






# RADYNVERSION: Learning to Invert a Solar Flare Atmosphere with Invertible Neural Networks

Christopher M. J. Osborne<sup>1</sup> , John A. Armstrong<sup>1</sup> , and Lyndsay Fletcher<sup>1,2</sup> <sup>1</sup> SUPA School of Physics and Astronomy, University of Glasgow, Glasgow, G12 8QQ, UK; [c.osborne.1@research.gla.ac.uk](mailto:c.osborne.1@research.gla.ac.uk)<sup>2</sup> Roseland Centre for Solar Physics, University of Oslo, P.O. Box 1029 Blindern, NO-0315 Oslo, Norway

Received 2019 January 28; revised 2019 February 15; accepted 2019 February 15; published 2019 March 13

## Abstract

During a solar flare, it is believed that reconnection takes place in the corona followed by fast energy transport to the chromosphere. The resulting intense heating strongly disturbs the chromospheric structure and induces complex radiation hydrodynamic effects. Interpreting the physics of the flaring solar atmosphere is one of the most challenging tasks in solar physics. Here we present a novel deep-learning approach, an invertible neural network, to understanding the chromospheric physics of a flaring solar atmosphere via the inversion of observed solar line profiles in H $\alpha$  and Ca II  $\lambda$ 8542. Our network is trained using flare simulations from the 1D radiation hydrodynamic code RADYN as the expected atmosphere and line profile. This model is then applied to single pixels from an observation of an M1.1 solar flare taken with the Swedish 1 m Solar Telescope/CRISP Imaging Spectropolarimeter instrument just after the flare onset. The inverted atmospheres obtained from observations provide physical information on the electron number density, temperature, and bulk velocity flow of the plasma throughout the solar atmosphere ranging from 0 to 10 Mm in height. The density and temperature profiles appear consistent with the expected atmospheric response, and the bulk plasma velocity provides the gradients needed to produce the broad spectral lines while also predicting the expected chromospheric evaporation from flare heating. We conclude that we have taught our novel algorithm the physics of a solar flare according to RADYN and that this can be confidently used for the analysis of flare data taken in these two wavelengths. This algorithm can also be adapted for a menagerie of inverse problems providing extremely fast ( $\sim 10 \mu\text{s}$ ) inversion samples.

*Key words:* line: profiles – methods: data analysis – Sun: atmosphere – Sun: chromosphere – Sun: flares – Sun: general

## 1. Introduction

The current and next generation of solar observations, with their high spatial, temporal, and spectral resolution, present a significant analysis challenge, as does the increasing complexity and realism of the models with which the data are confronted. The two go hand-in-hand: ever-increasing resolution reveals observational phenomena that cannot be understood using convenient theoretical simplifications, while the inclusion of “realistic physics” in models (often taken to mean, e.g., multifluid effects and nonequilibrium processes) motivates observational testing at higher and higher resolution. The challenge of model-data comparison grows accordingly and drives us to seek new approaches.

This paper deals specifically with combining models and observations to learn about the structure of the solar atmosphere during a solar flare. The underlying motivation for such investigations is to understand how the energy released in a flare is transported through and dissipated in the solar atmosphere, primarily in the solar chromosphere, where most of the flare’s radiation originates (appearing mostly in the optical and UV; e.g., Kretzschmar 2011; Milligan et al. 2014). However, the route to this is complicated. The observed chromospheric radiation—a combination of optically thin (mostly extreme UV) and optically thick (mostly UV to optical)—carries information about the temperature, density, and velocity structure of the solar chromosphere, which

evolves rapidly with time as it heats. This structure is determined by the pre-flare chromosphere and the characteristics of the flare energy input. The task is to work out the chromospheric structure from the radiation emitted and use this to constrain the properties of the energy input. The picture is complicated because the heating is very intense, between  $10^{10}$  and  $10^{12} \text{ erg cm}^{-2} \text{ s}^{-1}$  (Fletcher et al. 2007; Krucker et al. 2011), compared to the  $\sim 10^7 \text{ erg cm}^{-2} \text{ s}^{-1}$  (Withbroe & Noyes 1977) needed to balance radiative losses in the nonflaring chromosphere. In addition, there is abundant evidence for nonthermal particles and flows close to the sound speed, meaning that simplifying assumptions such as hydrostatic or local thermodynamic equilibrium are unlikely to be valid.

We focus here on optically thick emission lines from the upper photosphere and chromosphere. These lines encode information about the atmospheric structure; typically, the emergent radiation in the line core is formed higher up in the atmosphere than in the line wings. A number of techniques exist for “inverting” optically thick line profiles to recover the structure of the atmosphere that emitted them, though most have been developed for the inversion of spectropolarimetric information to also include the magnetic field, which is not our concern at present. These include analytic methods employing the Milne–Eddington approximation for frequency-independent opacity in an LTE atmosphere (e.g., Skumanich & Lites 1987), the non-LTE codes NICOLE (Socas-Navarro et al. 2000) and HAZEL (Asensio Ramos et al. 2008), and the non-LTE code STiC (de la Cruz Rodriguez et al. 2019), which can treat multiple atomic species and a complex atmospheric stratification. In essence, these all iterate the output of a forward model toward the observed spectropolarimetric line profiles (note that an alternative approach for solving the inverse problem for the chromospheric temperature structure from an integral form was

demonstrated by Metcalf et al. 1990). They have also not been developed with the flare chromosphere in mind, though NICOLE has been used by Kuridze et al. (2017, 2018) for flares. While non-LTE calculations are included in many codes, hydrostatic equilibrium is uniformly assumed. Instead, the most frequently used approach for flares is forward modeling with codes such as RADYN (Carlsson & Stein 1992, 1997; Allred et al. 2005, 2015) in an attempt to match with observed spectral lines. The energy input to the model is specified according to observed properties, when possible (i.e., the energy input by nonthermal electrons deduced from hard X-rays). This approach has produced some notable insights into the properties of the flare chromosphere from both line and continuum emission (e.g., Kuridze et al. 2015; da Costa et al. 2016; Kowalski et al. 2017; Simões et al. 2017). However, iterating these models toward agreement with observations is not practical, and in some cases, reproducing features of the observations pushes the models in ways that are difficult to justify observationally (e.g., the long beam injection times required by Kennedy et al. 2015). Also, while manageable for small samples of data, this “trial-and-error” approach cannot realistically be scaled up to take advantage of the high volumes of data from new instruments. Furthermore, in cases where the energy input by nonthermal electrons cannot be constrained because of a lack of complementary observations, it is hard to know where to start among the vast range of model possibilities.

Here we take a different track, exploiting developments in machine learning to efficiently recover RADYN-like atmospheres from spectral-line profiles. We design and train an invertible neural network (INN; similar to that introduced in Dinh et al. 2016; Ardizzone et al. 2018) to learn the output  $H\alpha$  and  $Ca II$  8542 Å spectral lines corresponding to many thousands of RADYN atmospheric solutions, and vice versa. The network proves capable of inverting model RADYN spectral-line profiles to accurately generate the corresponding RADYN atmospheric parameters, giving us confidence in its ability to recover reasonable, realistic atmospheres from observed flare spectral data. We demonstrate the method on data taken by the CRisp Imaging SpectroPolarimeter (CRISP) instrument on the Swedish Solar Telescope (Scharmer et al. 2003, 2008). The method is fast, producing both atmospheric parameters and a measure of their uncertainties in about  $44.7 \mu\text{s}$  measurement<sup>-1</sup> on a GPU. This makes application to large data sets feasible.

This initial paper is intended to demonstrate proof of concept, underpinning future in-depth analysis of flares. In Section 2, we describe the principles of INNs, and Section 3 covers how our network is trained and validated on RADYN models. In Section 4, we present the first inversion using this method of real flare data, and we end with discussion and conclusions in Section 5.

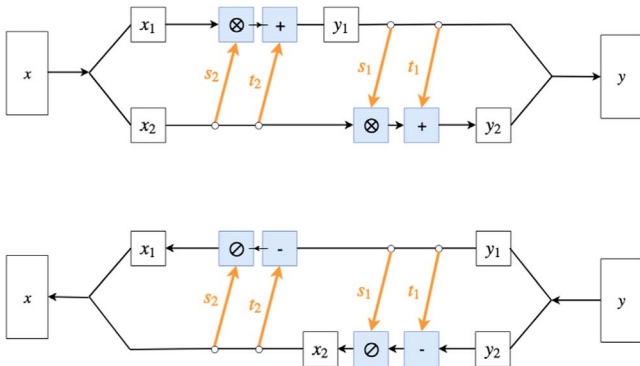
## 2. INNs

An inverse problem is one in which a set of measurements is used to deduce the properties of the system that caused them. It is usually the case that information about the system is missing because of the properties of the medium or the complexity of the physics involved. The example presented in this paper is that of deducing the plasma parameters of the chromosphere that are three-dimensional quantities, whereas we only observe the chromosphere as two-dimensional images at a given wavelength from an instrument such as the Swedish Solar Telescope CRISP instrument (Scharmer et al. 2003, 2008). We wish to learn about this missing information, as it will constrain

our model of the physical system producing the observations. Formally for any process, there exists a function  $y = f(x)$  that maps the input of physical parameters  $x$  to the output of observations  $y$ : this function is known as the forward process. The forward process does not define a bijective function, meaning that we cannot find a unique mapping from the output to the input; i.e., there are many possible  $x$  for a single  $y$ . This proves to be important, since a traditional neural network trained on such a problem will only learn to find one of the possible solutions or an average of multiple correct but physically incompatible solutions. Furthermore, with a traditional neural network, it is impossible to ever know if the connections being made are the correct ones, as the network is trying to learn an ill-defined problem.

We circumvent this issue in our work by introducing a latent space  $z$  that captures all of the information lost in the forward process (Dinh et al. 2014 and references therein). The latent space  $z$  represents the space of all information loss in the forward process, such that a sample from the latent space combined with the observation  $y$  will be able to be mapped to the correct input parameters  $x$ . As a result of the introduction of latent variables, we now have a bijective mapping  $x \leftrightarrow [y, z]$ . This means that we have transformed the inverse process into a deterministic function (a function that has a definite result for a set of inputs). Consequently, sampling different values from the latent space will lead to a sampling of the distribution of the input parameters corresponding to a given output observation. This deterministic function  $x = g(y, z)$  is thus invertible, and we can learn the function  $g^{-1}$  as the forward process and  $g$  as the inverse process that will directly track where the lost information is obtained from the latent space. This is characterized by our network assuming that the latent variables  $z$  are drawn from the unit multivariate Gaussian distribution  $\mathcal{N}(0, \mathcal{I}_N)$  for an  $N$ -dimensional data space in the reverse direction. Here  $g^{-1}$  will populate the true latent space  $z_{\text{true}}$  with the information lost in the forward process. Our network is then trained in such a way (see Section 3) as to learn this mapping from the true latent distribution to the unit Gaussian latent distribution. After sufficient training, sampling the unit Gaussian distribution will be equivalent to sampling the true latent distribution, since they differ by only a known mapping. The choice of drawing from the unit multivariate Gaussian is an arbitrary one. It is true that any distribution could be used here, but we choose a Gaussian because it is smooth and continuous. The architecture we choose to learn this is our INN.

Like traditional neural networks, INNs are composed of interconnected layers of neurons that aim to learn a function from input to output. The key difference is the composition of the hidden layers between the input and output. These take the form of *affine coupling* layers (Dinh et al. 2014, 2016). Affine coupling layers are simple yet powerful tools. By construction, in learning the function from the input to the output with an affine coupling layer, we get the inverse function learned for free. This is due to the reversibility of the blocks, illustrated in Figure 1. We base our layers on the form first presented in Ardizzone et al. (2018). The input  $x$  is split into two equal parts  $[x_1, x_2]$  that are propagated through the forward direction of the block. This leads to  $x_2$  undergoing an affine transformation before combining with  $x_1$  to obtain half of the output  $y_1$ . Then,  $y_1$  is subject to its own affine transformation and combination with  $x_2$  to get the second half of the output  $y_2$ . This is illustrated in the upper panel of Figure 1. There is now a



**Figure 1.** Affine coupling layer showing the affine transformation between input and output for the forward process (top) and reverse process (bottom). These form the building blocks of our INN, as they are easily invertible.

simple relation between the input and output for this layer,

$$y_1 = x_1 \otimes \exp(s_2(x_2)) + t_2(x_2), \quad (1)$$

$$y_2 = x_2 \otimes \exp(s_1(y_1)) + t_1(y_1), \quad (2)$$

where  $\otimes$  denotes the element-wise multiplication of two tensors (which are represented by matrices in our problem), and the functions  $s_i$ ,  $t_i$  are arbitrarily complex and differentiable ( $i \in \{1, 2\}$ ). After obtaining the pair of outputs  $[y_1, y_2]$ , they are then concatenated to give the total output  $y$ . The inverse of this operation is then simple, and we can also map from the output  $y$  to the input  $x$ ,

$$x_2 = (y_2 - t_1(y_1)) \oslash \exp(s_1(y_1)), \quad (3)$$

$$x_1 = (y_1 - t_2(x_2)) \oslash \exp(s_2(x_2)), \quad (4)$$

where  $\oslash$  denotes the element-wise division of two matrices. We have now defined a setup in which the inverse is easily calculable. This is extremely useful for inverse problems, as it is rarely easy to find the inverse function for a forward model. This means that the only problem we now need to deal with is learning what the latent space is to make sure that our network produces the correct inversion; see Section 3 for more information. Since the functions  $s_i$ ,  $t_i$  do not need to be inverted themselves to calculate the inversion, they can be as complex and arbitrary a function as needed. To fill this role, we look to fully connected artificial neural networks (ANNs).

ANNs can learn complex classification and regression problems via a method known as backpropagation, and are therefore universal function approximators (Rumelhart et al. 1986; Cybenko 1989). They are an example of supervised machine learning, meaning that the network is trained on a data set where the answers to the functions we want to learn are known. In backpropagation, the input data is fed through a neural network, where linearities and nonlinearities are applied to it until it reaches the output, where it is compared with the known answers. This comparison is then surmised by a loss function, which is minimized by changing the values of the weights in each layer of the network to produce a different result (Schmidhuber 2015). There have been innumerable successes of ANNs learning complex functions via this method, so we use randomly initialized ANNs as our complex  $s_i$  and  $t_i$  functions in the INN.

In our network, the functions  $s_i$  and  $t_i$  are defined by four-layer fully connected networks (FCNs). An FCN is a type of ANN where all neurons in the previous layer are connected to all neurons in the current layer. The basic architecture for the FCNs utilized in our network is shown in Figure 2. The activation function (the function that determines to what extent the nodes pass on information to the next layer) after the first three layers in our deep networks is a leaky rectified linear unit (ReLU),

$$\phi(x) = \max(x, 0.01x), \quad (5)$$

with the activation after the fourth given by a ReLU

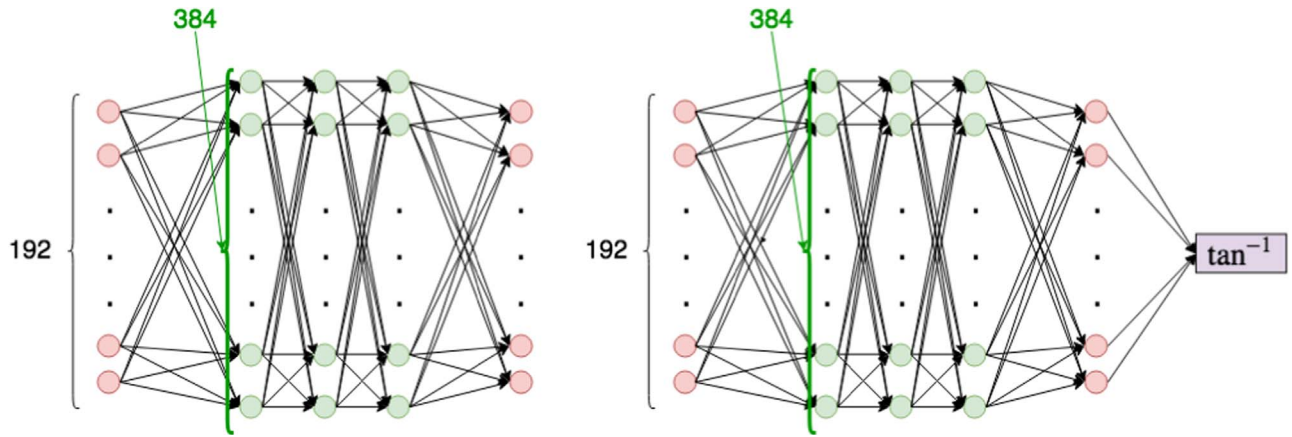
$$\phi(x) = \max(0, x), \quad (6)$$

where  $x$  is the input (in both cases). These activations are used, as they are sparse and thus speed up computation. Furthermore, ReLU activation and its variants are popular, as they are better at avoiding the vanishing gradient problem (when the gradients of the loss are small enough, they do not affect the update of the weights, leading to the optimizer getting stuck in the loss space). The functional forms of  $s_i$  and  $t_i$  differ by a clamping inverse tangent function applied at the end of the  $s_i$  networks. This clamping function stops the exponential terms dominating the affine transform while still being smooth (i.e., gradients are still easy to calculate). These networks are trained as normal via backpropagation (see Section 3), and they learn the optimal representation of the affine transform that will approximate the forward physical model. Then, this representation is also optimal for the inverse problem, as the FCNs apply to the inverse problem too.

Our network is comprised of five stacked affine coupling layers. Stacking these layers will allow us to approximate more complex tasks (this is the standard pillar of deep learning; Raschka 2015). This means that the network is dependent on 20 deep neural networks to approximate our inverse problem. Between each subsequent affine coupling layer, we have what is known as a permutation layer. This introduces channel mixing into our network by permuting the order of the inputs to each new layer. Channel mixing is when the inputs are shuffled into a different order. This is done as the input to the affine coupling layers is split in two, meaning that if there is no permutation, then these two halves remain independent throughout the network. The permutations are done by shuffling the input dimensions of our network in a random but fixed way (Dinh et al. 2014, 2016). Each permutation is different from the previous. This will increase the generalization properties of our network. The architecture of the INN is shown in Figure 3. The flow of the forward model is shown by the black arrows, and the flow of the inverse is shown by the cyan arrows.

### 3. Training an INN Using Synthetic Flare Data

This section describes the methods used to train and validate an INN to learn a bijective mapping between atmospheric profiles and two spectral lines. The training data consist of synthetic flaring solar atmospheres and spectral-line profiles generated from the one-dimensional radiation hydrodynamic model RADYN.



**Figure 2.** The FCNs for the  $t_i$  functions (left) and  $s_i$  functions (right). These are deep neural networks with four hidden layers. The network architecture for the  $s_i$  functions contains a smooth clamping function after output in the form of the inverse tangent. This clamps the output such that the exponential term in our affine transform does not overshadow the linear term (as this would make the linear term null). The input dimension is half the input dimension of the affine coupling layer due to the splitting of the input, as shown in Figure 1. The hidden layer depth is then double this.

### 3.1. Training Data

The state-of-the-art forward models for simulating the atmospheric response and radiation originating from solar flares are one-dimensional radiation hydrodynamic models that solve the equations of hydrodynamics coupled with the equations of radiative transfer (outside local thermodynamic equilibrium and statistical equilibrium). Among these models are RADYN (Carlsson & Stein 1992, 1997; Allred et al. 2005, 2015), FLARIX (Varady et al. 2010; Heinzel et al. 2016), and HYDRAD (Bradshaw & Cargill 2013). Due to the preexisting grid of RADYN simulations<sup>3</sup> and its widespread acceptance, we have chosen to use RADYN as the forward model for training here. These RADYN simulations all start from a modified VAL3C quiet Sun atmosphere (Vernazza et al. 1981).

For the simulations in the RADYN grid, the dynamic atmospheric response to an electron beam from a flare is computed, where

1. the distribution of electron energies in the beam is modeled as a power law with variable total energy flux (in the range  $3 \times 10^{10} - 1 \times 10^{12}$  erg cm<sup>-2</sup>),
2. the beam low-energy cutoff is  $E_c = \{10, 15, 20, 25\}$  keV,
3. the beam spectral index  $\delta = \{3, 4, 5, 6, 7, 8\}$ ,
4. the beam flux is a symmetric triangular pulse lasting for 20 s and peaking at 10 s, and
5. the simulation lasts for 50 s with data available every 0.1 s.

Some of the simulations with high total energy, lower values for  $E_c$ , and higher values for  $\delta$  did not complete and therefore are not available in the grid. This leaves 81 simulations, with 40,500 total timesteps to use as our training data. Of these timesteps, 20% are separated and used to independently verify the training.

RADYN uses an adaptive spatial grid (Dorfi & Drury 1987) to accurately represent the atmosphere, but due to the way in which our INN learns shapes, these data must first be interpolated onto a fixed, static grid. As we are primarily interested in the chromosphere and transition region, where the plasma parameters vary

rapidly in space, we interpolate onto 45 linearly spaced points below 3.5 Mm with a grid spacing of 79.2 km. Five further points are used to represent the rest of the corona, and these are spaced exponentially from 3.5 up to 10 Mm.

The plasma parameters extracted from the RADYN simulations are the electron density  $n_e$  [cm<sup>-3</sup>], the temperature  $T$  [K], and velocity  $v$  [cm s<sup>-1</sup>] as a function of altitude and time on the interpolated grid. The line profiles from these simulations, for H $\alpha$  6563 Å and Ca 8542 Å, are each interpolated onto 30 linearly spaced points in wavelength across wavelength ranges with half-widths of 1.4 and 1.0 Å, respectively. The assumption of energy input specifically by an electron beam originating in the corona results in a characteristic Coulomb-collisional energy deposition profile in the chromosphere determining  $n_e$ ,  $T$ , and  $v$ . For the spectral lines we will use, RADYN calculates both the thermal and nonthermal (i.e., direct beam–electron electron impact) collisional rates.

To reduce the dynamic range of these profiles and improve the performance of the INN, we first map  $n_e \mapsto \log_{10} n_e$ ,  $T \mapsto \log_{10} T$ , and  $v \mapsto \text{sign}(v) \log_{10} (|v|/10^5 + 1)$ . For each timestep in each simulation, the line profiles are divided by the maximal intensity in each profile, so that the profiles' relative intensities are preserved on a [0–1] scale.

### 3.2. MMD

Training the INN is made possible by the use of the maximum mean discrepancy (MMD). The MMD is a statistic used for computing the distance between two probability distributions based on a set of randomly drawn samples from each distribution by means of a high- or infinite-dimensional space through a nonlinear feature mapping. Our implementation is explained in depth in the Appendix, drawing on Gretton et al. (2012) and lectures given at the Machine Learning Summer School 2018.<sup>4</sup>

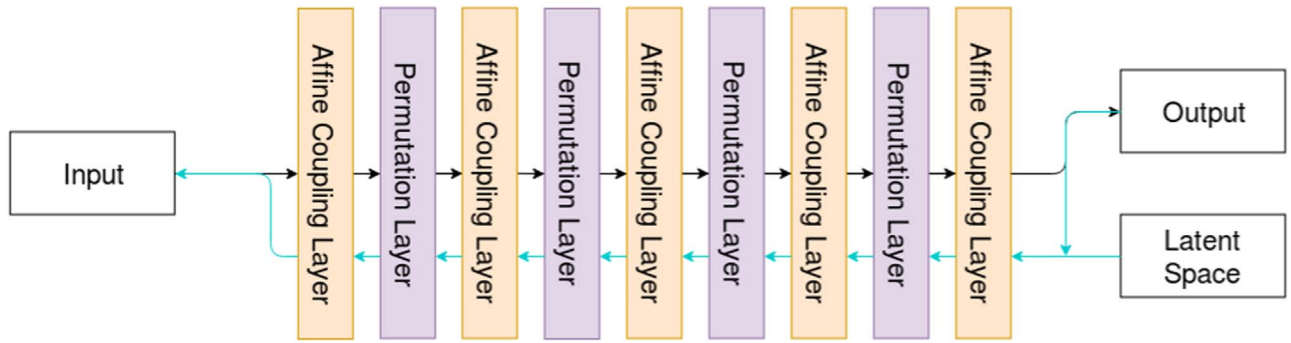
### 3.3. Training

Our INN is trained similarly to Ardizzone et al. (2018) and is based on their Framework for Easily Invertible Architectures (FrEIA).<sup>5</sup> Herein, we provide a more in-depth description of the

<sup>3</sup> Produced by the F-CHROMA project and available from <https://star.pst.qub.ac.uk/wiki/doku.php/public/solarmodels/start>.

<sup>4</sup> Available at <http://www.gatsby.ucl.ac.uk/~gretton/teaching.html>.

<sup>5</sup> <https://github.com/VLL-HD/FrEIA>



**Figure 3.** Architecture of our INN. We have five affine coupling layers with a permutation layer sandwiched between two affine coupling layers (four in total). The forward process mapping the input to the output is illustrated by the black arrows. The inverse process mapping a combination of the output and the latent space to the input is illustrated by the cyan arrows.

training method and the slight differences in the MMD loss used.

The INN is trained with the preprocessed simulation data alternating forward and backward iterations. We define the input  $x$  as the concatenation of the electron density, temperature, and velocity profiles at a certain timestep. The output  $y$  is the concatenation of the normalized line profiles at this timestep. The latent space  $z$  is currently defined to be the same length as  $x$ , although this is still an area of investigation tied to the intrinsic dimensionality of the problem. The output of the INN is then the vector  $[z, y]$ . Both the input and output vectors are zero-padded to provide the network blocks with additional dimensionality over which to represent the learned mapping, as well as to fix the input and output to the vectors to the same length, as the structure of the affine coupling layers requires this. We will write these zero-padded vectors  $x_p = [x, 0, 0, \dots]$  and  $y_p = [z, 0, 0, \dots, y]$ , and in our network, these are padded to have a length of 384.

The forward and backward training directions are both constrained by two loss functions. A loss function is a function that the neural network optimizer attempts to minimize during training so as to minimize the distance between the output from the ANN and the expected output. In the forward direction, we apply an L2 loss ( $\|y - y_{\text{true}}\|_2^2$ ) to the zero-padding and line profiles in the generated  $y_p$  vector against the expected  $y_p$  training vector from the forward model. An MMD loss is also applied between batches of  $[y, z]$  and  $[y_{\text{true}}, \mathcal{N}(0, \mathcal{I}_z)]$ . During backpropagation (modification of the weights in the ANN layers guided by the gradients at these nodes), the gradients on the generated  $y$  due to the MMD loss are ignored so as to train the neurons learning the mapping from the true latent distribution to the normal distribution without hindering the training of the forward model  $x \mapsto y$ . The convergence of this MMD loss ensures the independence of  $z$  from  $y$ .

The backward direction is trained similarly. The vector of  $y_{\text{true}}$  and the latents  $z$  generated by the forward iteration is propagated through the network in reverse, and an L2 loss is applied between  $x_p$  and a zero-padded vector containing  $x_{\text{true}}$ . Another vector of  $y_{\text{true}}$  with latents  $z$  drawn from the normal distribution is also propagated in reverse, and an MMD loss is computed between  $x$  and  $x_{\text{true}}$ . This second MMD loss serves to ensure that the distributions of  $x$  across the batch look alike (while taking into account internal variability within the true distribution).

The kernel used in our MMD loss is the same as that of Tolstikhin et al. (2017) and Ardizzone et al. (2018), the inverse multiquadric (IMQ) kernel

$$k_\alpha(x, y) = \frac{\alpha^2}{\alpha^2 + \|x - y\|_2^2}, \quad (7)$$

as it has been found most effective for comparing sample quality in these problems. In the example provided by Ardizzone et al. (2018), the kernel used is a sum of IMQ kernels with different  $\alpha$  (due to the properties of the reproducing kernel Hilbert space (RKHS) over which the MMD is defined, this sum is also a kernel); however, we had difficulty isolating a set of values for  $\alpha$  that were effective in training the latent distribution to match the expected distribution without dependence on  $y$ . By plotting the MMD for the same  $x$  and  $y$  samples but different values of  $\alpha$ , it was found that the biased sample estimate of the MMD between  $x$  and  $y$  drawn from similar but perturbed distributions produced a peak for certain values of  $\alpha$ . We therefore compute the value of  $\alpha$  at the turning point of the  $\text{MMD}^2(\alpha)$  (for which the MMD is maximal) during the training of the net and update  $\alpha$  every five epochs. This approach is supported by Sriperumbudur et al. (2009), as the kernel of a family that yields the greatest distinction between the two differing distributions is the one for which the MMD estimate is maximal.

Our INN is trained using the Adam optimizer (Kingma & Ba 2014) with  $\beta_1 = \beta_2 = 0.8$  and  $\epsilon = 1 \times 10^{-6}$ , where the  $\beta$  hyperparameters control the momentum of the first and second moments of the gradients and  $\epsilon$  prevents division by zero. A hyperparameter is a parameter that is set prior to training, possibly evolving in a predictable fashion, and is not optimized by the training process. The values of these parameters are typically determined empirically and may well not be optimal, but they have been chosen to lead to convergence of the model. The learning rate  $\eta$  (the size of the steps taken in descending the gradient) is initially set to  $1.5 \times 10^{-3}$  and decays by a factor of  $\gamma = 0.004^{1/1333}$  every 12 epochs; thus, for the model presented in this paper, trained for 11,400 epochs, the final learning rate is  $\eta \approx 3.38 \times 10^{-5}$ . This model does not appear to be very sensitive to variations in the learning rate, and multiple variations of  $\gamma$  have been used with success. We used a minibatch size of 500, with 20 minibatches per epoch, and the backpropagation took place every minibatch. In contrast to

traditional training, where the model is trained on the entire training set every epoch and accumulates gradients over the entire training set before backpropagation, minibatch training shows the model multiple small subsets of the data each epoch with gradient accumulation and backpropagation between each of these minibatches.

The two losses computed for each of the forward and backward iterations need to be combined into a single loss in each direction for the backpropagation. We use this as an opportunity to add additional hyperparameters with which to weight the various losses when combining them. We therefore define three weights:  $w_{\text{pred}}$ ,  $w_{\text{latent}}$ , and  $w_{\text{rev}}$ . Then, the loss from the forward process is produced by

$$\text{loss}_f = w_{\text{pred}}L2_f + w_{\text{latent}}\text{MMD}_f, \quad (8)$$

and the backward loss by

$$\text{loss}_b = 0.5w_{\text{pred}}L2_b + \xi(n)w_{\text{rev}}\text{MMD}_b, \quad (9)$$

where  $f$  and  $b$  represent the previously discussed forward and backward losses that are combined,  $\xi(n) = \left(\min\left(\frac{n}{0.4N_{\text{fade}}}, 1\right)\right)^3$ , where  $n$  is the current epoch and  $N_{\text{fade}}$  is the number of epochs in the initial training stage. The function  $\xi(n)$  helps to avoid the initially large gradients in  $\text{MMD}_b$  from steering the net away from the correct solution. In practice, it was found that this function was not strictly necessary but improved convergence. Additionally, the zero padding was set to  $5 \times 10^{-2}(1 - \xi(n))\mathcal{N}(0, 1)$  to increase the activations of these neurons during early training and therefore push their outputs toward zero. The exact values of these parameters were determined empirically but with an emphasis on minimizing the L2 losses.

The initial 800 epochs were treated as an initial fade-in stage as  $\xi(n)$  grew to 1 and the padding became zero. For this phase, the loss weightings were set to  $w_{\text{pred}} = 4000$ ,  $w_{\text{latent}} = 900$ , and  $w_{\text{rev}} = 1000$ . After this initial phase, the net was trained in batches of 400 epochs up to 4800 epochs, increasing  $w_{\text{pred}}$  by 1000 each batch. This process was then repeated with batches of 600 epochs up to a total of 12,000 epochs. Finally, the model that performed best on the unseen validation set was chosen as the final model. This model was trained for 11,400 epochs.

### 3.4. Validation

The first stage in validating the training of the model is to test the forward model against ground truths on the unseen testing data. Figure 4 shows the results of the forward model. The top panels are the electron number density, temperature, and flow speed from an unseen RADYN snapshot, and the bottom panels compare the ‘‘ground-truth’’ RADYN output line profiles with the network’s forward process. The mean squared error is  $5.73 \times 10^{-5}$  in the scaled intensity at each wavelength point. Note that for all figures in this paper, wavelength axes show the wavelength in a vacuum, and positive velocities represent upflows.

It is somewhat more difficult to evaluate the model’s ability to reproduce an atmosphere when given the line profiles due to the aforementioned ambiguity of the problem, as one set of line profiles may have been produced by a variety of atmospheres. To understand the range of solutions, we draw random samples from the latent space multiple times and use these samples with the line profiles to generate a histogram of atmospheric

properties predicted by the INN. Figure 5 shows the results and verification of the inversion of data from the unseen testing set. In the top panels, the input line profiles are plotted in dashed blue lines on top of horizontal bars representing the line profiles calculated using the recovered atmospheric solutions. The recovered solutions are shown in the bottom panels, plotted as two-dimensional colored histograms representing the probability density of the solution at each altitude node. The regions of highest density in these parameters are therefore the most likely values. Superposed on this are the ground-truth values for each parameter, plotted as dashed lines. The data in the histograms are accumulated for every solution for the atmospheric profile produced from different draws of the latent space and represent 10,000 sampled solutions.

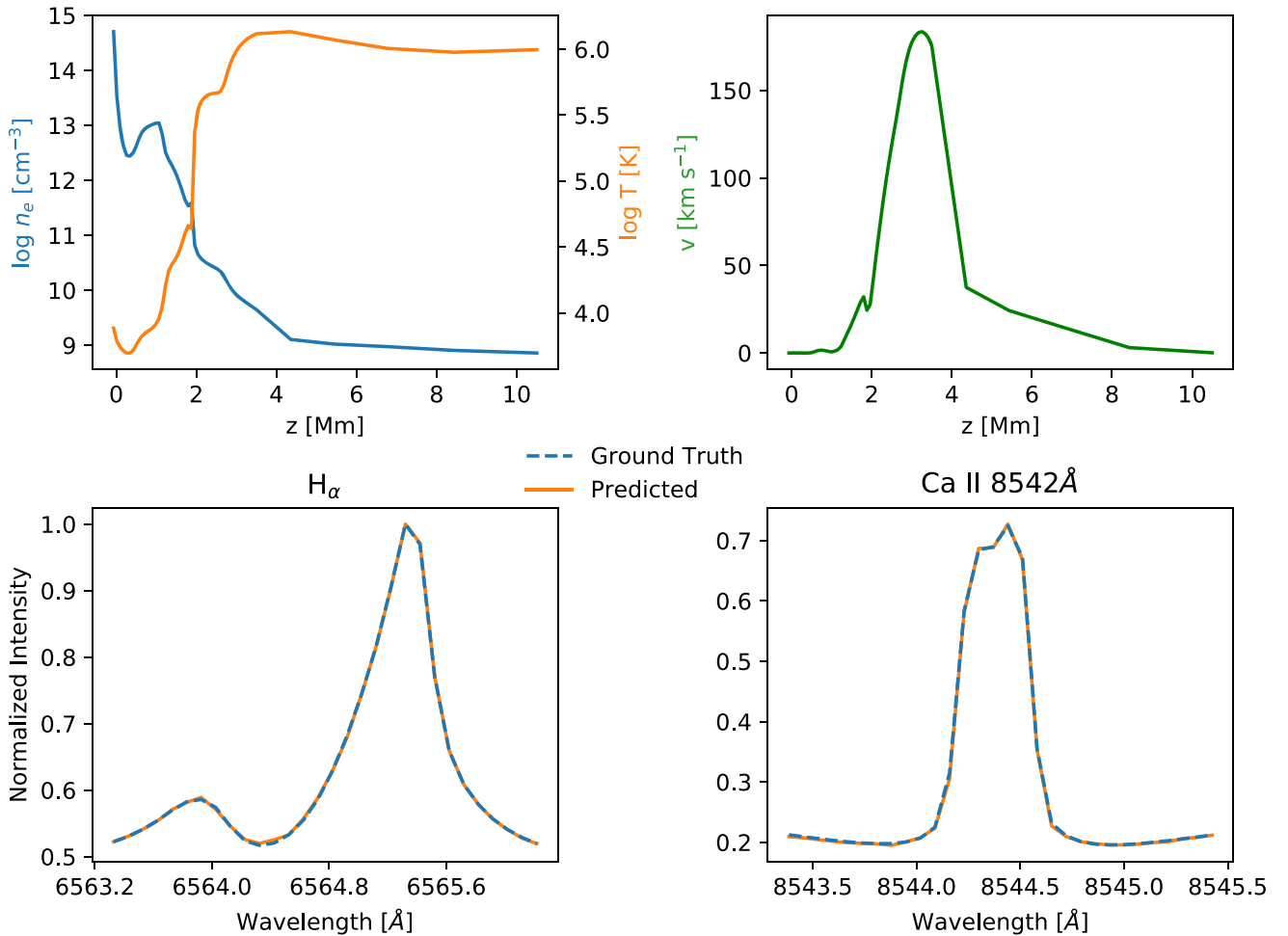
To better show the range of outlying solutions, all of the histograms were gamma-corrected (with  $\gamma = 0.3$ ) to reduce contrast. As can be seen from the dashed black lines in the bottom panels of Figure 5, the peak density of the solutions is close to the ground truth, and the narrowness of the histograms shows that the solution is well constrained through the atmosphere up to around 3 Mm above the photosphere. However, the spectral lines used do not constrain the problem well in the upper atmosphere, and although the solutions align very well with the ground-truth, the histograms are broader, particularly for the profile of velocity at 4 Mm and above. The histograms underneath the input line profiles in the top panels of Figure 5—so narrow as to look like single bars—are obtained by applying the forward model to each atmosphere produced by the inverse process and gamma-corrected in the same way. They reproduce the input line profiles very closely, demonstrating the self-consistency of the model’s solutions.

## 4. Single-pixel Inversion of Real Flare Data

We have demonstrated above that the INN has successfully learned the synthetic flare model from RADYN. The next step is to apply our learned model to real spectroscopic data, with the intention of characterizing the atmosphere that produced it, and eventually learning about the physics of a flaring chromosphere. As our problem is only defined in H $\alpha$  and Ca II 8542 Å, and these are mostly formed in the chromosphere (cores) and upper photosphere (wings), we will focus specifically on our results for atmospheric parameters below around  $z \approx 2$  Mm. We do not attach much significance to the results from the small number of points in the corona.

The flare data we use are from the M1.1 two-ribbon solar flare SOL 20140906T17:09, which occurred in NOAA AR 12157 with heliocentric coordinates  $(-732'', -302'')$ . Data were taken by CRISP (Scharmer 2006; Scharmer et al. 2008) mounted on the Swedish 1 m Solar Telescope (Scharmer et al. 2003) on La Palma. CRISP produced imaging spectroscopy data in both H $\alpha$  and Ca II. The H $\alpha$  data consist of 15 wavelength positions sampled at intervals of 200 mÅ from the line core, and the Ca II data consist of 25 wavelength positions sampled at intervals of 100 mÅ from the line core. The cadence of these observations is 11.54 s with a spatial sampling of  $0.057'' \text{ pixel}^{-1}$  (giving a spatial resolution of  $0.114''$ ). The data set is open access and available from the F-CHROMA solar flare database (Cauzzi et al. 2014),<sup>6</sup> where it has been preprocessed and reconstructed using Multi-Object Multi-Frame Blind Deconvolution (MOMFBD; Van Noort et al. 2005) and the CRISPRED data

<sup>6</sup> <https://star.pst.qub.ac.uk/wiki/doku.php/public/solarflares/start>



**Figure 4.** Output of the model’s forward process on unseen testing data. The top row shows the atmospheric parameters used as input to the network, and the bottom row shows the output of the model’s approximation of the forward process with the true results overlaid with the dashed line. Positive velocities represent upflows.

reduction pipeline (de la Cruz Rodríguez et al. 2015). We assume that the intensity calibration of the two lines is done as well as possible in the same way through the CRISPRED pipeline. Therefore, we are assuming that the relative intensities between the two lines are physically meaningful, as assumed by our inversion technique. This event was previously analyzed by Kuridze et al. (2015), who presented the time evolution of the  $H\alpha$  and  $Ca II$  8542 Å lines in small flaring regions and compared these with RADYN forward modeling, driven by an electron beam with properties deduced from the observed hard X-ray spectrum, commenting primarily on the relationship between plasma flows and line asymmetries.

Figure 6 shows the wing and core images of  $Ca II$  and  $H\alpha$  at  $\sim 16:56$  UTC, just after the onset of the flare at  $\sim 16:54$  UTC. These images clearly show the presence of two flare ribbons during the time of the observation. We chose two pixels to invert: one on the flare ribbon and one off the flare ribbon. These are indicated in Figure 6 by a circle and square, respectively. The spectral-line profiles from the two pixels are extracted, normalized to the maximum value of the two lines, and interpolated to the RADYN grid. These are shown in Figure 7.

The lines in the top panels of Figure 7 are from a point on the flare ribbon, and those in the bottom panels are from a point off the flare ribbon (the circle and square, respectively, in

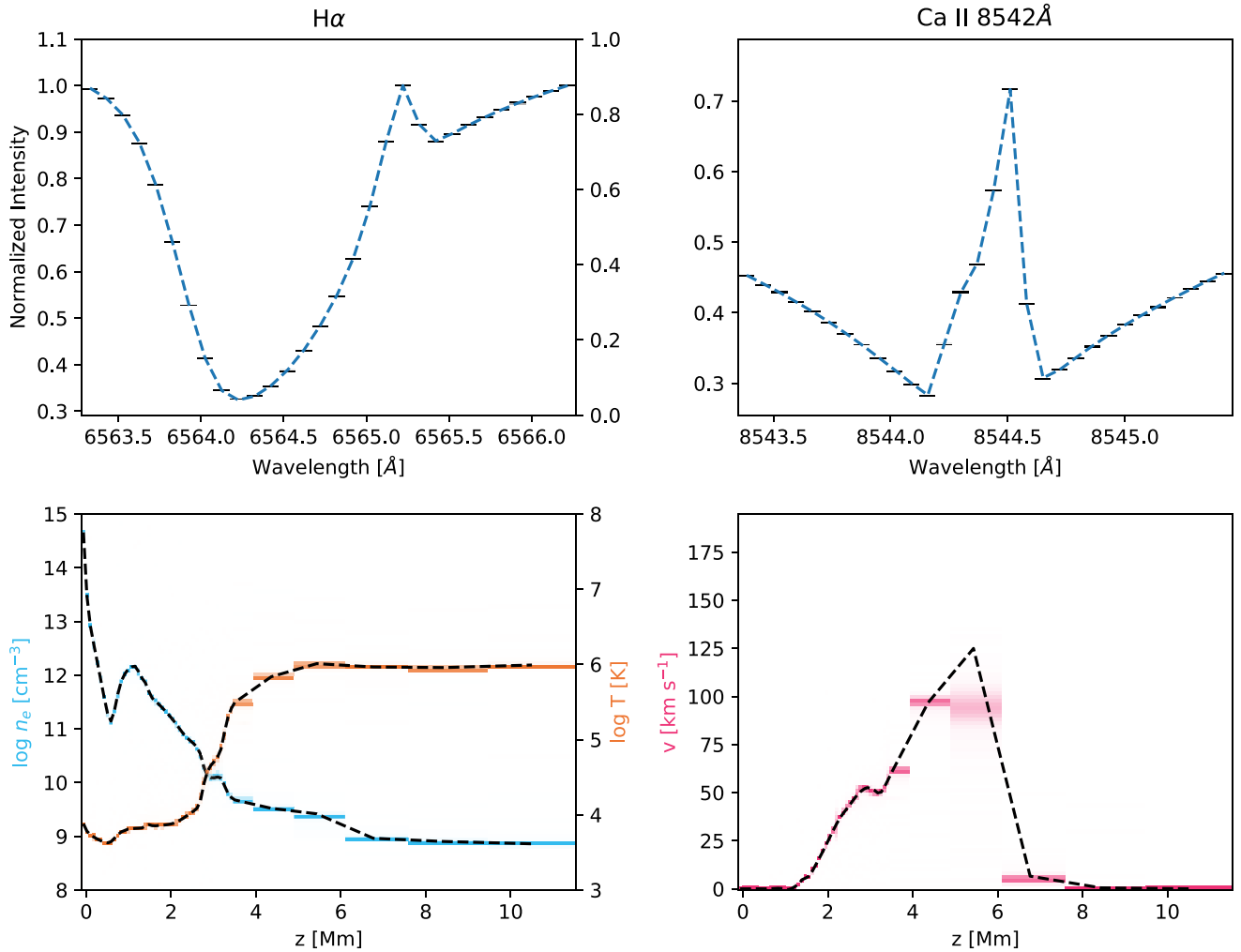
Figure 6). The  $Ca II$  8542 Å line profile for the circular point is characteristic during a flare. It is fully in emission, and the core is slightly blueshifted (with respect to the vacuum wavelength) by  $\sim 3.51$  km s $^{-1}$  with a slight wing asymmetry. The  $H\alpha$  profile is highly asymmetric, with the blue peak of the central reversal being much higher in emission than the red peak. For the square point, both profiles are heavily in absorption (indicative of the quiet Sun). The  $Ca II$  and  $H\alpha$  cores are slightly redshifted here (by  $\sim 1.26$  and  $\sim 2.18$  km s $^{-1}$ , respectively), and both profiles have some asymmetry between the wings.

To calculate the asymmetries in the profiles, we use a technique similar to that described in Mein et al. (1997), De Pontieu et al. (2009), and Kuridze et al. (2015),

$$I_B = \int_{\lambda_{0B}-\delta\lambda}^{\lambda_{0B}+\delta\lambda} I(\lambda) d\lambda, \quad (10)$$

$$I_R = \int_{\lambda_{0R}-\delta\lambda}^{\lambda_{0R}+\delta\lambda} I(\lambda) d\lambda, \quad (11)$$

where  $\lambda_{0B}$  and  $\lambda_{0R}$  are the center wavelengths of the blue and red wings, respectively, and  $\delta\lambda$  is the width of the wing from its center wavelength. The wings are defined as being the area of the line one standard deviation away from the calculated



**Figure 5.** Output of the model’s inverse process on unseen testing data. The dashed lines in the top panels show the input to the inverse process that is augmented with a randomly drawn latent space. The two-dimensional histograms in the bottom panels show the results of each inversion. The dashed lines in the bottom panels show the expected solution for the inversion. The two-dimensional histograms (narrow gray bars) in the top panels are the result of propagating each atmospheric solution from the inversion through the forward process.

intensity-averaged line core. The intensity-averaged line core is calculated via

$$\lambda_0 = \frac{\int I(\lambda) \lambda d\lambda}{\int I(\lambda) d\lambda}, \quad (12)$$

which leads to us calculating the variance of the profile,

$$\sigma^2 = \frac{\int I(\lambda) (\lambda - \lambda_0)^2 d\lambda}{\int I(\lambda) d\lambda}. \quad (13)$$

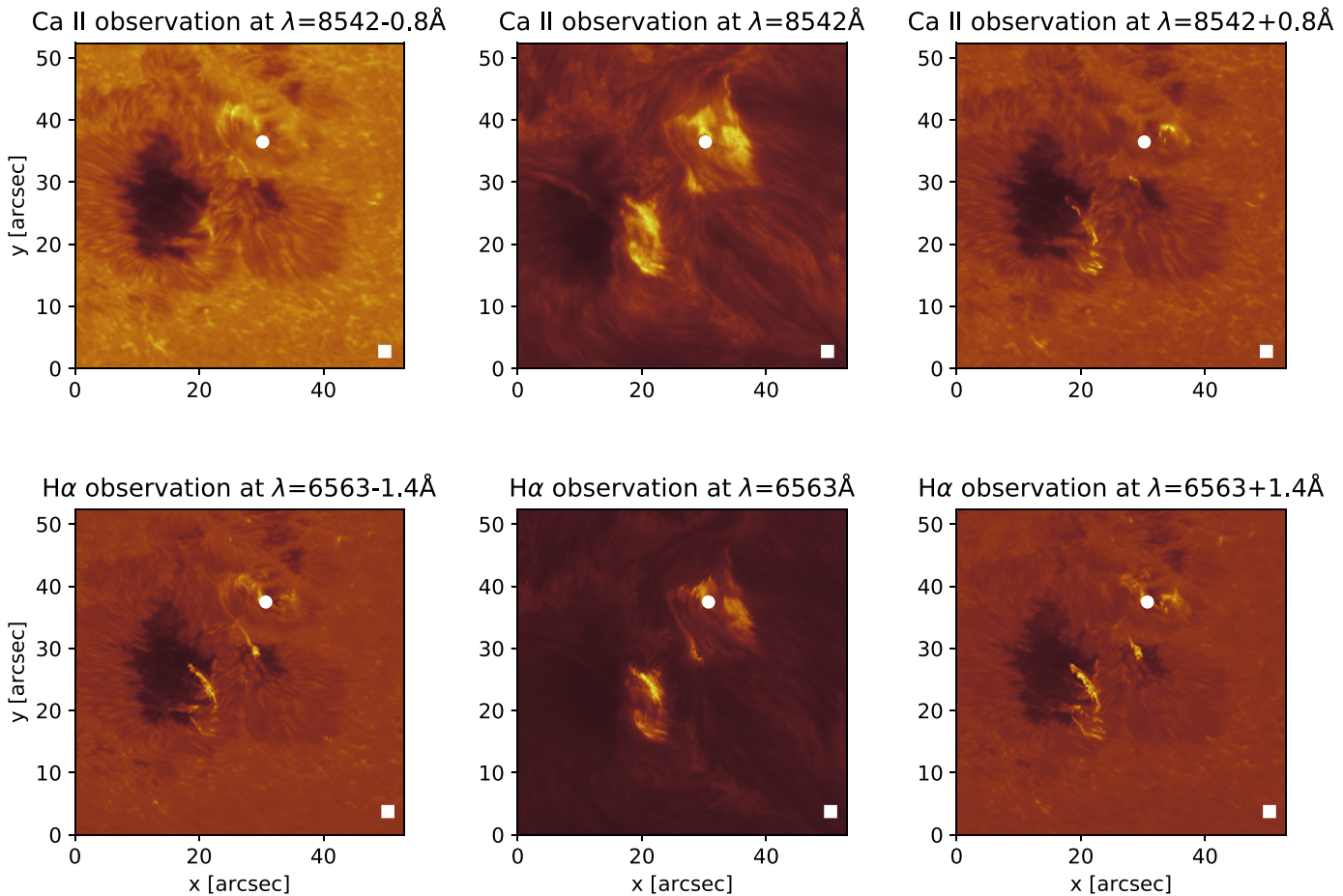
Then the end of the blue wing and the start of the red wing are defined by  $\lambda_0 - \sigma$  and  $\lambda_0 + \sigma$ , respectively, allowing us to calculate the central wavelengths for the wings and the half-width of the wings (i.e.,  $\lambda_{0B}$ ,  $\lambda_{0R}$ , and  $\delta\lambda$ ). These values, along with the intensity ratio of the wings,  $I_B/I_R$ , are presented in Table 1. The off-ribbon profiles both have red asymmetries of  $\sim 1.8\%$  for calcium and  $\sim 1.7\%$  for H $\alpha$ . This corresponds to small positive velocity gradients or downflows in the region where the wings of these lines are formed. The calcium profile on the ribbon has an  $\sim 3.2\%$  blue asymmetry, while the H $\alpha$

profile has a red asymmetry of  $\sim 0.4\%$ . This corresponds to small negative velocity gradients or upflows in the region where the wings of calcium are formed.

It has been shown that the spectral lines we are considering should be symmetric about the line core in a static atmosphere (Canfield et al. 1984; Fang et al. 1993; Cheng et al. 2006), implying that the velocity field in the flaring atmosphere is responsible for the observed asymmetries. This is likely linked to chromospheric evaporation (Neupert 1968; Fisher et al. 1985; Graham & Cauzzi 2015) and condensation (Ichimoto & Kurokawa 1984; Wulser & Marti 1989), which are the bulk expansion flows that occur in the rapidly heated flare chromosphere. However, mapping between the observed asymmetry and the flow direction is complicated by absorption and emission in the moving plasma. For example, a blue asymmetry, as is observed in the Ca II line on the flare ribbon, could be due to emission from upflowing plasma or absorption by downflowing plasma, as argued for this flare by Kuridze et al. (2015).

These observed spectral-line profiles were propagated in the backward direction through our INN (see Figure 3) 20,000 times each with different random draws from the unit Gaussian





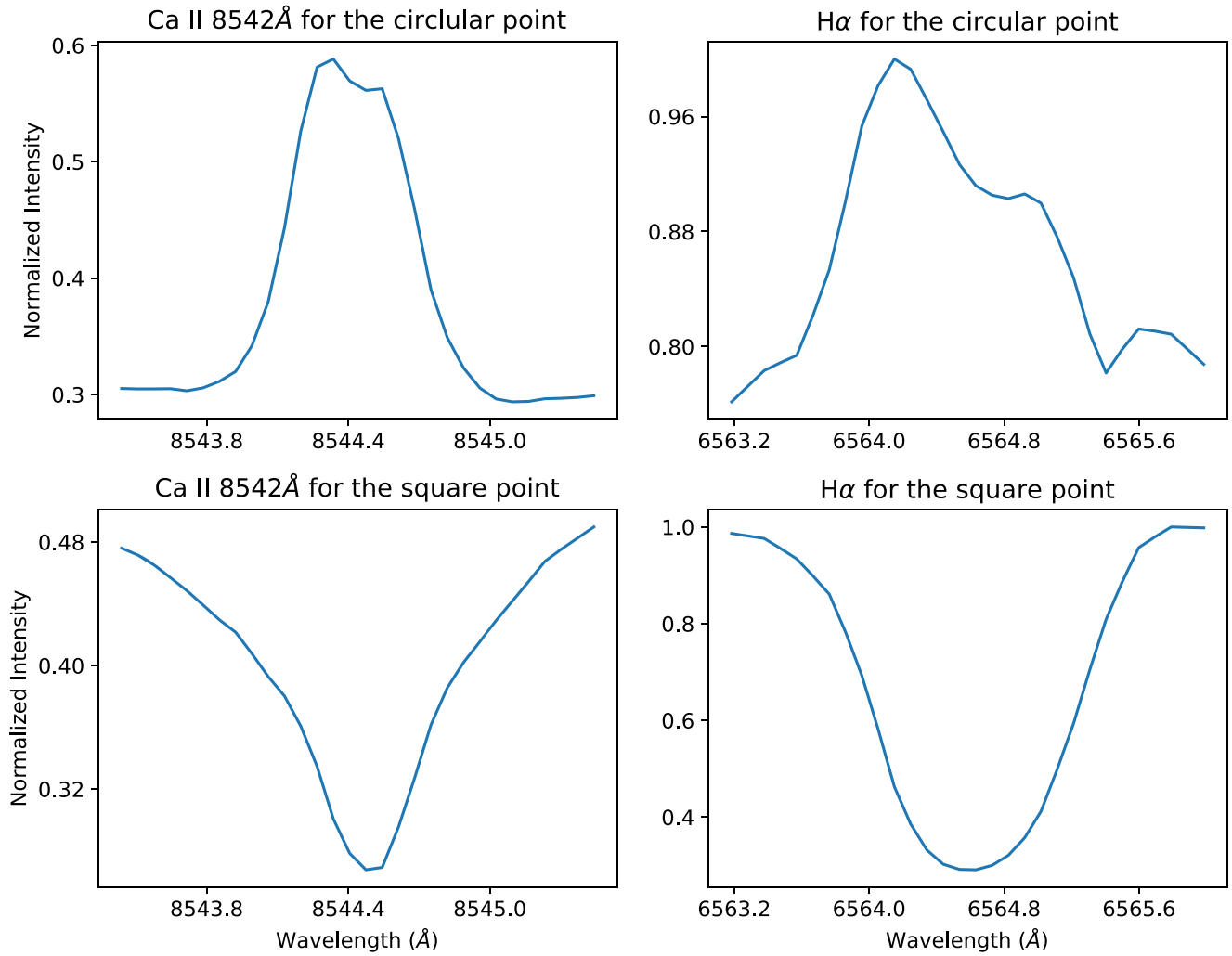
**Figure 6.** Observations of the M1.1 two-ribbon solar flare from AR 12157 on 2014 September 6. These images are from just after the onset of the flare at 16:56:13 UTC. The top row shows images taken in the Ca II 8542 Å band, with the left panel showing the blue line wing, the middle panel showing the line core, and the right panel showing the red line wing. The bottom row shows images taken in the H $\alpha$  band following the same convention as for Ca II. We select two pixels for our inversion test: one on the flare ribbon (circle) and one off the flare ribbon (square). These points are plotted on top of the images in each panel.

latent space (i.e., 20,000 inversions). The inversion of a single pixel takes  $\sim 893$  ms on an NVIDIA GTX 1050Ti and  $\sim 84.5$  s on an Intel Core i7-8700 CPU. The results of the inversions for the point on the flare ribbon are shown in Figure 8 and for the point off the flare ribbon in Figure 9. As in the case of the model validation in Section 3.4, the results are plotted as two-dimensional histograms (top panels of Figures 8 and 9). The dashed lines show the median profile for the parameters. This gives an approximation to the true solution from our inversion, as the median profile will pass through the bins with the highest densities. The bottom panels of these figures are plots of the observed spectral lines (dotted blue lines) and the densities of the round-trip profiles obtained by passing the results of the inversion back through the network in the forward direction. This shows that each of the atmospheres we produce are viable for the production of these spectral lines, with some curves being less likely due to the lack of density in the bins of the histogram (i.e., models with specific points in less dense bins are less likely to be the true solution).

Examining the atmospheric profiles obtained from the inversions helps us interpret the line profiles generated. Looking first at the line asymmetries, we have previously remarked that for the on-ribbon pixel, the Ca II line is slightly blueshifted, with a blue asymmetry in the wings. According to Kerr et al. (2016), the Ca II 8542 Å line during a flare is formed between 0.2 and 1.0 Mm above the photosphere, with the wings beyond  $\pm 0.3$  Å

from line center formed between 0.2 and 0.4 Mm, i.e., in the upper photosphere/lower chromosphere. The line core within  $\pm 0.3$  Å of line center is formed above that. A steep positive velocity gradient in the area of core formation (0.9–1 Mm) explains the blueshifted core of our flare ribbon calcium profile. In the region of formation of the wings of this line, we observe a small positive upflow that would cause the observed blue asymmetry due to the emitting material moving upward. Kuridze et al. (2015) indicated that the H $\alpha$  profile forms below 1.2 Mm, with the wings forming below 0.95 Mm and the core forming above this height. The wings of the on-ribbon H $\alpha$  profile are very slightly asymmetric in favor of the red wing. In the region where the wings are formed, there is a small positive velocity gradient. This leads us to believe that there has been chromospheric evaporation in this region leading to an increase in optical depth in the region of the blue wing, meaning that there will be more absorption in the blue wing.

For our off-ribbon pixel, both profiles have small red asymmetries. This can be explained in our inverted atmosphere due to a turbulent flow where the lines are formed, which would also explain the asymmetries. Our velocity solution here is quite oscillatory. RADYN has an underlying  $2 \text{ km s}^{-1}$  microturbulent velocity, so the line profiles it produces are not as broad as those observed. Having learned that flows produce shifted emission, this oscillation is our model's attempt at making the lines the correct width.



**Figure 7.** Spectral lines in Ca II 8542 Å and H $\alpha$  for the two points selected in the region of interest. The top panels show one point on the flare ribbon, and the bottom panels show one point off the flare ribbon. We perform inversions on both of these pairs of spectral lines.

**Table 1**

The Results of Calculating the Intensity-averaged Line Core and Line Standard Deviation from Moments Analysis and Using These Values to Calculate the Asymmetries in the Observed Lines from Figure 7

	$\lambda_0$ [Å]	$\sigma$ [Å]	$\lambda_{0B}$ [Å]	$\lambda_{0R}$ [Å]	$\delta\lambda$ [Å]	$I_B/I_R$
H $\alpha$ on ribbon	6564.57	0.78	6563.49	6565.68	0.31	0.996
Ca II on ribbon	8544.43	0.52	8543.67	8545.20	0.24	1.032
H $\alpha$ off ribbon	6564.58	0.93	6563.41	6565.75	0.23	0.983
Ca II off ribbon	8544.43	0.62	8543.63	8545.25	0.19	0.982

**Note.** Here  $\lambda_{0B}$  and  $\lambda_{0R}$  are the central wavelengths of the blue and red wings of the line, respectively;  $\delta\lambda$  is the half-width of the wings; and  $I_B/I_R$  is the wing intensity ratio.

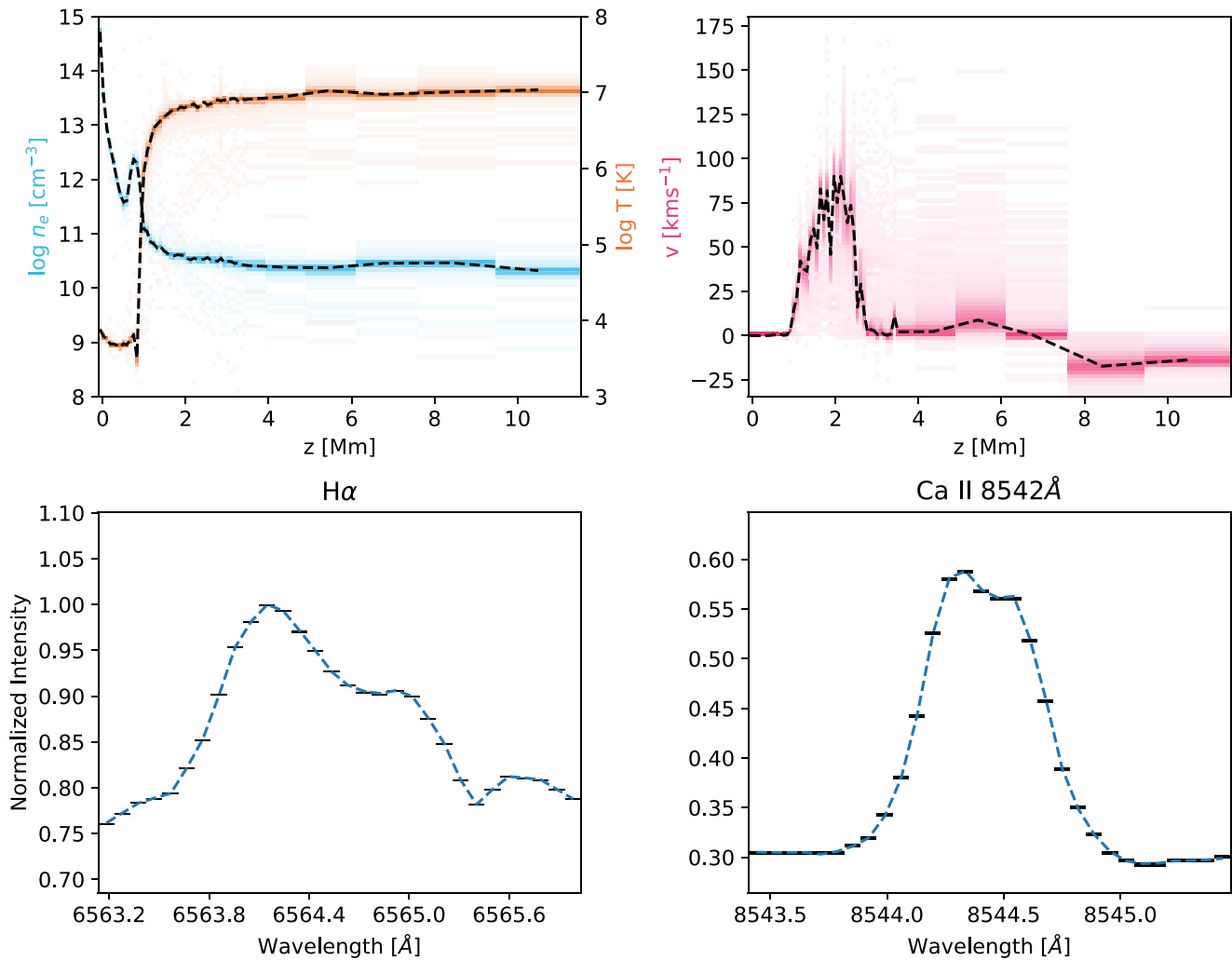
The other main feature is the lack of a strong central reversal in H $\alpha$  during the flare. This is likely due to the source function being closer to the blackbody in the regions of line core formation in the flaring atmosphere compared to the nonflaring atmosphere. This may, in turn, be a result of the order-of-magnitude increase in the electron density at the line formation

height in the flare, as indicated by the  $n_e$  curves in Figures 8 and 9.

## 5. Discussion and Conclusions

We have presented a novel approach to obtaining the distribution of solar atmospheric properties  $n_e$ ,  $T_e$ , and bulk flow speed  $v$  from observed H $\alpha$  and Ca II 8542 Å spectral-line profiles using an INN trained on RADYN flare models. The network learns a bijective approximation to the forward and inverse problems of mapping atmospheric snapshots to (observable) spectral-line profiles, and vice versa. Our initial results are very promising when tested on a flare previously analyzed by Kuridze et al. (2015), aligning well with their results, as discussed in Section 4.

The INN method of atmospheric inversion represents a significant theoretical step forward in the field of inversion. Taking the process of training and applying the INN as a whole, it is comparable to the process performed by existing non-LTE inversion tools, which are typically composed of a forward model for computing the line profiles from an atmosphere, such as RH (Uitenbroek 2001), and an “inversion engine” that is responsible for determining the necessary perturbations to the



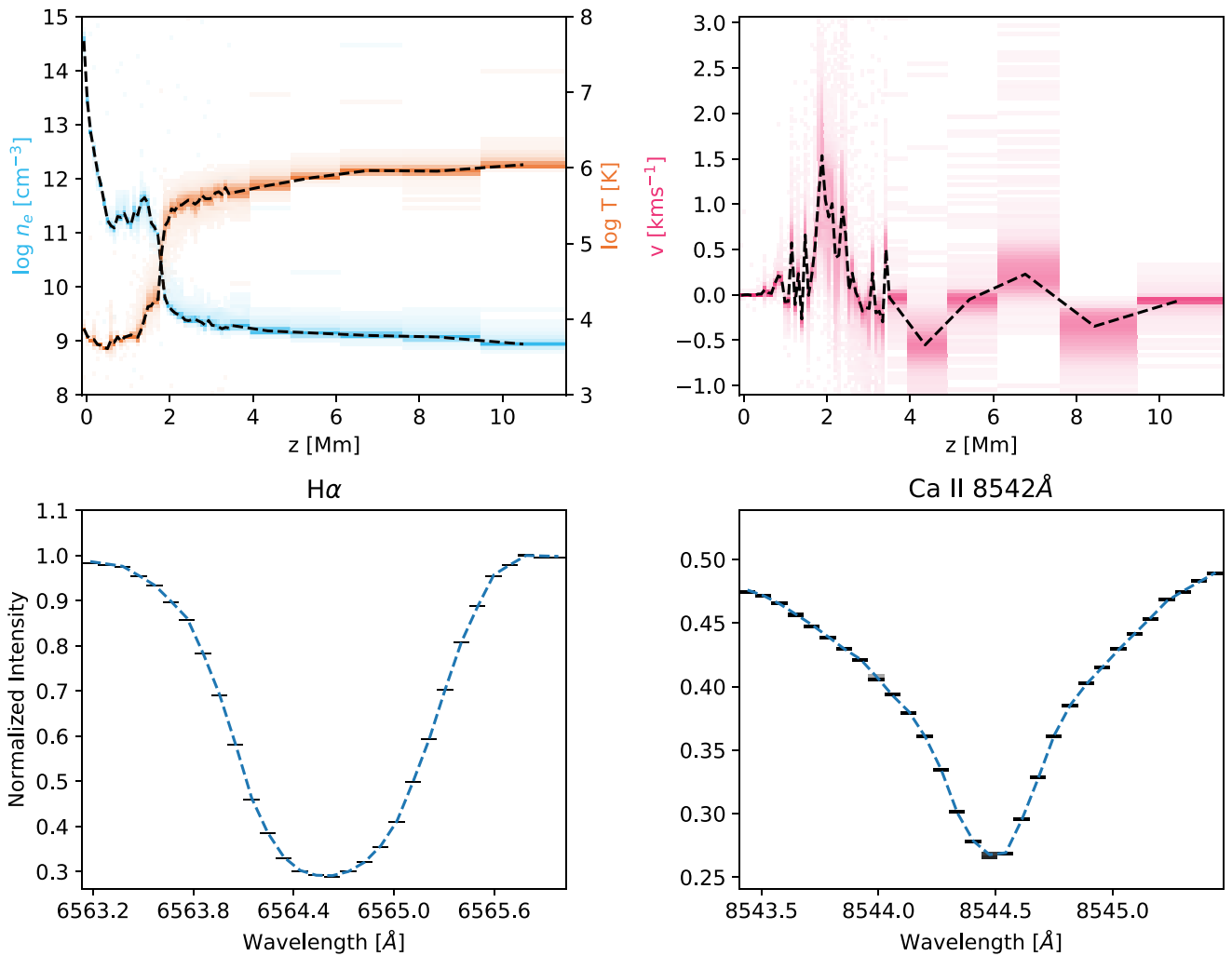
**Figure 8.** Inversion of the pixel on the flare ribbon. The top panels show the atmospheric parameters obtained from the inversion. The top left panel shows the electron density and temperature plotted on log scales, and the top right panel shows the net velocity flow in our plasma. The plots were made by sampling the latent space 20,000 times and plotting the results of the inversions as a two-dimensional histogram. The bins with the greatest density are the most likely values for the parameters at a certain height. The black dotted lines show the median profiles for each quantity. The bottom panels show the lines that were inverted. The blue dotted lines are the true line profiles. The black bins are the round-trip generation of the spectral lines produced by performing the forward process on the sets of atmospheric parameters we obtain from the inversion.

atmosphere to produce a best-fit line profile. Our INN first learns the forward process from our training data, but due to the bijective nature of the mapping, a perturbative solution approach is not required, as all of the information lost in the forward process can be restored through the latent space. The models that take this “inversion engine” approach, such as STiC (de la Cruz Rodriguez et al. 2019) and NICOLE (Socas-Navarro et al. 2015), are effectively performing a walk through the latent space guided by their “inversion engines.” There is no guarantee of solution uniqueness from those approaches, as the entire latent space is not visited. With the INN approach, the useful extent of the latent space is learned during training, and it is therefore trivial to span the latent space with multiple draws of the unit multivariate normal distribution.

As our INN was trained on RADYN data, it is important to stress that it can only generate RADYN-like solutions, and this should be taken into account when analyzing any atmospheric inversions performed. The RADYN training atmospheres also include the specific assumption of heating and nonthermal excitations by an electron beam from the corona. As a counterpoint to this, it is important to note that the INN does

not simply ingest the grid of RADYN simulations and return a closely matched or interpolated template (an approach used, for example, by Beck et al. 2015 in the fast inversion of Ca II 8542 Å spectropolarimetric data). Instead, the INN has learned a bijective mapping between the input space containing the atmospheric parameters and the output space containing the line profiles and the explicit latent space. In the inverse process, the line profiles are complemented by the latent space to remove ambiguities due to information lost in the forward process. The model’s validation on the unseen testing set should ensure that the atmospheres recovered are physically reasonable, and that the model has learned to relate the emergent line profiles with the properties of the atmosphere.

The INN method is fast, as it “front-loads” a large portion of the computational work by requiring a large training set in the form of RADYN simulations followed by approximately 1 day of training on an NVIDIA GTX 1050Ti GPU. The result of this precomputation is that inference is then extremely rapid while still drawing on a very complex physical model. The complex model is needed for the flare problem, where assumptions of hydrostatic and local thermodynamic equilibrium cannot hold,



**Figure 9.** Inversion of the pixel off the flare ribbon. The plots have the same format as Figure 8, and the latent space was also sampled 20,000 times.

and steep gradients are expected to form. This presents a further advantage of the INN method for flares, since, to reduce the size of the parameter space and allow an “inversion engine” to converge in a reasonable amount of time, all other inversion codes currently assume that the atmosphere is in hydrostatic equilibrium (Socas-Navarro et al. 2015; de la Cruz Rodriguez et al. 2019) and use  $<10$  nodes in the atmosphere where the parameters are computed, with various interpolation techniques used between these.

As found in Brown et al. (2018), the nonequilibrium level population and ionization effects present in RADYN, including those due to direct excitations by nonthermal electrons, cause significant deviations between the line profiles computed with these populations and those computed under the assumption of statistical equilibrium in RH (Uitenbroek 2001). Because our model is trained on RADYN data, the associated line profiles are based on RADYN’s nonequilibrium formalism and assumption of complete redistribution (i.e., the frequency of an absorbed photon that leads to an excited state and that of the resulting emitted photon are assumed to be independent). These effects are therefore learned by the INN. It is interesting that, even with limited atmospheric information, i.e.,  $n_e$ ,  $T$ , and  $v$ , which are a far-from-complete description of the state of the atmosphere, the INN was nevertheless able to very successfully reproduce the emission from the unseen RADYN snapshots

from the F-CHROMA grid. This implies that sufficient non-LTE and non-hydrostatic equilibrium information about local “microscopic” (ionization, level populations), “macroscopic” (gas pressure, opacity), and nonlocal physics (conduction, radiative back warming) must be encoded in these three parameters and their variation through the atmosphere.

Inversions of pixels on the flare ribbon performed in Section 4 suggest significant oscillations in the velocity profile in the transition region (e.g., Figure 8). These oscillations do not simply appear on the median line but appear with a similar form on many of the individual velocity profiles obtained from the inversion. This may in part be due to RADYN using a conservative  $2 \text{ km s}^{-1}$  microturbulent velocity throughout the atmosphere. Studies with the *Interface Region Imaging Spectrograph* (De Pontieu et al. 2014) have required significantly higher values to explain the nonthermal broadening in Mg II h & k in chromospheric plage. Carlsson et al. (2015) found a value of  $\sim 7 \text{ km s}^{-1}$ , and the inversions performed with STiC (de la Cruz Rodriguez et al. 2019) suggest a value of  $\sim 8 \text{ km s}^{-1}$  for the same observation. We suggest, then, that the INN needs to broaden the line to match observations and uses an oscillating velocity and higher temperature in the  $\tau = 1$  region to achieve this. To better constrain the parameters in the upper chromosphere and transition region requires computation of lines such as Mg II h

& k or Si IV 1403 Å, but these are not currently calculated in RADYN. While the emission from Mg II h & k could be computed from populations in statistical equilibrium using RH, it is essential to verify whether the nonequilibrium effects are important for these lines in flares.

There are several additional assumptions made during the training process that need to be considered when applying the INN.

1. Only the line profiles from the  $\mu \approx 0.9531$  ray angle were included in the training set. This is the emergent radiation at an angle  $\cos^{-1}\mu \approx 17^\circ.6$  to the normal of the atmospheric layers of the plane-parallel atmosphere used in RADYN. The emergent radiation detected from the flare discussed in Section 4 is approximately  $37^\circ$  from the local vertical. Assuming a plane-parallel atmosphere, the layers appear thicker by a factor of  $1/\mu$  than their depth along the normal to the atmosphere, so shallower layers may have a more significant effect than is predicted by the training set. The altitude stratification in the training set is perpendicular to the solar surface at this assumed  $\mu$ -ray angle to the observer.
2. Although different beam parameters are used, the simulations in the F-CHROMA RADYN grid all use the same 20 s triangular heating pulse, leading to a particular temporal sequence in the run of atmospheric properties that may not occur for different heating profiles (or indeed for different heating methods). As the inversions performed in Section 4 appear well constrained, this does not appear to be an issue.

To summarize, our novel technique using an INN trained with simulations from the radiation hydrodynamic model RADYN to solve the inverse problem of determining the solar atmospheric parameters given chromospheric spectral-line profiles lifts several restrictions that affect other inversion methods, such as enforcing hydrostatic equilibrium, that make these methods unusable for energetic atmospheres. The method is fast to train and very rapid to apply to data and has proven accurate on unseen validation tests; early results are very convincing and in broad agreement with previous analyses. This method of solving inverse problems is computationally tractable when a prior forward exists and could be leveraged to solve many other astrophysical problems. The code is available online under the MIT license<sup>7</sup> at <https://github.com/Goobley/Radynversion> and will soon be added to the RadynPy<sup>8</sup> (Osborne 2019) python package.

C.M.J.O. acknowledges support from the UK’s Science and Technology Facilities Council (STFC) doctoral training grant ST/R504750/1. J.A.A. acknowledges a data-intensive science studentship with the STFC “ScotDIST” center for doctoral training supported by grant ST/R504750/1. L.F. acknowledges support from STFC grant ST/P000533/1. The authors are grateful to M. Carlsson and the F-CHROMA collaboration for the production and availability of the grid of RADYN simulations. The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement No. 606862 (F-CHROMA) and from the Research Council of

Norway through the Programme for Supercomputing. The authors would like to thank P.J.A. Simões for helpful discussions and general constructive advice. The authors are also grateful to the reviewer for helpful comments and corrections.

## Appendix MMD

This section draws heavily on Gretton et al. (2012) and the lectures on this topic given at the Machine Learning Summer School Madrid 2018 (see footnote 4).

Training the INN is made possible by the use of the MMD. The MMD is a technique for determining the distance between probability distributions  $P$  and  $Q$  using observations  $X = \{x_1, \dots, x_m\}$  and  $Y = \{y_1, \dots, y_n\}$  drawn in an independent and identically distributed fashion from  $P$  and  $Q$ , respectively. The MMD can be mathematically expressed as

$$\begin{aligned} \text{MMD}^2 &= \|\mu_P - \mu_Q\|_{\mathcal{F}}^2 \\ &= \langle \mu_P, \mu_P \rangle_{\mathcal{F}} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{F}} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{F}} \end{aligned} \quad (14)$$

where  $\mathcal{F}$  is an RKHS known as the feature space, with elements known as features;  $\langle \cdot, \cdot \rangle_{\mathcal{F}}$  denotes the inner product in the feature space; and  $\mu_A$  represents the expectation vector of the features of  $\mathcal{F}$  evaluated for the distribution  $A$ .

Let  $X$  be a nonempty space with a positive definite kernel  $k : X \times X \rightarrow \mathbb{R}$  and  $\phi : X \rightarrow \mathcal{F}$  a feature map; then, for all  $x, y \in X$ ,

$$k(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathcal{F}}. \quad (15)$$

The feature spaces of kernels such as the Gaussian kernel

$$k(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, \quad \sigma > 0$$

are in fact infinite-dimensional, but the kernel trick of Equation (15) allows the inner product between vectors in this space to be written in closed form. The reproducing property of the RKHS states simply that under the inner product of features in  $\mathcal{F}$ , the kernel will always be recovered. For a positive definite kernel, there is a unique RKHS  $\mathcal{F}$  with a reproducing kernel  $k$  whose features are a subset of  $\mathcal{F}$ ; therefore, a feature map is not unique, but the kernel is.

Then,  $\mu_P$  from Equation (14) can be written in terms of the features of  $\mathcal{F}$ ,

$$\mu_P = [\dots \mathbb{E}_P[\phi_i(X)] \dots], \quad (16)$$

where  $\mathbb{E}_P$  denotes the expectation value of its argument with respect to  $P$  and  $\phi_i$  is the  $i$ th feature of  $\phi$ . From this definition, we can write

$$\langle \mu_P, \mu_Q \rangle_{\mathcal{F}} = \mathbb{E}_{P,Q}[k(x, y)], \quad (17)$$

where  $\mathbb{E}_{P,Q}[k(\cdot, \cdot)]$  denotes the expected kernel of  $P$  and  $Q$ , where  $x \sim P$  and  $y \sim Q$  (and  $a \sim A$  indicates that  $a$  is drawn in an unbiased way from  $A$ ).

<sup>7</sup> <https://opensource.org/licenses/MIT>

<sup>8</sup> <https://github.com/Goobley/radynpy>

Now, from the expansion in Equation (14), we have

$$\begin{aligned} \text{MMD}^2 &= \|\mu_P - \mu_Q\|_{\mathcal{F}}^2 \\ &= \mathbb{E}_P[k(x, x')] + \mathbb{E}_Q[k(y, y')] - 2\mathbb{E}_{P,Q}[k(x, y)]. \end{aligned} \quad (18)$$

For finite observations  $X$  and  $Y$  (of length  $n$ ), this then gives an unbiased sample estimate of the MMD:

$$\begin{aligned} \widehat{\text{MMD}}_u^2 &= \frac{1}{n(n-1)} \sum_{i \neq j} k(x_i, x_j) \\ &\quad + \frac{1}{n(n-1)} \sum_{i \neq j} k(y_i, y_j) - \frac{2}{n^2} \sum_{i,j} k(x_i, y_j). \end{aligned} \quad (19)$$

Due to the efficiency of the matrix operations used to compute the MMD loss in our training scheme, we compute a biased sample estimate of the MMD:

$$\widehat{\text{MMD}}_b^2 = \frac{1}{n^2} \sum_{i,j} (k(x_i, x_j) + k(y_i, y_j) - 2k(x_i, y_j)). \quad (20)$$

The bias on this statistic simply increases the expected MMD result but has the advantage of remaining positive even when  $P = Q$ , which works better with the optimizer used to train the INN.

### ORCID iDs

Christopher M. J. Osborne  <https://orcid.org/0000-0002-2299-2800>

John A. Armstrong  <https://orcid.org/0000-0003-1589-9365>

Lyndsay Fletcher  <https://orcid.org/0000-0001-9315-7899>

### References

- Allred, J., Hawley, S. L., Abbett, W., & Carlsson, M. 2005, *ApJ*, **630**, 573
- Allred, J. C., Kowalski, A. F., & Carlsson, M. 2015, *ApJ*, **809**, 104
- Ardizzone, L., Kruse, J., Wirkert, S., et al. 2018, arXiv:1808.04730
- Asensio Ramos, A., Bueno, J. T., & Degl'Innocenti, E. L. 2008, *ApJ*, **683**, 542
- Beck, C., Choudhary, D. P., Rezaei, R., & Louis, R. E. 2015, *ApJ*, **798**, 100
- Bradshaw, S. J., & Cargill, P. J. 2013, *ApJ*, **770**, 12
- Brown, S. A., Fletcher, L., Kerr, G. S., Labrosse, N., & Kowalski, A. F. 2018, *ApJ*, **862**, 59
- Canfield, R. C., Gunkler, T. A., & Ricchiazzi, P. J. 1984, *ApJ*, **282**, 296
- Carlsson, M., Leenaarts, J., & De Pontieu, B. 2015, *ApJL*, **809**, L30
- Carlsson, M., & Stein, R. 1992, *ApJ*, **397**, 59
- Carlsson, M., & Stein, R. F. 1997, *ApJ*, **481**, 500
- Cauzzi, G., Fletcher, L., Mathioudakis, M., et al. 2014, AAS Meeting, **224**, 123.39
- Cheng, J. X., Ding, M. D., & Li, J. P. 2006, *ApJ*, **653**, 733
- Cybenko, G. 1989, *Math. Control Signals Syst.*, **2**, 303
- da Costa, F. R., Kleint, L., Petrosian, V., Liu, W., & Allred, J. C. 2016, *ApJ*, **827**, 38
- de la Cruz Rodríguez, J., Leenaarts, J., Danilovic, S., & Uitenbroek, H. 2019, *A&A*, **623**, A74
- de la Cruz Rodríguez, J., Löfdahl, M. G., Sütterlin, P., Hillberg, T., & Rouppe van der Voort, L. 2015, *A&A*, **573**, A40
- De Pontieu, B., McIntosh, S. W., Hansteen, V. H., & Schrijver, C. J. 2009, *ApJL*, **701**, 1
- De Pontieu, B., Title, A. M., Lemen, J. R., et al. 2014, *SoPh*, **289**, 2733
- Dinh, L., Krueger, D., & Bengio, Y. 2014, arXiv:1410.8516
- Dinh, L., Sohl-Dickstein, J., & Bengio, S. 2016, arXiv:1605.08803
- Dorfi, E. A., & Drury, L. O. 1987, *JCoPh*, **69**, 175
- Fang, C., Henoux, J., & Gan, W. 1993, *A&A*, **274**, 917
- Fisher, G. H., Canfield, R. C., & McClymont, A. N. 1985, *ApJ*, **289**, 414
- Fletcher, L., Hannah, I. G., Hudson, H. S., & Metcalf, T. R. 2007, *ApJ*, **656**, 1187
- Graham, D. R., & Cauzzi, G. 2015, *ApJL*, **807**, L22
- Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. 2012, *J. Mach. Learn. Res.*, **13**, 723
- Heinzel, P., Kašparová, J., Varady, M., Karlický, M., & Moravec, Z. 2016, in *Proc. IAU Symp. 320, Solar and Stellar Flares and their Effects and Planets* (Cambridge: Cambridge Univ. Press), 233
- Ichimoto, K., & Kurokawa, H. 1984, *SoPh*, **93**, 105
- Kennedy, M. B., Milligan, R. O., Allred, J. C., Mathioudakis, M., & Keenan, F. P. 2015, *A&A*, **578**, A72
- Kerr, G. S., Fletcher, L., Russell, A. J. B., & Allred, J. C. 2016, *ApJ*, **827**, 1
- Kingma, D. P., & Ba, J. L. 2014, arXiv:1412.6980
- Kowalski, A. F., Allred, J. C., Daw, A., Cauzzi, G., & Carlsson, M. 2017, *ApJ*, **836**, 12
- Kretzschmar, M. 2011, *A&A*, **530**, A84
- Krucker, S., Hudson, H. S., Jeffrey, N. L., et al. 2011, *ApJ*, **739**, 96
- Kuridze, D., Henriques, V., Mathioudakis, M., et al. 2017, *ApJ*, **846**, 9
- Kuridze, D., Henriques, V., Mathioudakis, M., et al. 2018, *ApJ*, **860**, 10
- Kuridze, D., Mathioudakis, M., Simões, P., et al. 2015, *ApJ*, **813**, 125
- Mein, P., Mein, N., Heinzel, P., et al. 1997, *SoPh*, **172**, 161
- Metcalf, T. R., Canfield, R. C., Avrett, E. H., & Metcalf, F. T. 1990, *ApJ*, **350**, 463
- Milligan, R. O., Kerr, G. S., Dennis, B. R., et al. 2014, *ApJ*, **793**, 70
- Neupert, W. 1968, *ApJL*, **153**, 59
- Osborne, C. M. J. 2019, Goobly/radynpy: Contribution Function Update, Zenodo, doi:10.5281/zenodo.2547562
- Raschka, S. 2015, *Python Machine Learning* (Birmingham: Packt Publishing)
- Rumelhart, D., Hinton, G., & Williams, R. 1986, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, ed. J. Feldman, P. Hayes, & D. Rumelhart (Cambridge, MA: MIT Press), 318
- Scharmer, G. 2006, *A&A*, **447**, 1111
- Scharmer, G. B., Bjelksjö, K., Korhonen, T., Lindberg, B., & Petterson, B. 2003, *Proc. SPIE*, **4853**, 341
- Scharmer, G. B., Narayan, G., Hillberg, T., et al. 2008, *ApJL*, **689**, L69
- Schmidhuber, J. 2015, *NN*, **61**, 85
- Simões, P. J. A., Kerr, G. S., Fletcher, L., et al. 2017, *A&A*, **605**, A125
- Skumanich, A., & Lites, B. W. 1987, *ApJ*, **322**, 473
- Socas-Navarro, H., de la Cruz Rodríguez, J., Asensio Ramos, A., Trujillo Bueno, J., & Ruiz Cobo, B. 2015, *A&A*, **577**, A7
- Socas-Navarro, H., Trujillo Bueno, J., & Ruiz Cobo, B. 2000, *ApJ*, **530**, 977
- Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Lanckriet, G. R. G., & Schölkopf, B. 2009, in *Advances in Neural Information Processing Systems 22*, ed. Y. Bengio et al. (Vancouver: Curran Associates Inc.), 1750
- Tolstikhin, I., Bousquet, O., Gelly, S., & Schoelkopf, B. 2017, arXiv:1711.01558
- Uitenbroek, H. 2001, *ApJ*, **557**, 389
- Van Noort, M., Van Der Voort, L. R., & Löfdahl, M. G. 2005, *SoPh*, **228**, 191
- Varady, M., Kašparová, J., Moravec, Z., Heinzel, P., & Karlický, M. 2010, *ITPS*, **38**, 2249
- Vernazza, J. E., Avrett, E. H., & Loeser, R. 1981, *ApJS*, **45**, 635
- Withbroe, G., & Noyes, R. 1977, *ARA&A*, **15**, 363
- Wulser, J., & Marti, H. 1989, *ApJ*, **341**, 1088