [σ_{90b}]) are within $^{+2.29}_{-1.06}$ [$^{+8.96}_{-6.68}$] km s^{-1}, and for column density (in log-scale, σ_{68N} [σ_{90N}]) is within $^{+0.18}_{-0.16}$ [$^{+0.33}_{-0.35}$] cm^{-2}. The combined (σ_{90b} and σ_{90N}) outlier fraction with $|b_{\rm true}-b_{\rm pred}| > 7.8$ km s^{-1} along with $|N_{\rm true}-N_{\rm pred}| > 10^{0.33}$ cm^{-2} is less than 0.01%. This fraction reduces even further to only 0.0002% if we consider output only for samples with $S/N > 15$. The bottom panels of Fig. 8 show the histograms of true and predicted parameters. By construct, the count of true values is similar in each bin; however, CNN has relatively poor predictions of higher true b values. This discrepancy suggests that the CNN may have difficulty accurately predicting higher b-values, potentially due to the complexity of spectral features associated with broad absorption lines. However, we do not observe similar evidence for column densities, indicating that CNN's predictions for column densities are relatively more accurate across different parameter ranges.

Figure 9 demonstrates a few examples of the predictions by CNN for the single absorption test-simulated data. The input data, consisting of 301 pixels generated in Sect. 3, is shown in red dashed lines. The figures in the upper boxes demonstrate precise physical parameter predictions made by the CNN for low S/N data ($S/N < 10$; blue), while the second panel displays higher S/N data ($S/N > 10$; green). The two lower panels illustrate two instances of an inaccurate prediction made by CNN for low (purple) and high S/N (cyan).

The CNN results in similar accuracy for the test-simulated double absorption sample predictions as for the single absorption lines. The results comparing true and predicted values of b (b_1 and b_2), log $N_{\rm HI}$ (N_1 and N_2) and Δv are shown in the upper panel of Fig. 10. As seen from the upper panel of Fig. 10, the density of predicted values overlaps the true versus true one-to-one black line. The scatter of predicted values of log N for double lines has increased compared to the scatter for single absorption lines. However, we do not see any biases in the parameter's predictions, even in double absorption lines. This alignment demonstrates the algorithm's robustness in handling these more complex absorption profiles.

To test the algorithm's performance, we show the difference between the true and predicted values in the middle panel of Fig. 10. The histogram shows that σ_{68b} [σ_{90b}] for b_1 is within $^{+3.62}_{-3.87}$ [$^{+15.33}_{-16.47}$] km s^{-1}, and that for b_2 is within $^{+3.98}_{-3.76}$ [$^{+17.57}_{-14.77}$] km s^{-1}. Similarly σ_{68N} [σ_{90N}] for log N_1 is within $^{+0.43}_{-0.38}$ [$^{+0.97}_{-0.78}$] cm^{-2} and that for log N_2 is within $^{+0.28}_{-0.47}$ [$^{+0.73}_{-0.85}$] cm^{-2} and that for Δv is $^{+10.42}_{-10.43}$ [$^{+23.89}_{-26.82}$] km s^{-1}. The combined (σ_{90b} and σ_{90N}) outlier fraction with $|b_{\rm true}$-$b_{\rm pred}| > 16$ km s^{-1} along with $|N_{\rm true}-N_{\rm pred}| > 10^{0.90}$ cm^{-2} is less than 2.5%. This fraction reduces even further to only 1.5% if we consider output only for samples with $S/N > 15$. The lower panel of Fig. 10 illustrates that CNN predictions are less accurate for lines with extreme b-values, log $N_{\rm HI}$, and Δv. When b and $N_{\rm HI}$ values are low, accuracy decreases due to the signal being obscured by noise. Conversely, for higher b and lower Δv values, lines tend to blend together, leading to less accurate predictions. Similarly, high column densities result in poor predictions due to saturation effects. For higher Δv values, the predictions are affected because the two lines become nearly separate entities. In this scenario, either one line may be obscured by noise, or both lines may be nearly saturated, impacting the accuracy of predictions.

A few examples of the double absorption test-simulated data with their predicted Voigt profiles are shown in Fig. 11. Notably,

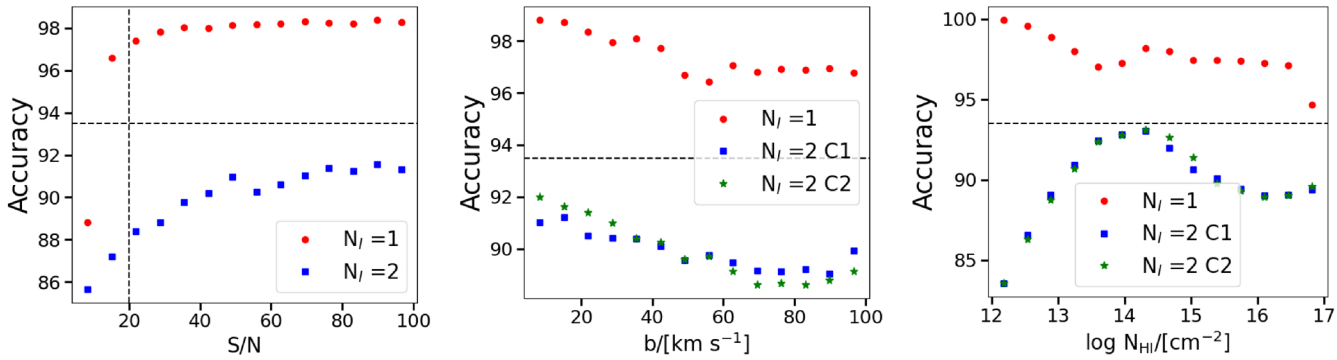

Fig. 6. Comparison of classification accuracy versus signal-to-noise ratio, Doppler width (b), and column density ($N_{\rm HI}$) for simulated data. $N_l = 1$ represents the single absorption lines, and $N_l = 2$ C1 and $N_l = 2$ C2 represent the first and second components of double absorption lines, respectively.

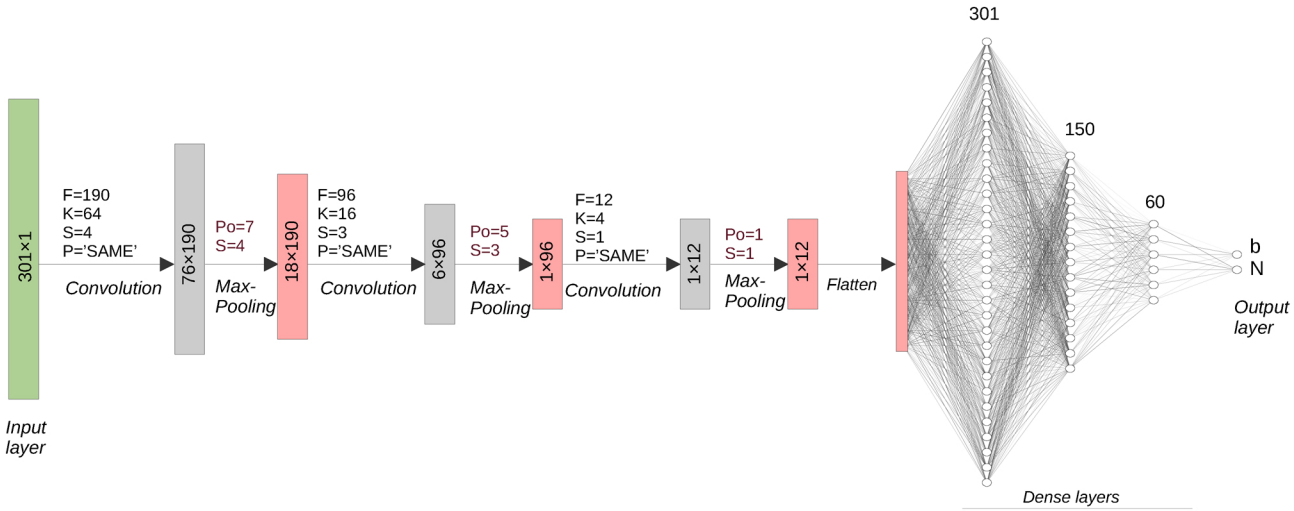

Fig. 7. Schematic diagram of the CNN model to predict the b and N values for a single absorption line. The CNN comprises three one-dimensional convolutional layers, each followed by three max-pooling layers; after flattening, the output is input to three dense layers with a decreasing number of neurons and two neurons in the output layer.

our ML model can accurately predict two parameters even if the double absorption lines are separated by small Δv. However, challenges arise when either the absorption lines are nearly saturated, or they are heavily obscured by noise, or there is a significant contrast in optical depth between two absorption lines with one almost buried in noise. These scenarios emphasize the CNN model's limitations under specific circumstances and scope for improvement.

Based on this analysis, we conclude that our CNN models accurately predict the column density and Doppler width for single and double absorption lines. The minimal outliers confirm that CNN can robustly extract b and $N_{\rm HI}$ of the single and double absorption lines for the simulated test data.

6. Application to real data

The preceding sections show that our ML algorithm excels in its performance on simulated absorption lines designed to emulate the characteristics of real data obtained from the HST-COS, as detailed in Sect. 3. To evaluate the real-world applicability of our ML algorithm (Sects. 4 and 5), we now test it on Lyα line profiles obtained from the COS data D16. To ensure compatibility, we chose observed data falling within a parameter range akin to the simulated data (as outlined in Sect. 3), in addition to 3σ detection according to the line detection criteria used in D16, resulting in 1364 absorption systems.

The line list files provided by D16[5] include 2400 Lyα absorption lines. We examine the corresponding spectrum for each line to determine whether the Lyα line consists of single or multiple components. Our approach involves searching each line to identify if adjacent lines share common wavelengths. If the span of common wavelengths for absorption lines continuously touches the spectrum's continuum (within error of the spectrum) for more than seven pixels (with each pixel being 0.035 Å), we classify these as single-component absorption lines. For the remaining lines, we count the number of lines meeting the aforementioned criteria. With this method, we find that out of 2400 components, approximately 1557 are Lyα lines with a single component, and 286 systems are identified as having double components. Following specific criteria ($5 \leq b/\,{\rm km\,s^{-1}} \leq 100$, $12 \leq \log N/{\rm cm^{-2}} \leq 17$, $5 \leq S/N \leq 100$ and discarding lines with significance below 3σ), we selected 1364 lines for further analysis.

To use these absorption lines as inputs for our ML algorithms, we select a segment of the spectrum encompassing the absorption line and an adjacent line-free region. This segment is centered within a 301-pixel chunk. If the spectral segment

[5] Available at https://archive.stsci.edu/prepds/igm/

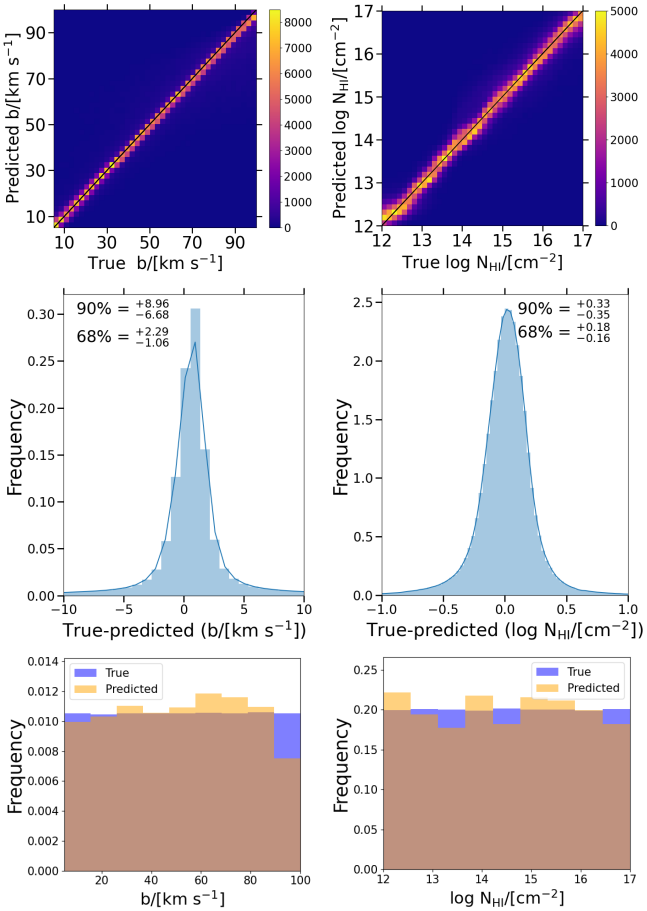

Fig. 8. Comparison and evaluation of the predicted and true parameters for simulated test single absorption line. The upper panel compares the actual and predicted values for the two parameters, b and $N_{\rm HI}$. The middle panel exhibits the normalized distribution of the differences between the predicted and true values, with markers for the 90% and 68% percentiles. The bottom panels show the normalized histogram of the CNN-predicted values in comparison to the true labels.

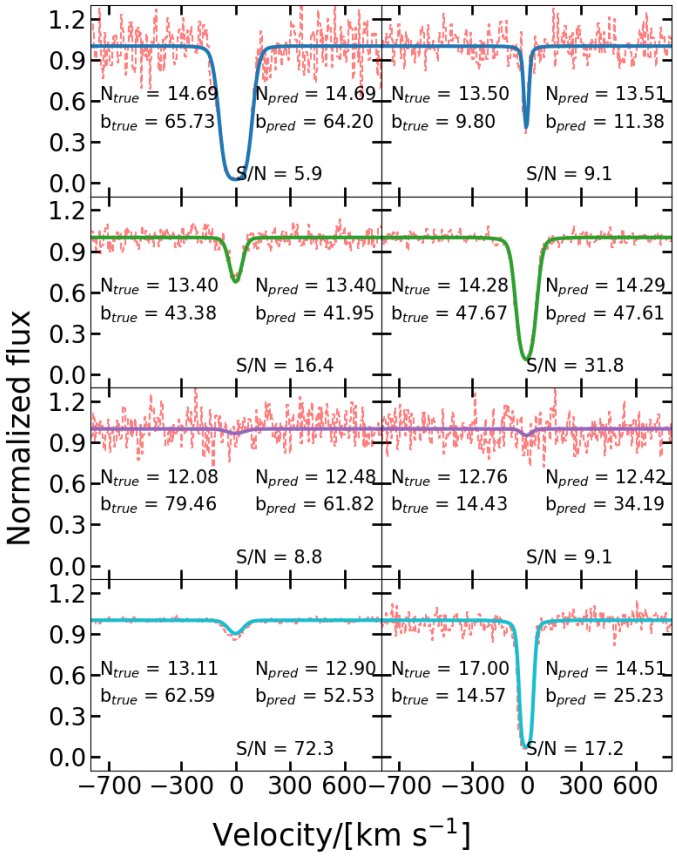

Fig. 9. Examples of the simulated test single absorption lines and the corresponding Voigt profile predicted by the CNN model (in colors). The input data, consisting of 301 pixels generated in Sect. 3, are shown as red dashed lines. The true and predicted parameters are written in each panel. The figures in the upper panel demonstrate precise physical parameter predictions made by the CNN for low-S/N data (blue), while the second panel displays higher-S/N data (green). The two lower panels illustrate two instances of an inaccurate prediction made by CNN for low- (purple) and high-S/N (cyan). To be considered an accurate prediction, the criterion is $|N_{\rm true}–N_{\rm pred}| < 0.33\,{\rm cm}^{-2}$ (σ_{90N} in log scale) and $|b_{\rm true}–b_{\rm pred}| < 7.8\,{\rm km\,s}^{-1}$ (σ_{90b}). Alternatively, $|N_{\rm true}–N_{\rm pred}| \geq 0.33\,{\rm cm}^{-2}$ and $|b_{\rm true}–b_{\rm pred}| \geq 7.8\,{\rm km\,s}^{-1}$ serve as the criterion for inaccurate prediction.

does not span the entire 301 pixels, we pad the remaining locations with a continuum added with Gaussian random noise. The standard deviation of the noise in this padded area is determined based on the median of the error vector within the line-free region surrounding the absorption line. To assess the performance of our ML algorithms on this dataset, we first apply our ML algorithm to these lines and record the results. For comparison purposes, we employ two distinct Voigt profile fitting algorithms. In addition to fitting done by D16, we also apply the VIPER algorithm (G17) to the same dataset.

Before analyzing the performance of our algorithms, it is important to highlight the differences in determining the number of components in absorption line systems between D16 and VIPER. Although VIPER adheres to the same significance level criteria for line identification as D16, their methods for determining whether lines are single or multicomponent differ. D16 used additional spectral information, such as higher-order lines or coincidental metal lines, to determine if lines are single or multicomponent. In contrast, VIPER employs the Akaike Information Criterion with Corrections (AICC) to decide the number of components. Furthermore, if AICC identifies more than one component, VIPER re-evaluates the significance level of all components, discarding any component with significance below 3σ. Due to these methodological variations, VIPER identifies 784 single lines and 296 double lines in the sample, differing from the 1137 single and 227 double lines found by D16. Given that VIPER, like our ML algorithm, does not utilize additional spectral information for component identifications and relies solely on Lyα lines as input, we anticipate our algorithm to exhibit improved performance in classification when using VIPER labels as compared to D16 labels.

6.1. Performance on HST COS data: Classification

First, we use the classification algorithm (see Sect. 4.1) to test the accuracy in predicting the number of absorption lines for the COS data. Similar to Sect. 4.1, we input the normalized flux to the model, and the output is the number of absorption lines. We visualize the model's performance by comparing the predicted labels with "true" labels obtained from the fits of D16 and VIPER.

In Fig. 12, we compare the results of our classification algorithm with the results of D16 (the left-hand panel) and VIPER

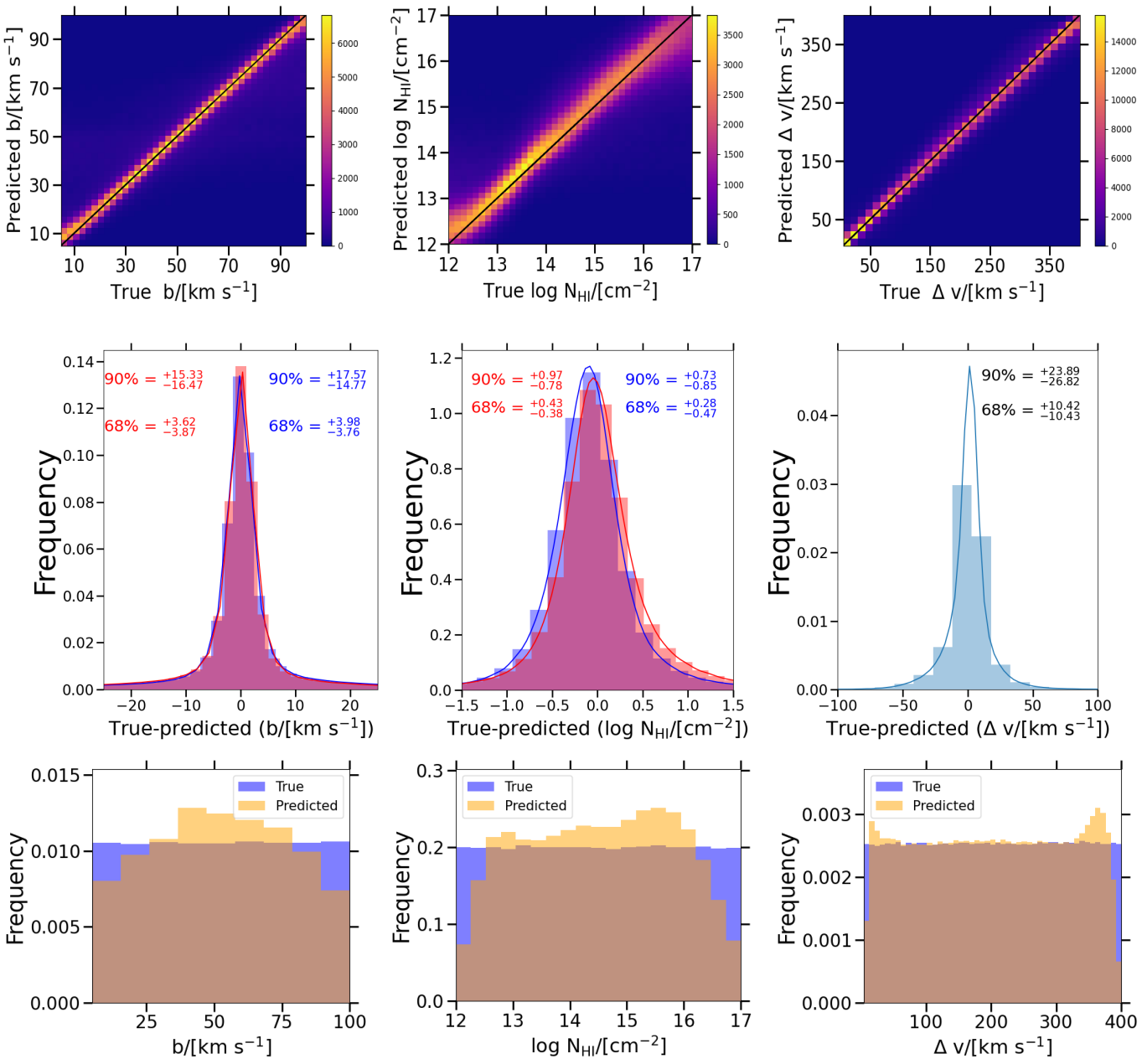

Fig. 10. Same as Fig. 8 but for a double absorption line. The upper left panel shows the true versus predicted values stacking b_1 and b_2. Similarly, the upper middle panels show column density. The right upper panel shows the velocity difference between the two absorption lines. The middle panels show the histogram of the difference between true and predicted values, with 90% and 68% values marked at the top for component 1 (red color) and component 2 (blue color) of the double absorption line. The lower panel shows the normalized histogram of true and predicted parameters.

(right-hand panel). In comparison with D16, we find that the double lines are identified accurately (93.4%). However, there are 22% of single lines identified as double lines, resulting in the accuracy for single lines to be just 78%. Upon further investigation, we found that most of the 22% single lines (152 out of 255) that are identified as double lines have $S/N < 20$. The overall accuracy of our algorithm with respect to the classification of single versus double lines identified by D16 is 80.21%, whereas other metrics such as recall is 77.57%, F1-score is 0.86.

In comparison with the results of VIPER (see the right-hand panel in Fig. 12), we find that 87% of the single lines are identified correctly, and only 13% were misclassified as double lines. For double lines, however, our algorithm could classify correctly for 80% of the lines and miss-classify 20% of the lines as single lines. 47 out of 58 misclassified double lines have $S/N < 20$. The overall accuracy of our algorithm with respect to the classification of single versus double lines identified by VIPER is 85.37%, which is better than the one obtained for identification D16, as per our expectations. Other metrics, such as sensitivity (recall), are 87.24%, and F1-score is 0.89.

Although our algorithm performs reasonably well, achieving an accuracy of 85% for labels from VIPER, this does not match the 93% accuracy obtained with our simulated data (refer to Fig. 4). The 8% decrease in accuracy is primarily due to a 10% reduction in true positives and true negatives for real data. We conducted a visual inspection of several instances where our

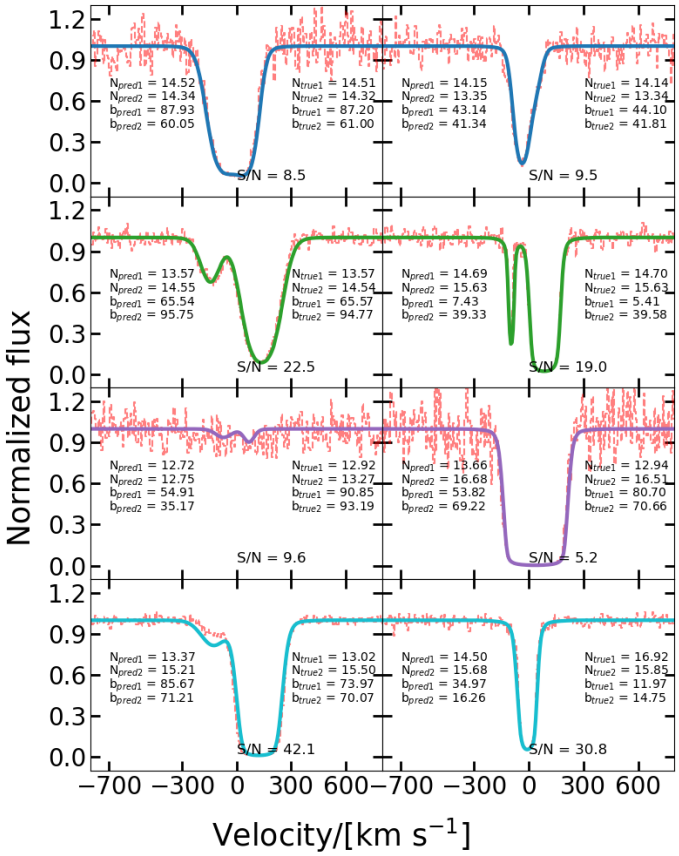

Fig. 11. Same as Fig. 9 but examples of the simulated test double absorption lines (red-dashed line) and the corresponding Voigt profile predicted by the CNN model (in solid colored lines). The parameters b and $N_{\rm HI}$ of two components (subscript 1 and 2) of the double line are written in each panel.

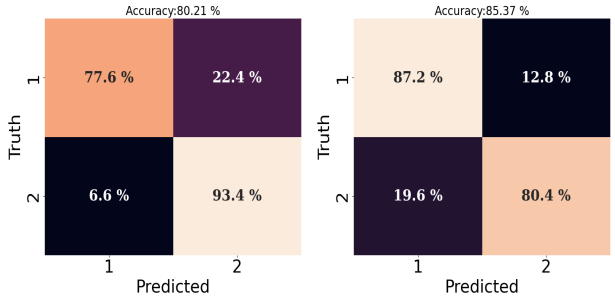

Fig. 12. The plot shows the prediction from the classification algorithm for the HST data with true labels from D16 (left) and VIPER (right). For D16 [VIPER], we find the accuracy = 80.21% [85.37%], sensitivity = 77.57% [87.24%], specificity = 93.39% [80.41%], precision = 98.33% [92.18%] and negative predictive value = 45.40% [70.41%].

algorithm was unsuccessful, yet we could not identify a clear reason for these failures. In many cases, the misclassifications appeared to stem from genuine confusion where double lines look single or are not prominent because of poor S/N. Examples of such cases, including those where the classification was accurate, are depicted in Fig. 13.

A potential reason for an 8–10% decrement in the performance of our algorithm on real data as compared to simulated data might be the inherent difficulty in emulating the real observations. To test this hypothesis, we decided to utilize our mock dataset (see Sect. 3.3) where we used exact same parameters of both D16 and VIPER fits, that is, identifications as well as b and $N_{\rm HI}$ values and modeled instrument effects, noises, and central wavelength. We then input those in our classification algorithm and compared them with D16 and VIPER labels. In this case, our results are shown in Fig. 14. We find that the accuracy [F1-score] for D16 fits is 96.13% [.97], and VIPER fits is 92.05% [.94]. These accuracies are comparable to the accuracy obtained in the simulated data. Therefore, it seems our hypothesis is correct, and there are some subtle differences between emulating the real data and the real data. Noise in real data originates from various sources, including detector noise and imperfections, cosmic rays, and photon noise. These noise patterns are inherently complex and can fluctuate over time and across different wavelengths. In contrast, simulated data incorporate noise patterns generated through simplified or idealized models and may not perfectly match the characteristics of real noise. This difference is even more apparent for regression analysis, as discussed in the following subsection.

6.2. Performance on HST COS data: Regression

After testing the classification of absorption lines, we now analyze the performance of the regression model on the identified absorption lines. This analysis entails utilizing flux data from D16 specifically for lines categorized as single (784) and double (296) by VIPER, totaling 1080 lines. Given the greater consistency observed in our classification algorithm with VIPER, we opt to utilize these line predictions. Among the 784 single lines, our classification model identified 684 as single lines, and among 296 double lines, our classification model identified 238 correctly.

First, we input the 620 (out of 684) single lines common across all studies (D16, VIPER and this work) into our algorithm to predict the values of b and $N_{\rm HI}$. In the upper panel of Fig. 15, we illustrate the distribution of b and $N_{\rm HI}$ from five datasets: D16 (purple lines), VIPER (red lines), CNN estimates of the real data (green lines), CNN estimates of D16 mocks (purple dashed lines), and CNN estimates of VIPER mocks (red dashed lines). We find a good consistency between the various studies. From our analysis of mock data, we discovered that centering the absorption lines significantly influences the accuracy of estimates. Therefore, all the absorption lines were centered before input into the ML model.

Our classification algorithm successfully predicted 882 single lines out of 1137 from the D16 dataset and 684 single lines out of 784 from the VIPER dataset. In the lower panel of Figure 15, we compare the CNN-predicted b and $N_{\rm HI}$ of these predicted single lines with the actual labels from D16 (882) and VIPER (684). The true versus true one-to-one line is overlaid in black for comparison. Interestingly, while classification results were more consistent with VIPER as compared to D16, this consistency does not extend to parameter estimation results. This discrepancy may stem from differences in how higher-order lines are utilized for classification in D16. However, fitting techniques remain independent of the use of higher-order lines in all the studies.

The similar inconsistencies between CNN prediction and D16 or VIPER is more clearly evident from Table 1. Table 1 provides details on the fraction of data exhibiting inconsistencies between the true (Data) and CNN-predicted estimates using σ_{90b} and σ_{90N} defined in Sect. 5.2. We mention the fraction of sample

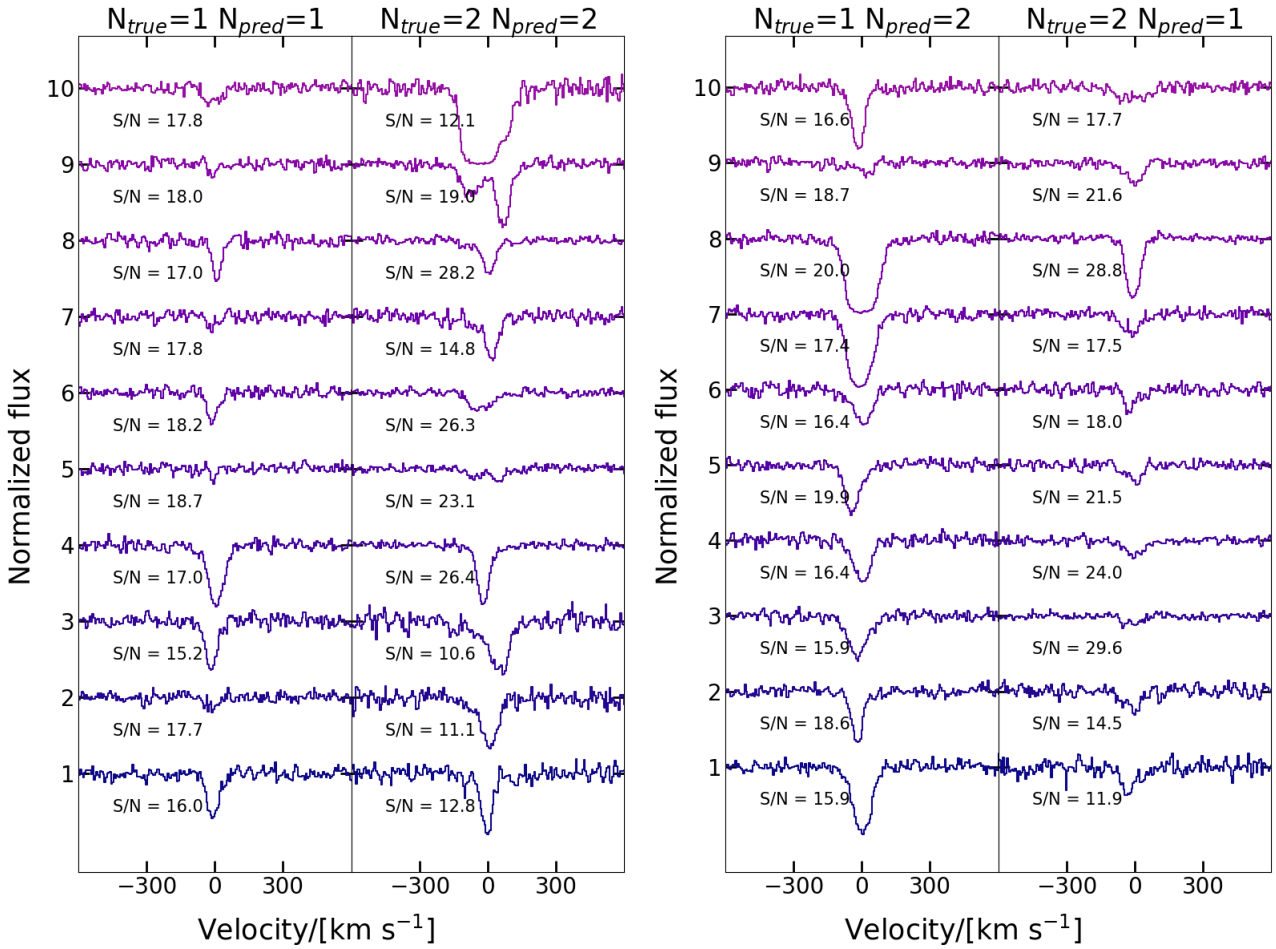

Fig. 13. Examples of classification performance of real observed absorption Lines. *Left panel*: Successful Classification – The panel displays instances where the classification algorithm accurately identifies single and double absorption lines, effectively matching the true labels. *Right panel*: Misclassified Cases – The two panels show examples of the misclassified single and double absorption lines, highlighting areas for improvement in certain challenging scenarios.

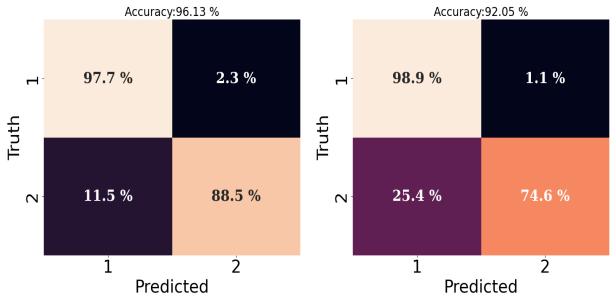

Fig. 14. Same as Fig. 12 but for the mock dataset. For D16 [VIPER], we find the accuracy = 96.13% [92.05%], sensitivity = 97.71% [98.94%], specificity = 88.50% [74.58%], precision = 97.62% [90.79%], and negative predictive value = 88.89% [96.54%].

with $\Delta b = b_{\rm true} - b_{\rm pred} > \sigma_{90b}$ or $\Delta N = N_{\rm true} - N_{\rm pred} > \sigma_{90N}$. In our analysis of simulated test data, we anticipate that the outcomes may vary depending on the S/N. Therefore, we examined the disparities of physical parameters between true and CNN predictions for the two S/N bins, that is, $S/N < 20$ and $S/N > 20$.

For the single lines, rows 1 and 3 of the table show the discrepancy between the CNN predictions for real data in comparison to the true parameters estimated from D16 and VIPER. Rows 2 and 4 show the same but for CNN predictions of the corresponding mock data. Notably, the mock tests exhibit reduced misestimations compared to real data, underscoring the challenge of accurately simulating real observations. Reiterating the fact that the real data are already centered before input to the CNN model, the better consistency between true and predicted values for mock datasets underscores the challenge of simulating real observations accurately. Nevertheless, these mock tests demonstrate the successful prediction of b and $N_{\rm HI}$ parameters by our ML algorithm, particularly when the simulated training dataset closely resembles the test dataset.

We observed that for $S/N > 20$, the results demonstrate greater consistency for column density, with a maximum of 2.7% [0.5%] of cases where predictions from the CNN do not align within σ_{90N} for the real [mock] dataset. However, for the prediction of b, we find similar results for $S/N < 20$ or $S/N > 20$. We find almost no absorption line at $S/N > 20$ that has both b and $N_{\rm HI}$ to be uncertain. Additionally, as indicated in Table 1, we observe that most of the inconsistencies arise from smaller values of b and $N_{\rm HI}$, particularly at higher S/N levels.

We extend our evaluation of the ML algorithm to encompass 118 (out of 238) common double absorption lines, which involve additional physical parameters including b_1, b_2, N_1, N_2, and Δv. In the upper-left panel of Fig. 16, we present stacked values of b_1 and b_2, while the upper-right panel illustrates stacked values of

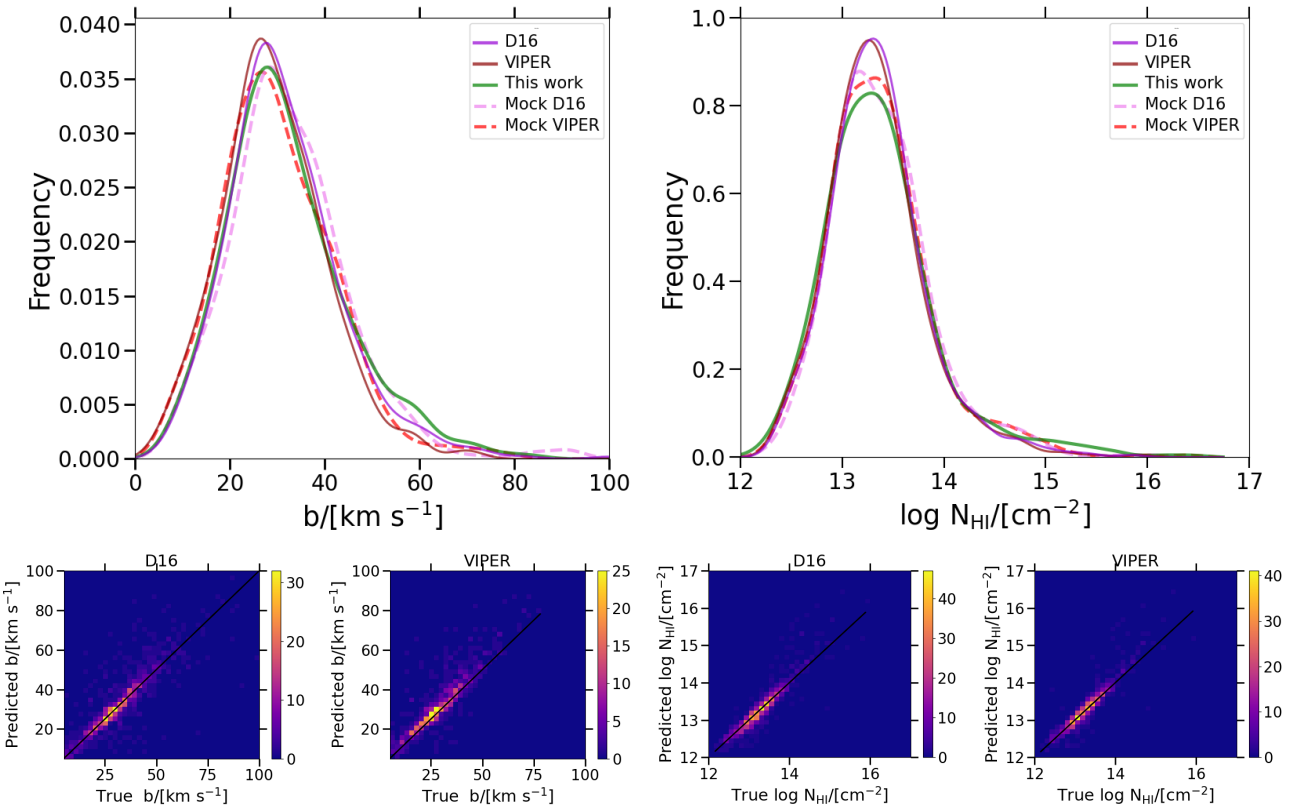

Fig. 15. Comparison of true parameter distributions and predictions for real observed single absorption lines. *Upper panels*: the histograms show the distribution of b and $\log N_{\rm HI}$ for a single absorption line common in all three studies estimated by the CNN model in this work (green) overplotted with distributions from D16 (purple) and VIPER (red). The distributions of the CNN predictions for the corresponding mocks are shown in dashed lines.*Lower panels*: the prediction of single line parameters from CNN compared to the true values from D16 and VIPER.

Table 1. Summary of the percentage of absorption lines with a discrepancy between true physical parameters and CNN predictions exceeding σ_{90} (90% percentile derived from simulated test data).

Sno.	Lines	Data	$\Delta b > \sigma_{90b}$[median N]		$\Delta N > \sigma_{90N}$[median b]		$\Delta b > \sigma_{90b}$&$\Delta N > \sigma_{90N}$	
			$S/N < 20$	$S/N > 20$	$S/N < 20$	$S/N > 20$	$S/N < 20$	$S/N > 20$
1	Single	D16	12.8% [13.35]	14.8% [12.70]	9.4% [32.50]	2.4% [34.60]	5.0%	0.3%
2	Single	D16 mock	6.2% [13.24]	7.5% [12.77]	2.8% [28.70]	0.5% [18.80]	0.4%	0.0%
3	Single	VIPER	14.1% [13.22]	13.0% [12.74]	7.6% [29.17]	2.7% [34.71]	3.5%	0.4%
4	Single	VIPER mock	6.6% [13.16]	5.2% [12.64]	5.2% [24.32]	0.0%	0.0%	0.0%
5	Double C1	D16	33.8% [13.34]	25.0% [12.84]	11.0% [35.90]	5.3% [29.80]	4.4%	2.6%
6	Double C1	D16 mock	8.0% [13.16]	8.0% [12.82]	4.0% [24.80]	0.0%	0.0%	0.0%
7	Double C1	VIPER	25.5% [13.23]	23.5% [12.76]	11.5% [25.20]	8.6% [27.34]	4.5%	4.9%
8	Double C1	VIPER mock	11.5% [13.28]	9.3% [12.55]	4.1% [26.21]	2.7% [38.12]	0.0%	1.3%
9	Double C2	D16	19.9% [13.32]	26.3% [12.78]	14.0% [32.70]	6.6% [38.40]	5.1%	3.9%
10	Double C2	D16 mock	5.6% [13.18]	4.0% [12.71]	7.2% [22.50]	1.3% [36.40]	0.0%	0.0%
11	Double C2	VIPER	26.1% [13.08]	29.6% [12.54]	12.7% [30.49]	11.1% [27.78]	3.8%	4.9%
12	Double C2	VIPER mock	15.5% [12.98]	16.0% [12.99]	5.4% [27.36]	4.0% [20.10]	0.7%	2.7%

Notes. σ_{90b} and σ_{90N} for single line is taken from Fig. 8 and for double line from Fig. 10. Single-line data are shown in rows 1–4, double-line component 1 (C1) in rows 5–8, and double-line component 2 (C2) in rows 9–12.

log N_1 and log N_2. Notably, our ML algorithm predicts b values significantly higher than those estimated by both D16 and VIPER. Similarly, the $N_{\rm HI}$ estimates generated by our ML algorithm consistently surpass those obtained by D16 and VIPER. Similar to single lines, the S/N and centering of double lines significantly impact the results. This impact is evident from our analysis of mock dashed lines. However, centering double absorption lines poses additional challenges and hence is not performed.

Our classification algorithm successfully predicted 212 double lines out of 227 from the D16 dataset and 238 double lines out of 296 from the VIPER dataset. In the lower panel of Fig. 16, we compare CNN predicted b and $N_{\rm HI}$ of these double lines with true estimates from D16 and VIPER. Interestingly, the scatter of predicted values of log $N_{\rm HI}$ for double lines increases compared to single absorption lines, consistent with results from simulated test data (Fig. 10). This suggests that predictions for double lines are less accurate than those for single lines.

Rows 5-12 in Table 1 lists the percentage of sample with discrepancy $\Delta b > \sigma_{90b}$ and $\Delta N > \sigma_{90N}$ for double lines. Here the σ_{90b} and σ_{90N} are taken from Fig. 10 for first and second component. Notably, we observe a 33% sample with $\Delta b > \sigma_{90b}$ for $S/N < 20$ compared to D16 true estimates (see Table. 1).

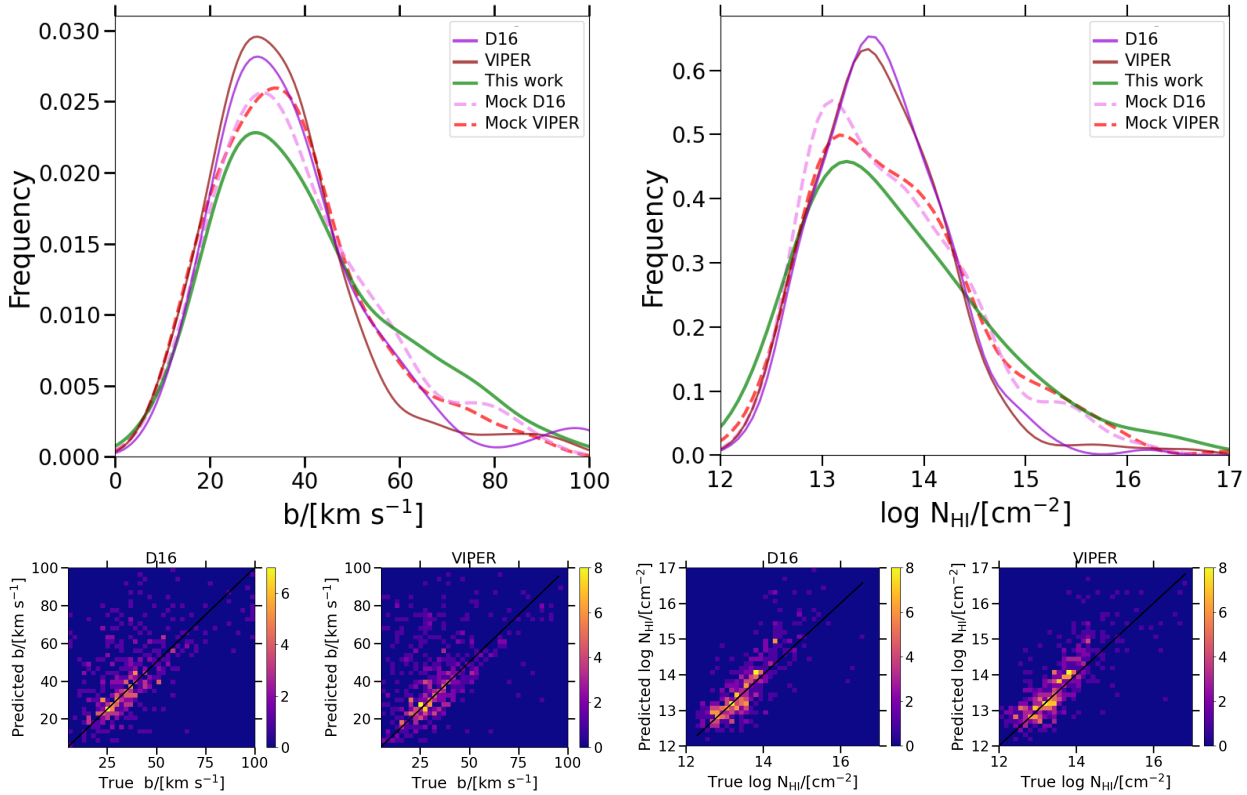

Fig. 16. Same as Fig. 15 but for parameters extracted from double absorption lines.

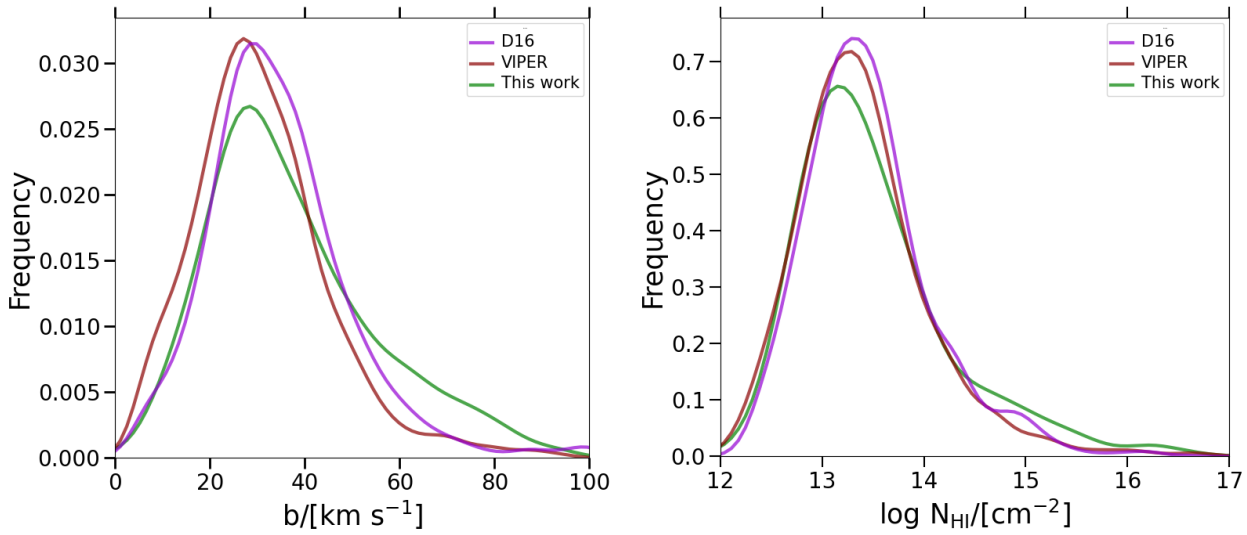

Fig. 17. b and log N_{HI} extracted from all the absorption lines. All the lines identified as single lines by our classification model are inputted into regression model 1, and all the lines identified as double lines by our classification model are inputted into regression model 2. The output from both of these models is plotted as the green line in comparison to the b and log N_{HI} estimates by D16 (purple) and VIPER (red) extracted from their single and double identified lines.

However, this reduces to 8% when utilizing "centered" mock data (rows 6). This trend of lower inconsistencies for mock data is evident for both components and both studies by D16 and VIPER. Consistent with single absorption lines, predictions for double lines exhibit fewer inconsistencies when $\Delta N > \sigma_{90N}$ and $S/N > 20$.

As mentioned previously, while plotting the b and N_{HI}, we considered only the common lines. However, in a real-life scenario, we would acquire single and double lines classified by our initial CNN model, which would then serve as input to the respective regression models, yielding distributions of b and N_{HI}. Out of the 784 single lines, our classification model correctly identified 684 as single lines and misclassified 58 double lines as single, resulting in a total of 742 single lines. Similarly, our model identified 238 double lines and misclassified 100 single lines as double, resulting in a total of 338 double lines.

Consequently, we input 742 single lines into our first regression model, resulting in 742 sets of b and $N_{\rm HI}$ values. Similarly, we input 338 double lines, as identified by our classification model, into our second regression model, resulting in 338×2 values of b and $N_{\rm HI}$ values. In Fig. 17, we depict this distribution of b and $N_{\rm HI}$ in comparison to D16 and VIPER. We observe that the distribution of column density remains comparable. However, the Doppler width values are affected by three issues: firstly, 30% of double lines are contaminated by single lines, and secondly, the parameter estimation for double lines is less accurate, and lastly, the centering of the lines is not performed for double lines.

Overall, our analysis reveals that while our algorithm performs reasonably well for single lines, it falls short of the performance achieved by D16 and VIPER in accurately reproducing the b and $N_{\rm HI}$ values for double lines. This discrepancy highlights that part of the issue lies within our classification algorithm, which does not perform as anticipated. To further investigate, we tested our regression algorithms on the mock datasets we had prepared. In these tests, as highlighted in the previous section, the classification algorithm shows performance in line with expectations. We also find from the mock analysis that the centering of the absorption lines plays a vital role in calculating the physical parameters using the regression algorithm. Moreover, the predictions are found to be more consistent for $S/N > 20$.

The successful application of regression analysis to both real observed data and mock data, in conjunction with the prior classification of absorption lines, validates the effectiveness of our comprehensive approach. Our study demonstrates that the ML algorithm yields statistically comparable results to traditional fitting methods. Furthermore, its minimal computational time renders it highly advantageous for large datasets. For instance, while manually fitting a Voigt profile takes at least 1–2 min and semiautomated codes like VIPER require 1–2 s, the ML algorithm can provide this information in just 0.0002 s. This efficiency underscores the potential of ML in efficiently managing complex absorption line analysis, especially considering the substantial demands of data processing.

7. Main results and discussion

The low-redshift ($z < 1$) Lyα forest is crucial for understanding the evolution of the IGM, galaxy formation, and unresolved baryon fractions. Despite its significant potential, extracting information from the Lyα forest presents considerable challenges, especially when using Voigt profile fitting for its numerous absorption lines. Therefore, developing an ML algorithm for Voigt profile fitting is essential for the success of upcoming large astronomical surveys.

In this study, we developed a two-part ML algorithm, FLAME, using CNNs to identify the number of Voigt profiles that best fit a given Lyα absorption system and then determine the b and $N_{\rm HI}$ for each profile (see Fig. 1). We trained these CNNs with approximately 10^6 low-z Voigt profiles, synthesized to mimic real data from HST-COS. Given that the majority (96%) of the Lyα lines in the existing high S/N HST-COS data can be fitted with single or double-line profiles (see D16 and G17), we designed our first ML algorithm to classify the lines as either single or double profiles. The second stage of our ML algorithm comprises two networks: the first targets single-line profiles to determine b and $N_{\rm HI}$, while the second is tailored for double-line profiles, determining b and $N_{\rm HI}$ for both components, as well as their velocity separation.

Evaluating the algorithms on simulated Lyα lines showcases its impressive performance. The classification algorithm (Fig. 3) correctly identifies ≥98% of the single absorptions and ≥90% of the double lines (Fig. 4). The regression algorithms in the second stage for single Voigt profiles determine values of b and $N_{\rm HI}$ robustly. For 68% of the single lines, the predicted b values lie within within $^{+2.29}_{-1.06}$ km s^{-1}, and log $N_{\rm HI}$ within $^{+0.18}_{-0.16}$ cm^{-2} (Fig. 8). Whereas for 68% of double lines, the regression algorithm predicts, b within $^{+3.80}_{-3.82}$ km s^{-1}, log $N_{\rm HI}$ within $^{+0.35}_{-0.43}$ cm^{-2} and velocity separation of both lines Δv within $^{+10.42}_{-10.43}$ km s^{-1} of the true values (Fig. 10). The model demonstrates a close match between predicted and actual parameters, confirming its accuracy. The minimal scatter and negligible bias in predictions underscore its reliability across a broad range of parameters. Nonetheless, the model shows limitations with nearly saturated lines, data with high noise levels, or significant optical depth contrasts, suggesting areas for further enhancement.

We evaluated the performance of our ML algorithms across different parameters, including S/N, b-parameter, $N_{\rm HI}$, and the velocity difference (Δv) between two absorption lines. Fig. 6 shows the classification accuracy trends with respect to these parameters. As expected, accuracies are notably lower for small values of S/N in simulated datasets. However, excluding cases with these parameters in the lower percentile ($S/N > 20$) consistently yields accurate accuracy above 94.2%, which is promising. Though in this analysis, we test for the wide range of parameters, however, if we established thresholds for these parameters, indicating conditions under which the model provides accurate estimates for b, $N_{\rm HI}$, and S/N above 97.5%. The identified thresholds are as follows: $S/N > 20$, b-parameter <40 km s^{-1} and $N_{\rm HI} < 14$ cm^{-2}.

We applied the algorithm to HST-COS data, focusing on a selected subset of 1,400 absorption lines, and evaluated its performance against two methods of Voigt profile fitting: one using the fits provided by D16 and the other using the automated Voigt profile fitting code VIPER G17. We observed that the accuracy of our classification algorithm decreased by approximately 10% compared to the simulated dataset. Specifically, we achieved an accuracy of 80% in comparison with the fits by D16 and 85% when compared with the fits from VIPER. Nevertheless, the regression algorithms showed reasonably good agreement in predicting b and $N_{\rm HI}$ values, closely matching their distributions from both VIPER and D16. Any discrepancies in the distribution mainly arise from the inherent difficulty in emulating the real observations.

In all our trials, we observed that the real dataset exhibits reduced accuracy compared to the simulated dataset. To understand the reasons behind the lower accuracy of our algorithms on real data, we generated mock data mirroring the parameters of real data fits from both D16 and VIPER. Applying our algorithm to this mock data, we found that the accuracy remains stable, showcasing an agreement between the predicted number of lines in the profiles and the values of b and $N_{\rm HI}$. This finding suggests that the differences in accuracies between simulated and real datasets can be to some extent ascribed to the inherent complexities in modeling real data, which often contain nuances difficult to replicate accurately in simulations.

The simulated data may fall short of capturing the complexities, variations, and noise inherent in the real data. For example, accurately replicating the inherent noise patterns and instrument-specific calibration errors in simulations is extremely challenging. Strategies such as incorporating a fraction of real data into the training set may not yield the desired effectiveness in this case, as the real data sample size is much lower compared to

the size of the simulated data used in this analysis. The ongoing and upcoming large-scale spectroscopic surveys, such as DESI, 4MOST, WEAVE, and PFS, are designed to enhance the availability of real spectra significantly. This influx of real data is expected to play a crucial role in enabling ML models to improve their performance further at higher redshifts. However, currently for lower redshifts, we do not have such large-scale spectroscopic surveys and hence rely on simulated data.

Our study highlights the effectiveness of ML in analyzing Lyα absorption lines in quasar spectra, showcasing capability for both classifying the number of lines and estimating physical parameters. Through evaluations with real and mock data, we show that ML not only matches the accuracy of traditional methods but also significantly reduces computational effort, proving especially beneficial for large datasets where traditional approaches falter due to time and labor demands.

8. Summary

We present FLAME, a two-part ML algorithm to fit low-z Lyα absorption lines using CNNs. This algorithm effectively determines the optimal number of Voigt profiles for Lyα absorption systems and determines the Doppler parameter b and neutral hydrogen column density $N_{\rm HI}$ for each profile. FLAME shows impressive accuracy, correctly identifying over 98% of single absorptions and over 90% of double lines in simulated data that mimic the real Lyα absorption lines from HST-COS. For 90% of single lines, the FLAME accurately predicts b values within $\approx\pm8\,{\rm km\,s^{-1}}$ and $\log N_{\rm HI}/{\rm cm^2}$ values within $\approx\pm0.3$. For double lines, it shows slightly lower accuracy, predicting b within $\approx\pm15\,{\rm km\,s^{-1}}$ and $\log N_{\rm HI}/{\rm cm^2}$ within $\approx\pm0.8$.

Despite its impressive performance on simulated data, FLAME's accuracy in classifying lines decreases by about 10% when applied to real HST-COS data. Nevertheless, there is good agreement between the predicted b and $N_{\rm HI}$ distributions with other Voigt profile-fitting algorithms, such as VIPER (G17), and the fits from D16 for single lines. However, the fits need to be improved for double lines.

Our analysis with mock HST-COS data, crafted to reflect the parameters of real data, shows that FLAME can maintain stable accuracy, mirroring its success with simulated datasets. This underscores that the primary discrepancies in accuracy when applied to simulated versus real data stem from the challenges in fully capturing the complexities of real data in the simulated training dataset. Despite these hurdles, FLAME successfully demonstrates the feasibility of employing ML to fit Voigt profiles, highlighting the potential for ML in analyzing absorption lines.

Moving forward, we aim to refine FLAME, particularly to improve its performance with real data. Our efforts will include examining the specific challenges posed by real data and assessing how these affect accuracy. Including additional spectral information, such as metal lines, similar to the approach taken by D16, appears to be a promising method for enhancing accuracy. Additionally, incorporating real data into our training samples is a key strategy we believe will help align FLAME more closely with practical applications. The upcoming influx of data from spectroscopic surveys like DESI, 4MOST, WEAVE, and PFS is expected to benefit ML models significantly by providing richer training datasets. This effort will lead to more precise characterizations of the Lyα forest by future developments of FLAME.

Acknowledgements. We thank our anonymous referee for their positive feedback and useful comments. PJ acknowledges Dr. hab Maciej Bilicki for useful discussion. The Polish National Science Center supported PJ through grant no. 2020/38/E/ST9/00395. VK is supported through the INSPIRE Faculty Award (No. DST/INSPIRE/04/2019/001580) of the Department of Science and Technology (DST), India, and by NASA through grant number HST-AR-17048.003 from the Space Telescope Science Institute, which is operated by the Associated Universities for Research in Astronomy, Inc., under NASA contract NAS 5-26555. A partial finances during the manuscript are supported by the Young Scientist Award 2023, won by PJ. PJ acknowledges the CFT and ARIES computer facility to provide high-performance computers. MV acknowledges support from DST-SERB in the form of a core research grant (CRG/2020/1657).

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, arXiv e-prints [arXiv:1603.04467]
Akhazhanov, A., More, A., Amini, A., et al. 2022, MNRAS, 513, 2407
Alam, S., Aubert, M., Avila, S., et al. 2021, Phys. Rev. D, 103, 083533
Bainbridge, M. B., & Webb, J. K. 2017, MNRAS, 468, 1639
Baur, J., Palanque-Delabrouille, N., Yèche, C., et al. 2017, JCAP, 2017, 013
Becker, G. D., Hewett, P. C., Worseck, G., & Prochaska, J. X. 2013, MNRAS, 430, 2067
Bolton, J. S., & Haehnelt, M. G. 2007, MNRAS, 382, 325
Bolton, J. S., Viel, M., Kim, T. S., Haehnelt, M. G., & Carswell, R. F. 2008, MNRAS, 386, 1131
Bolton, A. S., Schlegel, D. J., Aubourg, É., et al. 2012, AJ, 144, 144
Bolton, J. S., Puchwein, E., Sijacki, D., et al. 2017, MNRAS, 464, 897
Bolton, J. S., Gaikwad, P., Haehnelt, M. G., et al. 2022, MNRAS, 513, 864
Bosman, S. E. I., Fan, X., Jiang, L., et al. 2018, MNRAS, 479, 1055
Breiman, L. 2001, Mach. Learn., 45, 5
Busca, N. G., Delubac, T., Rich, J., et al. 2013, A&A, 552, A96
Carswell, R. F., & Webb, J. K. 2014, VPFIT: Voigt profile fitting program, Astrophysics Source Code Library [record ascl:1408.015]
Cheng, T.-Y., Cooke, R. J., & Rudie, G. 2022, MNRAS, 517, 755
Cortes, C., & Vapnik, V. 1995, Mach. Learn., 20, 273
Danforth, C. W., Keeney, B. A., Tilton, E. M., et al. 2016, ApJ, 817, 111
Davé, R., Anglés-Alcázar, D., Narayanan, D., et al. 2019, MNRAS, 486, 2827
de Dios Rojas Olvera, J., Gómez-Vargas, I., & Vázquez, J.A. 2022, Universe, 8
de Graaff, A., Cai, Y.-C., Heymans, C., & Peacock, J. A. 2019, A&A, 624, A48
de Jong, R. S., Bellido-Tirado, O., Chiappini, C., et al. 2012, in Ground-based and Airborne Instrumentation for Astronomy IV, eds. I. S. McLean, S. K. Ramsay, & H. Takami, SPIE Conf. Ser., 8446, 84460T
Eilers, A.-C., Davies, F. B., & Hennawi, J. F. 2018, ApJ, 864, 53
Fan, X., Strauss, M. A., Becker, R. H., et al. 2006, AJ, 132, 117
Flaugher, B., & Bebek, C. 2014, in Ground-based and Airborne Instrumentation for Astronomy V, eds. S. K. Ramsay, I. S. McLean, & H. Takami, Int. Soc. Opt. Phot. (SPIE), 9147, 91470S
Gaikwad, P., Choudhury, T. R., Srianand, R., & Khaire, V. 2017a, MNRAS, 474, 2233
Gaikwad, P., Khaire, V., Choudhury, T. R., & Srianand, R. 2017b, MNRAS, 466, 838
Gaikwad, P., Srianand, R., Choudhury, T. R., & Khaire, V. 2017c, MNRAS, 467, 3172
Gaikwad, P., Srianand, R., Haehnelt, M. G., & Choudhury, T. R. 2021, MNRAS, 506, 4389
Gholamalinezhad, H., & Khosravi, H. 2020, arXiv e-prints [arXiv:2009.07485]
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (MIT Press)
Gurvich, A., Burkhart, B., & Bird, S. 2017, ApJ, 835, 175
Hiss, H., Walther, M., Hennawi, J. F., et al. 2018, ApJ, 865, 42
Hopkins, P. F., Hernquist, L., Cox, T. J., & Kereš, D. 2008, ApJS, 175, 356
Hossin, M., & Sulaiman, M. N. 2015, Int. J. Data Min. Knowl. Manage. Process, 5, 01
Hu, T., Khaire, V., Hennawi, J. F., et al. 2023, arXiv e-prints [arXiv:2311.17895]
Huang, L., Croft, R. A. C., & Arora, H. 2021, MNRAS, 506, 5212
Iršič, V., Viel, M., Haehnelt, M. G., et al. 2017, Phys. Rev. D, 96, 023522
Khaire, V. 2017, MNRAS, 471, 255
Khaire, V., & Srianand, R. 2015, ApJ, 805, 33
Khaire, V., Walther, M., Hennawi, J. F., et al. 2019, MNRAS, 486, 769
Khaire, V., Hu, T., Hennawi, J.F., et al. 2023, arXiv e-prints [arXiv:2311.08470]
Kingma, D. P., & Ba, J. 2014, arXiv e-prints [arXiv:1412.6980]
Krogager, J.-K. 2018, arXiv e-prints [arXiv:1803.01187]
LeCun, Y., Bengio, Y., & Hinton, G. 2015, Nature, 521, 436
Lee, J., & Shin, M.-S. 2021, AJ, 162, 297
Liang, C., & Kravtsov, A. 2017, arXiv e-prints [arXiv:1710.09852]
Lidz, A., Furlanetto, S. R., Oh, S. P., et al. 2011, ApJ, 741, 70
Liu, T., Cao, S., Zhang, S., et al. 2021, Eur. Phys. J. C, 81, 903
Macquart, J. P., Prochaska, J. X., McQuinn, M., et al. 2020, Nature, 581, 391
Madau, P., & Dickinson, M. 2014, ARA&A, 52, 415

Matoba, K., Dimitriadis, N., & Fleuret, F. 2022, arXiv e-prints [arXiv:2203.01016]
McDonald, P., Seljak, U., Burles, S., et al. 2006, ApJS, 163, 80
McQuinn, M., Lidz, A., Zaldarriaga, M., et al. 2009, ApJ, 694, 842
Meiksin, A. A. 2009, Rev. Mod. Phys., 81, 1405
Parhi, R., & Nowak, R. 2019, IEEE Signal Process. Lett., 27, 1779
Parks, D., Prochaska, J. X., Dong, S., & Cai, Z. 2018, MNRAS, 476, 1151
Pieri, M. M., Bonoli, S., Chaves-Montero, J., et al. 2016, in SF2A-2016: Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics, eds. C. Reylé, J. Richard, L. Cambrésy, et al., 259
Rauch, M. 1998, ARA&A, 36, 267
Schaye, J. 2001, ApJ, 559, 507
Shull, J. M., France, K., Danforth, C. W., Smith, B., & Tumlinson, J. 2010, ApJ, 722, 1312
Shull, J. M., Harness, A., Trenti, M., & Smith, B. D. 2012, ApJ, 747, 100
Springel, V. 2005, MNRAS, 364, 1105
Stemock, B., Churchill, C. W., Lee, A., et al. 2024, AJ, 167, 287
Tanimura, H., Aghanim, N., Douspis, M., Beelen, A., & Bonjean, V. 2019, A&A, 625, A67
Tillman, M. T., Burkhart, B., Tonnesen, S., et al. 2023, ApJ, 945, L17
Vattis, K., Toomey, M. W., & Koushiappas, S. M. 2021, Phys. Rev. D, 104, 123541
Veiga, M. H., Meng, X., Gnedin, O. Y., Gnedin, N. Y., & Huan, X. 2021, arXiv e-prints [arXiv:2107.09082]
Viel, M., Schaye, J., & Booth, C. M. 2013, MNRAS, 429, 1734
Viel, M., Haehnelt, M. G., Bolton, J. S., et al. 2017, MNRAS, 467, L86
Walther, M., Oñorbe, J., Hennawi, J. F., & Lukić, Z. 2019, ApJ, 872, 13
Weinberger, R., Springel, V., Hernquist, L., et al. 2017, MNRAS, 465, 3291
Worseck, G., Prochaska, J. X., McQuinn, M., et al. 2011, ApJ, 733, L24
Yèche, C., Palanque-Delabrouille, N., Baur, J., & du Mas des Bourboux, H. 2017, JCAP, 2017, 047
You, Y., Gitman, I., & Ginsburg, B. 2017, arXiv e-prints [arXiv:1708.03888]