The Flare Necessities: Machine Learning Tools for Solar Flare Data Analysis

John Andrew Armstrong

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Physics & Astronomy
College of Science and Engineering
University of Glasgow

February 2022

This thesis is my own composition except where indicated in the text. No part of this thesis has been submitted elsewhere for any other degree or qualification.

May 4, 2022

I will stay behind, to gaze at the Sun. The Sun is a wondrous body. Like a magnificent father! If only I could be so grossly incandescent!

— Solaire of Astora

Dark Souls

Abstract

The study of the lower flaring atmosphere of the Sun is one facet of understanding the complex physics involved in solar flares and their effect on space weather and the Earth. Despite a rich history of investigation into the study of the lower flaring atmosphere, there are still many unanswered questions in regards to the mechanism of energy deposition and the response to such energy being injected into the atmosphere. This thesis aims to provide tools for future researchers to rigorously explore these problems. In particular, this thesis looks at how machine learning – with a particular focus on deep learning – can improve data storage and analysis pipelines as well as uncover new results from data that were not feasibly possible before.

In Chap. 1, the standard model of a solar flare is introduced and its extension to three dimensions explained. This allows for the definition of a flare ribbon – the brightest points in the lower solar atmosphere resulting from direct heating from a flare – which is a key observational feature whose origin is explored in later chapters. A brief history of study on flare ribbons is then given with a particular focus on the asymmetries in spectral lines that show clear flare ribbons. These asymmetries link directly to the velocity field in the flaring atmosphere as a static atmosphere would yield symmetric profiles. This gives a direct diagnostic of the motion happening in the atmosphere as it is heated and the ribbons evolve.

In Chap. 2, the field of deep learning is introduced from its inception to the current models used today. This chapter covers how to build and train deep neural networks and some best practices when implementing these tools.

The telescopes and detectors used to obtain the data analysed in Chaps. 4 - 6 are described in Chap. 3. In this chapter, the inner workings of the Swedish 1-m Solar Telescope's CRisp Imaging SpectroPolarimeter (SST/CRISP), Hinode's Solar Optical Telescope (Hinode/SOT) and Solar Dynamics Observatory's Atmospheric Imaging

Assembly (SDO/AIA) is described.

Chap. 4 introduces a deep convolutional neural network (CNN) trained on Hα images from Hinode/SOT for solar image classification. This is trained to distinguish between five classes of solar features prominent in Hα: filaments, flare ribbons, prominences, sunspots and the absence of any of the other four features. The final model has a validation accuracy of 99.2% misclassifying only one image in the validation dataset. The trained CNN is then tested with adversarial examples from SDO/AIA UV continua and EUV spectral line images where the features look perceptually different but still identifiable to the human eye. This demonstrates that the network cannot identify these features in different wavelengths well and to extend this network to non-visible wavelengths, the training set must be expanded to include such wavelengths. The trained CNN in this chapter is used further in Chap. 5 for transfer learning – the process of using a trained deep learning model to influence the training of another, related deep learning model.

In Chap. 5, a method based on deep learning for correcting the atmospheric effects in optical solar flare observations is presented. This takes the form of a fully convolutional autoencoder trained on data from SST/CRISP imbued with synthetic seeing described by the model developed in the first sections of the chapter. The trained model works well on the validation dataset showing accurate reconstruction of both spatial and spectral elements of the data. SST/CRISP data with real atmospheric seeing is then corrected by the trained model. The sources of error in this reconstruction are discussed with a coarse error estimate on the recovered intensity values used.

Then in Chap. 6 a novel deep learning method for estimating the parameters of the flaring atmosphere from observations is presented – an Invertible Neural Network (INN). The INN is trained on synthetic flare data produced by the one dimensional radiation hydrodynamics code RADYN with near-perfect restoration of the atmospheric paramters during validation. This is then applied to a single pixel from a CRISP image to show the power of this method in disentangling the ambiguity in the velocity field responsible for observed asymmetry in the spectrum. This method is then applied to flare ribbons as a whole – which are selected through a combination of a Gaussian Mixture Model (GMM) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) – to determine the specific motions of the flaring velocity field responsible for the observed spectral line asymmetries.

Contents

List of Tables

List of Figures

Acknowledgements

If the last four and a half years have taught me anything it is that solar flares are hard and machine learning is hard and combining the two is hard but it is a hell of a lot easier when you're surrounded by people you adore.

To steal a phrase from her previous two students, "first and foremost" I would like to thank Lyndsay Fletcher, my supervisor. Thank you for helping to shape me as a researcher and for always giving me time to ask about anything from flares to how to write a review for a paper. Thank you for stumbling through the dark with me as we tried to figure out what neural networks do. Thank you for all of your encouragement these past years. Thank you for everything.

Next I would like to send a heartfelt thank you to Alec MacKinnon. I did not get to have as many chats with you during my PhD as I had hoped but your guidance during my undergraduate degree cannot be understated as you ignited my passion for plasma physics and then got me interested in solar flares via our chats about energetic particles during meetings. I would also like to thank Paulo Simões who, probably inadvertently, sent me down the path I am on now. In 2013, you were my second year astronomy supervisor where one day discussing the work you were doing I remember thinking "this is what I want to do." Alec and Paulo, thank you.

To everyone in 604, thank you for providing a fantastic work environment during my PhD, I thoroughly enjoyed coming into the office (back when we were allowed to). Particularly, thanks to Chris for always humouring any discussion I fancied having whether work-related or not. To Aaron, Kris and David, I miss hunting legendary Pokémon, we should do that again some day.

Aaron and Craig, thank you for your friendship I truly treasure it.

To my family: Mum, Bekki and Dad – thank you for making me the person I am today and always encouraging me to follow my dreams and for all the support you provided. Sharon & Papa – I hope I'm making you proud.

To all of my little mischief-makers: Aria, Louise, Elena/Swoog, Peggy, Maisie, Annie, Sasha, Moggy, Mable, Daisy, Adore, Orlando, Jinkx, Bianca, Satan; I love you all, thank you for endless shenanigans and keeping me entertained over the years.

Magda, I never would have been able to do any of this without your support and love and your help through all of my obsessive, compulsive phases. You are my best friend and the person I want to spend my life with, I can't wait to get cats and rats and dogs and horses and grow old together (or as old as someone from the East End of Glasgow gets).

Declaration

With the exception of chapters 1, 2 and 3, which contain introductory material, all work in this thesis was carried out by the author unless otherwise explicitly stated.

1 | The Sun and its Flares

As viewed from the Earth, the Sun is the brightest star in the sky. Sporadically, the Sun expels up to 10% of its power from localised regions due to restructuring of its local magnetic field. This is known as a solar flare.

Solar flares are immense, explosive releases of energy occurring in the solar atmosphere producing radiation across the electromagnetic spectrum. As well as radiation, flares can also drive large quantities of material from the solar atmosphere into interplanetary space – this phenomenon is known as a coronal mass ejection (CME). Earthbound consequences of solar flares can have beautiful and catastrophic effects. The aurora polaris observed at high magnitudes of latitude on the Earth are formed via incident particles from the solar wind injecting their energy into the Earth's atmosphere. This injection of energy causes the emission of light of the aurorae. During a flare, if there is an associated CME, the interaction between the magnetic clouds expelled from the solar atmosphere and the Earth's magnetosphere can greatly modify the aurorae observed, both in colour and location (if directed Earthwards). One such example comes from the mid-19th century where on the night of 1st September 1859 where the aurorae extended further towards the equator than at any other time in observed history (Tsurutani et al., 2003). The aurora borealis in the northern hemisphere was observed as far south as Honolulu, Hawaii with the aurora australis observed as far north as Santiago, Chile[1]. The typical vibrant green light of the aurorae was turned a dark crimson on this night. This wondrous display of light was accompanied by electromagnetic disturbances all over the world. The accounts examined in Shea and Smart (2006) speak of sparking telegraph wires

[1]There have been reports of people in the Rocky Mountains, USA, being able to read their newspaper using only the light from the aurorae (Green et al., 2006).

setting nearby materials alight and telegraph systems' operators receiving electric shocks from their equipment. This event came to be known as the Carrington Event for a much tamer reason.

Approximately, 18 hours before the bright lights and explosive damages, Richard Carrington was observing a sunspot region before a large brightening lasting around five minutes happened above the group of sunspots (Carrington, 1859). This was in fact the first published observation of a solar flare.

The interaction between the Sun and the Earth and the effects that large solar events such as the Carrington event can have on modern satellite and Earth-based systems is a driving force behind solar flare research. This thesis focuses on the intersection between machine learning and solar flare research, particularly, leveraging machine learning methods to streamline flare data analysis and provide new insights into the physics of solar flares. Chap. 2 provides an introduction to deep learning – the field of machine learning involving the use of deep neural networks – covering the construction, optimisation and utilisation of such methods. Chap. 3 introduces the optical spectral lines used in later chapters to analyse flares and the instruments used to observe these lines. Chap. 4 presents a method using a deep convolutional neural network for identifying features in images of the Sun and how this can be used in data pipelines and to help train other deep learning models. Chap. 5 introduces a new method for correcting for atmospheric effects in solar flare observations using a neural network to learn how to perform these corrections by training it on examples with originally good seeing marred by a model of the seeing based on the statistics of turbulent media. Chap. 6 presents an application of an invertible neural network to learn the inverse problem between spectral line observations and atmospheric parameters from solar flare simulations. This is then used to explore the relationship between the asymmetry of the spectral lines and the flare velocity field. Chap. 7 recaps the research in the thesis and outlines next steps to be taken to further utilise machine learning in flare observations. The next section introduces the different layers of the solar atmosphere and what physics makes them distinct.

1.1 The Sun's Atmosphere

The solar atmosphere is considered to have three distinct layers: the photosphere, chromosphere and corona. Typically, the photosphere is defined as the region of

optical depth unity at $\lambda = 5000$Å and is referred to as the "surface" of the Sun. The photosphere is characterised by visible continuum with some strong absorption lines due to atomic/ionic species absorbing the emergent flux from below. The chromosphere is the region approximately 500km above the photosphere and is about 1000km thick before transitioning into the corona. The chromosphere is the least understood layer of the atmosphere and the focus of much modern solar physics research. The temperature profile in the solar atmosphere gives an indication of the state of each of the layers. The photosphere has a quasi-constant temperature around 5771K which decreases in the lower chromosphere to 4000K (the temperature minimum region) before increasing rapidly to 20000K (Vernazza et al., 1981). Semi-empirical models show that the mass density in the photosphere is of the order 10^{-7} g cm^{-3} which falls to 10^{-9} g cm^{-3} at the base of the chromosphere (pp. 155, 289; Foukal, 2004). Above this is the corona where there is an almost discontinuous temperature gradient, over a plasma width of ~50 km, and the temperature increases to a few million Kelvin and the plasma becomes very tenuous compared to the photosphere ($\rho \approx 10^{-11}$ g cm^{-3}).

1.2 Solar Flares

Solar flares are a phenomenon that affect all layers in the solar atmosphere. They are highly energetic (up to $\sim 10^{32}$ ergs) brightenings of the solar atmosphere across all wavelengths, occurring in and around active regions. Magnetic reconnection is hypothesised to take place at a region of twisted magnetic field in the corona which is thought to be the cause of a flare. Magnetic reconnection is the process where the direction of magnetic flux is discontinuously changed in a short period of time which causes the magnetic field lines to change their configuration into a lower-energy topology. The magnetic energy previously stored by the pre-reconnection field is then converted to heating of the chromosphere, particle acceleration and (sometimes) bulk plasma motion which is often referred to as a coronal mass ejection (CME; Fletcher et al., 2011).

An active region is an area from the photosphere to the corona accounting for all detectable radiation brought about by an extension of the magnetic field. This is caused by the emergence of twisted magnetic flux with strengths on the order of kilo-Gauss into the corona from below the photosphere (van Driel-Gesztelyi and Green, 2015). These emergent fields are driven by currents in the sub-photospheric plasma

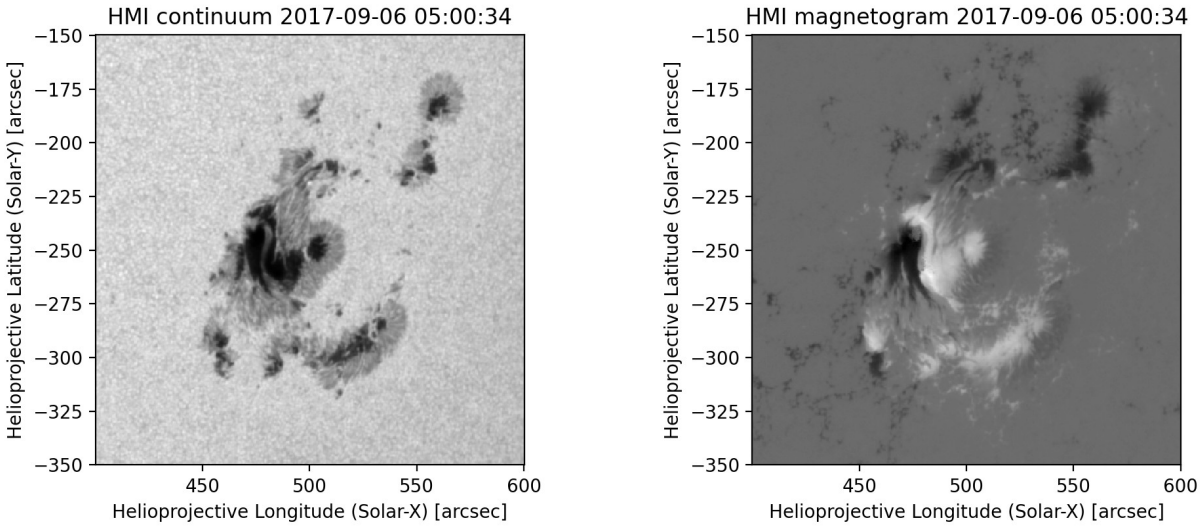

Figure 1.1: An example of an active region (NOAA AR12673) imaged in Fe I λ6173Å (left panel) by SDO/HMI highlighting the granular structure of the photosphere with an accompanying magnetogram (right panel) showing the emergent magnetic flux in this active region. Note that in the magnetogram, the darker and brighter areas correspond to opposite magnetic polarities showing the complexity of the geometry of the field.

and the emergence is thought to come from the magnetic buoyancy instability which drives the plasma upwards (Choudhuri, 1998). The plasma brings the twisted magnetic flux with it due to the plasma below the surface having a high magnetic Reynolds number (i.e. the magnetic flux is frozen in to the plasma; Alfvén, 1942). When a maximum energy threshold is exceeded that a field can store, it is explosively released – i.e. a flare. An example of an active region (NOAA AR12673) imaged by Solar Dynamics Observatory's Helioseismic and Magnetic Imager (SDO/HMI) is shown in Fig. 1.1. The left panel of Fig. 1.1 shows an image of the active region taken in Fe I λ6173Å and the right panel shows the magnetogram for the active region where the darker and brighter regions show opposite polarity magnetic fluxes. This shows the complexity of the geometry of the magnetic field in this active region.

The following description of a solar flare is based on the two-dimensional standard flare model known as the CSHKP model named after the authors who originated the model (Carmichael, 1964; Sturrock, 1966; Hirayama, 1974; Kopp and Pneuman, 1976) and its extensions into three dimensions. This is illustrated by the cartoon in Fig. 1.2 from Holman (2012)[2]. The temporal evolution of an active re-

[2]This cartoon was found in the "Grand Archive of Flare and CME Cartoons" compiled by H. S. Hudson with redesigned website by N. Chrysaphi available at https://www.astro.gla.ac.uk/cartoons/

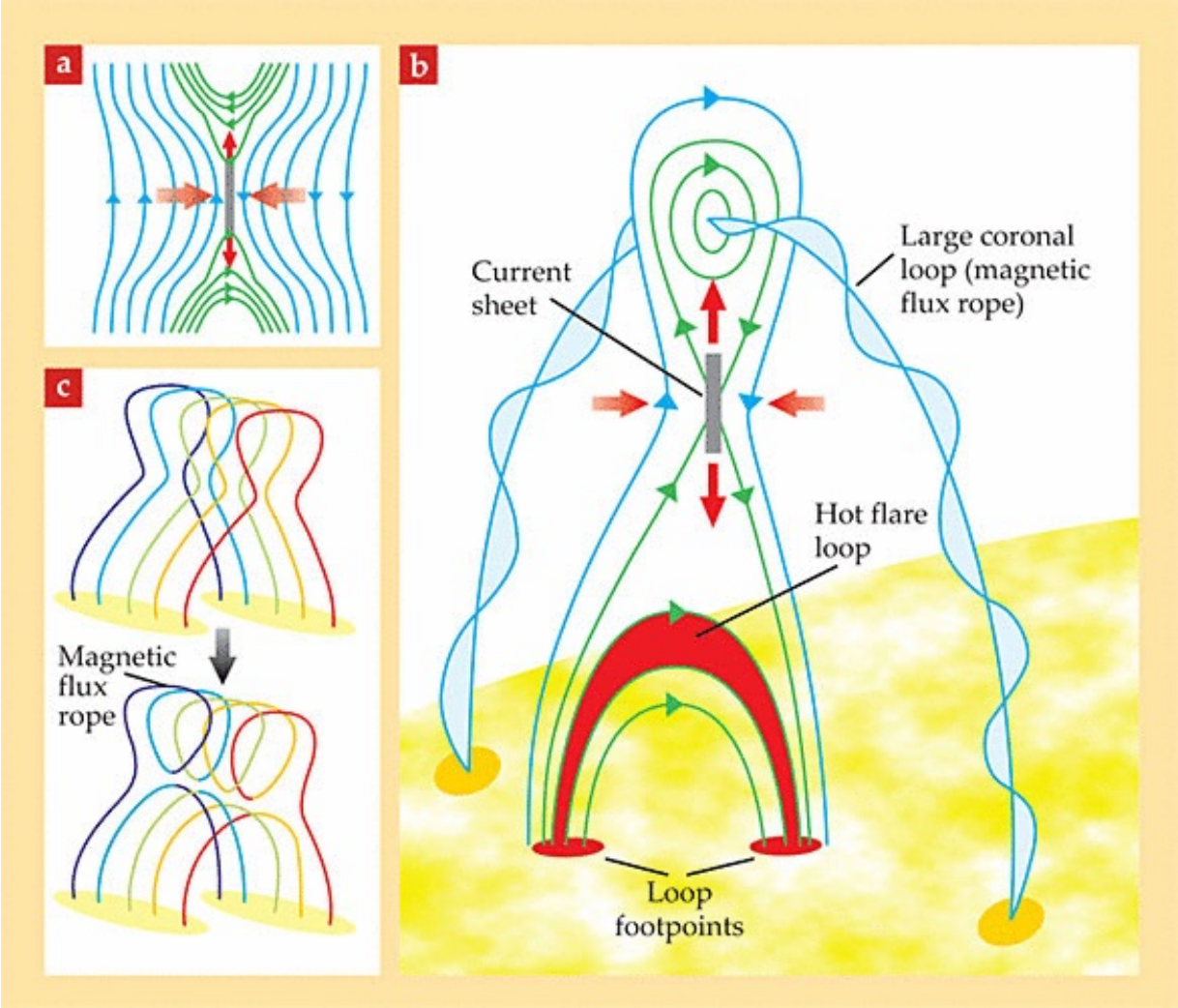

Figure 1.2: Cartoon of the standard CSHKP flare model and its extension to three dimensions from Holman (2012). Panel (a) shows a zoomed in view of the reconnection region where the oppositely directed magnetic field lines are swept into the diffusion region by the flows in the corona leading to the reconnection event and energy release. This panel shows the pre-reconnected field lines (blue) and post-reconnected field lines (green). Panel (b) shows the original 2D CSHKP model showing the reconnection in the corona with energy travelling outwards and inwards producing Earth-bound SEPs and CMEs and Sun-bound post-flare loops that have footpoints deep in the solar atmosphere. The magnetic flux rope perpendicular to this 2D structure is the extension into three dimensions giving electrons accelerated in the flare region the freedom to travel along this field producing gyrosynchrotron and X-ray emission. Panel (c) shows the full extension to 3D with multiple CSHKP structures connected by a flux rope undergoing reconnection with the footpoints of the post-flare loops forming elongated structures in the lower solar atmosphere: flare ribbons.

gion/flare system typically has three stages: preflare evolution, the impulsive phase and the gradual phase. Preflare evolution gives initial (albeit ambiguous) signs of activity on the Sun at an active region that a flare might occur. These are small-scale brightenings in ultraviolet (UV) to soft X-ray (SXR) wavelengths (4000-1Å which can occur on the order of tens of minutes before the flare eruption. High resolution observations of these preflare precursors indicate that they happen close to the flare site but not exactly where the flare occurs (Fárník et al., 1996). Non-thermal spectral line broadening due to plasma turbulence has also been observed in these regions up to the order of hours before the flare indicating turbulent flows near the active region (Harra et al., 2001, 2009).

The impulsive phase is the first stage of energy release and lasts a very short amount of time (of the order tens of seconds to tens of minutes). Many forms of radiation across the electromagnetic spectrum are observed throughout the impulsive phase giving indications of the processes and interactions occurring throughout. The bulk of the impulsive phase emission occurs in the chromosphere (Hudson, 1972) particularly at the footpoints (the base of the magnetic structure) and the ribbons (approximately the chromospheric intersection with the separatrix surfaces between the pre- and post-reconnected field). As the flare reconnection region is thought to be in the corona and ribbons brighten nearly simultaneously, the energy which causes the chromospheric emission must be transported extremely quickly. This phenomenon is not well understood and is arguably the biggest open question in flare physics. Kurokawa et al. (1986) showed that the heat conduction front in a solar flare moves too slowly to account for the nearly instantaneous heating of the lower solar atmosphere ruling out conduction as a possible heating mechanism of the lower atmosphere during the impulsive phase. The hypothesised explanation for the accelerated particles that cause heating in the chromosphere is beams of non-thermal electrons being accelerated into the solar atmosphere and the collisional transfer of their energy to the particles in the chromosphere causes heating (Brown, 1971). However, there is also a hypothesis of the energy being carried down the reconnected field and deposited by Alfvénic waves (Fletcher and Hudson, 2008; Hudson and Fletcher, 2009). A combination of these two energy transfer modes is also possible. This can be investigated by studying the type of emission observed during the impulsive phase.

Heating of the chromosphere during the flare leads to an increase in optical,

`index.html`.

near-infrared (NIR) and ultraviolet/extreme ultraviolet (UV/EUV) emissions. This accounts for a large portion of the released energy from a flare and causes emission from lines that are observed in absorption in the non-flaring Sun, such as hydrogen alpha (Hα) which is the Balmer series transition from level 3 to level 2, and singly-ionised calcium (Ca II) which has several prominent lines from UV to NIR. Hα and Ca II λ8542 are two of the most prominent lines in a flaring chromosphere and are therefore vital to chromospheric diagnostics (see Sec. 1.2.1 for more details). Deciphering these relationships is extremely important in understanding the chromospheric response to a flare. As well as optical lines, optical continua are produced by free-bound interactions (between electrons and ions where there is electron capture).

Acceleration of nuclei to high energies ($\geq$10 MeV) leads to nuclear recombination emitting gamma rays most notably from neutron capture (2.2 MeV) and electron-positron annihilation (511 keV) along with a continuum resulting from free-free interactions (between electrons and ions where there is no electron capture; Lingenfelter and Ramaty, 1967; Hua and Lingenfelter, 1987). Free-free interactions also lead to the emission of hard X-rays (HXR, E > 10 keV) at the footpoints. These HXRs are also observed at the loop-top.

Energy from the flaring region also accelerates particles into interplanetary space – these are known as solar energetic particles (SEPs). SEP electrons emit radio waves from mode conversion of Langmuir waves (Emslie and Smith, 1984). Electrons trapped by coronal magnetic fields emit SXR continuum ($E \leq$ 10 keV) due to thermal brehmsstrahlung and high-energy radio waves due to their gyrational motion about the magnetic field. The flux of these SXRs is how flares are classified from GOES (Geostationary Operational Environmental Satellite). The different classes of flares is shown in Tab. 1. Each has a corresponding emission measure which yields the amount of plasma emissivity along the line-of-sight (LOS) and is given by the equation:

$$\mathrm{EM} = \int n_e^2 \, \mathrm{d}\mathcal{V}, \tag{1.1}$$

where $\mathrm{d}\mathcal{V}$ is the emitting volume of plasma and n_e is the electron number density of the plasma.

The gradual phase can last up to several hours depending on the magnitude of the flare (Fletcher et al., 2011). This phase is indicated by its slowly-decaying SXR intensity profile. Arcades of loops form filled with what is hypothesised to be plasma

Table 1.1: GOES Classification System for Solar Flares.

GOES Class	Flux [erg cm−2 s^{-1}]	SXR Emission Measure [cm^{-3}]
X10	$> 10^{1}$	10^{51}
X	$10^{-1} < F < 10^{0}$	10^{50}
M	$10^{-2} < F < 10^{-1}$	10^{49}
C	$10^{-3} < F < 10^{-2}$	10^{48}
B	$10^{-4} < F < 10^{-3}$	10^{47}
A	$< 10^{-4}$	10^{46}

from the chromosphere which must expand due to rapid heating in the impulsive phase. This is known as chromospheric evaporation (Neupert, 1968; Fisher et al., 1985; Graham and Cauzzi, 2015). This causes an increase in gas pressure in the corona due to an increase in density and due to the upflowing plasma having flaring temperatures (~10–30 MK). Conservation of momentum requires there to be a downflow in the chromosphere to balance the evaporative upflow. This is known as chormospheric condensation (Ichimoto and Kurokawa, 1984; Wülser and Marti, 1987). As the coronal loops cool, the arcade begins to emit in lower temperature lines such as Hα and EUV. Once the loops cool to emitting Hα they begin to drain under gravity. This is known as coronal rain. These upflows into the corona are coupled to downflows from the corona as these processes happen in overlapping time periods.

1.2.1 Observational Signatures of Optical Flare Ribbons

Flare ribbons are observed across the optical, ultraviolet and infrared parts of the spectrum as bright elongated structures. Their formation during solar flares is directly linked to energy deposition in the lower atmosphere from electrons accelerated from the reconnection region downwards (and potentially Aflvénic wave heating). As time progresses, ribbons have an apparent motion associated as they appear to move in the solar atmosphere. This is not true motion of the ribbons themselves but rather new parcels of the quieter lower solar atmosphere being excited by newly reconnecting field lines while the old parcels return towards their quiescent state or are excited even further to the point where the excitation is no longer observable in the wavelength being examined. This is visualised in Fig. 1.3 where one can see as time progresses that the brightest points in the flare ribbons at a given wavelength are in different spatial locations due to the heating of different parts of the

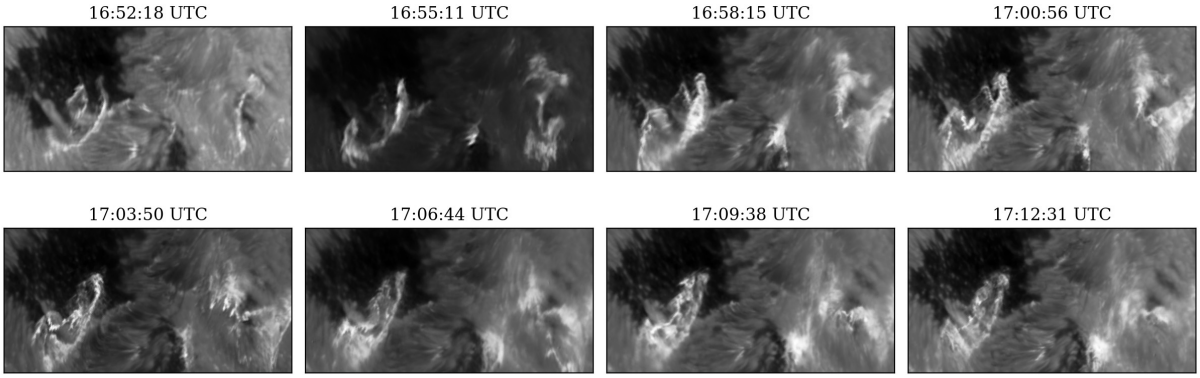

Figure 1.3: An example from the M1.1 SOL20140906T17:09 solar flare observed with SST/CRISP shown at eight different times to highlight the motion of flare ribbons as new parcels of the atmosphere are excited by reconnecting field lines. These images shown are taken in the red wing of Hα at $\Delta\lambda$ = +0.8Å.

atmosphere[3].

It is important to note that "true" ribbons are strictly defined as the emission directly caused by the heating of the lower atmosphere, not just any bright point that appears in an image. This is where spectroscopy becomes invaluable in the study of flare ribbons. The wings and the core of Hα are formed at different heights in the solar atmosphere with the wings forming in the upper photosphere/lower chromosphere and the cores forming in the mid-upper chromosphere (Vernazza et al., 1981). As such, images of the wings and core can be used as a method of detecting the "true" ribbon sources. There is a lot of complex motion in the chromosphere due to the fibril structure of the chromospheric plasma. As a result, when areas are heated by flare energy deposition, energy transfer through the fibrils can lead to bright points elsewhere in the chromosphere not directly heated by the flare. The photosphere does not contain this fibril structure and thus the bright points in the images of the line wing are considered the "true" ribbons – this will become important in Chap. 6 where the ribbon sources are isolated for analysis. An example of this effect is shown in Fig. 1.4 where a comparison of the flare ribbons at three different wavelengths in Hα (blue wing, line core, red wing) are shown along with the cotemporal observation from the SDO/AIA 1700Å UV image. This demonstrates the discussion above where there are bright points in the line core of Hα that do not

[3]Note, however, in this particular flare there is not much relative motion between the two ribbons, which is often observed in solar flares such as the flare shown in Svestka (1976, pp. 40, 41). This could be due to this being a shorter lived event compared to that in Svestka (1976) thus the ribbons do not experience as much lateral motion.

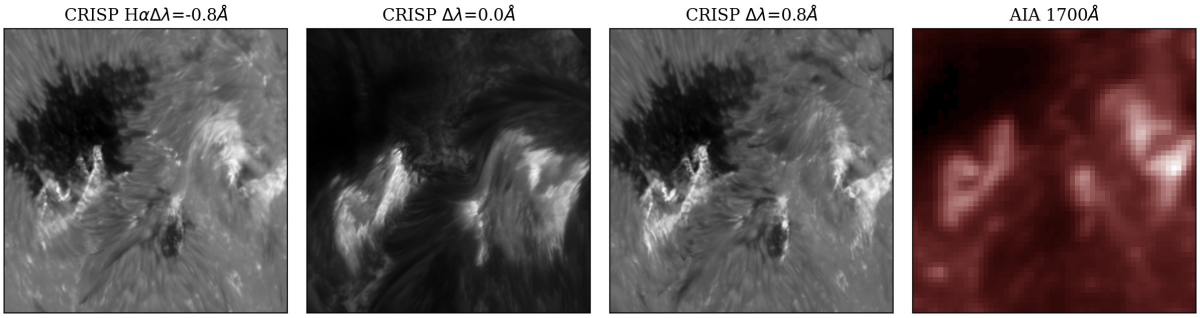

Figure 1.4: An example of the same flare as in Fig. 1.3 imaged at different wavelengths along the Hα spectrum (first three panels) and a comparison showing the ribbons in UV from SDO/AIA 1700Å (right panel). All observations are imaged at ~17:01 UTC. This demonstrates that the bright sources in the wing of the Hα spectrum are the "true" ribbon sources as they match well with some source in the line core and the ribbons as shown in the AIA image.

match up spatially with bright points in the Hα line wings or the AIA 1700Å image.

Studying the spectra of flare ribbons can uncover the dynamics of the flaring chromosphere and the energy source responsible. For instance, optical flare ribbon spectra are typically very asymmetrical in that the ratio of the blue wing to the red wing intensities differs from unity. This asymmetry is attributed to the flaring velocity field as spectral lines such as Hα and Ca II λ8542 would be symmetrical around the line core if there was no bulk motion of the plasma (Canfield and Gunkler, 1984; Fang et al., 1993; Cheng et al., 2006). This has been studied in great detail throughout the history of flare physics but a clear picture of ribbon dynamics is still elusive. This is partially due to the definition of asymmetry. In some literature, the asymmetry of a spectral line refers to ratio of the intensities of the spectral line wings defined close (within 1-2Å) of the line core (Canfield and Gunkler, 1984; Kuridze et al., 2015, and Chap. 6 of this thesis) while others refer to the asymmetry of the far wings of the spectral lines ($O(10)$Å away from the line core Ichimoto and Kurokawa, 1984). The ambiguity in the definition of asymmetry leads to discussion on different physics involved in the observations being analysed. The second reason for the difficulty in understanding ribbon dynamics is that the asymmetry itself (regardless of how it is defined) is ambiguous. Consider motion along the line of sight affecting the symmetry of an observed spectral line. If there is an excess of intensity on the blue side of the line, there can be either material emitting the observed radiation moving along the line of sight *towards* the observer of there could be material absorbing the observed radiation moving along the line of

sight *away* from the observer. Similarly, for an intensity excess on the red side of the line, there can be emitting material moving away from the observer along the line of sight or absorbing material moving towards the observer along the line of sight (Svestka, 1976; Ichimoto and Kurokawa, 1984; Heinzel et al., 1994). The disambiguation of the flare velocity field is important for understanding the dynamics of the flaring chromosphere and the conditions that lead to the observations. For example, Ichimoto and Kurokawa (1984) studied the enhancements of the far wings (up to ±15 Å from line centre) of Hα and determined that it was chromospheric condensation (the downwards motion of cooler flare material) responsible for the red asymmetries in their data. They also made estimations of these condensation velocities of ~40-100km s^{-1}. The data of Ichimoto and Kurokawa (1984) were taken using a slit spectrograph mounted at the 60cm Domeless Solar Telescope at Hide Observatory, University of Kyoto. They observed 30 flares in 1982 and analysed all in their study. The spectrograph was operated in a "sit and stare" mode meaning only a small area of each flare was observed. In these flares, only red asymmetries were observed at the onset which was in stark contradiction to Svestka (1976, and references therein) who reported on blue asymmetries being observed in the Hα spectrum at these times. Moreover, it was not until Canfield et al. (1990) that this difference was cleared up.

Canfield et al. (1990) showed that in a study of flares observed with the Sacramento Peak Observatory's Hα spectrograph that Hα spectra with a red asymmetry are more common, with the blue asymmetry profiles occurring only early in the impulsive phase of the flares. Similarly in this study, the spectra were taken with a wide passband (of ±4 Å) leading this discussion to being about the same class of asymmetries as Ichimoto and Kurokawa (1984). They also showed that Hα profiles with a red asymmetry show good spatial and temporal correlation to impulsive HXR emissions which they used to justify the Ichimoto and Kurokawa (1984) conclusion that the profiles are explained by condensation and evaporation processes (however, Canfield et al. (1990) noted that not all instances of red asymmetry are due to condensation). No physical interpretation of the blue asymmetry was concluded upon.

Moving on to the second definition of asymmetry – the relative intensity ratio of the line wings close to the vacuum wavelength – Canfield and Gunkler (1984) showed that the characteristic Hα flare spectrum with central reversal is the result of the Stark effect when nonthermal electrons heat the lower atmosphere. The asymmetry between the two enhanced wings then comes from the motion of the re-

gions forming the wings. In flare profiles with no central reversal in Hα, there was found to be a higher coronal pressure (Canfield and Gunkler, 1984). Canfield and Gunkler (1984) also remarked that other Hα flare characteristics are not as easily explainable without additional information – e.g. low Hα emission is not necessarily characteristic of low flare heating as Hα emission can be low when flare heating by nonthermal electrons is high but there is also a high value of the conductive flux.

Heinzel et al. (1994) studied how the flux of an electron beam heating the lower atmosphere affects the blue asymmetry in hydrogen Balmer lines α, β, γ, ϵ and the Ca II H spectral line. They found that the duration of the blue asymmetry decreases with increasing beam flux in their flare simulations due to heating (and thus the onset of condensation/evaporation) of high density regions quicker. However, no conclusive explanation is reached for the blue asymmetry.

Kuridze et al. (2015) revisited potential explanations for spectral line asymmetries during flares by comparing radiation hydrodynamic simulations with observations of Hα during a solar flare. They concluded that the cause of the asymmetries in Hα is inconclusive. Chap. 6 builds upon this work in analysing flare observations using an Invertible Neural Network (INN) and its properties to find an unambiguous solution to the asymmetry in the spectra.

This chapter has discussed the known physics behind solar flares as a whole split by the regions of the electromagnetic spectrum that are observed. A particular emphasis was placed on the formation of flare ribbons – the brightenings in optical and UV wavelengths in the lower solar atmosphere due to energy deposition – and the historical observations and analyses of these structures. This is due to the observations of flares in this thesis being primarily of optical wavelengths and thus the observed signature of flares are the ribbons. Moreover, the data analysis in Chap. 6 builds upon the historical work of determining the origin of spectral line asymmetry within and around flare ribbons.

2 | A Primer on Deep Learning

Rather than accounting for every edge case and boundary condition in a system, machine learning (ML) allows the computer to learn complex problems via high-dimensinoal optimisation. The main idea behind machine learning is that "data-driven" models are being created. That is, given a sufficient, diverse dataset, a model can be learned from the data to perform a specific task. What this task *is* determines what constitutes "sufficient" and "diverse" data, i.e. if one wanted a system to distinguish between images of dogs and stairs and the dataset contained only one breed of dog then the dataset would not be diverse enough to learn to distinguish all breeds of dogs from stairs. (Similarly, if the system is optimised using a single image of a dog and a single image of some stairs then it would not be well suited to tell the user about other dogs or stairs that are not the exact examples it has seen before). The “learning” that takes place by the system is the optimisation of parameters within the system to best estimate what the user desires.

Following this definition, machine learning can take many forms: from fitting analytical models through linear and logistic regression to modelling complex classification and regression tasks using neural networks. All machine learning algorithms can be expressed via the following equation,

$$Y = f(X;\Theta), \tag{2.1}$$

where Y is the desired output, X is the dataset, f is the process learned by the algorithm (e.g. the function that maps inputs to outputs or the determination of the underlying data distribution) and Θ is the set of parameters optimised within the system to learn the task at hand (henceforth Θ will be referred to as *learnable*

parameters).

There is a vast arsenal of different machine learning algorithms that can be leveraged to learn different tasks more effectively than others. There are two main distinctions when looking for an ML algorithm to use:

1. *How* the algorithm learns: supervised or unsupervised[1].

2. *What kind* of algorithm is being used: classical or deep.

"Classical" ML refers to the practice of using techniques not involving deep neural networks (DNNs) while "deep" is the opposite. Both classical and deep ML methods can be subdivided depending on how they learn the task: this can either be in a supervised or unsupervised manner. Supervised learning is when the dataset used to train an algorithm must be labelled such that the function learning the input to the output of the dataset has constraints within the optimisation space. For example, when learning what features are in an image (see Chap. 4 for more details), the dataset that the algorithm is learning from must describe what is in each image so that the algorithm can optimise its parameters to get as close as possible to the correct output for each input. Unsupervised learning is used when the data is to be explored for patterns and correlations (e.g. clustering algorithms), or when learning the distribution of the data. The former requires no extra information provided by the user to the algorithm and gives the algorithm freedom to distribute the data in the way it wants to. The latter requires a definite answer within the dataset used for learning but can produce probabilistic insights when used for inference (this is sometimes called *semi-supervised* learning).

One may construct the different families of ML algorithms by imagining a Punnett square with one parent carrying "Supervised" and "Unsupervised" genes while the other carries "Classical" and "Deep" genes. A visualisation of this is shown in Table 2.1. This table also gives examples of what algorithms constitute each type of machine learning.

Within this thesis, the main focus will be on *supervised deep learning* which consists of training deep neural networks (DNNs) in a supervised manner to learn the task at hand[2]. DNNs are often referred to as function approximators due to their satisfying of the Universal Function Approximation Theorem which states that a

[1]There is a third "how" known as reinforcement learning which is not explored in detail in this thesis.

[2]However, Chapter 6 will explore concepts in unsupervised classical and deep learning.

neural network of fixed depth and arbitrary width (Cybenko, 1989) or of arbitrary depth and fixed width (Lu et al., 2017) has the ability to approximate any continuous, well-defined function. To understand how DNNs came to be, how they learn and what purpose they can serve, we must visit the Cornell Aeronautical Laboratory in the 1950s and understand how Frank Rosenblatt's "perceptron" works.

Table 2.1: The four different families of machine learning algorithms: Supervised Classical, Unsupervised Classical, Supervised Deep and Unsupervised Deep. Each entry of the table shows examples of machine learning algorithms from each family.

	Supervised	Unsupervised
Classical	Decision Trees, Support Vector Machines, Shallow Neural Networks	Clustering, Dimensionality Reduction, Random Forests
Deep	Multi-layer Neural Networks, Convolutional Neural Networks, Residual Neural Networks	Generative Adversarial Networks, Variational Autoencoders, Invertible Neural Networks

Whilst notions of artificial intelligence have been around since antiquity, the first tractable example can be traced back to the 1950s. This was the invention of the perceptron (Rosenblatt, 1958). It was designed for image recognition[3] by mimicking the function of a neuron in an animal's brain: there are many inputs to a neuron in the brain with varying electrical signals that are summed together and if this amalgamated signal is larger than a threshold then the neuron fires. This translates to finding the vector inner product between an input x and the learnable parameters of the perceptron Θ which is then passed through a Heaviside step function to determine whether the neuron fires (= 1) or not (= 0). Whether or not a neuron fires is assigned a meaning in the problem one wishes to solve, e.g. a value of 1 may indicate dogs whilst a value of 0 indicates cats if the user's desire is to have a perceptron tell the different between dogs and cats. Given that input data is fixed, the deciding factor for whether the perceptron gives 1 or 0 comes down to the values of the learnable parameters. These parameters are changed until the desired result is achieved, with an automated algorithm to do this known as "backpropagation" being regularly used (see Sec. 2.3 for more details). Mathematically, this means that the desired output for the data can be represented as the composition of the linear inner product and the non-linear Heaviside step function:

[3] https://www.youtube.com/watch?v=cNxadbrN_aI

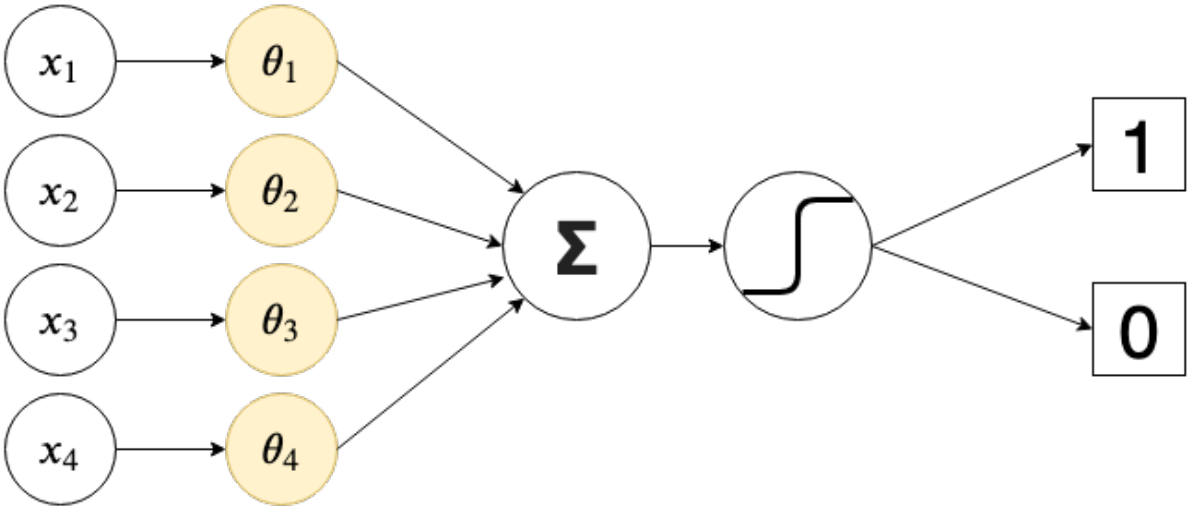

Figure 2.1: A schematic for Rosenblatt's perceptron. The input data $x = \{x_1, x_2, x_3, x_4\}$ are combined with the learnable parameters of the system $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ through the vector inner product Σ before being passed through the Heaviside step function to produce the output: 1 if the dot product is positive and 0 otherwise.

$$y = f(x;\Theta) = H(x \cdot \Theta) = \begin{cases} 1, \text{if } x \cdot \Theta > 0 \\ 0, \text{otherwise} \end{cases}, \tag{2.2}$$

where H is the Heaviside step function. An example of Rosenblatt's perceptron is shown in Fig 2.1. This type of system is referred to as a "feed forward" system because the input data is processed by the perceptron from start to finish and then a solution is given for the data at the end of the perceptron. Feed forward systems are how most modern DNNs work. By construction, Rosenblatt's perceptron can only learn binary classification problems due to the use of the Heaviside function, severely limiting its functionality. However, the vector inner product can be replaced as can the Heaviside function for other linear/non-linear function pairings allowing this perceptron to become more versatile. These modified perceptrons, known as *nodes*, are the starting building block for neural networks.

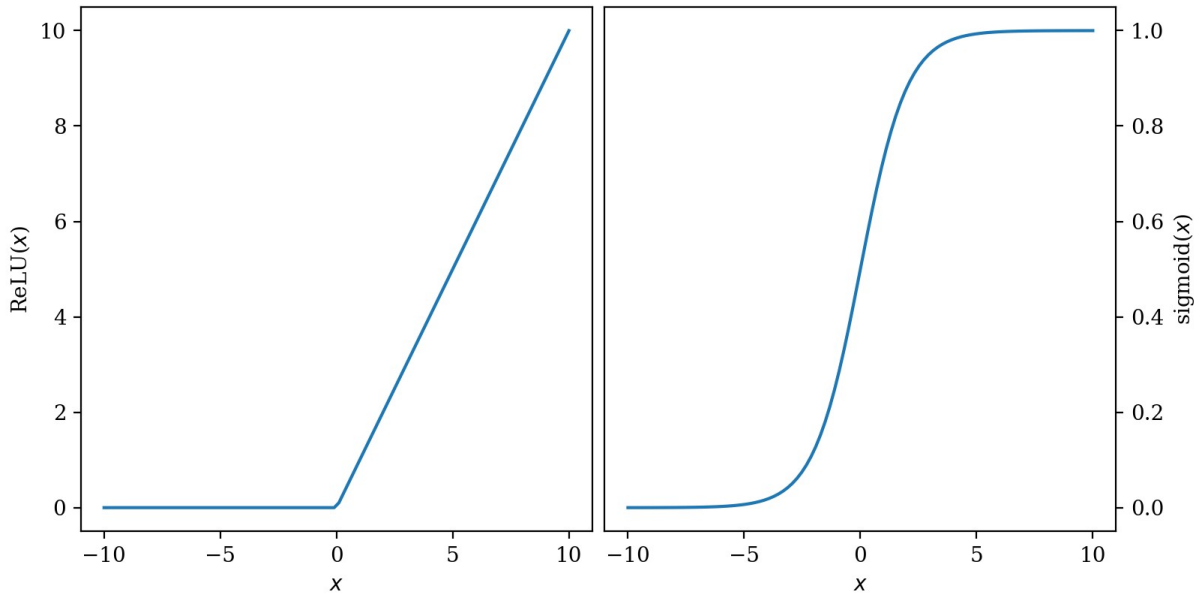

Figure 2.2: The two most common activation functions: the rectified linear unit (ReLU) on the left, which is linear when the input is positive and zero otherwise, and the sigmoid function on the right.

2.1 Nodes: The Generalisation of Rosenblatt's Perceptron

2.1.1 Activation Functions

One of the main limitations of Rosenblatt's perceptron is its inability to learn anything more complex than a binary classification problem. This can be rectified by modifying the non-linear function within the perceptron. That is, changing the Heaviside step function to a different non-linear function. In general, the non-linear function within an artificial neuron (generalised perceptron) setup is known as the *activation function*. While possible to use any differentiable non-linear function as the activation function within an artificial neuron, the Universal Function Approximation Theorem has been proven for sigmoid functions Cybenko (1989):

$$\phi(x) = \frac{1}{1 + e^{-x}}, \tag{2.3}$$

and for rectified linear unit (ReLU) functions (Nair and Hinton, 2010; Lu et al., 2017):

$$\phi(x) = \begin{cases} x, x > 0, \\ 0, \text{otherwise} \end{cases}, \tag{2.4}$$

As such, these are the two most common activation functions to use, with other commonly-used functions being linear combinations of sigmoids and ReLUs (e.g. the hyperbolic tan function is often used as an activation function and this can be expressed in terms of Eq.2.3). The shapes of these functions are shown in Fig. 2.2 which gives an indication of the advantages of using these activation functions beyond their satisfying of the Universal Function Approximation Theorem. The sigmoid activation function's codomain is $(0, 1)$ which makes it very useful for networks interested in calculating probabilities, as this can return an already normalised result. However, the sigmoid activation function is susceptible to the *vanishing gradient problem* (Hochreiter et al., 2001) as follows. When training a system using sigmoid activations, the gradient of the function with respect to the learnable parameters must be calculated as it is used to update the learnable parameters (see Sec. 2.3 for more details). If the output from the sigmoid for is close to either 0 or 1 then the gradients can become miniscule causing a halt in the training of the system.

On the other hand, the ReLU activation function avoids this due to the linear nature of the positive results leading to a constant gradient. Furthermore, the output of the ReLU function is sparse due to any negative values being mapped to zero, leading to an increase in computational efficiency. This increase is one of the main benefits of using the ReLU activation in DNNs. However, this is a double-edged sword. If all of the inputs are negative (or close to zero), then the activation will be zero everywhere. This causes a stagnation when training a model using ReLUs as the gradients become zero. This is known as the *dying ReLU problem* (He et al., 2015b).

The key point here is that all activation functions have strengths and weaknesses. The "correct" activation function to use will depend on two different factors. Firstly, how complex the neural network is. Typically, the sparsity of ReLUs (and ReLU-like functions) is preferred when using DNNs as the computational speed-up is significant compared to using sigmoids. However, sigmoids may be better suited to learning the function, meaning that even in DNNs they should be used as it

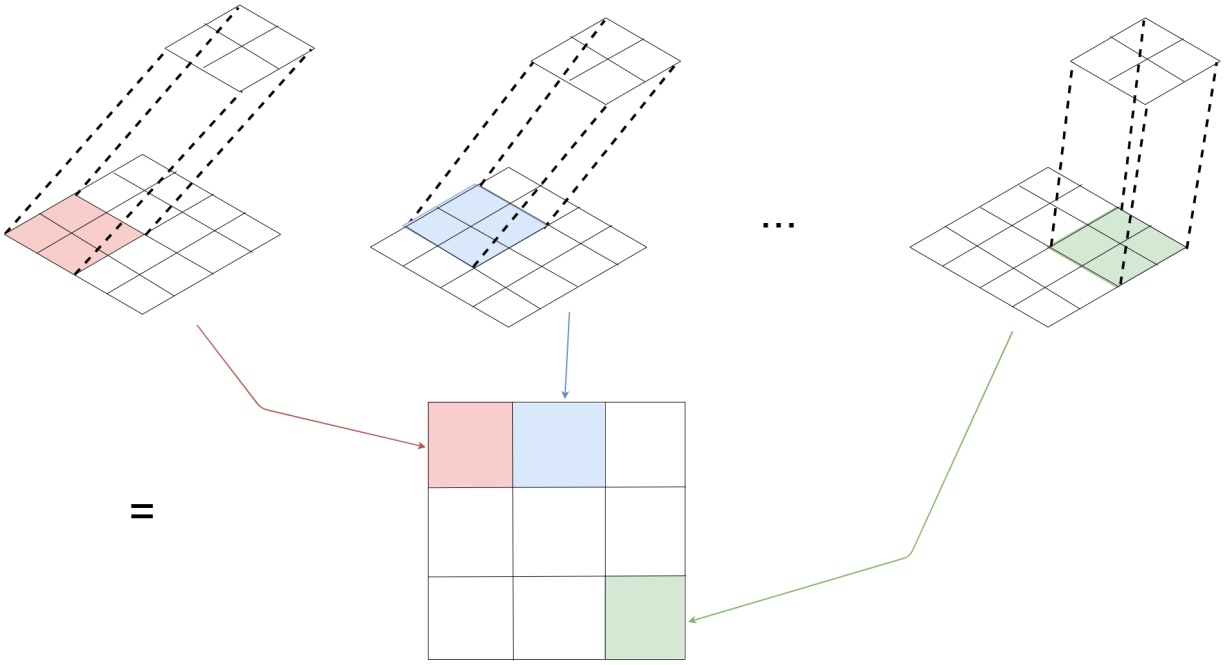

Figure 2.3: Illustration of how the convolution function is used in place of the vector dot product. In this example, the data is a 4×4 grid of values with the learnable parameters (the convolution kernel) being a 2×2 matrix. The output of this convolution operation is the 3×3 grid shown in the bottom row of the figure, known as a *feature map*. The convolutional kernel "slides" along the data with a defined stride of one with the discrete convolution given by Eq. 2.7 calculated at each stopping point. Each entry in the output is the result of these operations.

will learn faster than using ReLUs (nullifying the sparsity speed-up). Secondly, the dataset being used to train. A particular activation function may lead to a DNN learning the mapping from input data to output data more efficiently, however there is no empirical way to determine which activation function will fulfil this role (besides trial and error). In the end, it is possible that the choice of activation function can improve the learning of a DNN but there is no rigorous metric to be exploited to help one choose which activation function is best. Regardless, by the universal function approximation theorem, any activation function should be able to learn the desired function.

2.1.2 Linear Functions

The other part of Rosenblatt's perceptron is the vector dot product which combines the input data and the learnable parameters linearly. For n-dimensional data this

can be written:

$$\zeta(x) = x \cdot \Theta = \sum_{i=1}^{n} x_i \Theta_i. \tag{2.5}$$

The output of Eq. 2.5 is then passed as the input to the activation function giving the output for the system. The downside to using the vector inner product, is that the number of learnable parameters needed in a system scales linearly with the dimensionality of the input data. This can become particularly inefficient when dealing with image and video data as each pixel within a field of view needs to be treated independently and paired with its own learnable parameter. For example, a megapixel image would correspond to around a million inputs requiring the perceptron to have $O(10^6)$ learnable parameters. To counter this, LeCun et al. (1998) used the convolution function as the linear part of their nodes.

$$\zeta(x)(n_1, n_2, \ldots, n_M) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \ldots \sum_{k_M=-\infty}^{\infty} \Theta\left(k_1, k_2, \ldots, k_M\right) x\left(n_1 - k_1, n_2 - k_2, \ldots, n_M - k_M\right), \tag{2.6}$$

which for images can be simplified to the two-dimensional case:

$$\zeta(x)(m, n) = \sum_{h=0}^{H-1} \sum_{w=0}^{W-1} \Theta(h, w) \times x(m - h, n - w), \tag{2.7}$$

where H is the height of the convolutional kernel and W is the width and (n, m) denotes an arbitrary pixel in the image. The learnable parameters, Θ, in Eq. 2.6, 2.7 are an initialised matrix with predefined size known as a convolutional kernel. The output of Eq. 2.7 is known as a *feature map*. The idea is that rather than having a linear relationship between the number of inputs and the number of learnable parameters, each node has a singular convolution kernel that is convolved over the whole input in a sliding window manner (this is known as weight sharing, see Fig. 2.3 for an illustration). The weight sharing drastically reduces the number of learnable parameters per node (e.g. in Fig. 2.3, there are 4 learnable parameters compared to 16 if the vector dot product was being used as the linear function). Also, it incorporates interesting properties beneficial when working with images. Neighbouring pixels in an image are typically strongly correlated, with the correlation dropping as a func-

tion of Euclidean distance from the pixel. This important property is understood by the convolution function and is something that influences the size of the kernel chosen. Also, the convolution function, by construction, deals with shift-invariance, meaning that the relative positions of pixels in an image are not learned but rather the geometry of the features are, i.e. two images could be of the same object with the object in different orientations or magnifications and the feature map will have the same response to both images (Simard et al., 2003). Two important factors when defining a convolution kernel are the kernel size and the stride of the kernel.

The choice of the kernel's size is mostly a personal preference. Increasing the size of the kernel leads to a larger number of learnable parameters, allowing for more flexibility when learning a problem but, ultimately, increasing how long the optimisation process would take. Furthermore, having a larger convolution area could be beneficial depending on the dataset, if the user feels that including more data per convolution will lead to greater insight into the data for the algorithm. Kernel sizes are typically chosen to be square and typically have an odd numbered dimension. The odd numbered dimension makes it easier to pad the input if the user does not want to reduce the dimensions in the output and also has the nice property of centring one of the pixels in the middle of the kernel. N.B. even number dimensional kernels are often used in pooling layers, see Chap. 4.

The stride of the kernel is the size of the "step" the sliding window takes during the convolution. For example, in Fig. 2.3, the stride is equal to 1 as the kernel moves along by one column (and eventually down by one row) for each slide of the convolutional window. In most cases, this stride will be set to one as the user will want to evaluate the convolution centred on every pixel of an image. However, strided convolutions have been popularised as a way to downsample the input data, e.g. if the stride is two then the input dimensions will be halved; see Chap. 5 for more details.

Now that the perceptron has been generalised to being a single node (Fig. 2.4), many nodes can be combined in parallel and in series to create a neural network.

2.2 Building a Neural Network

As it turns out, many of these nodes can be stacked in parallel, creating what is known as a *layer*. Each node in a layer produces an output dependent on independent sets of learnable parameters. This allows for more complex problems to be learned as each node can learn a different feature of the problem. The mapping from

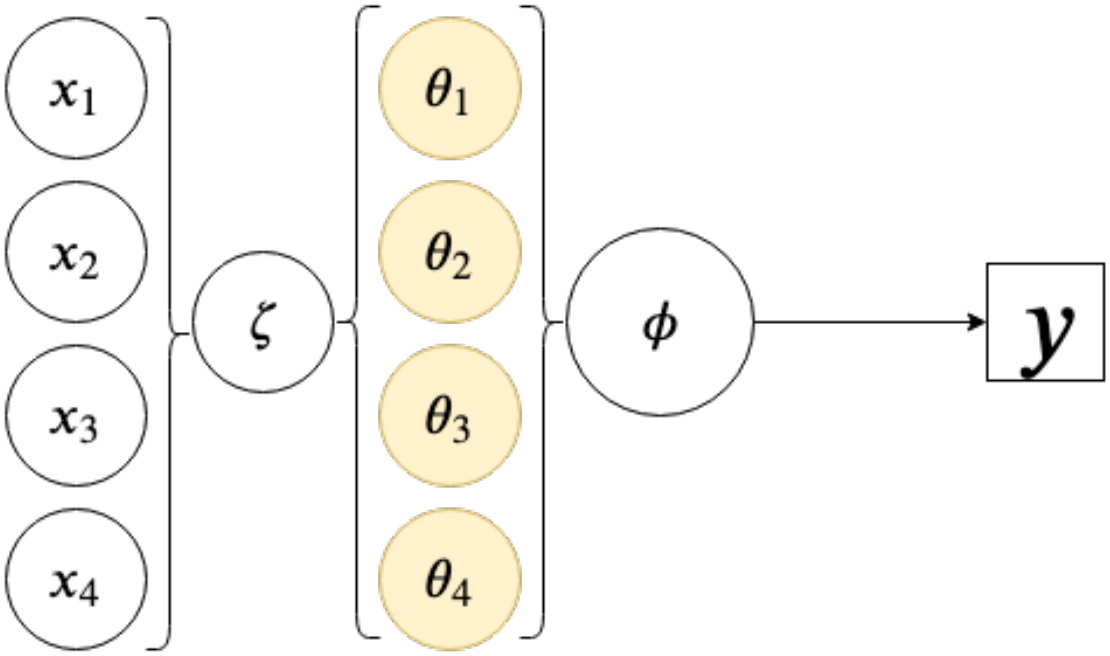

Figure 2.4: The generalised perceptron: a node. The set of inputs $\{x_1, x_2, x_3, x_4\}$ is combined with the set of learnable parameters $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ by a linear function ζ. The result of the function ζ is then passed through the non-linear activation function ϕ giving the output of the node.

input to a single layer to the output is, in fact, a neural network. This is the simplest kind of neural network known as a *shallow* neural network (SNN, see Fig. 2.5).

While it is true that SNNs – whose layer comprises of nodes made up using the linear/non-linear pairings in Sec. 2.1 – can learn any non-linear function (Jones, 1990) via the Universal Function Approximation Theorem, the timescales needed to learn more complex tasks becomes infinitely long. As such, it is more efficient to stack layers in series as each layer will learn a different part of the overall non-linear function (Hornik, 1991). These neural networks with *more than one* hidden layer are known as *deep neural networks*.

The idea behind using multiple layers is that with each successive layer, the representation of the data becomes more abstract. The inputs to one layer are the outputs from the previous layer. Rather than having the activations from each node in a layer combined to give an output, if they are passed as input to the next layer like the input dataset is passed to the first layer, then successive layers will perform operations on already-transformed data. This allows the NN to learn about the transforms of the transform, in a similar way that the second derivative of a function

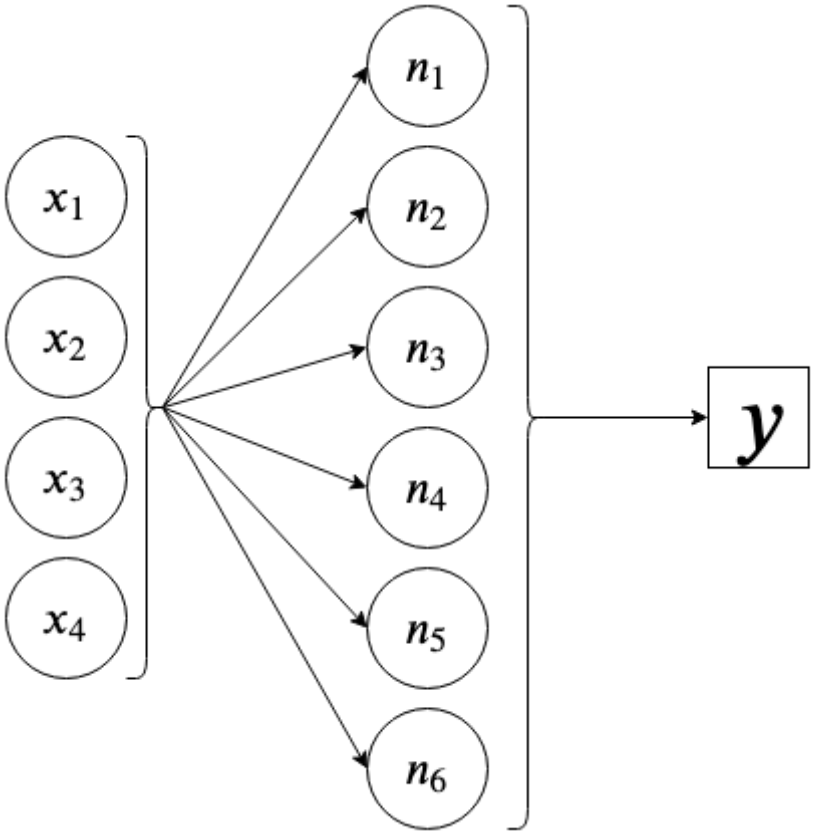

Figure 2.5: A schematic of a shallow neural network (SNN). SNNs consist of a set of inputs $\{x_1, x_2, x_3, x_4\}$ being passed through a layer of nodes $\{n_1, \ldots, n_6\}$ each providing their own set of learnable parameters $\{\Theta_1 = \{\theta_{11}, \theta_{12}, \theta_{13}, \theta_{14}\}, \ldots, \Theta_6 = \{\theta_{61}, \theta_{62}, \theta_{63}, \theta_{64}\}\}$. The outputs of these nodes are then combined to give the output of the SNN y. Note that the final connection transforms the outputs of the hidden layer to the output of the SNN and the result can be either a single number or a vector of outputs depending on the task being learned.

informs us about the nature of the first derivative. In essence, adding more layers allows the network to learn a hierarchical, abstract representation of the data, with earlier layers learning low-level information and later layers learning high-level information.

The number of layers, nodes and what architecture to use vary from problem to problem and are things to be experimented with when applying DNNs. This makes building a DNN difficult with the optimal architecture and training parameters challenging to find. A rigourous study of the combinations of these different architectures and parameters is ideal when applying a DNN to a task[4] (see Feurer

[4]However, in reality, the time may not always be there to do as rigorous a search as one would

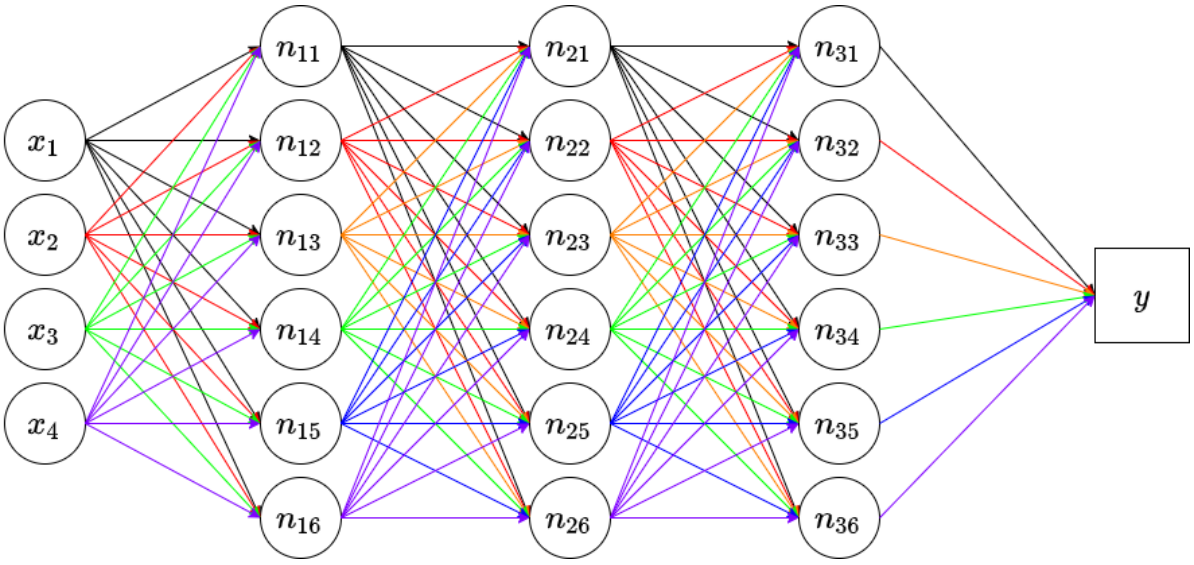

Figure 2.6: A fully-connected neural network. The outputs of the nodes in one layer are used as the inputs to the nodes in the proceeding layer. This is a typical setup that utilises the vector dot product as the linearity in the nodes. Each connection between nodes is a set of learnable parameters in this system.

and Hutter, 2019, and references therein for hyperparameter optmisation methods).

The following sections will discuss the two most commonly-used DNN architectures: fully-connected networks (FCNs) and convolutional neural networks (CNN) to illustrate how they are built and the benefits and drawbacks of each.

2.2.1 Fully-Connected Networks

Fully-connected networks consist of layers of nodes where every node of one layer is connected to every node in the proceeding layer (see Fig. 2.6). Each connection between nodes is a set of learnable parameters following a vector dot product linear function and some activation function. This leads to the number of learnable parameters scaling linearly with the number of inputs as discussed in Sec. 2.1.2. Consequently, FCNs are typically used for input data with lower dimensionality or after downsampling the input data using convolutional layers (see Sec. 2.2.2).

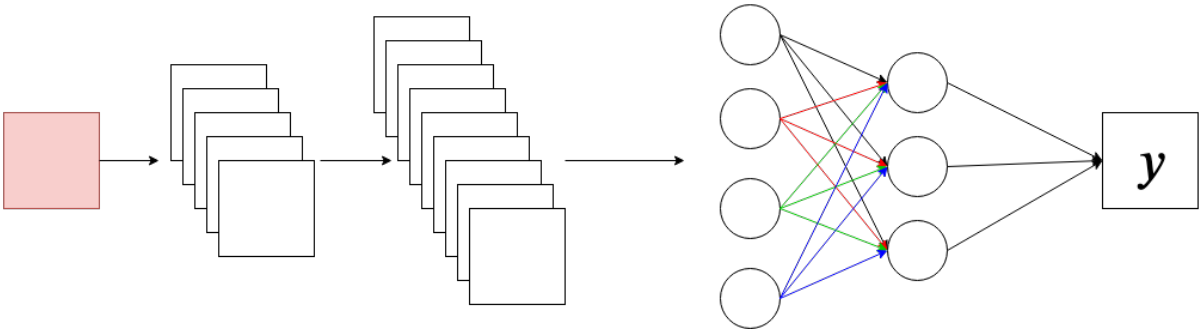

Figure 2.7: An example of a convolutional neural network. The first two layers indicate feature extraction layers with an increasing number of feature maps. These are then passed to a set of fully-connected layers to map to an output. The red square here is the input image to the network.

2.2.2 Convolutional Neural Networks

As mentioned in Sec. 2.1.2, images consist of $O(10^6)$ pixels. Using an FCN in this case would mean that there would be as many inputs as pixels and for the first layer with N nodes there would be $O(10^6 \times N)$ learnable parameters (and so on for any subsequent layers). As a result, using an FCN for image problems is unfeasible. The solution is to look at implementing a convolutional neural network.

CNNs utilise the convolution function as the linear function in its nodes (as described in Sec. 2.1.2) in the first several layers of the network before passing that transformed data to some fully-connected layers to find the output. The idea for CNNs came about by considering the visual cortices of animals. In loose terms, the visual cortex of an animal is a system of interconnected neurons, starting at the eye with an image and ending at the brain with an understanding of what is in the image, and passing a specific electrical signal between the connected neurons depending on the features that each group of neurons identifies. This is achieved in a hierarchical manner meaning that the first groups of neurons identify low-level features (e.g. colour, lighting, gradients) and the later groups identify high-level features (e.g. facial features). Thus the group of neurons within the visual cortex do not detect specific features of objects but rather each group identifies an abstract quantity whose signal will help the subsequent groups pick out other abstract quantities with the final combination telling the brain what the animal sees. The animal then subconsciously teaches its neurons how to react to different objects – it *learns*.

like.

Translating this to CNNs, each layer consists of N convolutional nodes producing N feature maps and aims to extract abstract features within image data to produce the desired output by learning the geometry of objects within the image. A set of N feature maps from one convolutional layer is then used as the input to the next with each new feature map generated by a kernel convolved with all previous feature maps. This, however, comes at the cost of interpretability as the abstract features of the image that the network chooses to learn are chosen by the computer itself. The earlier layers pick out coarse properties of an image with each feature map likely focussing on different features. The output feature maps can then be downsampled in some way for subsequent layers to learn more abstract features. This works as each downsampled pixel will represent more information, leading to more abstract features being extracted deeper in the network. Once features have been extracted by convolutional layers, they are passed through fully-connected layers to produce an output. The fully-connected layers here will be less computationally expensive than an FCN due to the downsampling of the data between layers.

The setup of a CNN is illustrated in Fig. 2.7. The first layer of this CNN takes a monochromatic image (red square) and transforms this into five feature maps. That is, each feature map is calculated via Eq. 2.7 using a different convolutional kernel each time. The five feature maps are then used as the input to the second convolutional layer which transforms the five feature maps into nine feature maps. This is achieved by using the previous five feature maps as an input to each new convolution. This adds an explicit third dimension to the convolution meaning that the kernels are now 3D rather than 2D – the initial convolutional layer is implicitly three-dimensinoal but since the image is monochromatic, the third dimension collapses. The resultant nine feature maps are then flattened into a one-dimensional signal to be passed to the first fully-connected layer. This fully-connected layer transforms and downsamples this 1D representation before a second fully-connected layer maps the resultant data to an output.

2.3 Training a Neural Network

Oftentimes, training is the most important and difficult process in any machine learning algorithm. Training is how a neural network learns what function it is trying to approximate. In practice, this is done through very high dimensional optimisation over the space spanned by the learnable parameters in the system. Initially,

a training dataset is defined with known input and output for the neural network to learn from. To perform the optimisation, a family of methods based on stochastic gradient descent (SGD) with backpropagation can be used – sometimes referred to as optimisers of a neural network. SGD is the numerical method used to update the value of the learnable parameters based on the output of the network in its previous state. In other words, the performance (a measure of how well the network estimates the correct output for a given input) of a DNN is evaluated by a *loss function*, $\mathcal{L}$ – a user-defined metric that measures the similarity of the network-generated output and the desired output. The gradient of the loss function is used to iteratively update the learnable parameters in the system until it converges to an acceptable approximation of the function of interest – this occurs when the loss function reaches a suitable local minimum. Mathematically, gradient descent (GD) can be written as

$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\tilde{y}, y), \tag{2.8}$$

where θ_{t+1} and θ_t refer to the values of the learnable parameter θ at the $(t+1)$-th and t-th iteration. Here, y is the true output and $\tilde{y}$ is the estimate from the DNN. The stochasticity of the optimisation comes in the form of batching the data when training. There is a user-chosen parameter (herein referred to as a *hyperparameter*) known as the batch size which defines how many groups the data is split into for training. An iteration in the system is then the update of the learnable parameters based on the gradient of the loss function *per batch of data*. That is, for Eq. 2.8 to describe SGD and not just GD y, $\tilde{y}$ must represent a batch of true outputs and network-generated outputs. The loss over a batch is then averaged to provide an estimate of the loss on the entire batch. The batch loss then updates the learnable parameters via Eq. 2.8. Iterating through all of the batches of data is referred to as an *epoch* when training a neural network.

The reasoning behind batches and SGD is twofold. Firstly, providing an averaged parameter update of a batch can be beneficial in traversing the space being optimised over. Using the gradient calculated for every single input would result in an oscillatory path throughout the learnable parameter space, as the loss of each input would change the direction of travel slightly. In averaging over a batch, the traversal will be smoother allowing the algorithm to avoid the incorrect local minima. Also, rather than pulling the parameters towards a solution for one input, they are moved generally in the direction of a solution that works the best for most of

the inputs. Secondly, and maybe unintentionally, choosing the correct batch size can speed up the training of the network depending on the hardware being used. Loading more data onto the hardware device at once will allow for fast, parallel computations allowing for fewer I/O operations between storage and device. However, a correct balance between what batch size should be and what it can be needs to be investigated as increasing the batch size too high lowers the stochasticity of the SGD.

The estimate $\tilde{y}$ is a function of the input to the neural network and the learnable parameters of the system and can be written as the composition of the layers of the network:

$$\tilde{y}(x;\Theta) = (\phi_M \circ \zeta_M \circ \cdots \circ \phi_1 \circ \zeta_1)(x), \tag{2.9}$$

where the linear functions encompass the learnable parameters Θ of the system. Due to the construction of the network output, calculating the gradients required for SGD is made simple since the linear and non-linear functions comprising the neural network are differentiable. Updating the learnable parameters at every iteration in this manner is known as *backpropagation* (Rumelhart et al., 1986a,b) and is the pillar of training in deep learning.

Consider an arbitrary learnable parameter in a DNN attributed to node j of layer k with data being fed from node i of the previous layer $k-1$, θ^k_{ij}. The gradient term on the right-hand side of Eq. 2.8 can then be written as $\partial\mathcal{L}/\partial\theta^k_{ij}$ which can be estimated via the chain rule:

$$\frac{\partial\mathcal{L}}{\partial\theta^k_{ij}} = \frac{\partial\mathcal{L}}{\partial o^k_j}\,\frac{\partial o^k_j}{\partial\theta^k_{ij}}, \tag{2.10}$$

where o^k_j is the output of the node j of layer k. o^k_j can be written

$$o^k_j = \sum_{r=0}^{r^{k-1}} \phi_k(\zeta_k(o^{k-1}_r)), \tag{2.11}$$

where r^{k-1} is the total number of nodes in the previous layer $k-1$. That is the output of node j at layer k is the layer applied to all of the outputs from previous layer $k-1$. This means that when calculating the gradient due to learnable parameter θ^k_{ij}, the

second term on the right-hand side of Eq. 2.10 can be written as

$$\frac{\partial o_j^k}{\partial \theta_{ij}^k} = \frac{\partial}{\partial \theta_{ij}^k}\left(\sum_{r=0}^{n^{k-1}} \phi_k(\zeta_k(o_r^{k-1}))\right) = \phi_k(o_i^{k-1}), \tag{2.12}$$

The first term on the right-hand side can also be expanded using the chain rule

$$\frac{\partial \mathcal{L}}{\partial o_j^k} = \sum_{l=1}^{n^{k+1}} \frac{\partial \mathcal{L}}{\partial o_l^{k+1}} \frac{\partial o_l^{k+1}}{\partial o_j^k}, \tag{2.13}$$

where n^{k+1} is the number of nodes in layer $k+1$. Using Eq. 2.11 the second term on the right-hand side of Eq. 2.13 can be written

$$\frac{\partial o_l^{k+1}}{\partial o_j^k} = \theta_{jl}^{k+1} \phi'_{k+1}(o_j^k), \tag{2.14}$$

meaning Eq. 2.13 can be written

$$\frac{\partial \mathcal{L}}{\partial o_j^k} = \phi'_{k+1}(o_j^k) \sum_{l=1}^{n^{k+1}} \theta_{jl}^{k+1} \frac{\partial \mathcal{L}}{\partial o_l^{k+1}}, \tag{2.15}$$

Combining Eqs. 2.12 & 2.15 means Eq. 2.10 can be written

$$\frac{\partial \mathcal{L}}{\partial \theta_{ij}^k} = \phi_k(o_i^{k-1}) \phi'_{k+1}(o_j^k) \sum_{l=1}^{n^{k+1}} \theta_{jl}^{k+1} \frac{\partial \mathcal{L}}{\partial o_l^{k+1}}, \tag{2.16}$$

Now the update of the learnable parameter depends on the output of the previous layer's node, the output of the current node and the gradient of the loss function with respect to the outputs of the next layer. The outputs are known when the data is fed through the network in the former direction leaving only the gradients to be calculated. Since the parameter update depends on the gradients from the next layer, the gradients must be calculated in reverse leading to the inception of backpropagation.

Going back to Eq. 2.8, η is known as the learning rate and is a hyperparameter which determines the rate at which the parameter space is traversed. This can be vital; a learning rate too large can lead to a network that will never converge because

it will continually hop over the troughs of minimal loss that it is looking for. On the other hand, a learning rate too small will result in the network falling into the first local minimum that it encounters and never being able to escape. The idea is to find a medium place between these two extremes and hope that the algorithm can land in a local minimum that is an area of minimal loss. SGD says that the value of the weights at iteration $t + 1$ is then the value at the previous iteration plus some correction depending on how close the model is to convergence.

Typically, convergence is determined when the value of the loss function settles below a certain threshold. This is a good metric for measuring performance in traditional numerical methods where a statistical model is the framework used. However, the goal in machine learning is to produce a data-driven model which interpolates to unseen data. Therefore, the loss function being below a certain threshold on the training data can be useful to the performance of the hyperparameters, but can actually be detrimental to generalisation if it is too low. If the training loss is too low, the model will be over-fitted and will not interpolate to any data outside of the training domain. To combat this a validation set of data is used which is a subset of the training data (typically 10-20%) that the network has never seen before. The values of the loss function for this validation set are not used during backpropagation, but rather as a metric of how well the model generalises i.e. the validation loss does not update the learnable parameters of the network. That is, if the network is overfitting the loss function will increase. As a result, convergence in an ML framework is defined as the minimum of the validation loss function. Furthermore, the number of epochs used for training is an important hyperparameter. If it is too small, the network will not have enough time to learn and generalise, whereas if it is too large, the network can overfit. The typical way of testing for the correct number of epochs without overfitting is using a validation dataset.

2.3.1 Aiding Training Through Initialisation

Like any optimisation problem, initialisation of the learnable parameters in a DNN can be beneficial to its performance. The type of initialisation to be used is entirely dependent on the problem to be learned. Zero initialisation or initial values drawn from a unit normal distribution are commonly implemented in DNN frameworks but do not have any problem-specific reasoning behind their use. Furthermore, the variance of the learnable parameters is not considered in these initialisations which

can lead to exploding or vanishing gradients immediately in a DNN. As such, He et al. (2015b) proposed an initialisation scheme for DNNs which takes into account the architecture of the network. This is known as *He (or Kaiming) initialisation*.

Consider trying to initiliase learnable parameter Θ^k from layer k. Before the activation function in layer k, Θ^k will be combined linearly with input x^k such that

$$y_k = \Theta^k x_k, \tag{2.17}$$

where $x_k = \phi_{k-1}(y_{k-1})$ is the vector of inputs to layer k and y_k is the vector of outputs of the linear functions in layer k. The following assumptions are made about the distributions of the data and the learnable parameters:

1. All learnable parameters are drawn from the same distribution and are statistically independent from one another – similar assumption is made about the input data and the output data.

2. The learnable parameters and the input are statistically independent of one another.

3. The distributions of the learnable parameters and the output of the linear function (Eq. 2.17) have mean zero and are symmetrically distributed about this mean.

Under these assumptions, the variance of Eq. 2.17 can be investigated:

$$\text{Var}\left[y_k\right] = n_l \text{Var}\left[\Theta^k x_k\right], \tag{2.18}$$

where n_l is the total number of nodes in layer k. Using the properties of the product of variances and the assumptions about the distributions, the right-hand side of Eq. 2.18 can be written

$$\text{Var}\left[\Theta^k x_k\right] = \text{Var}\left[\Theta^k\right] \mathbb{E}\left[x_k^2\right]. \tag{2.19}$$

Given that the input to layer k can be written in terms of the output of layer $k-1$, the variance on the output of the layer can be written as a recurrence relation

$$\text{Var}\left[y_k\right] = \frac{n_l}{g^2} \text{Var}\left[\Theta^k\right] \text{Var}\left[y_{k-1}\right] \tag{2.20}$$

where g is known as the *gain* of the activation function. Equation 2.20 can then be applied to the variance on the output of the whole network

$$\text{Var}\left[y_L\right] = \text{Var}\left[x_1\right]\left(\prod_{k=2}^{L}\frac{n_l}{g^2}\text{Var}\left[\Theta^k\right]\right), \tag{2.21}$$

where L is the total number of layers in the DNN, y_L is the output of the network and x_1 is the input to the first layer. The point in the initialisation is to avoid exploding/vanishing gradients from the start which would be the case if the variance of the output of the network is equal to the variance of the input to the network. This puts the condition on the variance of the learnable parameters that $\forall k \in [1,\ldots,L]$

$$\frac{n_l}{g^2}\text{Var}\left[\Theta^k\right] = 1. \tag{2.22}$$

From assumption 3 above, this leads to a good initialisation of learnable parameters being drawn from a normal distribution with mean zero and standard deviation

$$\sigma = g/\sqrt{n_l}, \tag{2.23}$$

i.e. the initialised parameters are drawn from the normal distribution $\mathcal{N}(0,\sigma)$. This result can also be derived from backpropagation arguments as shown in He et al. (2015b). This initialisation led to the first machine learning algorithm that outperformed a human in image classification.

The distribution for which Eq. 2.22 holds can also be varied and other common implentations use a uniform distribution rather than a normal distribution to intialise the parameters with the bounds on the normal distribution given by

$$b = g\sqrt{\frac{3}{n_l}}, \tag{2.24}$$

with the initialised parameters drawn from the distribution $\mathcal{U}(-b,b)$.

The concepts discussed in this chapter will be applied in Chaps. 4, 5 & 6 where

a variety of different deep learning tools will be exploited for the data processing and analysis of solar flare data. The next chapter discusses the telescopes and instruments used to collect the data that is then used for training, validation and application of the deep learning models.

3 | Instrumentation

The following chapter will provide an overview of the instrumentation and detectors whose data is analysed throughout Chaps. 4–6.

3.1 Swedish 1-m Solar Telescope's CRisp Imaging SpectroPolarimeter

The Swedish 1-m Solar Telescope (SST; Scharmer et al., 2003) is a refracting telescope whose primary lens has a diameter of 1m. SST is located in the Observatorio Roque de los Muchachos, La Palma, Spain at an altitude of approximately 2.4km. Built to replace the Swedish Vacuum Solar Telescope (SVST) in the same tower, the SST is also a vacuum telescope meaning it avoids telescopic seeing and artifacts that may appear in data due to dirty mirrors. A diagram of the optical path adapted from Scharmer et al. (2003); de la Cruz Rodríguez et al. (2015) is shown in Fig. 3.1. Light enters the telescopes at the 1m primary lens (PL) before being reflected down the tower by two flat mirrors configured such that the angle of incidence is 45°. The beam is focussed at the bottom of the tower (inset A) where light is reflected by a mirror towards the Schupmann corrector (SC, inset B). The PL produces an image with chromatic aberration which is corrected for by the SC. The beam then undergoes corrections for atmospheric seeing by the adaptive optics system (inset C) consisting of a deformable (DM) and a tip-tilt mirror (TT). The corrected wavefronts are then passed through a reimaging lens (RL) and onto the optical bench.

The optical bench of the SST can consist of up to three instruments: the TRI-Port Polarimetric Echelle-Littrow (TRIPPEL; Kiselman et al., 2011), the CRisp Imaging

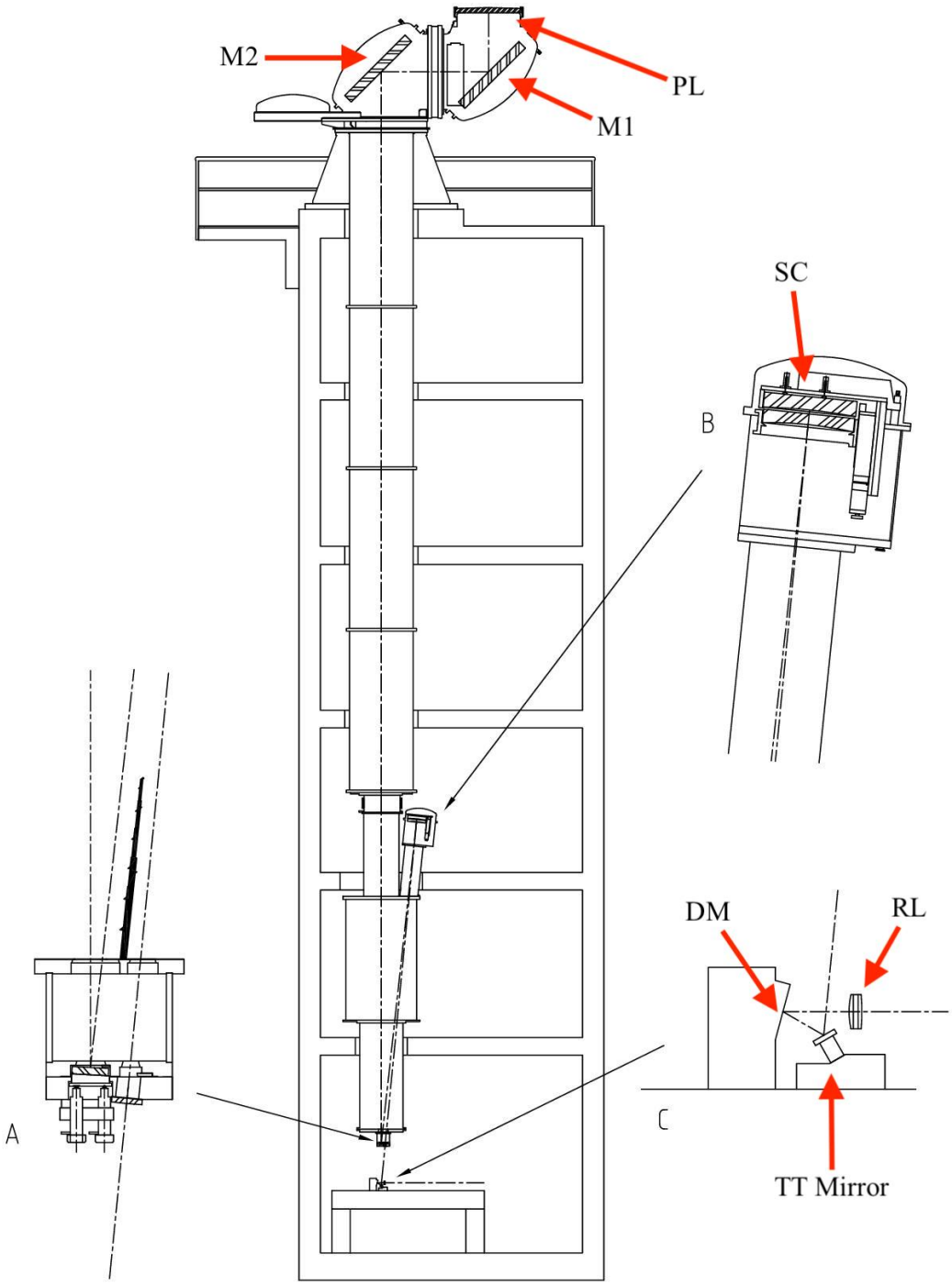

Figure 3.1: Diagram of the optical path of the SST adapted from Scharmer et al. (2003); de la Cruz Rodríguez et al. (2015). Light enters the telescope via the 1m primary lens (PL) before being reflected by two flat mirrors (M1, M2) and sent down the tower. Inset A shows where the light is focussed onto the optical bench. Inset B shows the Schupmann corrector (SC) and inset C shows the adaptive optics system containing the deformable mirror (DM), tip-tilt mirror (TT) and reimaging lens (RL).

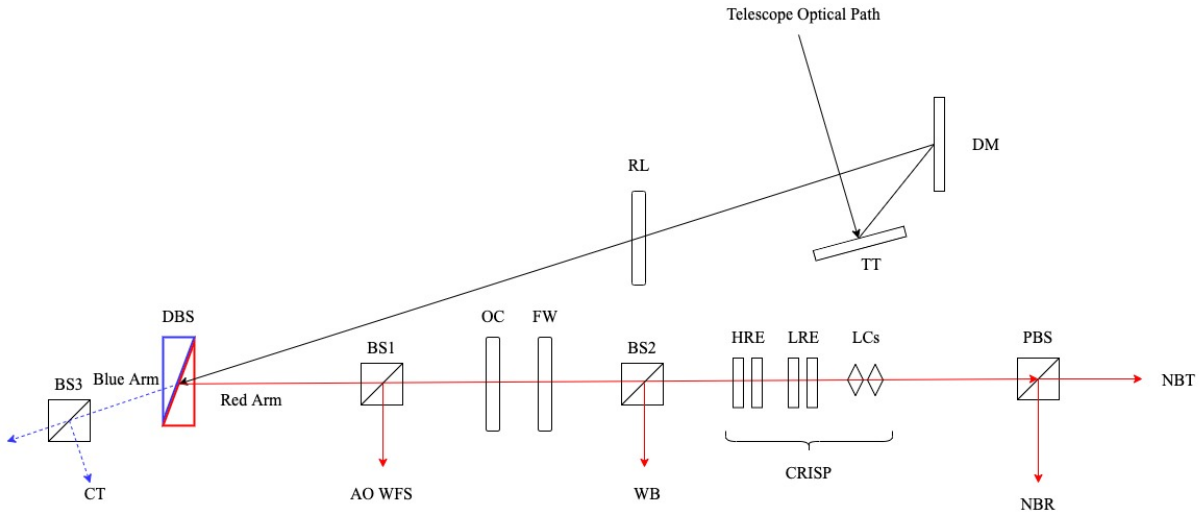

Figure 3.2: Layout of CRISP on SST's optical bench. After being corrected by the adaptive optics system, the light is split into a red and blue component via the dichroic beamsplitter (DBS) where the blue light goes through another beamsplitter (BS3) where the reflected light is used in the correlation tracking (CT) and the transmitted light may be used for CHROMIS if it is in use. The red light encounters a beamsplitter (BS1) where the reflected light is fed to the adaptive optics wavefront sensor (AO WFS) which gives updates to the DM (same meanings as in Fig. 3.1). The remaining light passes through an optical chopper (OC) and a filter wheel (FW) before arriving at a second beamsplitter (BS2). The reflected light from BS2 is imaged by the wideband (WB) camera with the transmitted light fed into the CRISP instrument. The CRISP instrument itself consists of a high resolution etalon (HRE), low resolution etalon (LRE) and liquid crystals (LCs). The HRE & LRE splits the light by wavelength depending on settings from the FW and polarises the light using the LCs. Finally, the light is split by the polarisation beamsplitter (PBS) to record two narrow-band (NB) signals: reflected (R) and transmitted (T) each of which measures an orthogonal polarisation state (e.g. $I+V$, $I-V$).

SpectroPolarimeter (CRISP; Scharmer, 2006; Scharmer et al., 2008) and the CHROMospheric Imaging Spectrometer (CHROMIS; Löfdahl et al., 2021). TRIPPEL is a Littrow spectrograph capable of observing three different spectral regions at the same time with a tunable wavelength range 3800-11000Å. CRISP is a Fabry-Pérot tunable filter (also known as an *etalon*) system working in the red end of the visible spectrum (5100-8600Å) with polarimetric capabilities provided by liquid crystals. Similarly, CHROMIS is a Fabry-Pérot system but working in the blue end of the visible spectrum (3800-5000Å) and does not have polarisation capabilities. The data analysed in this thesis mainly came from CRISP and so the layout of the CRISP instrument will be described in more detail.

The path of light through CRISP's area on the optical bench is shown in Fig. 3.2.

After the light from the telescope passes through RL, a dichroic beamsplitter (DBS) is used to split the light into a blue and a red component where the blue light is transmitted and the red light is reflected. The blue light passes through another beamsplitter (BS3) where the reflected light is passed onto the correlation tracker (CT). This is part of the AO system of the SST and measures the motion of the images in the image plane and sends instructions to the TT mirror to keep the image steady. The transmitted component of the blue light can be used with CHROMIS if it is being used.

The red light goes through a beamsplitter (BS1) where the reflected light encounters the AO wavefront sensor (WFS) which provides the updates needed to the deformable mirror to correct the wavefronts for atmospheric seeing. The transmitted red light then passes through an optical chopper (OC) and filter wheel (FW). The OC is a rotating plate with holes which periodically interrupts the light beam. The FW is then used to select the spectral line the observer wishes to record. After the FW, the remaining light passes through a second beamsplitter (BS2) where this time the reflected light is imaged by a wideband (WB) camera to produce a set of complimentary wideband observations used for alignment and restoration of the narrowband data via the Multi Object Multi Frame Blind Deconvolution (MOMFBD; Van Noort et al., 2005) algorithm. The light transmitted from BS2 then enters the CRISP instrument passing through a high resolution etalon (HRE) and a low resolution etalon (LRE). An etalon is made up of two reflecting optical flats with the light undergoing multiple reflections in the cavity between the two. Each etalon will have a transmission profile corresponding to peaks where the light being reflected interferes with itself. The HRE will produce a transmission profile with multiple narrow peaks which is refined to the wavelength the observer wants to measure by the wider transmission profile of the LRE (see Fig. 3 of de la Cruz Rodríguez et al., 2015). Even with both etalons, there can still be some unwanted transmission peaks which the choice of the wavelength observed in the FW deals with.

After the etalons are the nematic liquid crystals (LCs). These are used to encode the polarisation information into the intensity with four linear combinations of the Stokes parameters. Finally, a polarising beamsplitter (PBS) is used to split the light into two orthogonally polarised light beams which are imaged on two synchronised CCDs: one for the narrowband reflected (NBR) light and one for the narrowband transmitted (NBT) light. The separation between the flats in the etalons is changed to alter the wavelength being observed. This is used to step through the spectral line

an observer is looking at and image in many different wavelengths. One drawback to observing using FPIs is that the different points along the spectrum cannot be sampled at the same time which puts a constraint on the number of wavelength points that can be sampled – it is recommended that the observation time be less than the solar evolution time for the feature being observed to avoid smearing effects introduced by the evolution of the feature. For flares particularly, this can pose an issue due to the subsecond evolution of the flaring atmosphere.

3.2 Hinode's Solar Optical Telescope

Hinode (Kosugi et al., 2007), formerly known as Solar-B, is a spaced-based solar observatory in a Sun-synchronous orbit (SSO) launched on 22nd September 2006 from Uchinoura Space Centre, Japan. It was originally planned to run for three years but as of writing, most of the instruments on-board are still taking data. Hinode was launched with three instruments: the Solar Optical Telescope (SOT; Tsuneta et al., 2008), the X-ray Telescope (XRT; Golub et al., 2007) and the Extreme ultraviolet Imaging Spectrometer (EIS; Culhane et al., 2007). SOT is a 50cm Gregorian telescope used for imaging and spectropolarimetry across the visible spectrum to study the solar photosphere and chromosphere, XRT uses grazing incident optics to capture images of the whole Sun measured in different X-ray filters and EIS is an imaging spectrometer able to capture two EUV spectral bands to study the corona. The data used from Hinode in Chap. 4 of this thesis is entirely from SOT therefore the rest of the focus of this section will be on how SOT deals with the light it collects.

Figure 3.3 shows a detailed diagram (adapted from Fig. 5 in Tsuneta et al., 2008) of the optical path of Hinode/SOT. Light collected by the telescope is passed to a beam splitter via a tip tilt mirror with one of four possible destinations. A small amount of light is passed to the correlation tracker which keeps the images steady in the field of view. The three main science destinations are then the slit spectropolarimeter (SP), broadband filter imager (BFI) and narrowband filter image (NFI). The SP observes the neutral iron (Fe I) line doublet at 6302Å taking measurements of the four Stokes parameters of these lines. The BFI images several spectral windows of interest in SOT's highest spatial resolution (0.0541″ px^{-1}): spectral lines from neutral cyanogen (CN I) 3383.5Å, Ca II H 3968.5Å and neutral methylidyne (CH I) 4305.0Å, and continua from the blue (4504.5Å), green (5550.5Å) and red (6684.0Å) parts of the visible spectrum. The NFI collects data in the green and red parts of

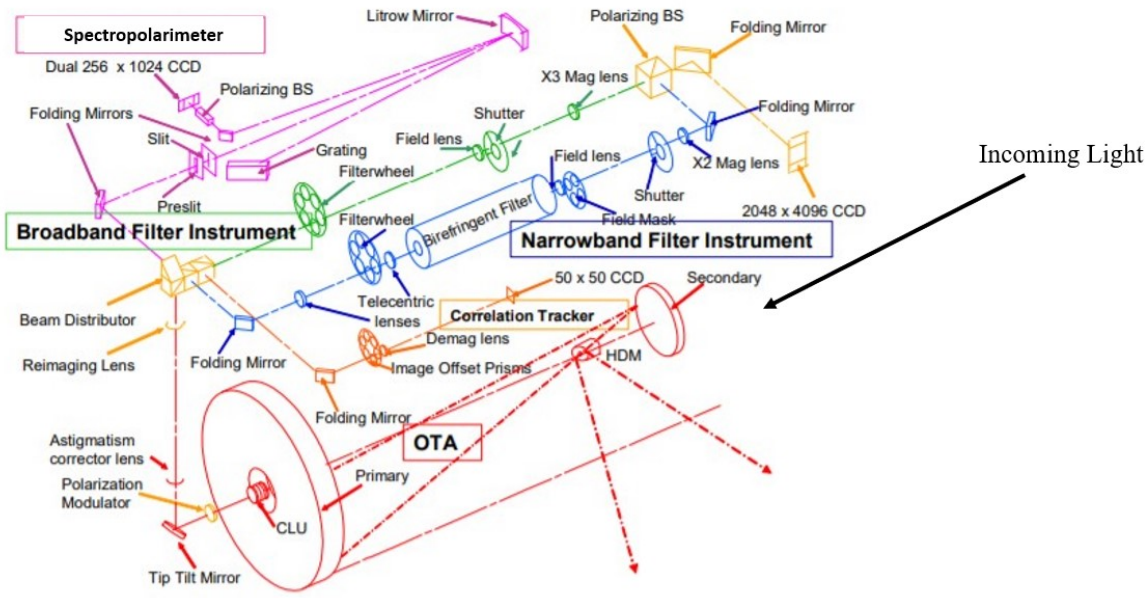

Figure 3.3: Diagram of light path in Hinode/SOT adapted from Tsuneta et al. (2008). The incoming light indicated in the diagram is incident on the primary mirror before reflection on the secondary mirror passes it to the instruments. The light is then split between four paths: the correlation tracker which keeps the image steady; the slit spectropolarimeter (SP) which observes the Fe I λ6302Å doublet and records the four Stokes parameters over these lines; the broadband filter imager (BFI) used to observe continua and some spectral lines in the blue end of the spectrum in the solar atmosphere; and the narrowband filer image (NFI) used to observe spectral lines in the green and red part of the visible spectrum.

the visible spectrum with a high spatial resolution of 0.08″ px^{-1}. The NFI has four different imaging modes:

1. Filtergram: a high resolution image taken in one of the observable spectral windows.

2. Dopplergram: an image of the Doppler shift of a spectral line by taking several filtergrams centred on different wavelengths and subtracting them from the image centred on the rest wavelength of the line.

3. Longitudinal magnetogram: The ratio of the Stokes V/I images.

4. Stokes I, Q, U, V: high resolution images of the four Stokes parameters[1].

[1]Up to the four Stokes parameters can be selected in these modes.

The imaging of the NFI is taken through the Lyot (birefringence) filter and can observe in passbands focussed on the following spectral lines and centred on the given wavelengths: neutral magnesium b (Mg I b) 5172.0Å, Fe I triplet 5250.0Å, Fe I 5576.0Å, neutral sodium D (Na I D) 5896.0Å, Fe I doublet 6302.0Å and Hα 6563.0Å. The data used in Chap. 4 from Hinode/SOT is Hα filtergrams from the NFI.

3.3 Solar Dynamics Observatory's Atmospheric Imaging Assembly

The Solar Dynamics Observatory (SDO; Pesnell et al., 2012) is a spaced-based solar observatory launched on 11th February 2010 from Cape Canaveral Air Force Station in Brevard County, Florida, USA. SDO is in a geosynchronous orbit chosen to keep constant communication with the ground station receiving the data. It is currently scheduled to take data until the year 2030 and has three main instruments on-board (two are still fully operational). The three instruments on SDO are the Heliospheric and Magnetic Imager (HMI; Scherrer et al., 2012), Extreme ultraviolet Variability Experiment (EVE; Woods et al., 2012) and the Atmospheric Imaging Assembly (AIA; Lemen et al., 2012). HMI focusses on observations of the Fe I 6173Å spectral line and its polarisation, using this information to produce intensity images and longitudinal and vector magnetograms. EVE measures the irradiance from the Sun across EUV wavelengths treating the Sun as a single spatial point in the sky. AIA consists of four 20cm normal-incidence telescopes each sensitive to two of the eight wavelength ranges AIA observes[2] The passbands observed by AIA span the visible, UV and EUV spectral ranges with the seven EUV channels covering spectral lines sensitive to tempeatures from 50kK-20MK: continuum (4500, 1700 & 1600Å), Fe XVI 335Å, He II 304Å, Fe XIV 211Å, Fe XII,XXIV 193Å, Fe IX 171Å, Fe XIII,XXI 131Å and Fe XVIII 94Å.

The layout of the AIA telescopes along with which filters each telescope observes is shown in Fig. 3.4. Telescope 1 images in 335Å and 131Å, telescope 2 in 211Å and 193Å, telescope 3 in 171Å and the UV continua and telescope 4 in 304Å and 94Å. Telescopes 1, 3 and 4 have filter wheels to select which passband to image while telescope 2 makes use of an aperture blade. A mechanical shutter is used

[2]There are technically 10 passbands AIA observes but the 1600, 1700 & 4500Å continua are all imaged by the same filter.

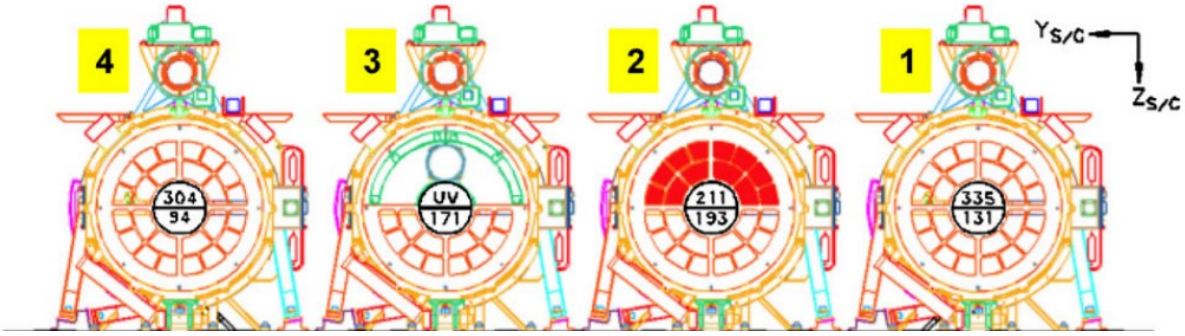

Figure 3.4: The setup of the four AIA telescopes on board and the filters each one observes adapted from Fig. 2 of Lemen et al. (2012). Telescope number 1 (right) observes the 335Å and 131Å filters, with telescope number 2 (middle right) observing the 211Å and 193Å passbands, telescope number 3 (middle left) observing 171Å and the UV continua and telescope number 4 (left) observing 304Å and 94Å.

to regulate the exposure time. A full set of AIA exposures takes approximately 12 seconds to acquire. This full set contains an image in each of the EUV passbands and an image in one of the three optical/UV bands. That is the UV band used, 1600 or 1700Å, is alternated every 12 seconds unless the timestamp falls on an integer minute wherein the continuum exposure is of the optical continuum at 4500Å. AIA images with a spatial resolution of 1.2″and a field of view size of 41′. AIA data imaged in 304, 1600 & 1700Å is used in the adversarial testing in Chap. 4.

4 | Classification of Solar Images Using Convolutional Neural Networks

The following chapter is based on the work in Armstrong and Fletcher (2019) with Sec. 4.1 following the same logic as the introduction of Armstrong and Fletcher (2019) but updated for values in 2022. Section 4.2 takes a more in-depth look at how the CNN known as Slic was constructed and how the training data was put together. Sections 4.3 & 4.4 cover the training and validation of Slic in more depth than in the paper while Sec. 4.5 covers adversarial examples described in the paper as well as new examples not covered in the scope of the paper.

4.1 Exponential Growth in Solar Physics Data

With each new solar physics mission/telescope, instruments are improving in spatial, temporal and/or wavelength resolution. Increased resolution in any of these three categories equals greater volumes of data. This has led to an exponential increase in the amount of data acquired in the past decade, from < 10 TB per year from Hinode/Solar Optical Telescope (SOT) in 2006 (Tsuneta et al., 2008) to 500 TB per year from the Solar Dynamics Observatory (SDO) in 2012 (Pesnell et al., 2012) to 10 000 TB per year expected from the Daniel K. Inouye Solar Telescope (DKIST) which saw first light in 2021 (Elmore et al., 2014). On top of this, the Hinode and SDO data is all archived totaling 4.1 PB (petabytes, 1PB = 1000 TB) of data which

will only keep growing with each passing year.

This is a huge amount of data, and sorting through it is not a task which can be given to humans. For an efficient alternative, automation of data preparation and sorting must be explored with machine learning. This is the kind of automation that can save data analysts time and effort when acquiring and traversing their data and in the age of data-intensive solar physics these techniques can prove invaluable. Motivated by this, an efficient machine learning algorithm for the classification of solar images is proposed: a convolutional neural network (CNN). This is designed to learn the different geometry of large-scale features on the Sun such that, after the model has been trained, a dataset of solar images can be passed to the network and it will identify which images contain which relevant feature in a very short time.

The proposed algorithm will allow the user to easily identify the images of most importance to the study they are carrying out. Furthermore, having a pre-trained CNN that understands the geometry of solar features can be very beneficial for "transfer learning". Transfer learning is when a previously trained neural network is used for initialisation and/or training for a new network which aims to learn a different but related task. This can be beneficial during training a new network as the old network teaches the new network what it knows about the physical system it has learned about and can steer the optimiser towards a better solution[1]. This particular trained DNN is used in transfer learning in Chap. 5 to help the DNN proposed there to correct for atmospheric seeing in ground-based solar flare observations.

4.2 Constructing a Convolutional Neural Network and Training Set for Solar Image Classification

The work in this thesis is mostly concerned with optical wavelengths and the use of the classifier in this chapter will focus on images within this range. The CNN is trained using images from the Hinode/SOT instrument taken by the Hα filter. The images are sorted into 5 classes: filaments, flare ribbons, prominences, sunspots and the quiet Sun (i.e. lack of any of the other four features). While filaments and prominences are the same physical feature (dense, cool plasma that runs parallel to a magnetic neutral line and is suspended in the atmosphere by a coronal magnetic

[1]Note that the converse is also true: if a new network is taught by an existing network that knows about a system not beneficial to what is trying to be learned, the training of the new network can take longer.

field; Labrosse et al., 2010), just in different locations (prominences off-limb; filaments on- disk), their geometries in the Hinode/SOT Hα images are vastly different leading to the split in classification. This split can easily be consolidated when using the network by asking for images with both filaments and prominences. The Hα flare ribbons are intense brightenings in the solar atmosphere which are interpreted as the base of the coronal magnetic field structures to which flare energisation is attributed. The images of sunspots either contain one or multiple sunspots such that our network learns what a singular sunspot looks like but can still understand if there is a group. Thus, the network learns the geometry of these features when observed at this wavelength. One of the goals is to see if the computer perceptually understands what these features are. That is, if it can identify the same features correctly when they are imaged in different (UV and extreme ultraviolet) wavelengths e.g. sunspots observed in 1600/1700Å and prominences observed in 304Å.

The training set itself is a catalogue of 13175 Hα images from SOT's NFI that were classified by hand into one of the five classes[2]. This dataset was constructed to be as diverse as possible for each class i.e. the sunspot class contains images with single sunspots, multiple sunspots, different shapes and sizes of sunspots; the flare ribbon class contains flares with two ribbons, flares with different ribbon geometry and confined flares; the filament class consists of filaments of different geometries taken at different viewing angles; the prominence class features prominence images taken on both limbs of the Sun at different positions; and, finally, the quiet Sun class features images taken from a variety of viewing angles from disk centre out to the limb so that the trained CNN will not get confused by the limb brightening observed in Hα. All images in the training and validation are resized to 256×256 with anti-aliasing using the `resize` function from scikit-image's `skimage.transform` module (van der Walt et al., 2014).

The neural network architecture used is a 13 layer CNN inspired by the VGG networks[3] (Simonyan and Zisserman, 2014) and is shown in Fig. 4.2. The network seeks to model the function that maps an image of the Sun in Hα to a vector of probabilities of the images containing a specific feature. Therefore, the input of the network will be the pixel intensities of the image and the output will be a vector of class probabilities with each element corresponding to the probability of a feature.

[2]The Hinode/SOT data is available from http://sdc.uio.no/sdc/

[3]Named after the Visual Geometry Group at the University of Oxford who developed these architectures.

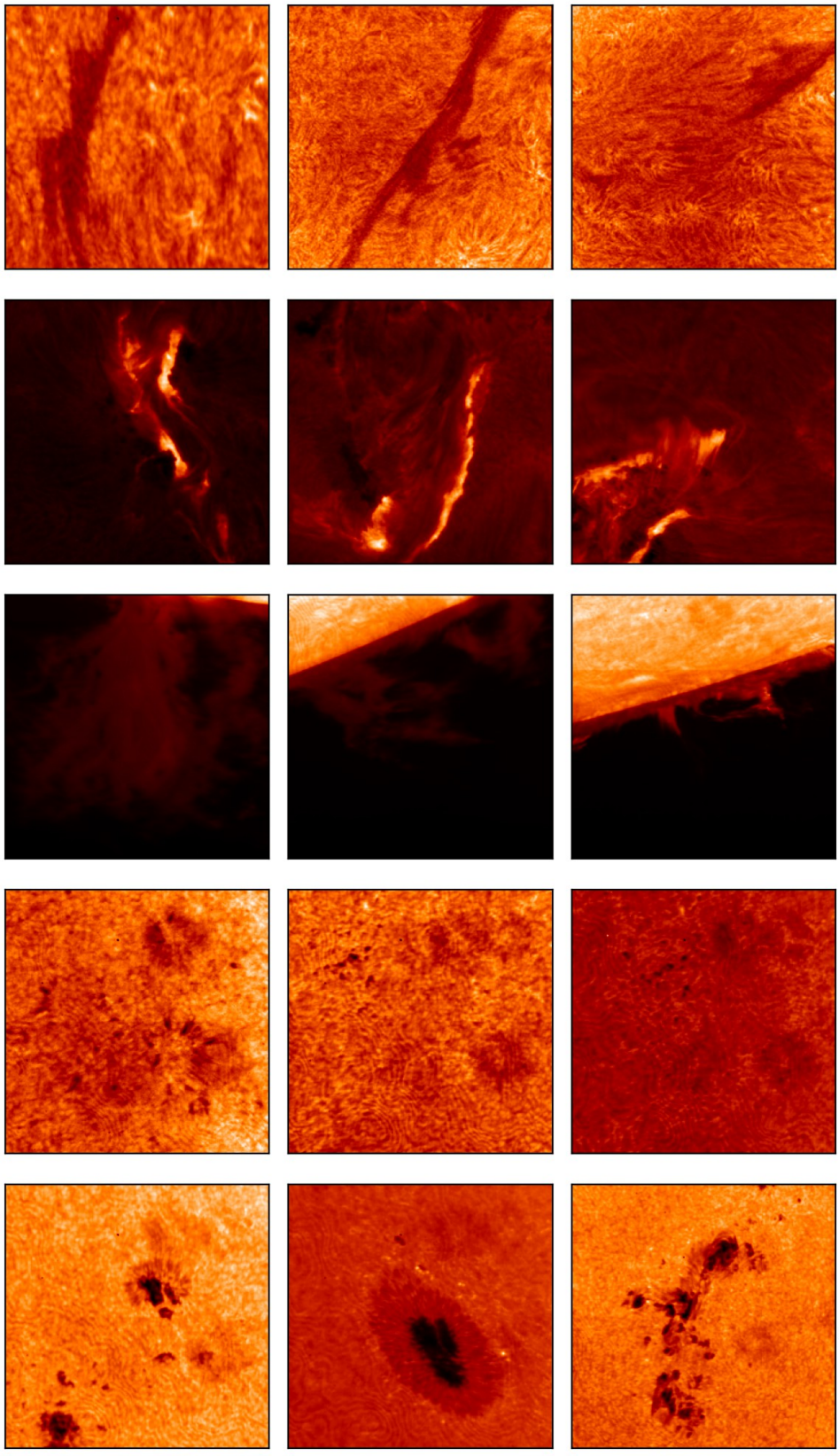

Figure 4.1: Examples of images used for each class during the training of the CNN. Each column shows three different examples of each class with (going from left to right) the first column showing filaments, the second flare ribbons, the third prominences, the fourth quiet Sun and the fifth sunspots. These examples were picked to show the diversity in the training set.

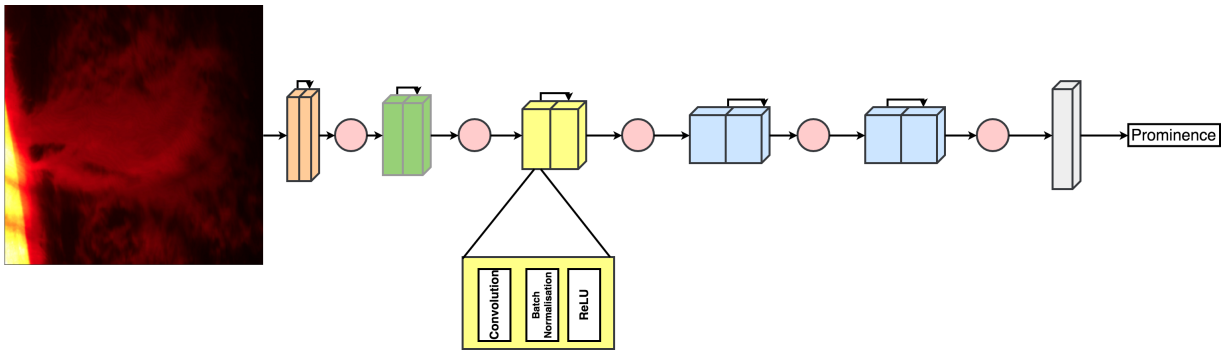

Figure 4.2: The setup of the 13 layer CNN inspired by VGG networks (Simonyan and Zisserman, 2014) where the arrows between each block indicate the flow of data in the feed-forward process. The blocks are colour-coded to reflect their purpose. Orange, green, yellow and blue are all convolutional layers which have 64, 128, 256 and 512 trainable feature maps, respectively. The inside of one of the convolutional layers is shown which is the same for all convolutional layers – the data undergoes a convolution followed by batch normalisation followed by the activation via a ReLU function. The red circles correspond to the max pooling layers. The grey block corresponds to the classifier at the end of the network. The example here is of a prominence in Hα from Hinode/SOT being classified correctly.

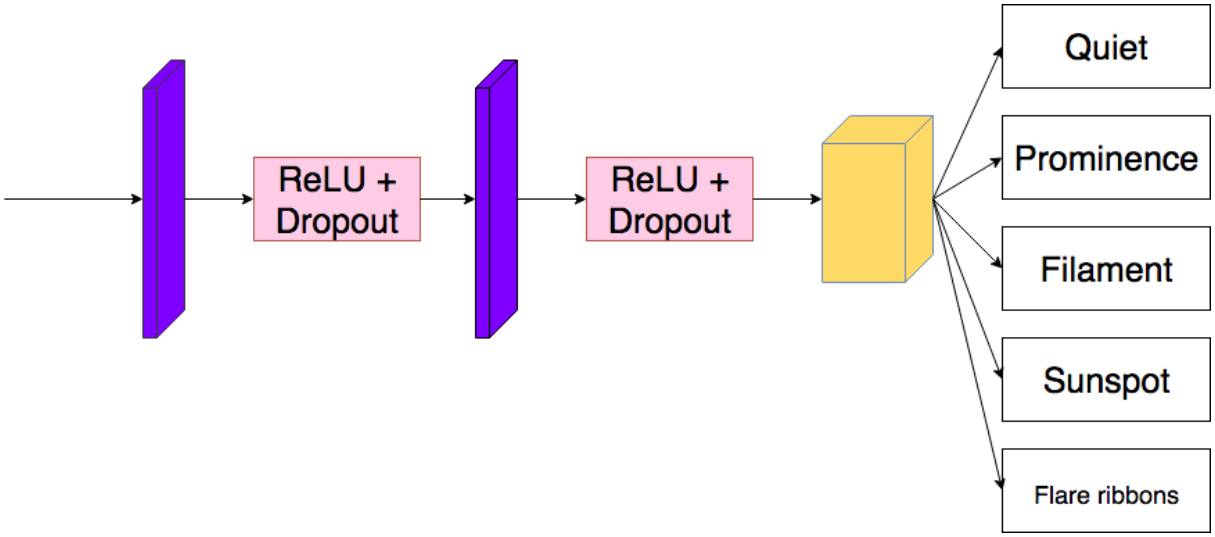

Figure 4.3: The classifier mini-network. The 3D blocks represent fully-connected layers which map the output feature maps from the last maxpooling layer to the class labels with a certain probability. The pink boxes refer to rectified linear unit (ReLU) activation followed by dropout regularisation. The input dimension to the classifier mini-network is 512×8×8 = 32768, with the output dimension being 4096.

If the network learns the features correctly then the highest probability (i.e. the maximum value element of this vector) will correspond to the correct class for the image.

The layers shown as cuboids in groups of two in Fig. 4.2 between the input and the output are convolutional layers as discussed in Sec. 2.1. The convolution kernels in each of these layers is composed of 3×3 pixels initialised by He initialisation described in Sec. 2.3.1. The number of feature maps in each of these groups of layers increases towards the output of the network as the model is detecting more and more complex features and a larger number of convolutions to look at will help to distinguish between these features. In the first group of convolutional layers 64 feature maps are used which is doubled at each new group of two until the fourth where the fourth and fifth groups of two use 512 feature maps. The insides of these layers is shown also in Fig. 4.2 where the eagle-eyed may notice that there is something extra compared to the convolutional layers discussed in Sec. 2.1: *batch normalisation*.

Batch normalisation (Ioffe and Szegedy, 2015) is applied to the output from the convolution operation. This is a technique used to increase the stability of our network and normalises the output of the convolution calculation around a batch mean (β) and standard deviation (γ) via the equation

$$y = \gamma \times \frac{x - E[x]}{\sqrt{\sigma + \epsilon}} + \beta, \tag{4.1}$$

where x is the output feature maps and y is the batch normalised feature maps, ϵ is a small positive constant used to stop the denominator going to zero. σ is the sample standard deviation and $E[x]$ is the sample mean of the feature maps being normalised. Both of these depend on the batch size hyperparameter introduced in Sec. 2.3. This is beneficial as it reduces the dynamic range of the data at the cost of two extra trainable parameters (β, γ) and speeds up training sufficiently (if the batch size is large enough). That is, it helps with what is known as the internal covariate shift (ICS) of a DNN. The ICS refers to the change in a layer's input distribution when a preceding layer has been updated while training. Updating the learnable parameters in a DNN layer will lead to changes in activations within nodes and layers which will change what the input to a following layer looks like to the network. This causes the network to try to adapt to this new form of the input which can prolong training. The role of batch normalisation is to mitigate this by stabilising the input distributions to layers based on a learned mean and standard

deviation, i.e. the feature maps are normalised by a learned mean and standard deviation (β, γ) such that the input distribution to the next layer appears to be the same as the previous epoch despite the learnable parameters in the system being updated. This is achieved by (β, γ) themselves being additional learnable parameters. This will speed up training by reducing the ICS in a network. Moreover, batch normalisation reduces the dependence of the gradients calculated during optimisation on the scale of the inputs and their initial values. This means that higher learning rates can be used without the risk of divergence (which is beneficial in learning rate scheduler methods, see Sec. 5.5). Equation 4.1 can then easily be manipulated during backpropagation to return x such that the true feature maps can be recovered from the batch normalised feature maps. The activation function used in each of the convolutional layers is the ReLU (see Sec. 2.1.1).

The red circles in Fig. 4.2 represent *maxpooling* layers used to downsample the data by a factor of 2 in their spatial dimensions. This downsampling works by parsing the image into segments of four pixels (2×2 grids) and taking the maximum of those pixels. This means that one pixel in a downsampled image is representative of the four pixel block it came from. This is, in a sense, how the network learns more complex features – as the resolution of the input is decreased, each pixel represents more information from the original input and thus each operation is performed on a larger fraction of the original image (e.g. four pixels rather than one) which will highlight more complex, larger features via the convolution operation. Other types of pooling exist, such as average pooling (taking the average of the group of pixels being downsampling), but maxpooling is the prevailing method in this case due to its benefits for reducing over-fitting since the same pixel out of the four may not be the maximum after every learnable parameter update. Furthermore, downsampling of data in a network is typically tied to increasing the number of feature maps, since decreasing the dimensions of the feature maps as the number of them is increased is computationally efficient.

The grey cuboid at the end of the network in Fig. 4.2 is the classifier of our network: after the features within the images are identified by the convolutional layers, they are passed to the classifier which decides what class to assign to the images. This can be described by a mini-network shown in Fig. 4.3. The 3D blocks in Fig. 4.3 represent fully-connected layers to classify the final set of 512 feature maps from the last convolutional layer. The output of the first and second fully-connected layers go through a ReLU activation and a regularisation technique known as *dropout*

(Srivastava et al., 2014). This assigns a probability, p, to each input node in a layer such that for each training epoch there is a probability that the network will ignore that node and connection and thus train on an approximate model. Training on a set of approximate models and then averaging them at validation time works well as a regularisation technique – i.e. helps reduce over-fitting – whilst still preserving (and actually improving, in many cases) results as shown in Srivastava et al. (2014). In the CNN, $p = 0.5$ (i.e. 50% chance of the connection being dropped). The third fully-connected layer (in gold in Fig. 4.3) determines which class each image should be assigned. Normally, there would need to be a final activation function here but the class labels are inferred in this CNN via the choice of loss function which implicitly adds this final activation layer (see Sec. 4.3).

4.3 Training the Convolutional Neural Network

The loss function chosen to minimise for this network is the standard for image classification tasks: the *cross-entropy loss* (CEL). In information theory, the entropy, H, of a random variable x is the uncertainty in the random variable's outcome from its probability distribution, p,

$$H(x) = -\sum_{x_i} p(x_i) \log p(x_i), \tag{4.2}$$

where $\{x_i\}$ are all possible values of x. That is, a random variable $x \sim p$ can take one of the values $\{x_i\}$, if the probability distribution p is skewed towards any values x can take then the value of Eq. 4.2 will be lower (less entropy, more certainty in values x can take) whereas if p is uniform then the value of Eq. 4.2 will be higher (more entropy, less certainty in values x can take). The concept comes from information theory describing the amount of information given by certain events. Events with a high probability are less surprising than those with low probability meaning there is less entropy in the system and thus less information is needed to describe these events. Extending this idea, *cross entropy* can be introduced to give a measure between two probability distributions p and q both characterised by random variable x

$$H(p, q) = -\sum_{x_i} p(x_i) \log q(x_i). \tag{4.3}$$

The value of $H(p,q)$ will then be small for two similar probability distributions. That is, when the distributions p and q are similar there is less entropy in the system. It is Eq. 4.3 that gives way to the CEL. The output of the classifier CNN is the vector of probabilities of an input being of a certain class, $\tilde{y} = q(x_i;\Theta)$ and the ground truth is a similar probability vector with a 1 in the entry corresponding to its class and 0s elsewhere, $y = p(x_i)$. This means that by minimising the cross entropy between these two probability vectors, the network will learn how to classify images correctly – this is because when the two distributions are dissimilar, the cross entropy value will be high communicating to the network that there is a lot of entropy in the system. That is, the CEL can be written

$$\mathcal{L} = -\sum_{i}^{C} p(x_i)\log(q(x_i;\Theta)), \tag{4.4}$$

where C is the number of classes in the classification problem and the estimate from the network depends on the learnable parameters of the system Θ. The probability distribution q is modelled as a softmax function

$$q(x_i) = \frac{\exp(x_i)}{\sum_k \exp(x_k)}. \tag{4.5}$$

This softmax function is the last non-linear activation after the final fully-connected layer. Due to the truncating nature of the exponential function, if the network correctly thinks an image is a certain class then $q \to 1$ giving a low value of Eq. 4.4[4].

In the training of this CNN, a variant of vanilla SGD is used known as SGD *with Nesterov momentum* (Sutskever et al., 2013). Rather than updating a learnable parameter θ by Eq. 2.8, Nesterov momentum updates it by the following

$$\theta_{t+1} = \theta_t + v_{t+1}, \tag{4.6}$$

where v_{t+1} is referred to as the velocity term and

$$v_{t+1} = \mu v_t - \eta\nabla\mathcal{L}(\theta_t + \mu v_t), \tag{4.7}$$

[4]The softmax function is the multinomial generalisation of the logistic function used in binary logistic regression. Thus the use of the softmax distribution and CEL here characterises the classifier as a non-linear variant of multinomial logistic regression.

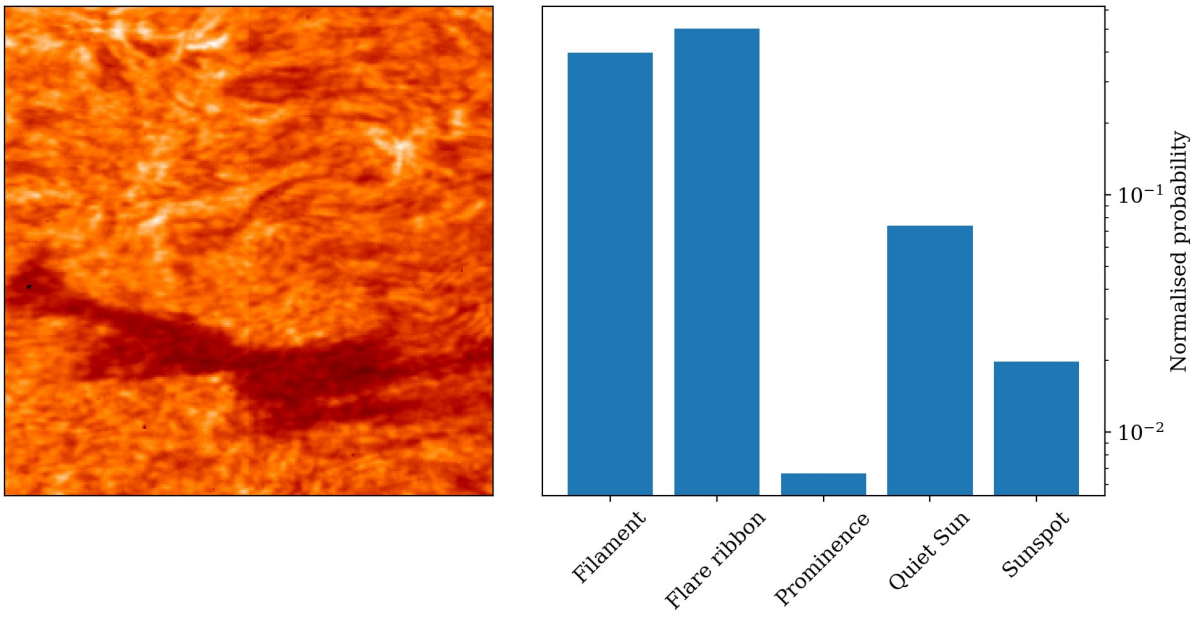

Figure 4.4: The single misclassified case from our validation set. The network identifies the image as containing flare ribbons likely due to the brightenings at the top of the image. The image was truly classified by eye as containing a filament. However by inspection of the probability distribution for the classes (right), the second most likely class is the correct one. This could point to using the entire distribution rather than just the maximum to inform the classification.

where μ is called the momentum coefficient and is a new hyperparameter introduced to the system. v_t is the velocity for the previous epoch. The term in the argument of the gradient allows this method to correct the velocity term in a faster way if the current prediction is not good. For example, if the product μv_t results in a poor update for the learnable parameter then the gradient function calculated will be steeper and thus tend back towards θ_t such that the optimiser can try again in another direction. Thus SGD with Nesterov momentum allows the traversal of the loss space at an accelerated rate but, by construction, since areas with flatter curvature will be closer to the minima, the acceleration will slow as the minimum is approached and thus the optimiser will not overshoot.

The CNN is trained over 100 epochs at a constant learning rate $\eta = 5{\times}10^{-4}$ with a momentum coefficient of 0.9 and a batch size of 32. After 4 epochs, the CNN achieves 99.92% classification on the validation set and this is taken as the final model (1 out of the 1318 validation images are misclassified, Fig. 4.4). The near-perfection of this model is impressive and not to be understated, as a perfect classifier for image data

	Filaments	Flares	Prominences	Quiet	Sunspots
Filaments	175	1	0	0	0
Flares	0	270	0	0	0
Prominences	0	0	304	0	0
Quiet	0	0	0	242	0
Sunspots	0	0	0	0	326

Table 4.1: The confusion matrix for the trained CNN. This is a representation of the network's performance on the validation set where each element in the confusion matrix is the number of images classified as containing a feature compared to the true feature contained in that image.

is difficult to come by due to the possibility of distortions and artefacts leading to misclassification. Training takes approximately three hours on an NVIDIA Titan Xp and the validation step takes 4.66 seconds per epoch on the same hardware (3.54ms per image per epoch). Note that the speed with which the CNN learns the task does not represent the complexity of the task but rather that the initialisation of the parameters was particularly good. The initialisation is determined by a random seed that can be user-chosen or picked by default by the library being used. This changes where the solution begins in the loss space and means it is likely that the training starts in a place close to a local minimum giving excellent results[5]. This highlights that retraining the same network can lead to different results unless everything is kept the same. An important takeaway from this is if the model gets a high validation accuracy, but not to the user's requirements, then retraining by altering single hyperparameters can lead to finding the increase in performance desired.

4.4 Validation and Confusion Matrix

Classification percentage on a validation set is, however, not statistically robust enough to determine whether or not a classifier has actually learned what it is set up to do. This can be a result of having an uneven split in the validation set between the classes or having a strongly biased classification task. To deal with this, the "confusion matrix" is calculated for the CNN. This is a matrix whose elements correspond to what class an image actually belongs to compared to what class the

[5]Note that ideally the network would find a global minimum in the loss space for the task it is trying to learn but attaching such certainty to a converged solution is difficult hence the use of the term "local" when describing the minimum

network classified it in. This is shown in Table 4.1. This quantifies the types of errors made by a classifier. The predictions a classifier makes can now be split into four categories for each of the features:

i) *True positives*: the number of images containing the feature of interest correctly identified as containing that feature. That is, for a feature i that is of interest:

$$\mathrm{tp}_i = c_{ii}, \tag{4.8}$$

where c_{ij} is an element of the confusion matrix.

ii) *False positives*: the number of images not containing the feature of interest that are identified as containing that feature

$$\mathrm{fp}_i = \sum_{k=1}^{n_{\mathrm{rows}}} c_{ki} - \mathrm{tp}_i. \tag{4.9}$$

iii) *False negatives*: the number of images containing the feature i that are misclassified as not containing the feature.

$$\mathrm{fn}_i = \sum_{l=1}^{n_{\mathrm{cols}}} c_{il} - \mathrm{tp}_i. \tag{4.10}$$

iv) *True negatives*: the number of images not containing feature i that are correctly classified as not containing feature i

$$\mathrm{tn}_i = \sum_{l=1}^{n_{\mathrm{cols}}} \sum_{k=1}^{n_{\mathrm{rows}}} c_{kl} - \mathrm{tp}_i - \mathrm{fp}_i - \mathrm{fn}_i. \tag{4.11}$$

From these measures two statistics can be defined that can probe how well a classifier works. The first is known as *precision* and this is a measure of the fraction of the images that the model classified as having feature i that truly contain feature i

$$\rho_i = \frac{\mathrm{tp}_i}{\mathrm{tp}_i + \mathrm{fp}_i}. \tag{4.12}$$

The second is known as *recall*. This is a measure of the fraction of images containing feature i that were correctly identified as containing feature i. This can be thought

of as the ability of the model to find all of the images of interest

$$r_i = \frac{\mathrm{tp}_i}{\mathrm{tp}_i + \mathrm{fn}_i}. \tag{4.13}$$

Ideally precision and recall will both be equal to one for all classes. The precision for flare ribbons deviates from one as the misclassified image is misclassified as a flare ribbon. This corresponds to the image not containing a flare ribbon but the network deciding it does. The precision for all other classes is one, meaning that the network does not classify any images not containing these features as actually containing them. The recall for filaments is the only recall different from one as it is an image containing a filament that is misclassified. This means that the network thinks this image containing a filament actually contains another feature (in this case a flare ribbon). The recall being equal to unity for all other classes means that the network never classifies any of those images as having a feature different from the feature they contain. Overall, the misinterpretation of the CNN is not detrimental to its performance.

Figure 4.5 shows examples of the network classifying images. These are images from SOT in Hα which the network has not seen during training. This provides a test to ensure our network is not "memorising" the training data i.e. adjusting its learnable parameters to classify only the training set correctly.

The second column of Fig. 4.5 shows an image with clear flare ribbons that are classified correctly by the network. However, there is also a sunspot in this image which the network picks up on in the probability distribution (second column, bottom). There is a non-negligible probability that the important feature in this image is a sunspot. This means that the network can be used for classification of multiple large-scale features in a single image. However, a more precise way to do multi-label classification would be more beneficial and is discussed in Sec. 4.6. The other images are classified correctly in Fig. 4.5 showing that the model has learned the geometry of these features.

4.5 Application to SDO/AIA Wavelengths

Having trained the network on Hinode/SOT Hα data, adversarial tests on the network are performed. These are tests in which the input to the network is designed to be confusing to the network. Adversarial examples are used where the answer is

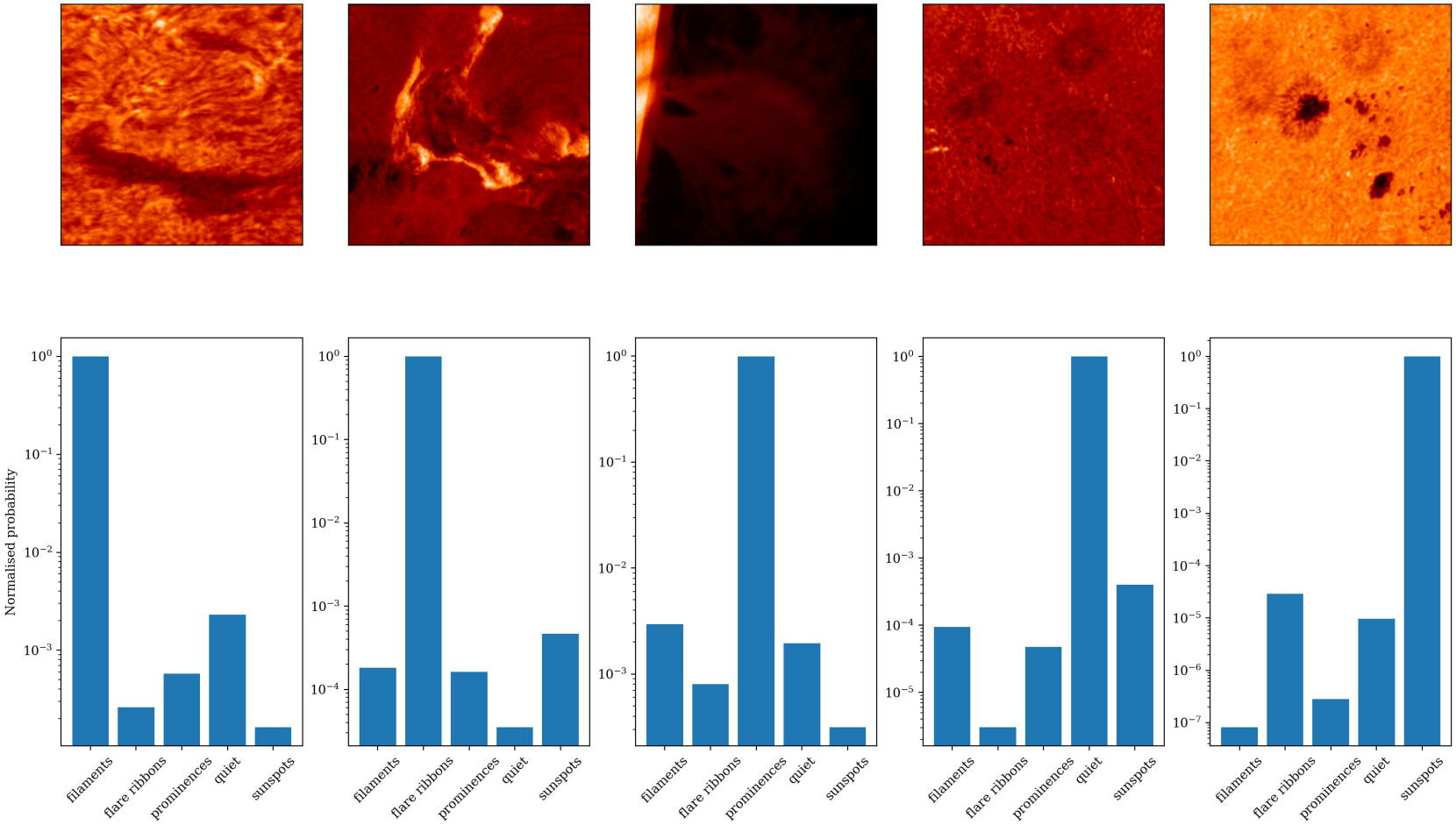

Figure 4.5: Validation on unseen images from Hinode/SOT imaged in Hα. This shows our network correctly identifying images with a filament (first column), flare ribbons (second column), prominences (middle column), quiet Sun (fourth column) and sunspots (fifth columns) in images it has never seen before.

obvious to the user but not necessarily to the network. That is, datasets of sunspots and prominences are used in different wavelengths given that they look perceptually similar to the features in Hα. The sunspot datasets come from the UV wavelengths (1600 & 1700Å) of SDO/AIA and the prominence datasets come from the 304Å EUV channel. This gives an idea of whether or not the trained CNN can generalise to other wavelengths without retraining.

4.5.1 Sunspots in UV

Three different sunspot datasets observed in UV are used as adversarial tests: AR11638 from 2013/01/01, AR12665 from 2017/07/10 and AR12674 from 2017/09/06. Each dataset used is over a one hour time range (12:00:00–13:00:00 UTC).

The 1600Å sunspot data was not classified well by the trained CNN, with the 1700Å data faring better. In 1600Å, every image was classified incorrectly as containing either flare ribbons or a prominence with results shown in Figs. 4.6, 4.7 & 4.8. It is hypothesised that there could be two possible reasons for this:

i) There are small-scale UV brightenings around sunspots. This can be attributed to plage (dispersed brightenings in an active region). While these brightenings occur in the optical and the ultraviolet; they are more noticeable in the UV due to the background UV quiet Sun being dimmer than in the optical. This implies that the contrast between plage and quiet Sun in UV wavelengths will be higher which can impact the network's classification ability by convincing it that the brightenings are the important feature. Furthermore, the plage can often look like elongated bright regions and this elongation may be further proof to the network that this image should be classified as something other than a sunspot.

ii) The lack of spatial resolution in the AIA images. In Hα, SOT has a spatial resolution of 0.33″whereas for the AIA UV channels the spatial resolution is 1.2″. This disparity could be another cause (or composite cause) of the misclassification of the UV sunspot observations. Due to the nature of convolutional feature extraction, the extracted features from two images of the same object but with different resolutions can be vastly different. This would affect the feature maps being passed through the trained CNN and thus the end classification result.

The first hypothesis is tested using these three datasets and the results are illustrated in Figs. 4.6, 4.7 & 4.8. Due to the incorrect classifications being either flare ribbons or prominences for these active regions it is believed that the brightness and elongation of the plage region is responsible for this. As can be seen in Figs. 4.1 & 4.5, in Hα both flare ribbons are bright, elongated structures on a darker background which is what leads us to believe the first hypothesis is responsible for incorrect classification. Moreover, for the 1600Å sunspot observations that are misclassified as prominences (namely Fig. 4.7 third column) the image background is darker than the other images possibly leading the network to be confused into thinking this image contains a prominence. It could also be that in this image the bright plage region resembles more of the geometry of the prominences that the CNN is trained on than the flare ribbons.

To test for the second hypothesis, sunspot observations that are cotemporal with observations from SOT in Hα must be used. The results of this test are shown in Fig. 4.9. The dataset chosen was from a single-sunspot active region AR11108 from 2010/09/25. The observations used from AIA were taken from 08:05:00 – 09:20:00 UTC. An example is shown in the left column of Fig. 4.9 where the sunspot was misclassified as a flare ribbon. The observations used from SOT were taken contemporaneously with AIA in Hα. These Hα images are downsampled by a factor of 3 to AIA resolution. The full resolution and low resolution images are then passed to the network. Both sets of images are classified perfectly by the network as shown by the middle and right columns of Fig. 4.9. This result invalidates the second hypothesis and leads to the conclusion that resolution is not a determining factor in misclassifications. Thus, it is concluded that the plage is the feature confusing our network from understanding sunspots in 1600Å.

In 1700Å, despite the sunspots in each active region not evolving much over the observed time range, the network sometimes classifies these sunspots correctly whilst sometimes incorrectly classifying them as either flare ribbons or prominences (Figs. 4.10, 4.11 & 4.12). In general, the trained CNN performs much better in 1700Å than 1600Å as it actually gets most classifications correct. The results shown in Figs. 4.10, 4.11 & 4.12 show confident sunspot classifications in the 1700Å data. It seems as though the bright plage regions in 1700Å have less of an effect on the classifications than they do in 1600Å leading to some interesting possibilities of how the network interprets the data. There appears to be a lower contrast between the plage and the background in 1700Å than in 1600Å and comparing to SOT Hα sunspot ob-

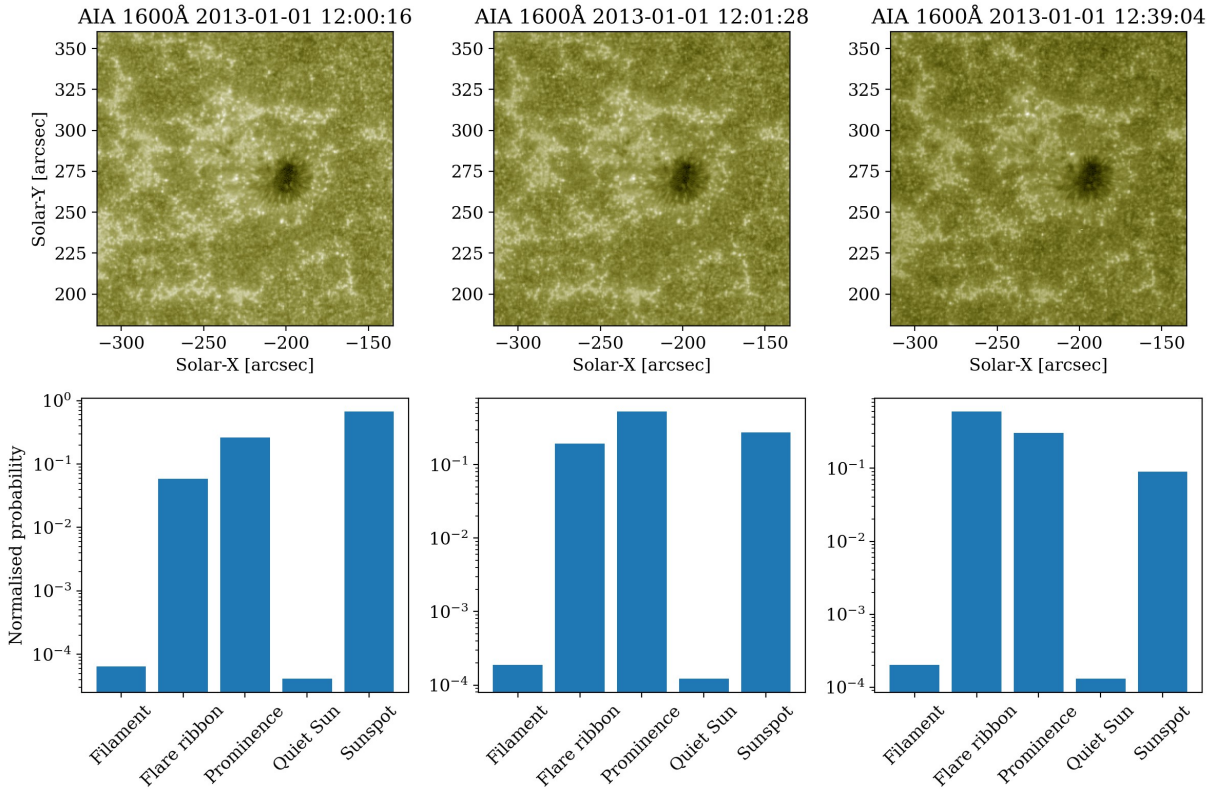

Figure 4.6: The three images analysed here are of the same sunspot (AR11638) imaged in SDO/AIA 1600Å a few minutes apart. These are shown to highlight the confusion of the network when dealing with 1600Å sunspots as the sunspot is never classified correctly. The sunspots in 1600Å are always either classified as flare ribbons or prominences. It is hypothesised that this is due to the elongated, bright plage in the images.

servations this points to a potential reason why the trained CNN classifies 1700Å correctly but not 1600Å. Looking at the last columns of Figs. 4.1 & 4.5, perceptually it makes sense for images with less contrast between the plage and the background to be classified more accurately as the network is trained on images without these bright plage regions. Methods such as saliency maps or SHAP (SHapely Additive exPlanations; Lundberg and Lee, 2017) for exploring the important features the network picks out from the images when doing these classifications would confirm these hypotheses.

4.5.2 Prominences/Filaments in 304Å

Two different datasets for three different prominences are used to investigate if the trained CNN can classify prominences in other wavelengths. SDO/AIA 304Å ob-

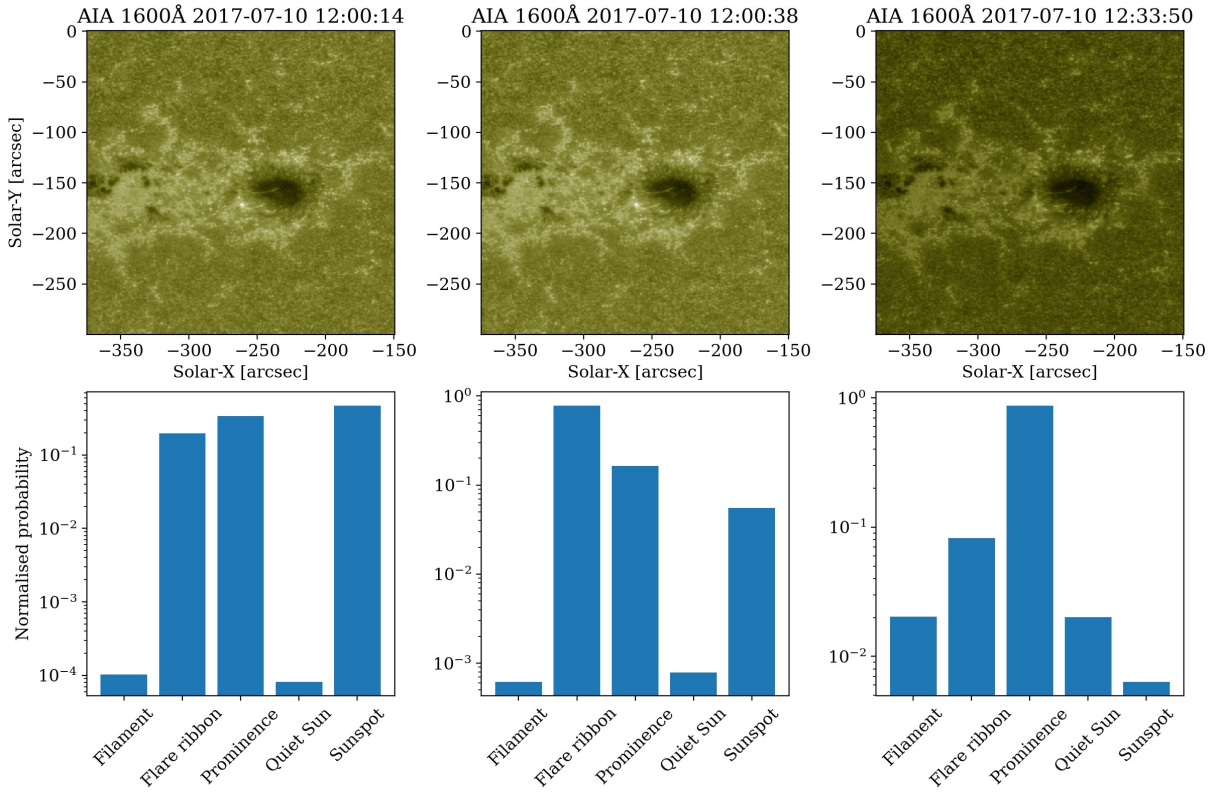

Figure 4.7: Same layout as Fig. 4.6 but for the sunspot in AR12665.

servations are used to look at prominences which correspond to HeII emission at ≈50,000 K. These datasets were taken from 2012/08/31 12:00:00–13:00:00 UTC and 2013/01/01 10:00:00–11:00:00 UTC. The 2012/08/31 dataset has the prominence located off the eastern limb of the Sun and is shown in the right column, top row of Fig. 4.13. The 2013/01/01 dataset has two prominences: one located off the eastern limb north-east from disk centre and another located off the eastern limb south-east from disk centre. These are shown in the left and middle columns, top row of Fig. 4.13.

As shown in the bottom row of Fig. 4.13, none of the structures are predicted correctly. This is thought to be caused by the noisy coronal background emission at the heights of the prominence. This is seen in the images in Fig. 4.13, where there is emission in the region of the prominence that is not directly from the prominence. In contrast, the Hα images from Hinode/SOT do not have emission except in the prominence at the heights of the prominence (as can be seen in the third columns Figs. 4.1 & 4.5).

All of the images in Fig. 4.13 are misclassified as flare ribbons. For the two

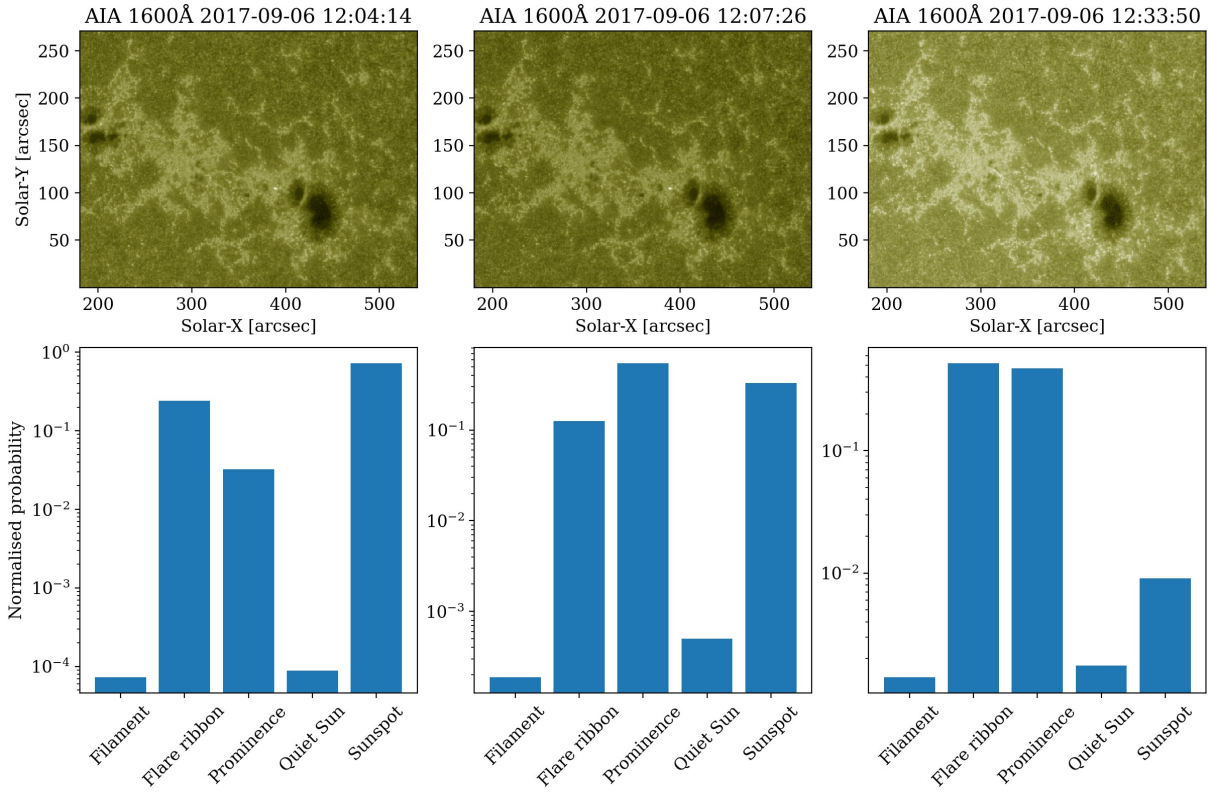

Figure 4.8: Same layout as Figs. 4.6 & 4.7 but for the sunspot in AR12674.

prominences from 2013/01/01, this is caused by the background HeII emission as this causes the prominence to appear bright against an emitting background which is similar to the flare ribbon images used for training in Hα. The filament from 2012/08/31 has comparable probabilities of the image containing a flare ribbon or a prominence. In the image of this filament, it is assumed that for the flare ribbon classification that the trained CNN chooses the bright point in the middle of the image as the most important feature. Interestingly, though, the trained CNN picks up the geometry of the filament as a different feature and is almost equally confident that this image contains a prominence. Also the same argument as in Sec. 4.5.1 follows: that the difference in resolution does not impact the CNN's classification ability. Therefore, the conclusion that only the coronal emission affects the classification ability of the trained CNN is reached. Again, the saliency maps or SHAP method could be used to confirm this hypothesis.

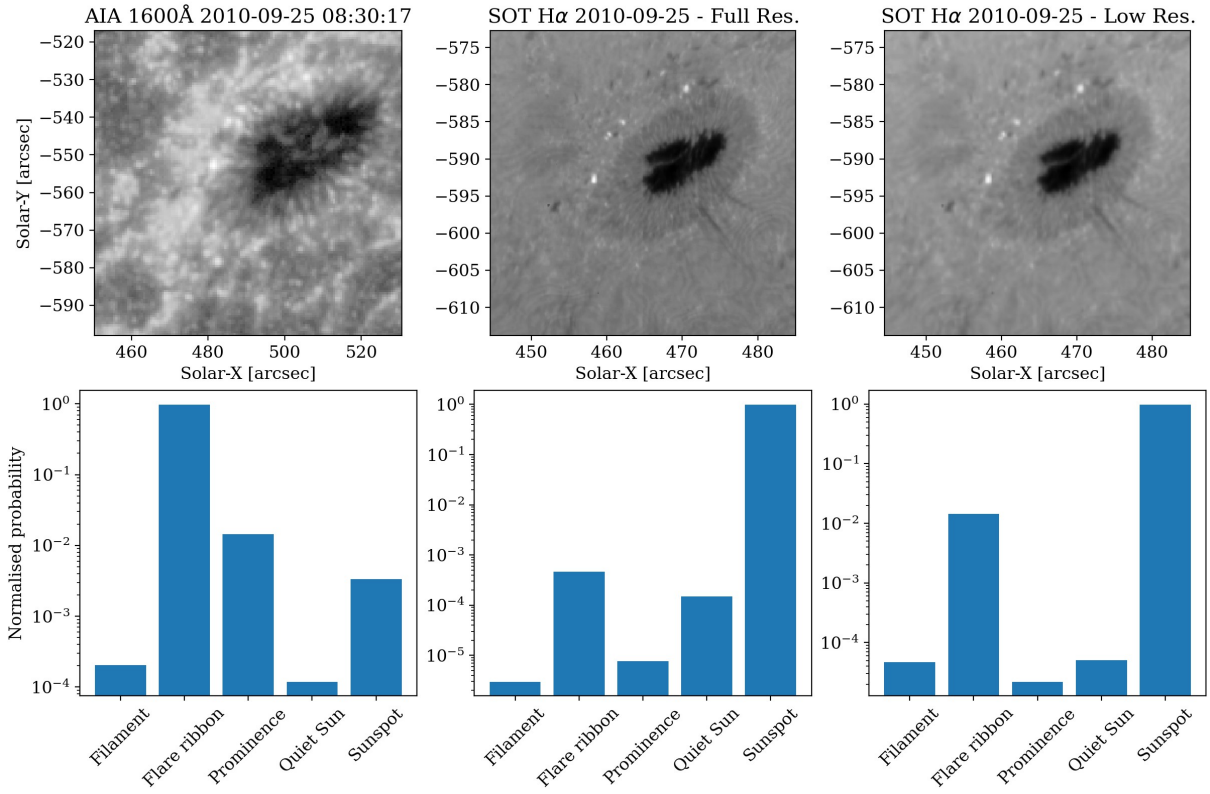

Figure 4.9: Comparison of the sunspot from AR11108 images in SDO/AIA 1600Å (left column) and Hinode/SOT Hα at full resolution and degraded to AIA resolution (middle and right columns, respectively). This illustrates that the resolution does not play a significant role in skewing the classification of the trained CNN as both the full resolution and low resolution SOT images are classified correctly whilst the AIA image is not.

4.6 Conclusion and Further Work

The deep convolutional neural network presented in this chapter has been shown to be able to learn the geometry of features on the Sun. This works very well for the wavelength that the network is trained on but does not always generalise to other wavelengths (which is to be expected due to some emission mechanisms occurring in some wavelengths and not others). This leads to a discussion of how the network can be improved through more detailed classification and multi-wavelength training regimes that could produce a classifier that generalises better to unseen data. Also increasing the depth of training can lead to more efficient uses of transfer learning from one network to another.

Further improvements to the network will make it more versatile and precise. In

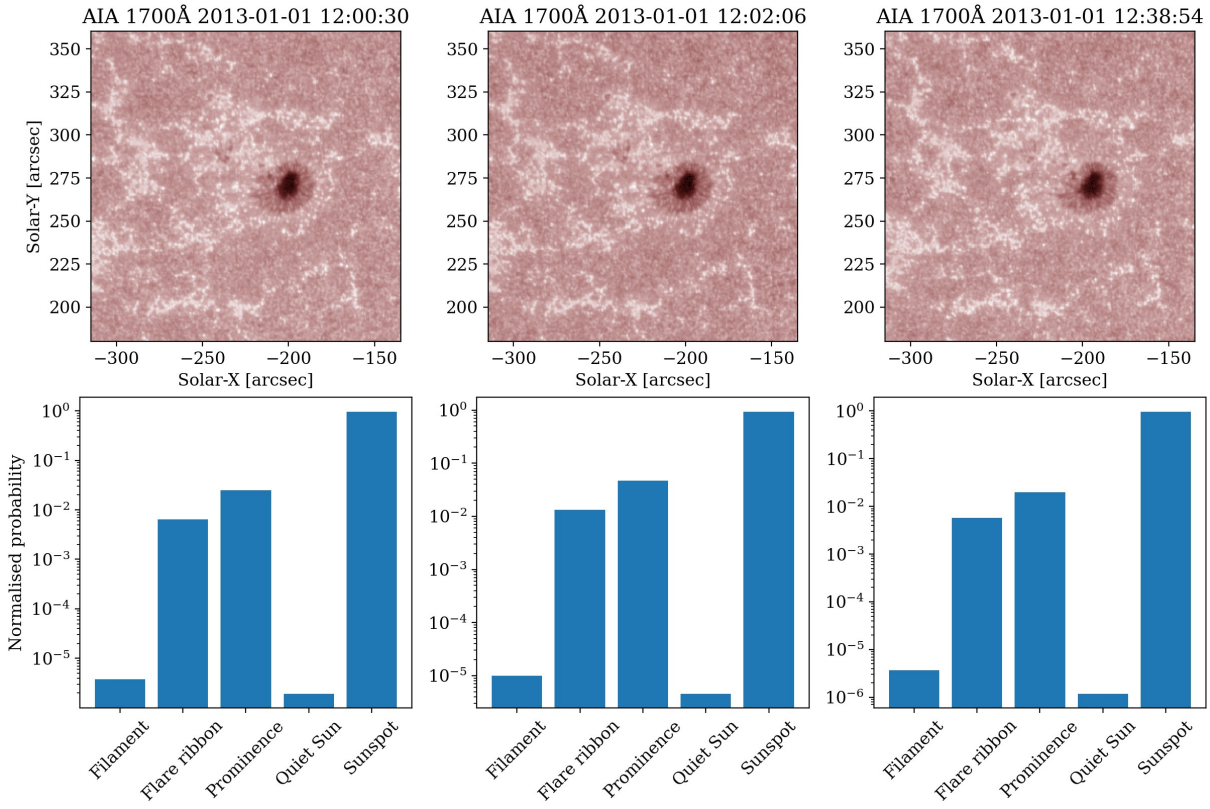

Figure 4.10: SDO/AIA 1700Å observations of the sunspot in AR11638 (top row) and the resulting classifications by the trained CNN (bottom row). This performs much better than the same dataset observed in 1600Å as all three examples here are classified correctly as sunspots.

the versatility direction, multi-label classification can be used. This means that each image will have more than one label e.g. n sunspots or single flare ribbon; this can be analysed sequentially. One way to do this is by using multiple binary classifier CNNs on the images and using the results from the binary networks to determine what features are in an image, e.g. one network to detect sunspots, one to detect flare ribbons and so on (Read et al., 2011). Another is using an ensemble method where there is a set of multiclass classifiers that each assign one label to the image. These predictions are then combined with each class getting a certain percentage of a vote from each classifier and the labels with a percentage above a certain threshold are used as the multi-label for the image (Rokach et al., 2014). This can be done using a recursive neural network (RNN). An RNN is a network that is specialised at processing sequential data. RNNs do this by using the previous layer's output as the dependency for the current layer's input – there is some function that connects

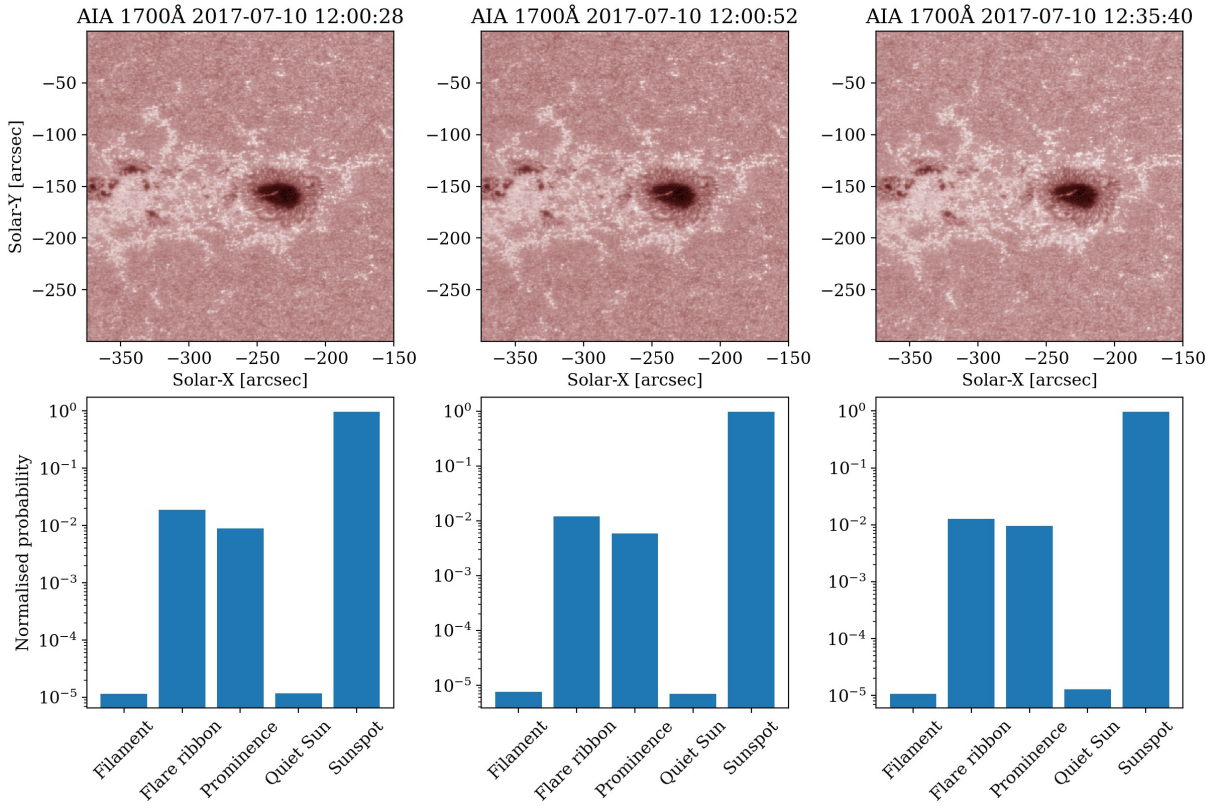

Figure 4.11: Same layout as Fig. 4.10 but for sunspots observed in AR12665. The data here is also classified correctly as sunspots despite its 1600Å counterpart being misclassified.

the output of the previous layer to the input of the current layer in a specified sequence (a "recurrence relation"). Following Bui et al. (2016), a convolutional RNN (C-RNN) which takes the feature maps from the last convolutional layer in the original network (after activation) as an input and outputs a compact representation of each feature over many convolutions can be used. This allows the C-RNN to learn a general form for the features (i.e. over many convolutional filters). For multi-label classification, the C-RNN architecture network will generate N RNNs to describe each image by N labels. For example, if an image contains at least one sunspot and the user wants to know if it has a single sunspot or multiple sunspots then two RNN blocks will be used – one to predict that the image contains a sunspot and the second to predict how many sunspots are in the image. This has seen great success in other image classification cases (Bui et al., 2016; Wang et al., 2016) and could work well for solar images.

There are many changes that can be implemented to improve precision. The

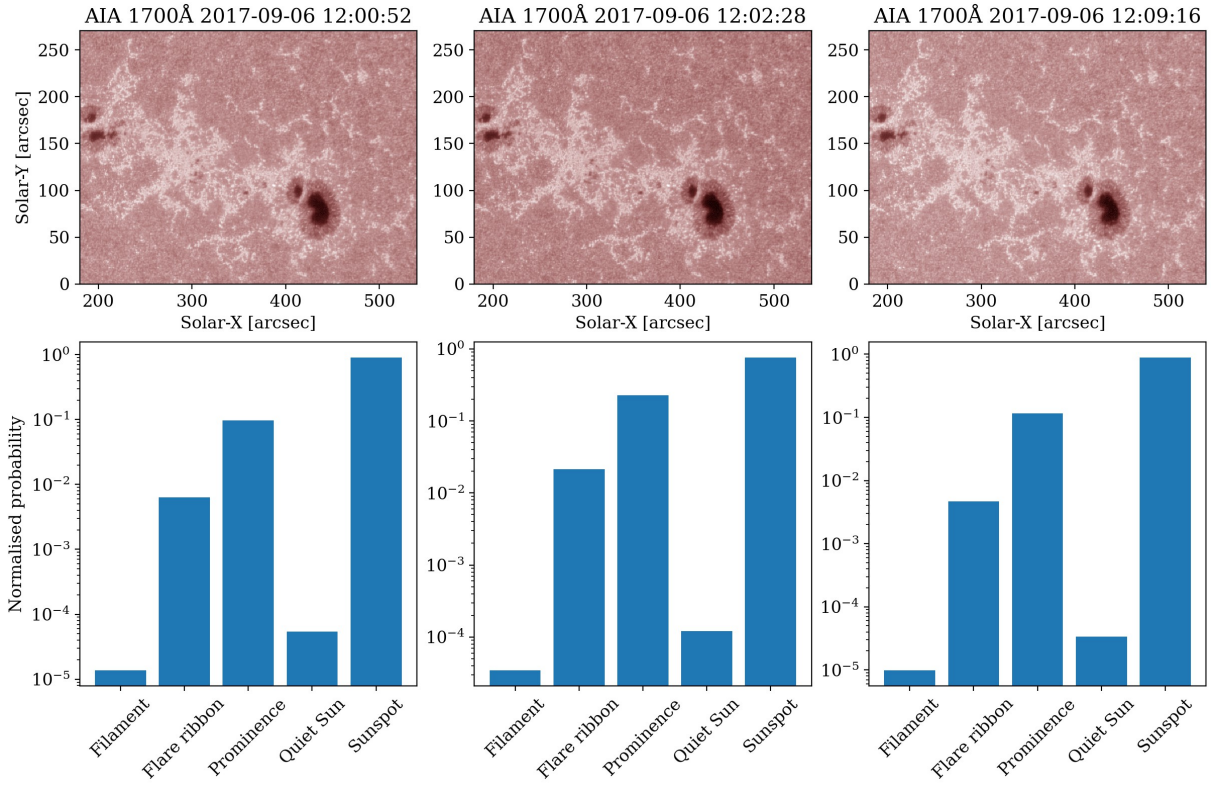

Figure 4.12: Same layout as Figs. 4.10 & 4.11 but for sunspots observed in AR12665. The data here is also classified correctly as sunspots despite its 1600Å counterpart being misclassified.

dropout layers could be replaced with max-dropout or stochastic dropout proposed in Park and Kwak (2017) which has improved performance on standard datasets. Another possibility is to change the convolution blocks to residual blocks (He et al., 2015a) wherein the network learns the residual of a function (the difference between the function and the input) rather than the function itself (see Sec. 5.4 for more information). This has been shown to improve speed, performance and how deep a network can be before suffering from vanishing gradients.

Another interesting property of the trained CNN is that it is based on a series of very successful deep CNNs known as VGG networks which were made to learn the ImageNet dataset (Deng et al., 2009; Simonyan and Zisserman, 2014). These deep architectures are necessary for solar image classification as shallower networks did not yield sufficient results (even for a simple task such as image classification). The ImageNet dataset is a well-known database of millions of images that has been classified into thousands of classes. This has been an incredibly successful approach

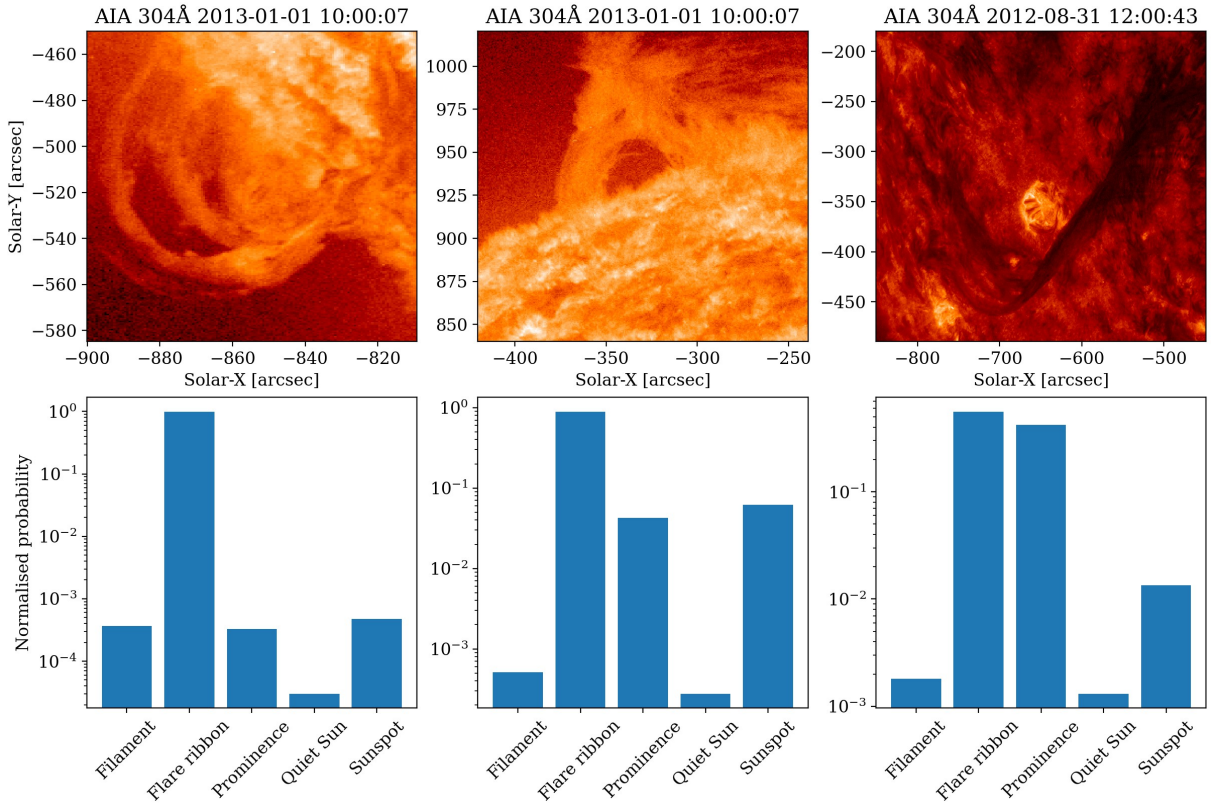

Figure 4.13: Examples of incorrect classification of prominences/filaments observed in SDO/AIA 304Å. The left and middle columns are quite confidently classified as flare ribbons which is thought to be due to the background coronal HeII emission visible in these images but does not have an analogue in the Hα training set. The right column shows the trained CNN thinking that the image is nearly as likely to contain a prominence as a flare ribbon. It is assumed that the trained CNN identifies the bright patch in the middle as a flare ribbon but also picks up the filament above it. This shows the trained CNN's capability of giving a good idea if there are multiple features in a single image without being explicitly taught to do so.

and is useful in transfer learning. The pre-trained VGG networks have proven to be extremely useful for transfer learning for real-world images (Kupyn et al., 2017; Johnson et al., 2016). This leads to the proposal of an analogous solar dataset i.e. a solar ImageNet (SIN). SIN would be a huge dataset containing features imaged in different wavelengths from different instruments. A classifier could then be trained to learn what these features look like in different solar contexts. (Or a series of classifiers to identify features in different wavelengths before the overall result is combined at the other end). The classification network presented here can be used as a building block for SIN and acts like a VGG network trained on a subset of

ImageNet.

This would make a transfer learning approach to solar machine learning extremely plausible and could lead to increased accuracies in deep learning tasks in solar physics compared to the same networks initialised without transfer learning. For example, this kind of network would be useful in data pipelines for creating catalogues of data and picking up on observations that were targeted at a specific feature but picked up something else too. Furthermore, the network presented can be used in conjunction with already existing data pipelines where the data may not have a specific target specified in the meta information. Due to its speed and accuracy, this model will be useful for anyone having to sift through terabytes of data. Lastly, networks of this design could be utilised in automating telescope pointing. With more detailed training, a sufficient network could parse synoptic observations of the observer's target and calculate where the target will be when the observations will be occurring.

5 | Correcting for Atmospheric Seeing in Solar Flare Observations

The following chapter presents a more in-depth view of the research published in Armstrong and Fletcher (2021) on the Seeing AUtoeNcoder (Shaun). Particularly, Sec. 5.2 details the mathematical derivation of the synthetic seeing model used while Sec. 5.6 provides more examples of the trained model in action.

5.1 Atmospheric Seeing and the Current State of the Art

Atmospheric scintillation (also referred to as seeing) is the refraction of incoming wavefronts of light from astronomical sources by the turbulent Earth's atmosphere. This causes the observations to become noisy and degraded as the incoming light is no longer coherent. This is ubiquitous in ground-based astronomy. It poses a problem for all observers, particularly those studying highly variable phenomena. It has become the norm for observing facilities to use adaptive optics (AO) systems in their optical path to correct for the wavefront deviations introduced by the atmosphere. AO systems use a wavefront sensor to detect changes in the photon's trajectory from the plane-parallel direction. The sensor then sends commands for how to correct the photons back to being plane-parallel to a deformable mirror which applies this correction to incoming photons. There are two important atmospheric parameters

that characterise an AO system: the isoplanatic angle, θ_0, and the coherence time, t_0. Both of these quantities depend on the Fried parameter r_0 (Fried, 1966, discussed further in Sec. 5.2) which is a quantity that describes the length scales over which turbulent areas of the Earth's atmosphere will produce coherent refractions to photons.

The isoplanatic angle is the maximum distance in the sky that two plane waves can be separated and still pass through the same turbulent cell before reaching the detector. The isoplanatic angle is directly proportional to the Fried parameter and inversely proportional to the height of the turbulent layer. This means that the isoplanatic angle is smaller for smaller values of the Fried parameter, i.e. less coherent turbulence leads to smaller source separation passing through the same turbulence. That is, when the seeing is worse, it is harder to estimate the statistics characterising the atmosphere and thus harder to correct for the seeing. Mathematically, the isoplanatic angle is given by

$$\theta_0 = \frac{r_0}{h}, \tag{5.1}$$

where h is the height of the turbulent layer that the wavefronts are passing through. With modern imagers looking at extended sources, θ_0 is smaller than the field of view being imaged (known as *anisoplanatism*). θ_0 is estimated from the centre of the field of view meaning that the statistics may not hold for observations outside of a disc with radius θ_0 centred on the image centre. The same wavefront corrections are used over the entire field of view meaning that the corrections outside of the disc will not be as accurate.

The coherence time is the evolution timescale of the turbulence for a given Fried parameter. That is, within the coherence time, an area of the atmosphere will produce the same refractions to wavefronts passing through it. Therefore, once a coherence time has passed there will no longer be the same deformations to the light. This is due to the wind speed in the atmosphere and the coherence time can be written as

$$t_0 = \frac{r_0}{v}, \tag{5.2}$$

where v is the average wind speed in the atmosphere. Moreover, this means that the AO system must work on timescales shorter than the coherence time to make sure that the corrections sent to the deformable mirror are correct for incoming wavefronts. This can be difficult as typical values for t_0 are $O(10\text{ms})$.

As a result of constraints by θ_0 and t_0, when the seeing conditions are particu-

larly bad, the object being observed evolves faster than the speed of the AO system, or the field of view is much larger than the isoplanatic patch, and post-processing techniques must be introduced to correct for seeing. The two most common post-processing techniques used in solar physics are Speckle interferometry and Phase Diversity (PD) methods.

Speckle interferometry is the process of using 2D power spectra of many short exposures to estimate the atmosphere's effects on the observations and correct for these (for solar applications these methods are developed in von der Lühe and Dunn, 1987; von der Lühe, 1993). As the turbulence changes on small time scales, t_0, analysing many short exposures will lead to any errant motions within the data being solely due to the atmospheric turbulence. This process consists of two passes through the set of short exposures. In the first pass, the Fourier transform of the images is found before averaging these transforms and their power spectra. The ratio of the average transforms to average power spectra is independent of the static structure within the field of view and thus only depends on the atmospheric turbulence present (averaging the observations will remove high frequency data due to the atmospheric turbulence but the average power spectrum will maintain the higher frequency information so that the ratio can be used to estimate the turbulence). The information learned about the turbulence is used to produce a noise filter characterising the atmospheric noise. This noise filter is then used to remove the seeing effects by passing through the data again. However, due to anisoplanatism, the field of view in each short exposure frame must be divided into overlapping subfields before being corrected individually. Once the subfields are corrected, they are combined back into the whole field of view by examining their cross spectra, and an inverse Fourier transform is performed to recover the corrected image in real space.

For the most dynamic of processes (e.g. solar flares), the evolution time will be shorter than the cumulative length of the exposures required to obtain the frames necessary for the reconstruction. This means that the ratio of averaged Fourier images to averaged power spectra will depend also on the dynamics of the processes being imaged if all short exposures are considered. Therefore, the atmospheric parameters will need to be estimated from a number of consecutive short exposures which is suboptimal for the algorithm. This will lead to greater uncertainty in the atmospheric parameters and a poorer restoration, as a result.

PD methods jointly estimate the restored image and the distortions responsible for the aberrated image in a maximum likelihood estimation. The state of the art PD

method in solar physics is multi-object multiframe blind deconvolution (MOMFBD; Van Noort et al., 2005). MOMFBD implements a simple model of the optics and detectors used in the observations, eliminating the need to rely on the atmospheric statistics as in Speckle reconstruction. Synthetic images are then generated by different pupil functions until a maximal likelihood pupil function is found. As explained in Van Noort et al. (2005), PD methods work best when contrast is high, noise is low, and exposure time is short. This is difficult to achieve in narrow-band solar observations and, as a result, wide-band data collected simultaneously must be used to aid in the MOMFBD restoration. It is this that poses the biggest problem for the restoration of flare data. Chromospheric energy deposition in a flare is mostly seen through the enhancement of optical and near-infrared spectral lines and not necessarily strong continuum enhancements (Fletcher et al., 2011). This can lead to the objects being studied looking very different in the wide-band and narrow-band observations. Given that the wide-band is used to help the optimisation of the restoration, in cases where there is no continuum enhancement in a flare, it can actually be a hindrance to the restoration[1].

Furthermore, both Speckle and PD methods have a limit to their restoration capabilities (as all methods will). This is detrimental to flare observations due to their sporadic nature meaning observers cannot wait for optimal seeing conditions to observe. For these reasons, a dedicated flare seeing-correction tool is proposed based on training a deep neural network (DNN) on diffraction-limited narrow-band flare data synthesized with artificial seeing. Note that this model is for the application to data that is time-integrated i.e. has already been processed by one of the aforementioned methods. This is due to the assumption of the azimuthally-symmetric point-spread functions in the model which makes it unsuitable for use on raw frames of data. This use and how it can be expanded upon are discussed further in Chap. 7.

[1]Note that MOMFBD also provides near-perfect image alignment between the images taken at different wavelengths which is something that is assumed by the neural network model described below. Inclusion of this kind of registration between images would be hugely important for processing the data completely MOMFBD-free.

5.2 Development of a Seeing Model From the Statistics of Turbulent Media

The following section discusses the development of the model to simulate the effects of synthetic atmospheric seeing on an image, dependent on the wavelength the data is observed at. This model is used to generate the training dataset in Sec. 5.3 which is in turn used to train the DNN to correct for real seeing (Secs. 5.5, 5.6). In any photon-collecting system, any change in medium that the observed light travels through has an effect on the light when observed by a detector. This is typically characterised by a *point spread function* (PSF) of the system which "spreads" the photons in a smearing pattern either spatially or spectroscopically. The PSF of different instruments within a photon-collecting system are well characterised before construction of said system, allowing for the removal of these effects during post-processing of the data to present the data as true to the emitted light from the source as possible. The removal of such artefacts is posited as a deconvolution problem as the observed data, O at a detector can be written:

$$O = I * P + \mathcal{N}, \tag{5.3}$$

where $*$ represents the convolution of I and P, I is the light emitted from an astronomical source, P represents the PSF of the system and $\mathcal{N}$ represents random noise occurring within components of the system such as readout or thermal noise. In modern systems, even the random noise attributed, $\mathcal{N}$, can be well characterised allowing for accurate reconstruction of emitted light. The problem then becomes observing astronomical sources from the ground.

Much like observing through an instrument, observing through the Earth's atmosphere alters the incoming photons that observers wish to detect. These alterations in reality are the incoming photons being refracted by random variations within the atmosphere's density and temperature structure. Such refractions change the path length of the photons greatly impacting ground-based observations by introducing a large smearing effect. While the instruments used can be well characterised, it is much more difficult to characterise the atmosphere as it is a turbulent system meaning the changes within the system are random. The effects of

the atmosphere can be written in the same way as Eq. 5.3 as a PSF:

$$O = I * P_{\text{atmos}} + G, \tag{5.4}$$

where everything retains its same meaning from Eq. 5.3, P_{atmos} represents the PSF of the Earth's atmosphere and G is random Gaussian noise. As discussed in Van Noort et al. (2005)'s Sec. 4.4.3, the use of Gaussian noise in favour of Poissonian noise in Eq. 5.4 is likely not the best assumption for spectral lines with low signal-to-noise ratio (SNR). However, since the cameras at the SST have a high SNR, Van Noort et al. (2005) claim that the assumption of Gaussian noise "is probably not a bad one at least in the wavefront sensing step". As such, the model here also uses this assumption of Gaussian noise given the majority of the data this method is applied to is from the SST.

The model being developed here boils down to finding a form for P_{atmos} that can be populated for various atmospheric conditions to create a diverse training set. Racine (1996) showed that the atmospheric PSF can be written generally as the Hankel transfer of the modulation transfer function (MTF) of the atmosphere:

$$P_{\text{atmos}}(\rho) = \int_0^\infty J_0(\rho\nu) \exp\{-0.5 D_{\text{S}}(\nu)\} \, \nu \mathrm{d}\nu, \tag{5.5}$$

where ρ is the 2D spatial coordinate, ν is the 2D spatial frequency coordinate and J_0 is the zeroth order Bessel function. The MTF depends solely on the *structure function* of the atmosphere, D_S, which is where the form of the seeing comes from.

To find a form for this structure function, the assumption is made following Tatarski (2016) that the Earth's atmosphere is a medium with smoothly varying turbulence. Media with smoothly varying turbulence have phase-structure functions of the following form:

$$D_{\text{S}}(\rho) = 2.91 k^2 \rho^{5/3} \int_{\vec{\ell}} C_n^2(\vec{r}) \mathrm{d}\vec{r}, \tag{5.6}$$

where k is the wavenumber of the photon, $\vec{\ell}$ is the path travelled by the photon and C_n^2 is proportional to the ratio of the rate of dissipation of inhomogeneities in the atmosphere, $\bar{N}$, to the cube-root of the mean energy dissipation per unit mass, ε, which describes the structure of the atmosphere at a point $\vec{r}$. To understand how Eq. 5.6 comes to be, the statistical description of continuous random fields must be

understood.

For Sec. 5.2.1, the mathematical definitions of random functions, stationary random functions and random functions with stationary increments along with their associated two-point correlation and structure functions are adapted from Tatarski (2016) Part 1 "Some Topics from the Theory of Random Fields and Turbulence Theory" with added description around some of the more dense mathematics. This is similarly the case when introducing random fields in Sec. 5.2.2. Deriving the expression for the structure function of the Earth's atmosphere follows the arguments presented in Tatarski (2016) Part 3 "Parameter Fluctuations of Electromagnetic and Acoustic Waves Propagating in a Turbulent Medium" under the assumption that the Earth's atmosphere is a medium with smoothly-varying turbulence. This results in a comprehensive derivation of the Kolmogorov structure function which is then used to estimate the entries in the seeing disk whose size is defined by the Fried parameter. Many of the mathematical steps shown in these sections are not presented within Tatarski (2016) (left as "exercises to the reader" some might say) but are presented here for transparency of the mathematical rigour required to reach the model.

5.2.1 What Actually *is* a Structure Function?

Consider a random function f. At a given fixed point in space, x, the value of the function $f(x)$ is a random variable (i.e. can assume a set of different values) and there exists a definite probability $F(x, \xi_1)$ such that $f(x) < \xi_1$. However, since f is random, to be able to be completely described, one must know the definite probabilities and in all dimensions making it very difficult to fully describe a random function in reality. Therefore, statistics of the random function are used instead to describe the random function. Tatarski (2016) makes use of two important statistics used to describe random functions: the mean of the random function $\overline{f(x)}$ and the correlation function within the field:

$$B_f(x_1, x_2) = \overline{\left(f(x_1) - \overline{f(x_1)}\right)\left(f^*(x_2) - \overline{f^*(x_2)}\right)}, \tag{5.7}$$

where f^* is the complex conjugate of f (however, only real functions are considered here so $f^* = f$). The correlation function of a random function gives a measure of how a change at point x_1 affects the random function at point x_2 and vice versa. If

x_1 and x_2 are statistically independent then the value of Eq. 5.7 will approach zero. The correlation function (Eq. 5.7) and the structure function (Eq. 5.6) turn out to be related for a specific kind of random function known as a "random function with stationary increments".

A random function is known as *stationary* if the mean of the random function is constant at different points in space and the two point correlation function depends only on the distance between the two points. This can simplify the description of the function as only the distance between two points being considered is important. The assumption is made (unless otherwise explicitly stated) that the mean value of a stationary random function is zero[2]. Therefore, Eq. 5.7 can be written

$$B_f(x_1, x_2) = \overline{f(x_1) f^*(x_2)}. \tag{5.8}$$

Furthermore, a stationary random function f can be written as a Fourier-Stieltjes integral with the following form:

$$f(x) = \int_{-\infty}^{\infty} e^{ikx} \, \mathrm{d}\varphi(k), \tag{5.9}$$

where $\mathrm{d}\varphi(k)$ are random complex amplitudes describing the fluctuations in the random process and k is the wavenumber of the fluctuation. Substituting Eq. 5.9 into Eq. 5.8, an expression for the correlation function depending only on the difference between the two arguments can be constructed

$$B_f(x_1 - x_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left[i(k_1 x_1 - k_2 x_2)\right] \overline{\mathrm{d}\varphi(k_1) \mathrm{d}\varphi^*(k_2)}. \tag{5.10}$$

As f is a stationary random function, B_f can only depend on the difference between x_1 and x_2 and the differential in Eq. 5.10 has the following form

$$\overline{\mathrm{d}\varphi(k_1) \mathrm{d}\varphi^*(k_2)} = \delta(k_1 - k_2) W(k_1) \mathrm{d}k_1 \mathrm{d}k_2, \tag{5.11}$$

where $\delta(\cdot)$ represents the Dirac delta function. Eq. 5.11 says that the correlation function only exists when the parameters $k_1 = k_2 = k$. Eq. 5.10 can then be written

[2] In the case where $\overline{f(x)} \neq 0$, there always exists a random function $g(x) = f(x) - \overline{f(x)}$ where $\overline{g(x)} = 0$.

as:

$$\begin{aligned} B_f(x_1 - x_2) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left[i(k_1x_1 - k_2x_2)\right] \delta(k_1 - k_2) W(k_1) \mathrm{d}k_1 \mathrm{d}k_2 \\ &= \int_{-\infty}^{\infty} \exp\left[ik(x_1 - x_2)\right] W(k) \mathrm{d}k, \end{aligned} \tag{5.12}$$

since $\int_{-\infty}^{\infty} \delta(x-a)f(x)dx = f(a)$. B_f and $W(k)$ are thus Fourier transforms of one another. The function $W(k)$ is the spectral density of the random function $f(x)$ describing how the turbulent fluctuations are spread on different spatial scales.

However, for the Earth's atmosphere, the mean value of the density and temperature change in time meaning it cannot be considered a stationary function. What is investigated instead, is whether there are length scales under which the Earth's atmosphere can be considered a stationary random function and whether or not it is feasible to evaluate it in this way. A random function that can be considered stationary under certain length/time scales is referred to as a *random function with stationary increments*. Treating the quantities (e.g. density, temperature, pressure) in the Earth's atmosphere as random functions with stationary increments allows the quantities to be described by a *structure function* rather than the two point correlation function. For example, if the random function f is not stationary, then consider small scales ϵ such that the difference

$$F_\epsilon(x) = f(x+\epsilon) - f(x), \tag{5.13}$$

is not largely affected by slow changes in f. That is, the function F_ϵ is approximately a stationary random function (consequently implying that f is a random function with stationary increments). If the two-point correlation function for the stationary random function F_ϵ is considered and under the assumption that the mean of the stationary functions are zero then:

$$\begin{aligned} 2B_F(x_1, x_2) = \overline{(f(x_1+\epsilon) - f(x_2))^2} + \overline{(f(x_1) - f(x_2+\epsilon))^2} - \\ - \overline{(f(x_1+\epsilon) - f(x_2+\epsilon))^2} - \overline{(f(x_1) - f(x_2))^2}, \end{aligned} \tag{5.14}$$

using Eq. 5.13 and the algebraic identity:

$$2(a-b)(c-d) = (a-d)^2 + (b-c)^2 - (a-c)^2 - (b-d)^2. \tag{5.15}$$

The two point correlation function (Eq. 5.14) can then be written as a linear combination of functions of the form:

$$D_f(x_i, x_j) = \overline{(f(x_i) - f(x_j))^2}, \tag{5.16}$$

which are referred to as the structure functions of the random process F_ϵ. From here on out Eq. 5.16 will be used as a proxy for Eq. 5.14. If the structure functions depend only on the difference between the two points being considered then the two point correlation function of the system will also only depend on this, satisfying the criteria that within scales of ϵ, the system is a stationary random function. As such the structure function is the standard characteristic used to describe a random function with stationary increments. It acts like a proxy for the two point correlation being specifically constructed to consider the process on scales smaller than or equal to ϵ. Similarly to how a stationary random function can be expressed using a Fourier-Stieltjes integral, a random function with stationary increments can be represented in the form

$$f(x) = f(0) + \int_{-\infty}^{\infty} (1 - e^{ikx}) \mathrm{d}\varphi(k), \tag{5.17}$$

where $f(0)$ is a random variable and the amplitudes $\mathrm{d}\varphi(k)$ follow Eq. 5.11. Eq. 5.16 can then be written

$$\begin{aligned} D_f(x_i, x_j) &= \overline{(f(x_i) - f(x_j))^2} = \overline{(f(x_1) - f(x_2))(f^*(x_1) - f^*(x_2))} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left(e^{ik_1x_1} - e^{ik_1x_2}\right) \left(e^{-ik_2x_2} - e^{-ik_2x_1}\right) \overline{\mathrm{d}\varphi(k_1)\mathrm{d}\varphi^*(k_2)}, \end{aligned} \tag{5.18}$$

and following the same logic as Sec. 5.2.1, $D_f(x_1 - x_2)$ can be expressed by the following integral and the spectral density of f

$$D_f(x_1 - x_2) = \int_{-\infty}^{\infty} \left[1 - e^{ik(x_2 - x_1)} W(k)\right] \mathrm{d}k. \tag{5.19}$$

Since the structure function and spectral density are real-valued functions, on the real part of Eq. 5.19 will yield a physical solution, therefore, it can be written

$$D_f(x_1 - x_2) = 2 \int_{-\infty}^{\infty} \left[1 - \cos(k(x_2 - x_1)) W(k)\right] \mathrm{d}k. \tag{5.20}$$

5.2.2 From Random Functions to Random Fields

A random field is simply a random function considered in three spatial dimensions. That is, the quantities to be described in the Earth's atmosphere are random fields. As with random functions in Sec. 5.2.1, the mean and two-point correlation function can be defined for points within a random field.

$$B_f(\vec{r}_1, \vec{r}_2) = \overline{(f(\vec{r}_1) - \overline{f(\vec{r}_1)})(f(\vec{r}_2) - \overline{f(r_2)})}, \tag{5.21}$$

where f is the random field and $\vec{r}_1$, $\vec{r}_2$ are vector positions within the field being considered.

The concept of a random field being stationary is now referred to as the field being *homogeneous* as the mean value does not vary as the field is moved through. A homogeneous random field also has the property that the correlation function depends only on the displacement between the two points. The field can also be referred to as *isotropic* if the values of the Eq. 5.21 depend on the separation between two points but not the direction in which they are separated. The fields to be emulated in the model are not globally homogeneous or isotropic but can be formulated to be locally homogeneous and isotropic meaning the structure function can be defined for these random fields as

$$D_f(\vec{r}_1, \vec{r}_2) = \overline{(f(\vec{r}_1) - f(\vec{r}_2))^2}, \tag{5.22}$$

following the same logic as in Sec. 5.2.1. Along the same lines, the structure function for a random field which is locally homogeneous and isotropic can be expressed in terms of a three dimensional spectral density $\Phi_f(\vec{k})$

$$\begin{aligned} D_f(\vec{\rho} = \vec{r}_1 - \vec{r}_2) &= \overline{(f(\vec{r}_1 + \vec{\rho}) - f(\vec{r}_1))^2} \\ &= 2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(1 - \cos\vec{k}\cdot\vec{\rho})\Phi_f(\vec{k})\mathrm{d}\vec{k}. \end{aligned} \tag{5.23}$$

Given that the random field f is assumed to be locally isotropic, the structure function will now depend on the magnitude of the distance between the two points $\rho = |\vec{\rho}|$ meaning

$$D_f(\rho) = 8\pi\int_{-\infty}^{\infty}(1 - \cos k\rho)\,\Phi_f(k)k^2\mathrm{d}k, \tag{5.24}$$

where $k = |\vec{k}|$.

Having argued that a structure function can be used to describe the nature of the random processes in the Earth's atmosphere, an analytical form for this function will be derived to be used to generate synthetic atmospheric seeing in solar flare observations.

5.2.3 Creating a Model of the Earth's Atmosphere

Plane parallel photons entering the Earth's atmosphere from a distant astronomical source undergo refraction due to inhomogeneities in the refractive index in the Earth's atmosphere. The refractive index inhomogeneities are present due to the turbulent nature of the density, temperature and pressure in the atmosphere and result in the refractive index being well-described by a random field. The assumption is made that the refractive index field is locally homogeneous and isotropic such that it can be described by the structure function given by Eq. 5.22. The characterisation of the field then comes from the form that this structure function takes when related to physical parameters.

Consider the plane parallel optical photons entering the top of the Earth's atmosphere from the Sun. Since there is negligible scattering of optical photons in interplanetary space, the assumption is made that at the top of the Earth's atmosphere, the photons are unchanged compared to when they were emitted. Therefore, an effective source of photons at the top of the Earth's atmosphere is assumed. The model of the Earth's atmosphere then aims to describe how the phase of the light wave changes as it is refracted by the inhomogeneities. That is, given a photon travels a distance L from the effective source to the detector, can the changes to this photon caused by the atmosphere be estimated?

Following Kolmogorov's theory of turbulent flows (Kolmogorov, 1941), the assumption is made that the Earth's atmosphere consists of a turbulent flow made up of eddies of varying sizes where there exists two important length scales: L_0 known as the outer length scale of the turbulence and l_0 known as the inner length scale of the turbulence. These two quantities are bounds on the size of the eddies in the turbulent atmosphere. The inner length scale is the lower bound on the size of the eddies and depends on the kinematic viscosity of the atmosphere — below scales of l_0, eddies are dissipated by viscous motions. The outer length scale is the upper bound on the size of the eddies and is known as the turbulence correlation length — below scales of L_0 but above scales of l_0, the field is locally homogeneous and

isotropic as it is within one eddy, allowing for the description in Sec. 5.2.2 to be used to describe the properties of the medium. Since the turbulent flow affects the atmospheric parameters, it will also affect the refractive index within the atmosphere leading to the refraction of incoming photons.

As described in Sec. 5.2.2, the structure function of a random field (in this case, the refractive index) is given by Eq. 5.22 and can be rewritten in terms of the refractive index of the atmosphere n as

$$D_n(\vec{r}_1, \vec{r}_2) = \overline{(n(\vec{r}_1) - n(\vec{r}_2))^2}, \tag{5.25}$$

To understand how the structure of the refractive index field affects the incoming radiation, the wave equation must be solved assuming the incoming photon is a plane parallel monochromatic wave. Under the assumption that the wavelength of the light $\lambda << l_0$ (which is a reasonable assumption as the discussion concerns optical light), the wave equation can be written in terms of the photon's electric field as:

$$\nabla^2 \vec{E} + k^2 n^2(\vec{r}) \vec{E} = 0, \tag{5.26}$$

where here k now refers to the photon wavenumber. This has a simple plane wave solution of:

$$\vec{E}(\vec{r}) = \vec{E}_0 e^{iS(\vec{r})}, \tag{5.27}$$

where S is the phase of the wave. Assuming that the amplitude of the wave remains unchanged by the refraction then the wave equation can be written in terms of the change of the phase as:

$$(\nabla S)^2 = k^2 n^2(\vec{r}). \tag{5.28}$$

Since n is a random field, there is no exact solution to Eq. 5.28 and instead a solution can be characterised by the fluctuations in n causing fluctuations in S, that is S and n can be replaced by

$$n = 1 + \delta n(\vec{r}), \tag{5.29}$$

$$S = S_0 + \delta S, \tag{5.30}$$

where the average refractive index of the Earth's atmosphere is 1 and $\delta n(\vec{r})$ is how this differs for some position $\vec{r}$. $S_0 = \vec{k} \cdot \vec{r}$ is the phase of the light at the top of the Earth's atmosphere and δS is the phase after moving through the turbulent

refractive index field. Assume the fluctuations are small, i.e. $|\delta n(\vec{r})| << 1$ and $\delta S << S_0$. Substituting Eqs. 5.29 & 5.30 into Eq. 5.28 gives

$$(\nabla S_0)^2 + 2\nabla S_0 \cdot \nabla \delta S + (\nabla \delta S)^2 = k^2(1 + 2\delta n(\vec{r}) + \delta n^2(\vec{r})). \tag{5.31}$$

Equating first order terms and neglecting second order terms, Eq. 5.31 becomes

$$\nabla S_0 \cdot \nabla \delta S = k^2 \delta n(\vec{r}), \tag{5.32}$$

and from equating first order terms in Eq. 5.31, it can be seen that $\nabla S_0 = k$. Assuming a Cartesian coodinate system where the (x, y)–plane is the plane of the sky and the z-direction is the direction of propagation of the waves, normal to this plane, then Eq. 5.32 can be rewritten as

$$\frac{\mathrm{d}\delta S}{\mathrm{d}z} = k\delta n(z). \tag{5.33}$$

The solution to this equation is then:

$$\delta S(x, y, D) = k \int_0^D \delta n(z)\, \mathrm{d}z, \tag{5.34}$$

where the limits $z = 0$ represents the top of the atmosphere and $z = D$ represents the position of the detector.

Now consider the change in phase at two different points (x_1, y_1), (x_2, y_2) in the sky at the plane $z = D$. The difference between the two changes in phase can be written as

$$\delta S(x_1, y_1, D) - \delta S(x_2, y_2, D) = k \int_0^D \delta n(x_1, y_1, z) - \delta n(x_2, y_2, z)\, \mathrm{d}z. \tag{5.35}$$

Going back to the assumption that the Earth's atmosphere is homogeneous and isotropic on scales between l_0 and L_0, Eq. 5.35 can be squared and averaged to give an expression for the structure function of the phase changes

$$\overline{(\delta S(x_1, y_1, D) - \delta S(x_2, y_2, D))^2} = k^2 \left(\int_{\mathcal{V}} \overline{\delta n(x_1, y_1, z) - \delta n(x_2, y_2, z)}\, \mathrm{d}z \right)^2. \tag{5.36}$$

Using the property of square integrals:

$$\left(\int_a^b f(x)\,dx\right)^2 = \int_a^b f(x)\,dx \times \int_a^b f(y)\,dy = \int_a^b dy \int_a^b dx\, f(x)f(y), \quad (5.37)$$

Eq. 5.36 can be written as

$$\overline{(\delta S(x_1,y_1,D) - \delta S(x_2,y_2,D))^2} = \\ = k^2 \int_0^D \mathrm{d}z_2 \int_0^D \mathrm{d}z_1 \overline{(\delta n(x_1,y_1,z_1) - \delta n(x_2,y_2,z_1))(\delta n(x_1,y_1,z_2) - \delta n(x_2,y_2,z_2))}. \quad (5.38)$$

Using Eq. 5.15, Eq. 5.38 can be rewritten as:

$$\begin{aligned} D_S(\rho) &= \overline{(\delta S(x_1,y_1,D) - \delta S(x_2,y_2,D))^2} \\ &= k^2 \int_0^D \mathrm{d}z_2 \int_0^D \mathrm{d}z_1 \left[D_n\left(\sqrt{\rho^2 + (z_1-z_2)^2}\right) - D_n\left(|z_1 - z_2|\right)\right], \end{aligned} \quad (5.39)$$

where $\rho = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$. A locally homogeneous and isotropic random field's structure function will be an even function so the property of even functions

$$\int_0^D \mathrm{d}z_2 \int_0^D \mathrm{d}z_1 f(z_1 - z_2) = 2\int_0^D (D-z) f(z)\mathrm{d}z, \quad (5.40)$$

can be used to simplify Eq. 5.39

$$D_S(\rho) = 2k^2 \int_0^D (D-z)\left[D_n(\sqrt{z^2+\rho^2}) - D_n(z)\right]\mathrm{d}z. \quad (5.41)$$

The positions and distances considered in the model will be smaller than the outer length scale of the turbulence which in turn is much smaller than the distance travelled by a photon in the Earth's atmosphere i.e. $z, \rho < L_0 << D$. Therefore, $(D-z) \approx D$ and Eq. 5.41 can be written

$$D_S(\rho) \approx 2k^2 D \int_0^D \left[D_n(\sqrt{z^2+\rho^2}) - D_n(z)\right]\mathrm{d}z. \quad (5.42)$$

In addition to being able to describe this field by Eqs. 5.17 & 5.24, an expression can also be written for the random field in two dimensions (i.e. in the plane where

z = const.).

$$n(x,y,z) = n(0,0,z) + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{1-\exp\left[i(k_1x+k_2y)\right]\}\,\mathrm{d}\varphi(k_1,k_2,z), \tag{5.43}$$

where $n(0,0,z)$ is a random function describing the refractive index and $\varphi(k_1,k_2,z)$ follows a relation similar to Eq. 5.11

$$\overline{\mathrm{d}\varphi(k_1,k_2,z)\mathrm{d}\varphi^*(k_1',k_2',z')} = \delta(k_1-k_1')\delta(k_2-k_2')F(k_1,k_2,|z-z'|)\mathrm{d}k_1\mathrm{d}k_2\mathrm{d}k_1'\mathrm{d}k_2', \tag{5.44}$$

where F here is the two-dimensional spectral density of the random field. Following Eq. 5.18 but expanding it to the two-dimensional case and using Eq. 5.15:

$$D_n(\sqrt{z^2+\rho^2}) - D_n(z) = 2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{1-\cos\left[k_1(x_1-x_2)+k_2(y_1-y_2)\right]\}\times \times F(k_1,k_2,|z_1-z_2|)\mathrm{d}k_1\mathrm{d}k_2. \tag{5.45}$$

Given that the difference in refractive index structure functions can also be given by Eq. 5.23, the expression for $F(k_1,k_2,|z-z'|)$ can be obtained

$$F(k_1,k_2,z) = \int_{-\infty}^{\infty}\cos(k_3z)\Phi(k_1,k_2,k_3)\,\mathrm{d}k_3. \tag{5.46}$$

Equations. 5.45 & 5.46 can then be substituted into Eq. 5.42 to give two give two equivalent expressions for $D_S(\rho)$

$$D_S(\rho) = 2\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[1-\cos(k_1(x_1-x_2)+k_2(y_1-y_2))\right]F_S(k_1,k_2,z)\,\mathrm{d}k_1\,\mathrm{d}k_2, \tag{5.47}$$

$$D_S(\rho) = 2\pi k^2 D\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\{1-\cos[k_1(x_1-x_2)+k_2(y_1-y_2)]\}\,\Phi_n(k_1,k_2,z)\,\mathrm{d}k_1\,\mathrm{d}k_2, \tag{5.48}$$

where F_S is the two dimensional spectral density of the phase fluctuations and Φ_n is the three dimensional spectral density of the refractive index fluctuations. It can then be posited that

$$F_S(k_1,k_2,z) = 2\pi k^2 D\Phi_n(k_1,k_2,z), \tag{5.49}$$

and in a locally isotropic field this reduces to

$$F_S(\kappa, z) = 2\pi k^2 D \Phi_n(\kappa, z), \tag{5.50}$$

where $\kappa = \sqrt{\kappa_1^2 + \kappa_2^2}$.

Following Tatarski (2016) Chap. 3 "Microstructure of the Concentration of a Conservative Passive Additive in a Turbulent Flow" and originally reported by Obukhov (1970), the structure function of the refractive index for the scales considered here can be written using the "two-thirds law"

$$D_n(z) = C_n^2 z^{2/3}. \tag{5.51}$$

Using Eq. 5.51 to solve Eq. 5.24, an expression for the three dimensional spectral density of the refractive index field can be found

$$\Phi_n(\kappa, z) = 0.033 C_n^2 \kappa^{-11/3}, \tag{5.52}$$

where C_n^2 is a constant describing the model of the Earth's atmosphere used. F_S can then be found for this model using Eq. 5.50

$$F_S(\kappa, z) = 0.21 k^2 D C_n^2 \kappa^{-11/3}. \tag{5.53}$$

Eq. 5.47 can be rewritten in terms of κ:

$$D_S(\rho) = 4\pi(0.21) k^2 D C_n^2 \int_0^\infty \left[1 - J_0(\kappa\rho)\right] \kappa^{-8/3} \, \mathrm{d}\kappa, \tag{5.54}$$

where J_0 is the zeroth order Bessel function of the first kind. This gives a solution for $D_S(\rho)$ as

$$D_S(\rho) = 2.91 k^2 D C_n^2 \rho^{5/3}, \tag{5.55}$$

using the identity (Tatarski, 2016):

$$\int_0^\infty \left[1 - J_0(ax)\right] x^{-p} dx = \pi a^{p-1} \left\{ 2^p \left[\Gamma\left(\frac{p+1}{2} \right) \right]^2 \sin\left(\frac{\pi(p-1)}{2} \right) \right\}^{-1}, \tag{5.56}$$

where $1 < p < 3$ and Γ represents the Gamma function. Thus, an expression has been derived to measure the structure function of the phase fluctuations of photons

travelling through the Earth's atmosphere assuming they arrive as plane parallel waves from astronomical sources within regions smaller than the outer length scale of the turbulence L_0.

This model would work perfectly if an observer could look through exactly one eddy of size $< L_0$ but unfortunately this cannot be the case. Therefore, the previously assumed constant C_n^2 is constant only within an individual eddy. If the photon travels through more than one eddy then the contributions of each of those eddies must be summed to give the change in the phase of the photon. This is what was referred to earlier as a medium with *smoothly-varying turbulence*. On length scales of L_0 the turbulence changes according to some atmospheric model encompassed in the C_n^2 term in Eq. 5.55. Modifying the model is relatively easy by making C_n^2 a function of the altitude meaning Eqs. 5.51 & 5.52 become:

$$D_n(z) = C_n^2(z) z^{2/3}, \tag{5.57}$$

$$\Phi_n(\kappa, z) = 0.033 C_n^2(z) \kappa^{-11/3}. \tag{5.58}$$

This results in the solution to Eq. 5.42 being:

$$D_S(\rho) = 2.91 k^2 \rho^{5/3} \int_{\text{path}} C_n^2(\vec{r})\, \mathrm{d}\vec{r}, \tag{5.59}$$

which is Eq. 5.6 quoted above as the model for the phase fluctuations. For a photon incident to the plane at the top of the Earth's atmosphere, $z = 0$, at arbitrary angle of incidence θ, the Eq. 5.59 can be simplified by writing the C_n^2 profile as

$$\int_{\text{path}} C_n^2(\vec{r})\, \mathrm{d}\vec{r} = \sec\theta \int_0^D C_n^2(z)\, \mathrm{d}z. \tag{5.60}$$

Equation 5.59 can then be rewritten as

$$D_{\mathrm{S}}(\rho) = 2.91 k^2 \rho^{5/3} \sec\theta \int_0^D C_n^2(z)\, \mathrm{d}z. \tag{5.61}$$

To simplify Eq. 5.61 further, a characteristic scale length for the turbulence is introduced known as the Fried parameter (Fried, 1966). Also known as the Fried coherence length (or just the coherence length), the Fried parameter gives a measure of the length scales over which the distorted light wave can still be assumed to

be a plane wave. That is, it gives an estimate of the length scales over which random turbulence inhomogeneities will affect the wavefront. Empirically, the Fried parameter was shown to be related to the turbulence structure in the atmosphere via

$$r_0 = \left(0.423k^2 \sec\theta \int_0^D C_n^2(z)\, \mathrm{d}z\right)^{-3/5}, \tag{5.62}$$

which can be used to simplify Eq. 5.61 to

$$D_{\mathrm{S}}(\rho) = 6.88\left(\frac{\rho}{r_0}\right)^{5/3} = 6.88\left(\frac{\lambda\nu}{2\pi r_0}\right)^{5/3}, \tag{5.63}$$

where λ is the air wavelength of the photon and ν is the spatial frequency expressed in units of radians of phase per radian field of view (Racine, 1996). Equation 5.63 is used when emulating synthetic seeing as it provides a model independent of the atmospheric model used (C_n^2) with r_0 being the free parameter defining the system. A variety of different, typical values for r_0 are used when generating the training dataset (see Sec. 5.3) to estimate realistic data marred by seeing.

r_0 has another important property: the Fried parameter gives the size of the effective aperture that the observation is taken through. That is, regardless of telescope aperture size, for turbulence characterised by a value r_0, the observations will appear as if taken through a telescope with aperture size r_0. Thus the angular size of the atmospheric PSF can be found using

$$\alpha = 2.021 \times 10^5 \times \frac{\lambda}{r_0}, \tag{5.64}$$

where α is given in arcseconds. The size of the PSF in detector pixels (n_{pix}) can then be calculated by dividing the angular size of a single pixel α_{pix}

$$n_{\mathrm{pix}} = \frac{\alpha}{\alpha_{\mathrm{pix}}}. \tag{5.65}$$

n_{pix} is then the size of the PSF array to be convolved with the image. This PSF is populated using Eqs. 5.5 & 5.63.

5.3 Construction of Training Data

The model developed in Sec. 5.2 depends on two main parameters: the wavelength of the light observed and the Fried parameter of the atmosphere. Defining these two parameters allows the angular size of the PSF in the sky to be defined by Eq. 5.64 and allows for the calculation of the structure function by Eq. 5.63 (and via Eq. 5.5 the PSF) at a distance ρ from the centre of the PSF kernel. Once a PSF kernel is constructed for a set of parameters (r_0, λ), it is convolved following Eq. 5.4 with data observed at the same wavelength that were taken in good seeing conditions such that the assumption can be made that the data before artificial seeing is imprinted on them is diffraction limited. This gives a dataset comprising data without bad seeing and data imprinted with bad seeing from the model. The DNN described in Sec. 5.4 is then trained with the bad seeing data as input and the images with good seeing as output.

The data used for training the network is imaging spectroscopy from three different flares observed using SST/CRISP in two different optical lines: hydrogen-α (Hα) and the infrared triplet line of singly-ionised calcium observed at λ = 8542Å (Ca II λ8542). Each of these sets of observations have a pixel size of 0.057″ resulting in a theoretical spatial resolution of 0.114″. However, the diffraction limit for both Hα and Ca II λ8542 for the SST (0.162″ and 0.211″, respectively) are greater than the theoretical spatial resolution meaning that the observations are diffraction-limited. The three datasets used are:

1. The M1.1 solar flare SOL20140906T17:09, which took place in NOAA active region (AR) 12157 with helioprojective coordinates (-732″, -302″). In these observations, the Hα line was sampled at 15 different wavelengths and the Ca II λ8542 line was sampled at 25 different wavelengths. Both lines are sampled uniformly through a 1.2Å bandpass filter with Hα being sampled every 200mÅ and Ca II λ8542 being sampled every 100mÅ. The cadence of these observations is 11.54 seconds.

2. The X2.2 solar flare SOL20170906T09:10, which took place in NOAA AR 12673 with helioprojective coordinates (537″, -222″). Here the Hα line was sampled at 13 wavelengths non-uniformly over a passband of 1.2Å being more densely sampled in the core than the wings. Similarly, the Ca II λ8542 line was sampled at 11 wavelengths through the same passband in the same way. The cadence

of these observations is 15 seconds.

3. Lastly, the X9.3 solar flare SOL20170906T12:02, which took place in the same NOAA AR as SOL20170906T09:10 and was the most energetic solar flare of solar cycle 24. The two spectral lines were observed identically to the observations of SOL20170906T09:10 and the cadence remained unchanged.

All data has been pre-processed using the CRISPRED data reduction pipeline (de la Cruz Rodríguez et al., 2015) that includes all alignment, instrument calibration and image restoration using MOMFBD. Therefore, the ground truth to be recovered makes the assumption that images without bad seeing are completely corrected for seeing and other aberrations by the CRISPRED pipeline.

Given that the properties of the atmospheric PSF are wavelength-dependent, different PSFs should be calculated for the two different lines under consideration here. This is due to the sizes of the atmospheric PSF kernels being used varying largely between the two spectral lines for the same atmospheric conditions i.e. since $r_0 \propto \lambda^{6/5}$, $\alpha \propto \lambda^{-1/5}$ and $\alpha_{\mathrm{H}\alpha}/\alpha_{\mathrm{Ca\,II}} \approx 1.054$. However, the size of the kernel does not change substantially across the line leading to the same kernel being used per passband. This means that there are two models (one for each spectral line) and so two DNNs with the same architecture (see Sec. 5.4) are trained to approximate separately the corrections in each spectral line. The use of two different DNNs and how they may be consolidated into one are discussed in Sec. 5.6.

A range of Fried parameters $r_0 = \{1, 2.5, 5, 7.5, 10, 12.5, 15\}$ cm is used to generate many different PSFs to convolve with the good seeing images following Eq. 5.4. This creates a diverse training data set for the neural network to learn from.

Figures 5.1–5.6 show examples of the training data for Hα. Figures 5.1 & 5.2 show how the seeing models for a variety of Fried parameters ($r_0 = \{5, 10, 15\}$ cm) affect the M1.1 solar flare Hα observations, particularly an observation from approximately 19 minutes before the GOES SXR peak. Figure 5.1 focuses on the line core images and the effects of the seeing model on the spectral line as a whole. Panels (a), (b), (c) & (d) show the ground truth and models with $r_0 = 5$ cm, $r_0 = 10$ cm and $r_0 = 15$ cm, respectively. This highlights the smearing effect that seeing has over a field of view, as the models with a smaller r_0 to have their light spread more across the image. This is complemented by panel (h) which shows the azimuthally-averaged (also known as radially-averaged) power spectrum for each of the images. The power spectrum gives information of the strengths of different features at different spatial

scales. This is characterised by taking the square of the Fourier transform of the image in question

$$P(u,v) = |\mathcal{F}\left(O(x,y)\right)|^2, \tag{5.66}$$

where the (u,v) spatial frequency plane is the Fourier pair to the (x,y) spatial plane and $\mathcal{F}$ represents the Fourier transform. Then starting from the centre of the image and defining annuli of thickness d, the power spectrum within a radius ν_d can be calculated

$$P(\nu_d) = \frac{1}{N_d} \sum_{i=0}^{N_d-1} P(\nu_{d,i}), \tag{5.67}$$

where $\nu_d = \sqrt{u_d^2 + v_d^2}$ is the radius of an annulus centred on the image centre in Fourier space and N_d is the number of power spectra within the annulus. Eq. 5.67 gives the azimuthally-averaged power spectrum. Lower spatial frequencies represent larger spatial scales which describe the larger scale structures in the image, with the opposite being true for the higher spatial frequencies. A large value of the power spectrum at a spatial frequency will point to a strong signal at that spatial scale indicating defined features on these scales. The higher the spatial frequency, the lower the spatial scale meaning a higher power here represents that small scale features are well-represented in an image i.e. the images are sharper. Going back to the effects of seeing on observations and looking at the effects of the seeing models on the sharp observation's power spectrum in Fig. 5.1(h), it is apparent that the seeing models have an impact at the higher end of the spatial frequency scales with each of the models losing a lot of the highest resolution information. In Figs. 5.1–5.12, the ground truth is represented by circles, with the r_0 = 5 cm model being represented by triangles, r_0 = 10 cm represented by squares and r_0 = 15 cm represented by pentagons. Therefore, looking at panel (h), high resolution information is lost in all three seeing models with lower spatial frequencies being impacted more the worse the seeing becomes. This is what is expected of the seeing model as the main effect of seeing is the apparent effect of observing through a smaller aperture equal to r_0.

Furthermore, panels (e) & (g) show the change to spectral lines caused by the seeing models. The two spectral lines shown are indicated by the cross and the plus markers in panels (a)–(d). A point near the flare ribbon and away from the flare ribbon are chosen and shown in panels (e) & (g) respectively. These spectra exhibit the kind of behaviour that is expected in response to the seeing models: for the point

near the flare ribbon, the intensities at different wavelengths are slightly increased (moreso for worse seeing conditions) which is to be expected due to the light from the flare ribbons being spread into these darker pixels. Moreover, the spectrum for the quieter part of the atmosphere is not changed significantly by the seeing models which is also expected due to there being smaller gradients in intensity around these areas. The intensities for data affected by seeing are increased slightly which could be down to the averaging-like effect of the seeing PSF.

Finally, panel (f) shows a spatial slice through each of these images which displays the intensity as a function of distance along the y-axis. This shows the smoothing effect of the seeing models well as, overall, the brighter points appear fainter and the dark points appear brighter due to the spatial spreading of the light.

Figure 5.2 shows the application of the three seeing models listed above to the same observation as Fig. 5.1 but this time showing the blue and red wing images ($\Delta\lambda = \mp 600$mÅ, top and bottom row, respectively). In the last column, there is the azimuthally-averaged power spectrum for each of the images to show how the image is affected by the seeing models. The points in the last column retain their same meaning from Fig. 5.1. As with the line core images, high resolution information is lost in all seeing models with lower resolution information lost in the seeing models with smaller r_0.

Figures 5.3 & 5.4 show how the three seeing models discussed for Figs. 5.1 & 5.2 affect the Hα observations of the X2.2 solar flare. In this case, an observation 1.5 minutes after the GOES SXR peak is examined. The panels in Fig. 5.3 are equivalent to those in Fig. 5.1 (similarly for Fig. 5.4 and Fig. 5.2). The spectral line near the flare ribbon (panel (e)) shows a great enhancement due to the spreading of the flare ribbon light to surrounding pixels and there is a larger increase than in the previous flare as this flare is about an order of magnitude more energetic (also, because this observation is after the main reconnection event whereas the observations looked at in Figs. 5.1 & 5.2 were before). The spectral lines, spatial distributions and power spectra shown in Figs. 5.3 & 5.4 exhibit the same effects as described for the M1.1 flare above. Similarly, Figs. 5.5 & 5.6 present data in the same way from the X9.3 event.

Due to its longer wavelength, the size of the atmospheric PSFs for the Ca II λ8542 line will be larger for the same Fried parameter meaning that the effect of each seeing model will be worse. This is shown in Figs. 5.7–5.12 where these figures are cotemporal with Figs. 5.1–5.5 but showcasing the Ca II λ8542 observations. The

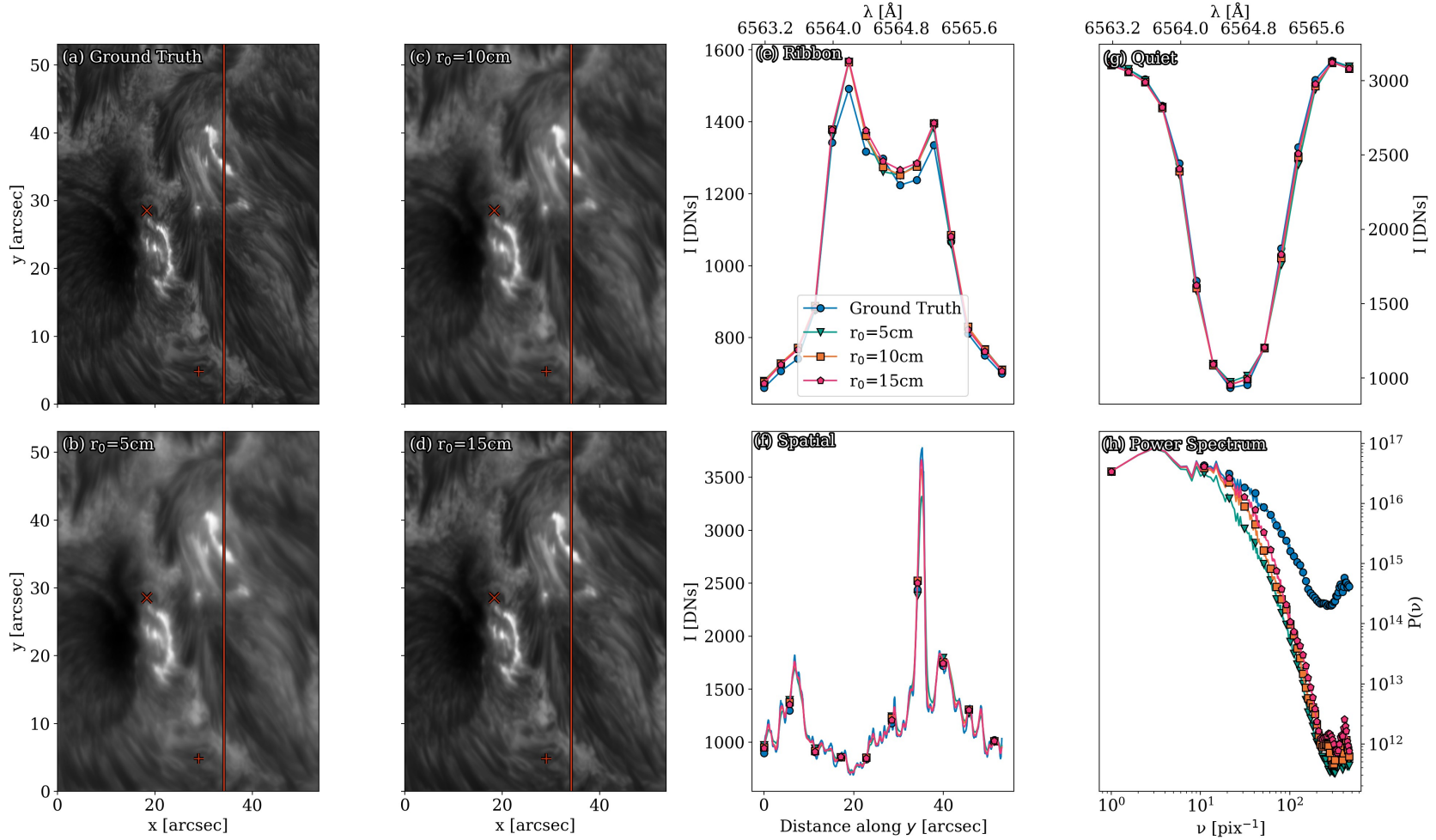

Figure 5.1: Example of the seeing model described in Sec. 5.2 applied to one of the good seeing observations from the M1.1 solar flare dataset described in Sec. 5.3. This observation is from 16:50:57UTC approximately 19 minutes before the flare soft X-ray peak, shown in panel (a). The image used here for demonstration is of the Hα line centre. The seeing model is applied for three different Fried parameters as can be seen in (b) r_0 = 5 cm, (c) r_0 = 10 cm, and (d) r_0 = 15 cm. (e), (f), and (g) show the change in the spectral line on the flare ribbon, a spatial slice, and the spectral line in a quieter part of the atmosphere, respectively. The flare ribbon line is indicated by the cross in panels (a)–(d) with the quiet point being the plus sign and the slice shown by the vertical line. (h) shows the azimuthally averaged power spectrum across the images. In panels (e), (f), (g), and (h), the circles correspond to the ground truth, the triangles to r_0 = 5 cm, the squares to r_0 = 10 cm, and the pentagons to r_0 = 15 cm.

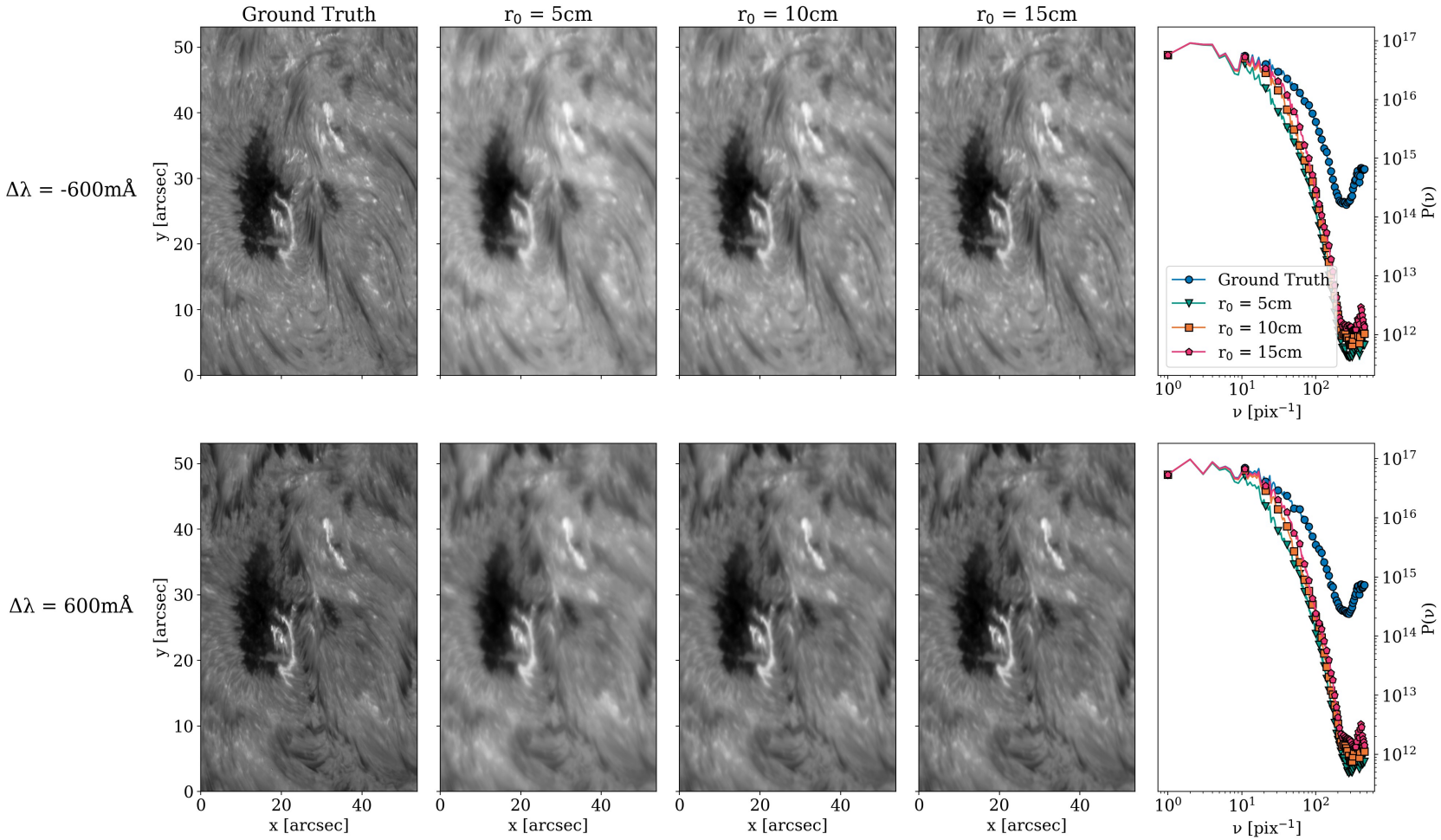

Figure 5.2: Data from the same observation as Fig. 5.1 but this time showing the blue and red line wing observations (top and bottom row, respectively) and their changes due to the synthetic seeing. In Fig. 5.1(e), there is a clear change in the intensity of the wings of the spectral line indicating that the seeing has an effect across the whole spectral line. In this figure, the first column represents the ground truth for both wings, the second shows the wings affected by seeing characterised by r_0 = 5 cm, the third r_0 = 10 cm and the fourth r_0 = 15 cm. The last column shows the azimuthally-averaged power spectra over the images with the symbols retaining their definition from Fig. 5.1. The power spectra in both wings show a loss of high frequency information consistent with the line centre observations and what is to be expected by the effects of seeing. The blue and red wing observations shown are taken at $\Delta\lambda = \mp 600$mÅ, respectively.

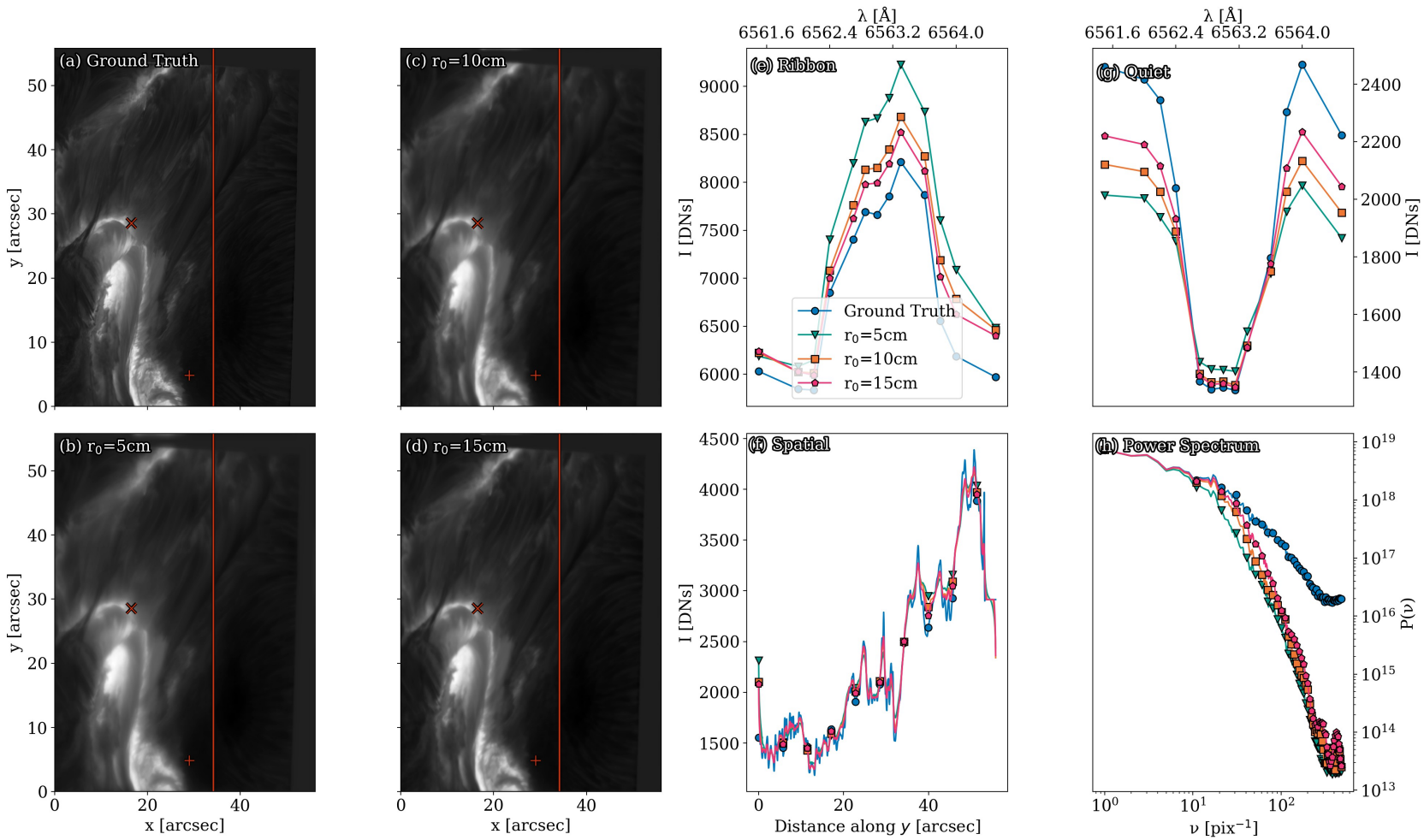

Figure 5.3: An equivalent figure to Fig. 5.1 (everything retains their meaning) but for the line centre observations of the X2.2 flare SOL20170906T09:10. The observation used is from approximately 1.5 minutes after the SXR peak.

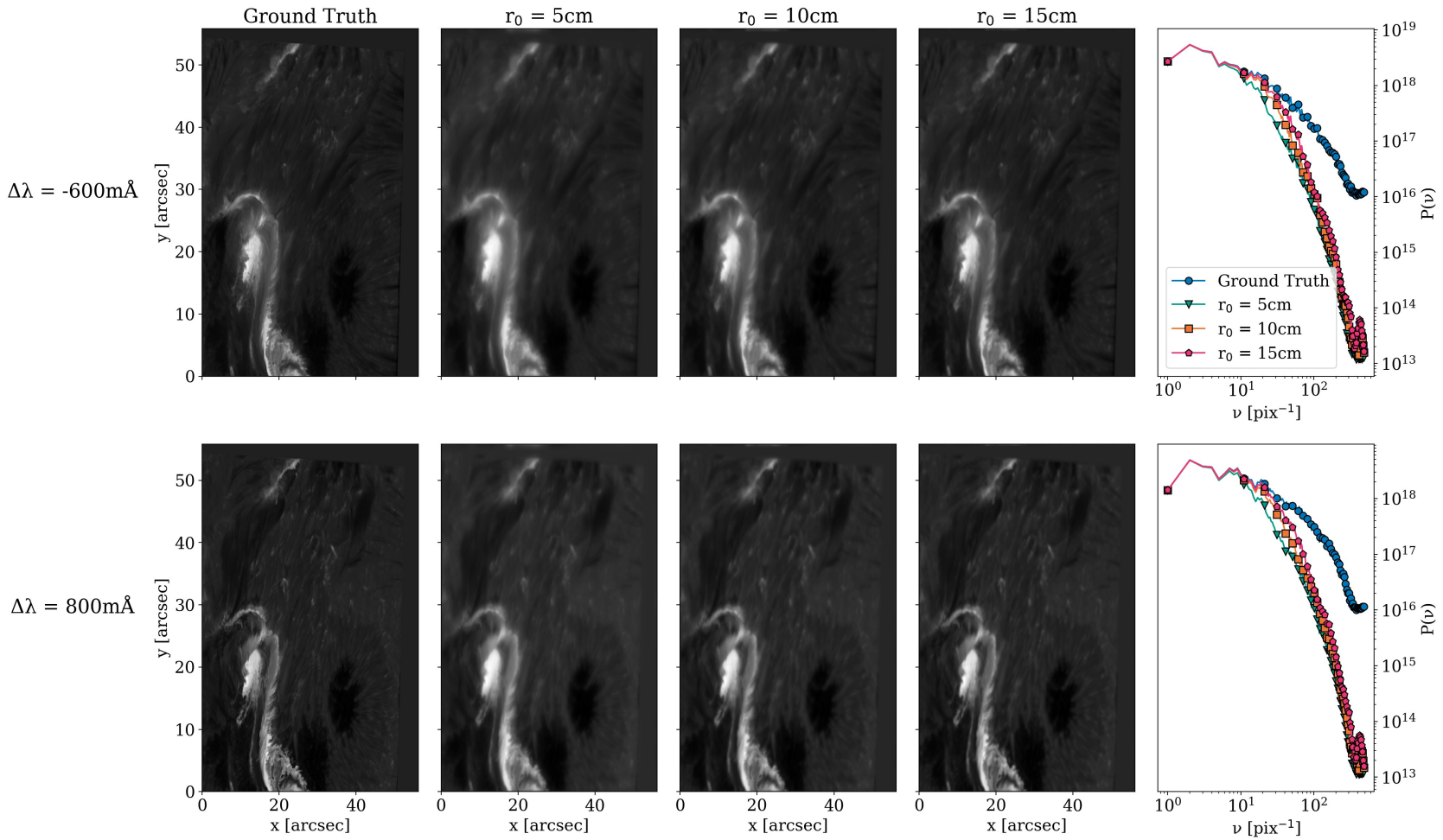

Figure 5.4: An equivalent figure to Fig. 5.2 but for the line wing observations of the flare from Fig. 5.3. The blue wing observation shown here is at $\Delta\lambda = -600$mÅ with the red wing observation being at $\Delta\lambda = 800$mÅ.

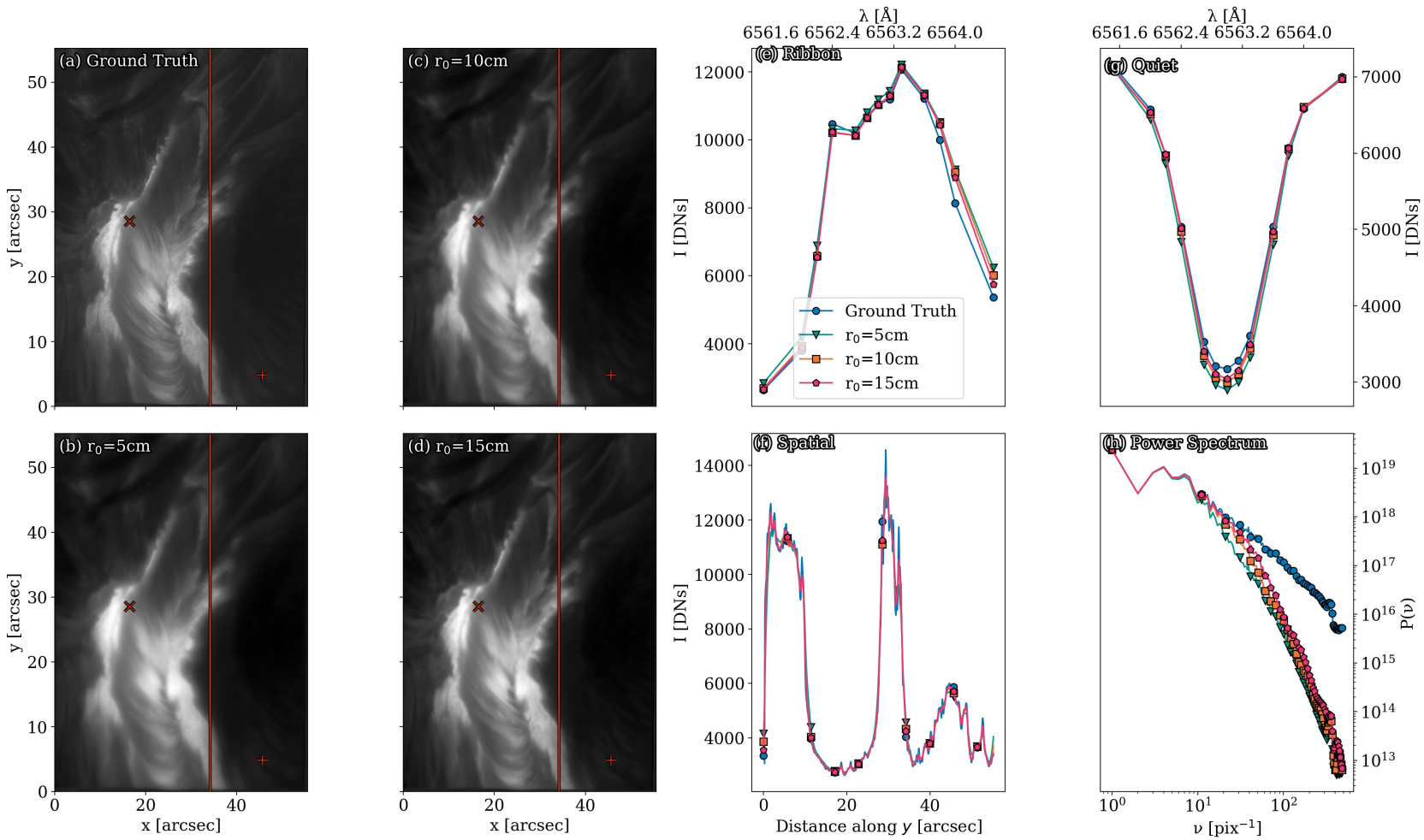

Figure 5.5: An equivalent figure to Figs. 5.1 & 5.3 (everything retains their meaning) but for the line centre observations of the X9.3 flare SOL20170906T12:02. The observation used is from approximately 9 minutes after the SXR peak.

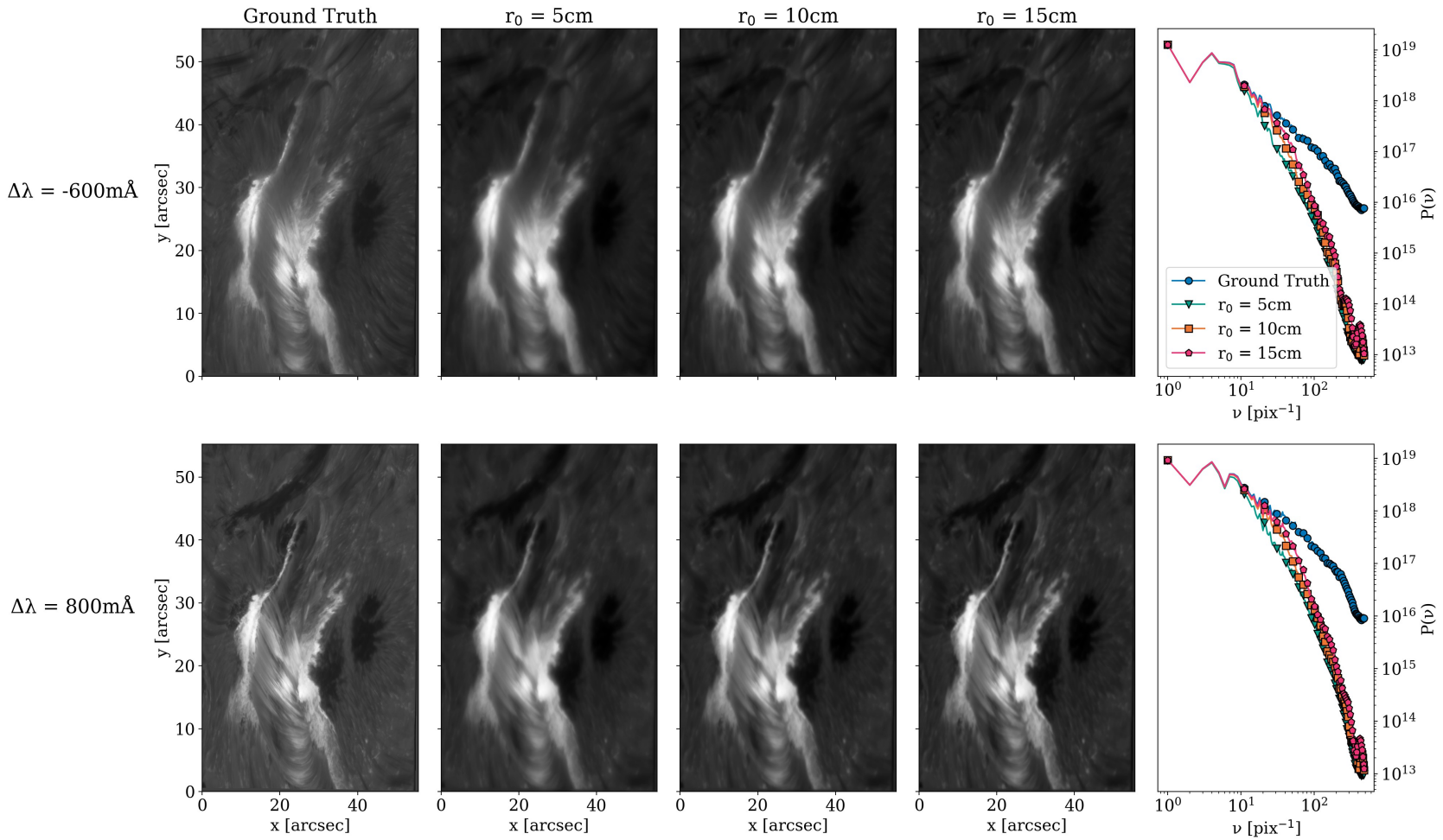

Figure 5.6: An equivalent figure to Figs. 5.2 & 5.4 but for the line wing observations of the flare from Fig. 5.5. The blue wing observation shown here is at $\Delta\lambda = -600$mÅ with the red wing observation being at $\Delta\lambda = 800$mÅ.

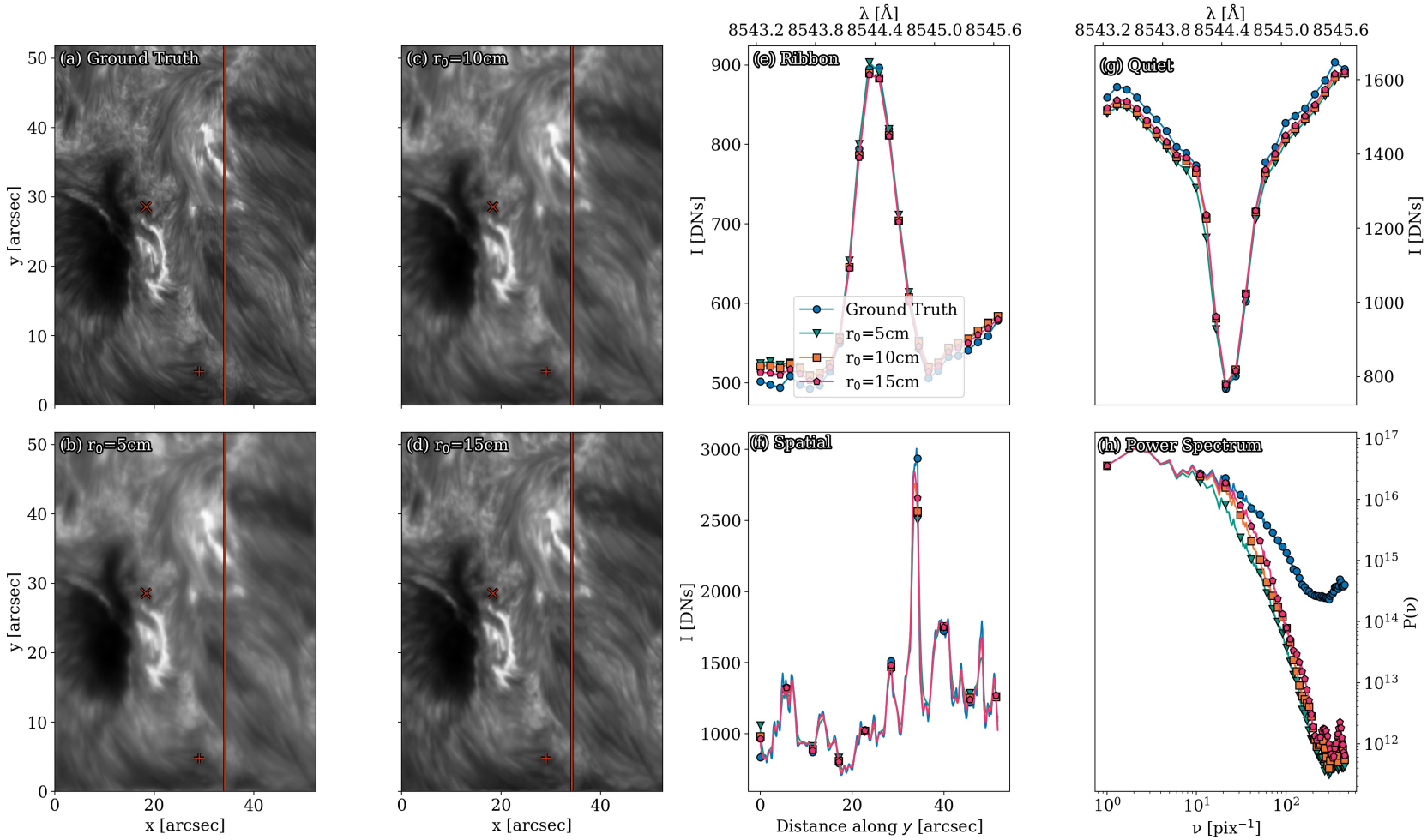

Figure 5.7: Same figure layout as for Figs. 5.1, 5.3 & 5.5 with everything retaining the same meanings. This time, though, the observations shown are the line centre of the Ca II λ8542 line. The observation used here is cotemporal with the Hα observation in Fig. 5.1.

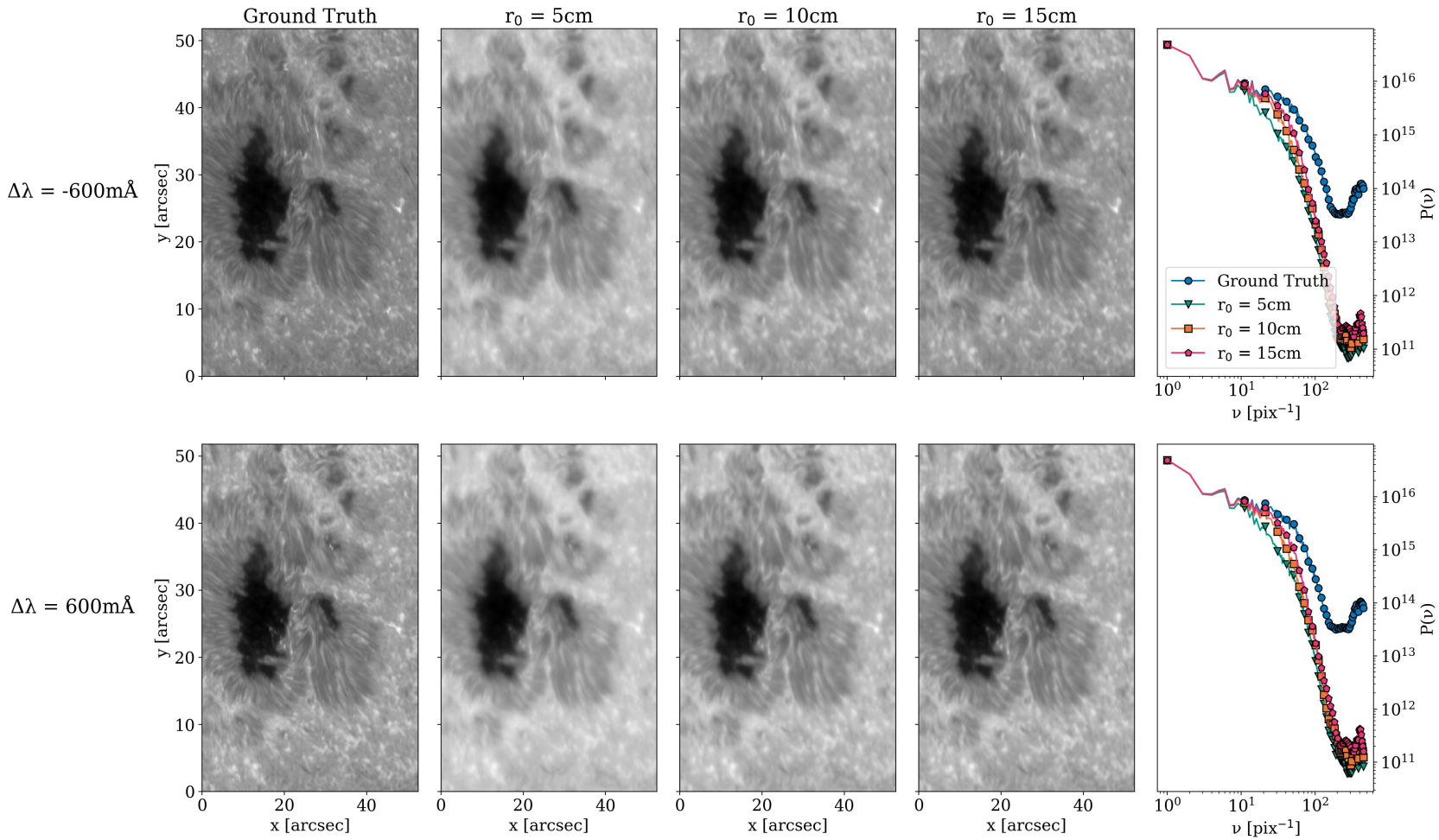

Figure 5.8: Same figure layout as for Figs. 5.2, 5.4 & 5.6 showing the complimentary line wing observations to Fig. 5.7. The blue and red wings are observed at $\Delta\lambda = \mp 600$mÅ, respectively.

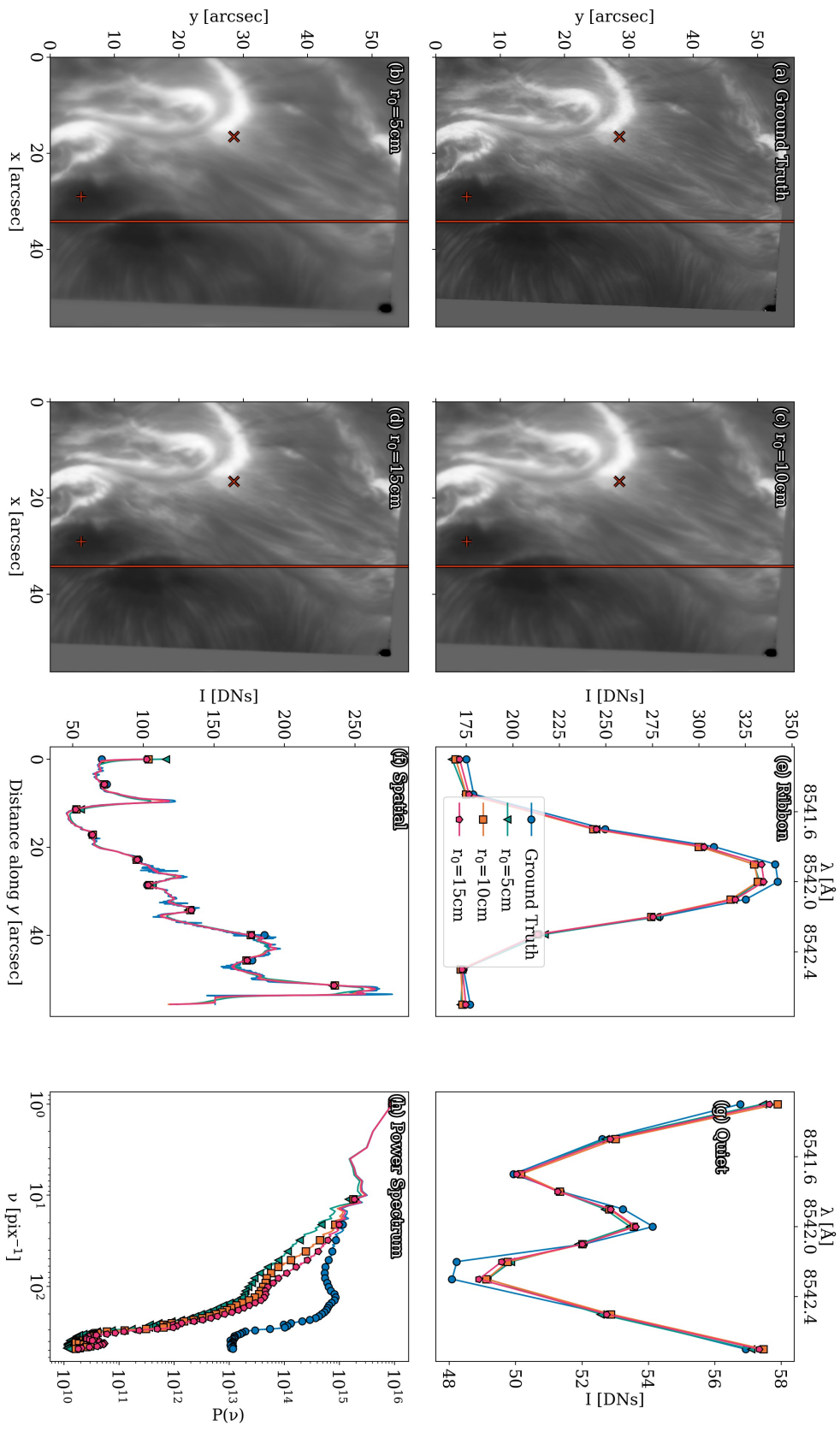

Figure 5.9: An equivalent figure to Fig. 5.7 where the observations shown here are cotemporal with the observations shown in Fig. 5.3.

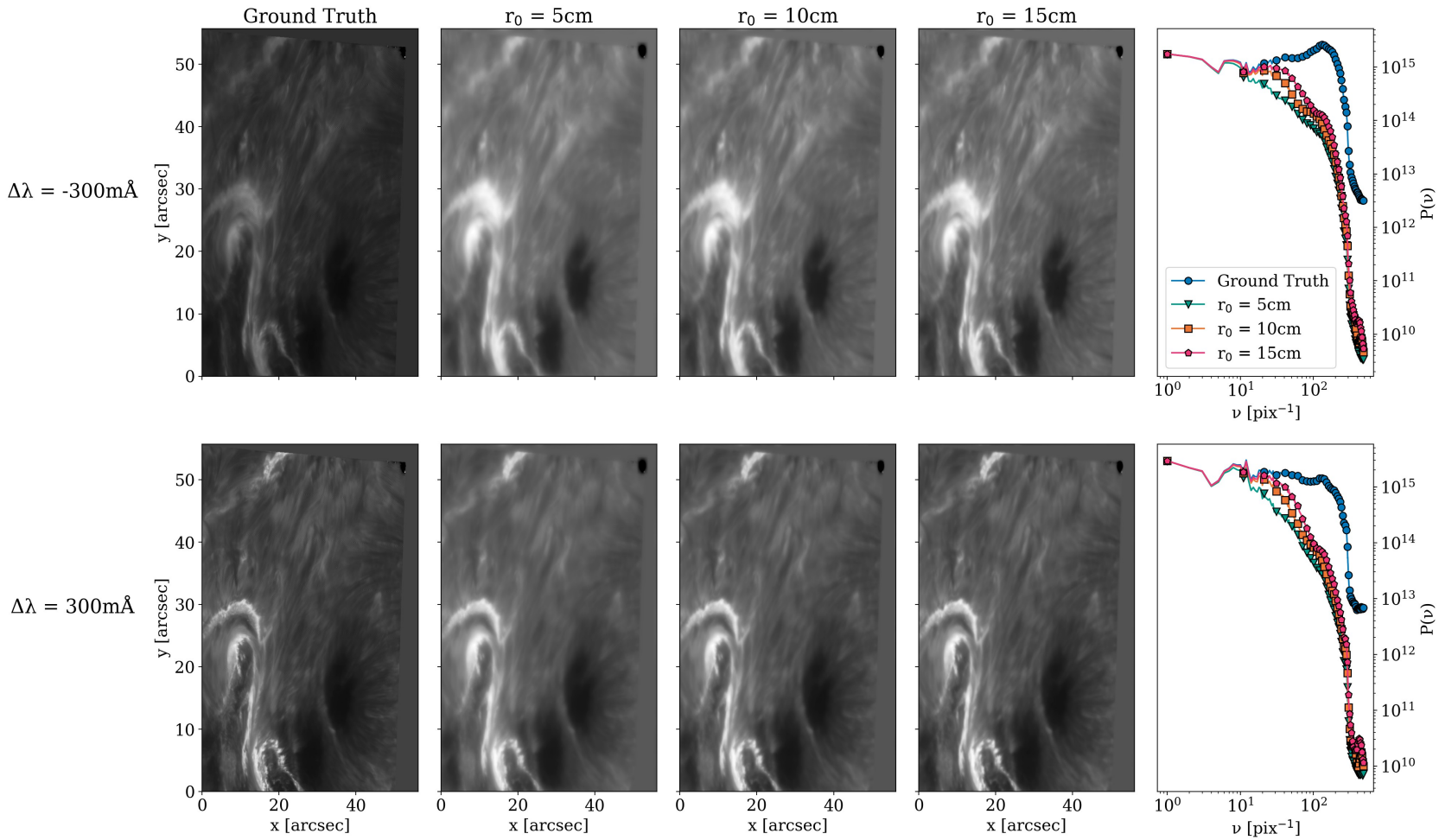

Figure 5.10: An equivalent figure to Fig. 5.8 but in this figure, the complimentary wing images are from the data used in Fig. 5.9.

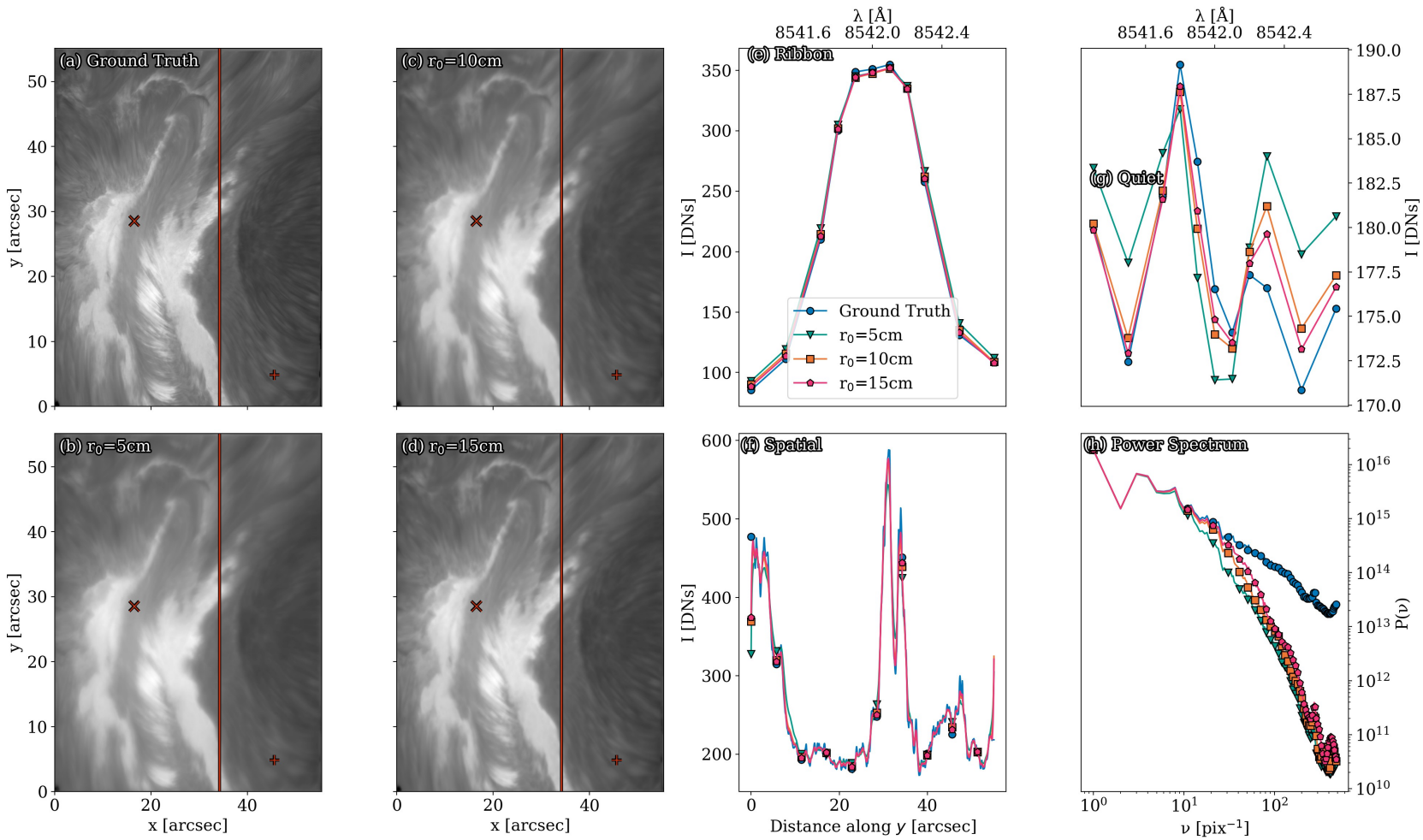

Figure 5.11: An equivalent figure to Figs. 5.7 & 5.9 where the observations shown here are cotemporal with the observations shown in Fig. 5.5.

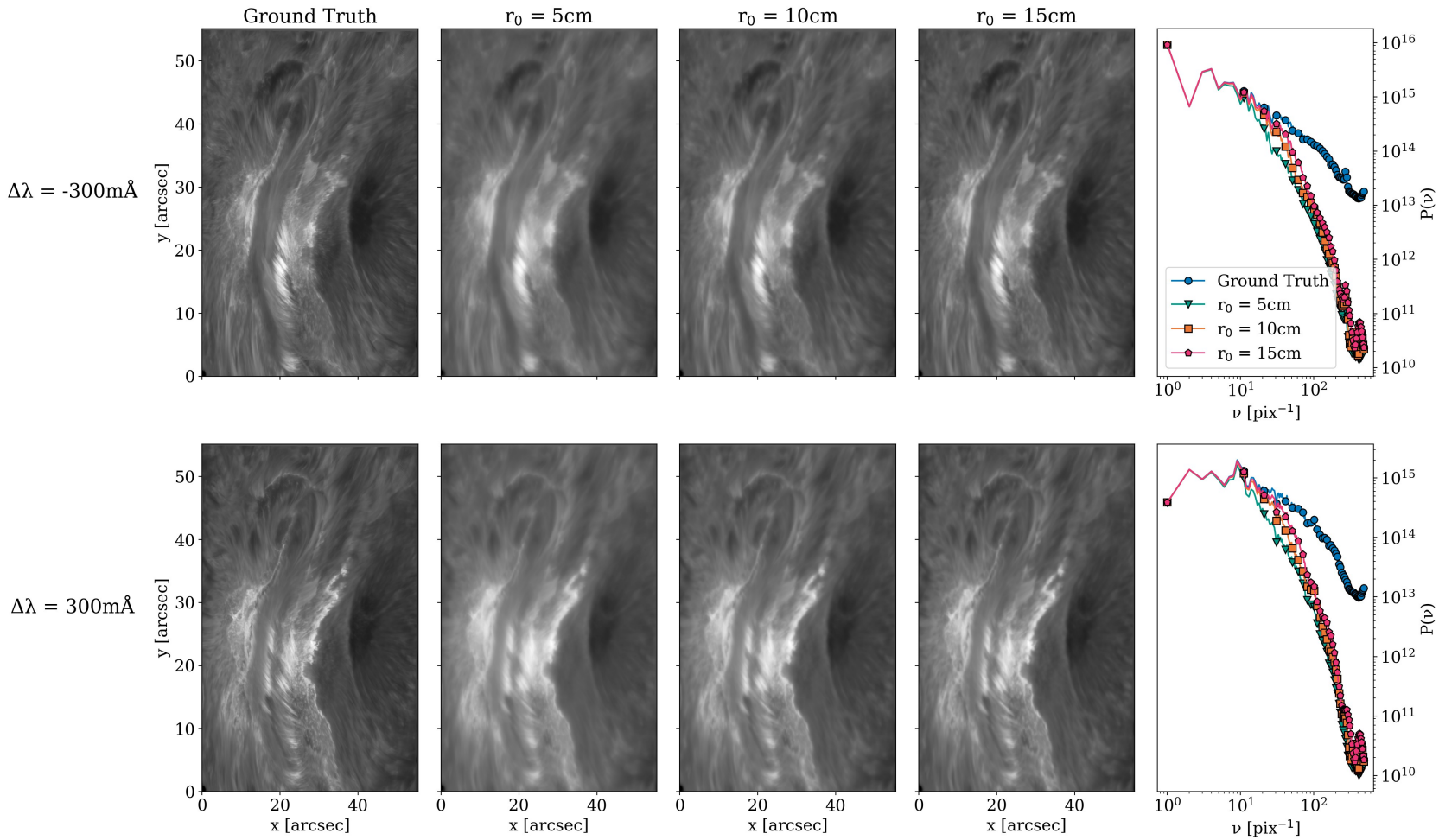

Figure 5.12: An equivalent figure to Figs. 5.8 & 5.10 but in this figure, the complimentary wing images are from the data used in Fig. 5.11.

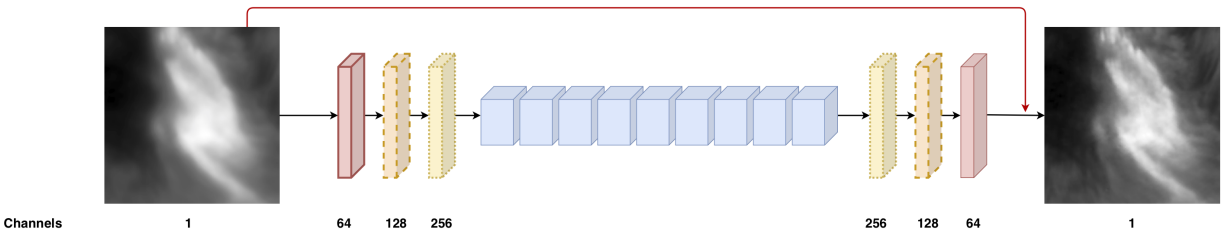

Figure 5.13: Schematic of the DNN used to learn seeing correction. This network consists of six convolutional layers and nine residual layers The picture on the left of the network shows the input that is an image from the data set generated in Sec. 5.3 of an image marred with synthetic seeing. The picture on the right is the ground truth the network is trying to recover. The first block (with the solid lines) is a convolutional layer using a 7×7 kernel and generating 64 feature maps. The block with the dashed lines downsamples the feature maps produced by the first block by a factor of 2 using a strided convolution of 3×3 kernel and produces 128 feature maps. The dotted line block downsamples the feature maps by a further factor of 2 using a strided convolution of 3×3 kernel and produces 256 feature maps. The shorter blocks in the middle are the residual layers that all consist of 3×3 kernel convolutions with 256 feature maps. The inner structure of the residual layers is shown in Fig. 5.14. The next dotted line block upsamples the feature maps by a factor of 2 using nearest neighbour interpolation and reduces the number of feature maps to 128. The second dashed line block then upsamples by a further factor of 2 using the same method while reducing the number of feature maps to 64. The last block in the network is a convolutional block that reduces the number of feature maps to the number of output channels using a 7×7 kernel convolution before passing the output through a hyperbolic tangent (tanh) function. This is then combined with the input to the network (red arrow) to produce the output of the network. In each of the convolutional and residual layers, the normalisation is batch normalisation and the activation is ReLU.

same spreading and smearing effects are observed when applying the seeing models to the Ca II λ8542 data.

5.4 Construction of Neural Network

The DNN architecture used is illustrated in Fig. 5.13 and inspired by the generator network used in Kupyn et al. (2017). The network follows an encoder-decoder framework wherein the input data – in this case, the images with bad seeing, divided into 256×256 pixel segments – are downsampled in the spatial dimensions to a lower dimensional, abstract representation of itself while increasing the number of feature maps which can then be reconstructed without the bad seeing by the learned network; by upsampling the representation at the other end of the network. This is accomplished using a combination of convolutional layers and residual lay-

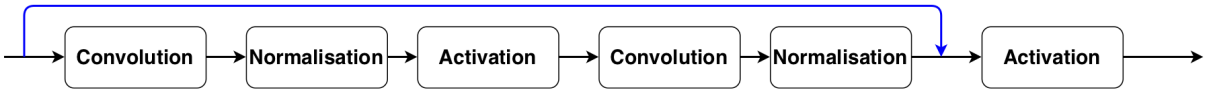

Figure 5.14: Inside a residual layer (short boxes in Fig. 5.13): two convolution layers applied to the input like traditional convolutional neural networks but with a skip connection (blue arrow) adding the input of the layer to the output before the second activation. This allows residual networks to be deeper than traditional networks as it prolongs the onset of the vanishing gradient problem.

ers (Fig. 5.14, He et al., 2015a). Residual neural network layers are an ingenious solution to the *degradation problem* in DNNs. This is when networks that are too deep (have too many layers) fail to learn the problem at hand. DNNs with too many layers[3] apply so many functions to the data that by the time the network needs to construct the output, there is no clear link between what the input was and what the output should be. This causes a stagnation in training. Residual layers combat this problem by not learning a specific function per layer but rather the residual to a function. For example, consider an arbitrary DNN layer looking to approximate some function $f(x)$. For input x, the *residual* of this function, $h(x)$ can be given by:

$$h(x) = f(x) - x. \tag{5.68}$$

The goal of the residual layer (like any other layer or DNN) is to approximate the function $f(x)$, therefore, the learnable parameters within a residual layer are used to approximate $h(x)$ and then $f(x)$ is calculated by $h(x) + x$ by definition of the residual. What this means conceptually is that the residual layer is shown what the input looks like after it has performed the transformations to it. This constrains the layer to learning the residual which was shown by He et al. (2015a) to improve how deep DNNs can be. How the residual layer works is illustrated in Fig. 5.14. A residual layer is two convolutional layers stacked with an extra connection known as a *skip connection* before the second activation which adds the input to the output of the computations. This can be thought of as "reminding" the layer what it originally took as an input and making sure the solution to learning the function remains on the right track. In this problem, residual layers aim to learn the complexities of the downsampled abstract representation of data with bad seeing and transform this

[3]The number of layers before arriving at the degradation problem is a hyperparameter and dependent on the problem and the DNN architecture. However, He et al. (2015a) found for CNNs that anything above 20 layers suffered from the degradation problem.

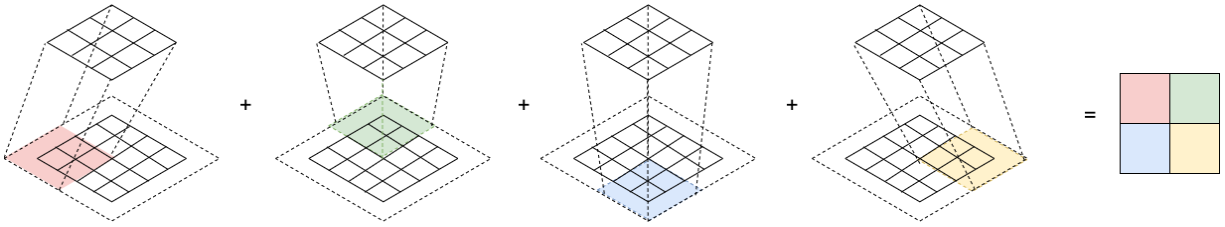

Figure 5.15: Example of a strided convolution used for downsampling in DNNs. The input to the convolution layer is the 4×4 solid grid with the 3×3 solid grid representing the convolutional kernel. This differs from Fig. 2.3 in that the stride is set to two meaning that the centre of the kernel moves two places between computations. This results in a 2×2 output shown on the right as there are only four locations where the kernel will stop. The dotted box indicates the padding applied to the input so as not to reduce the dimensionality due to the size of the kernel but due to the value of the stride.

into an abstract representation that can be upsampled to produce data corrected for bad seeing. Nine residual layers were found to be the optimal number to learn this problem.

In total there are six convolutional layers: three before the set of residual layers used to downsample and transform the input and three after to upsample the corrected abstract representation into an image with seeing corrected. The first convolutional layer (shown with bold vertices in Fig. 5.13) convolves the image with a 7×7 kernel and transforms the input to 64 feature maps. The dashed line layer convolves these feature maps with a 3×3 kernel, using a stride of 2 to downsample (the convolution kernel only performs the convolution operation when it is centred on every second value, see Fig. 5.15 for an illustration) and doubling the number of feature maps to 128. The dotted line layer convolves the 128 feature maps in the same way as the previous layer, downsampling by a factor of 2 and doubling the number of feature maps to 256.

After this, these feature maps are passed to the nine residual layers, shown as the shorter blocks in Fig. 5.13. Each of these layers has the structure shown in Fig. 5.14. The convolution kernel sizes are all 3×3 with each residual layer keeping the number of feature maps at 256.

Subsequently, the feature maps are given to the second dotted line layer that upsamples the feature maps by a factor of 2 using a *transpose convolution* with stride of 2 (see Fig. 5.16 for an illustration) and reduces the number of feature maps by a factor of 2 to 128 using a convolution with a 3×3 kernel. Mathematically, a transpose convolution is just a convolution with some extra padding used to upsample

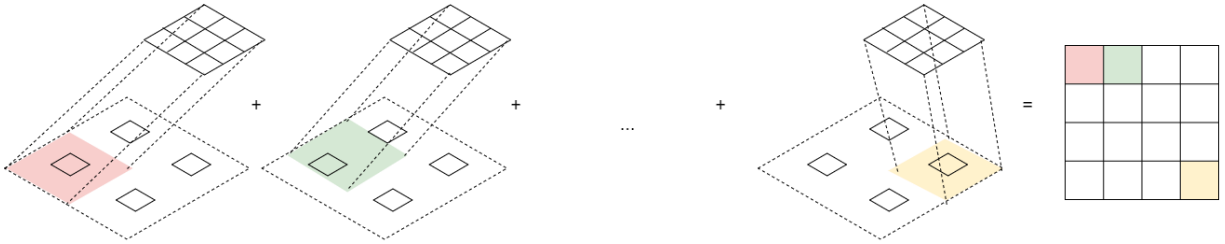

Figure 5.16: Example of a transpose convolution with stride 2 used for upsampling an input. Each of the four solid line boxes represent one of the data values of the output of Fig. 5.15. There is padding between each value (which is the input for the transpose convolution) as "stride" in transpose convolutions refers to how many steps the kernel needs to take to get between inputs rather than how often it will perform computations. The 3×3 solid line grid represents the convolution kernel with the dotted box representing padding around the outside to account for the size of the convolution kernel. The output shown on the right is the 4×4 input in Fig. 5.15.

some data. This is exemplified in Fig. 5.16, where the 2×2 output from Fig. 5.15 is upsampled back to its original 4×4 resolution. In this situation, the use of the word "stride" takes on a different meaning: while in normal convolutional layers "stride" is used to downsample by dictating the stopping points of the convolution kernel, in transpose convolutions, the stride refers to the distance the kernel would have to travel to get between two of the inputs. Essentially, a convolution is performed on data that has padding between the values to allow the learnable parameters in these layers to learn the optimal upsample. There is also still padding around the data to not have the kernel's size interfere with the dimensionality of the output. Then, the second dashed line layer follows the same process resulting in there being 64 feature maps a factor of 2 larger than those before, which are then passed to the final layer. The final layer transforms the feature maps to the number of output channels (in this case, 1) using a convolution with a 7×7 kernel.

The normalisation used in the convolutional and residual layers is called *instance* normalisation (Ulyanov et al., 2016), and the activations used are ReLUs (see Sec. 2.1.1). Instance normalisation differs from batch normalisation (see Chap. 4) as it normalises each input individually with its own learnable mean and standard deviation. This introduces orders of magnitude more learnable parameters than batch normalisation but having these parameters per input rather than per batch can improve training efficiency[4].

[4]It can also do the opposite. Like with all DNN options, choosing the correct normalisation to use in layers can be considered a hyperparameter. In this case, instance normalisation provided better

The output of this layer is then operated on by a hyperbolic tangent (tanh) function before being combined with the input to the network (shown by the red arrow in Fig. 5.13). Being combined with the input is what Kupyn et al. (2017) coined as a "ResOut" connection. The philosophy behind this is analogous to how a residual layer works but for the whole network: the network learns the residual needed to correct for the bad seeing before being combined with the input before the final output.

5.5 Training the Neural Network

The network described in Sec. 5.4 is then trained using the data generated in Sec. 5.3 with the images synthesised with artificial seeing as the input to the network and the corrected images as the output. Training follows the skeleton given in Sec. 2.3 with 90% to 10% split of the training data into training and validation with a few extra modifications used to improve the learning of the network.

Aside from generation of a good training data set, training a DNN using a meaningful loss function is crucial. For this problem, a loss function which can account for changes in both the overall look of the image and in the intensity values stored within each pixel is needed. This loss function takes the form of two individual loss functions in a linear combination[5]: perceptual loss and mean square error (MSE) loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{P}} + \mathcal{L}_{\mathrm{MSE}}. \tag{5.69}$$

Perceptual loss (introduced by Johnson et al., 2016) is a measure of similarity between two images based on how they are perceived by a different neural network from the one being trained. This is an example of transfer learning: the process of using a previously trained neural network to influence the learning of a new network. The network from Chap. 4 (henceforth, referred to as Slic) is used here due to it being trained to classify features in the solar atmosphere. The argument is that a network trained sufficiently well on recognizing features should produce the same feature maps for *two identical images*. Therefore, using a measure of the difference

convergence than batch normalisation as tests were carried out with both.

[5]Aside: linear combinations of loss functions can have scalar multipliers to weight each individual loss function accordingly. In this work, it was found that weighting both loss functions equally provided the best results. These weightings would be further hyperparameters to tune in the training of the network.

of these features maps produced deep within the Slic network will give a measure of the similarity in the two images. This works by taking the network-generated image I_G and the ground-truth image I_S and applying Slic to them. The output is then cut after the eighth layer and the feature maps compared using an MSE metric. This can be written as

$$\mathcal{L}_{\mathrm{P}} = \frac{1}{W_j H_j C_j} \sum_{x=0}^{W_j} \sum_{y=0}^{H_j} \sum_{c=0}^{C_j} \left(\psi_j \left(I_{\mathrm{S}} \right)_{x,y,c} - \psi_j \left(I_{\mathrm{G}} \right)_{x,y,c} \right)^2 , \tag{5.70}$$

where W_j are H_j are the width and height of the feature maps in the j-th output layer of Slic, respectively, and C_j is the number of feature maps after the $j-$th output layer of Slic. ψ_j is the function resulting from feeding the images through Slic and taking the output after the eighth layer. The definition of ψ_j is a hyperparameter when training this network. A different cutoff layer of Slic could be used for different purposes. In this case, both lower and higher cutoff layers resulted in worse models being trained potentially due to the lower-level feature maps not encoding enough complex information for an accurate reconstruction and the higher-level feature maps not encoding enough coarse information for reconstruction. The MSE loss is the N-dimensional Euclidean distance function squared where the data are N-dimensional.

$$\mathcal{L}_{\mathrm{MSE}} = \| I_{\mathrm{S}} - I_{\mathrm{G}} \|^2 . \tag{5.71}$$

This loss function compares the intensity values in each pixel of an image to ensure that the generated image does not violate the conservation of energy. If only constrained by perceptual loss, one can imagine a trained DNN being able to produce an image that looks similar to the target image (a low value of Eq. 5.70) but doing so by introducing larger intensity discrepancies between pixels to get the required sharpness. This may lead to unphysical data being produced which is why the inclusion of the MSE error is used as this should make sure the network does not produce vastly over- or under-estimated photon counts. (i.e. the DNN should learn to redistribute the photons within the image to preserve conservation laws before changing the total number of photons collected by the detector). Both loss functions are minimised simultaneously using the Adaptive Moment Estimation (Adam) variant of SGD with backpropagation (Kingma and Ba, 2014). Rather than using the gradients of the loss function to update the learnable parameters in the system, Adam uses the first and second moments of the loss function under the assumption that

the gradient of the loss function evaluated for a batch of data can be considered a random variable. As such Eq. 2.8 can be rewritten as the following:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{\mu}}{\sqrt{\hat{\sigma}} + \epsilon}, \tag{5.72}$$

where $\epsilon \sim O(10^{-8})$ to avoid division by 0 and:

$$\hat{\mu} = \frac{\beta_1}{1-\beta_1}\mu_t + \nabla\mathcal{L}, \tag{5.73}$$

$$\hat{\sigma} = \frac{\beta_2}{1-\beta_2}\sigma_t + (\nabla\mathcal{L})^2, \tag{5.74}$$

are estimates of the first and second moments of the gradient of the loss function, respectively. β_1 and β_2 are new hyperparameters introduced by Adam that control the influence of the previous moments on the updates to those moments for the next epoch. Much like Nesterov momentum discussed in Chap. 4, the previous first and second moments of the gradient of the loss function is used as a velocity term to control the speed through which the optimisation space is navigated. At the same time, the gradient of the loss function for the current epoch is used to steer the optimiser in the right direction. The optimal solution for Adam would be for the gradient of the loss' distribution to be a standard normal distribution. As discussed previously, the choice of variant of SGD to be used when optimising a neural network is a problem-dependent conundrum. In this work, it was found that convergence was achieved more efficiently using Adam than using SGD with Nesterov momentum while the opposite was true for the network trained in Chap. 4.

On top of using the Adam optimiser, *minibatch* training (minibatching) was utilised when training this network. Minibatching consists of not using the entirety of the training and validation data sets while training the network. Instead, 10% of the training and validation data are used randomly per epoch for training. This increases the speed of the epoch that can speed up the convergence of the network (diversity across the data will lead to better generation as the network does not see the same data every epoch and having more but quicker epochs leads to more parameter updates and thus faster learning). However, minibatching can lead to the need for more epochs of training due to only a percentage of the training and validation sets being seen each epoch. It is useful to circumvent hardware limitations as loading/unloading the entire dataset can be costly. In training, a batch size of 12

is used with 100 minibatches per epoch for the training data and 10 minibatches for the validation data – this constitues 4% of the training dataset and 27% of the validation dataset.

Furthermore, using an adaptive learning rate was found to aid the convergence of this network. Changing the learning rate while training a network can provide the network with the ability to descend stably into a local minimum but also escape a local minimum not suited to the task that is to be learned. One such example of changing the learning rate during training is to use *cosine annealing*:

$$\eta_t = \eta_{\min} + \frac{1}{2}\left(\eta_{\max} - \eta_{\min}\right)\left(1 + \cos\left(\frac{T_{\text{cur}}}{T_{\max}}\pi\right)\right), \tag{5.75}$$

where η_t is the current learning rate, $\eta_{\min}$ is the minimum learning rate, $\eta_{\max}$ is the starting learning rate, T_{cur} is the current epoch number, and $T_{\max}$ is the number of epochs to get to the minimum learning rate. The change in learning rate follows a smooth decrease (since the cosine function is differentiable) as the number of epochs progresses, allowing the network to slow down and explore local minima, and is then followed by a "warm restart" after $T_{\max}$ is exceeded. A "warm restart" refers to the increase of the learning rate back to $\eta_{\max}$ from $\eta_{\min}$ after $T_{\max}$ epochs allowing for an escape of any local minima the optimiser may have been stuck in. The network here has $\eta_{\max} = 5\times10^{-3}$, $\eta_{\min} = 1\times10^{-6}$, and $T_{\max}$ =100. As mentioned in Sec. 5.3, separate DNNs are trained for each spectral line in the training dataset. The hyperparameters for each DNN are kept the same, except the number of epochs it is trained for. The Hα network is trained on an NVIDIA Titan Xp for 1900 epochs and the Ca II λ8542 network is trained on the same hardware for 1513 epochs. The results are shown in Sec. 5.6 below.

5.6 Results

5.6.1 Validation Results

The following section is dedicated to the trained DNNs being applied to data from the validation dataset (i.e. originally good-seeing data with the bad-seeing models applied, which the network does not see during training).

An ad-hoc error on output estimates from the trained neural networks is obtained by evaluating the whole training set by the trained models and averaging the results

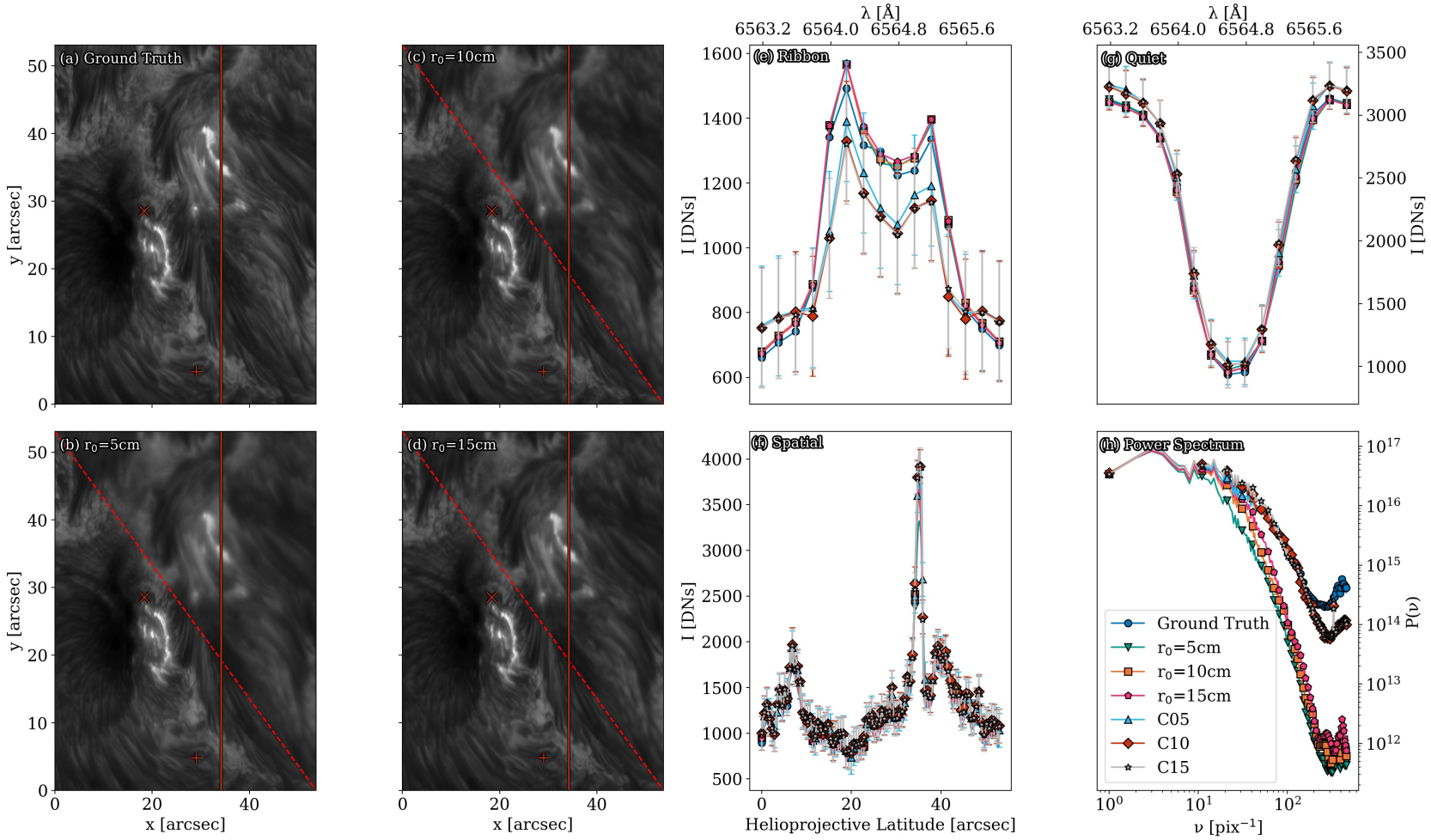

Figure 5.17: Using the trained Hα neural network to correct the data shown in Figs. 5.1 & 5.2. This figure has the same format as Fig. 5.1 with panel (a) being identical between both figures. For panels (b)–(d), the data above the dotted diagonal line is the data with bad seeing and below the line is the data corrected by the trained DNN. Panels (e), (f) & (g) again show the spectrum for a point near the flare ribbon indicated by a cross in panels (a)–(d), a spatial slice indicated by the vertical solid line in panels (a)–(d) and a spectrum for a point in the quieter part of the atmosphere shown by the plus in panels (a)–(d). As with Figs. 5.1–5.8, the circles indicate the ground truth, the upside down triangles represent the seeing model with r_0=5 cm, the squares are the seeing model with r_0=10 cm and the pentagons are the r_0=15 cm seeing model. On top of this, these panels now show the correction of each of these light curves for each seeing model: the upward facing triangles are the correction to the seeing model with r_0=5 cm, the rhombi are the correction to the seeing model with r_0=10 cm and the five-sided stars are the correction to the seeing model with r_0=15 cm. Panel (h) shows the azimuthally-averaged power spectrum across each field-of-view.

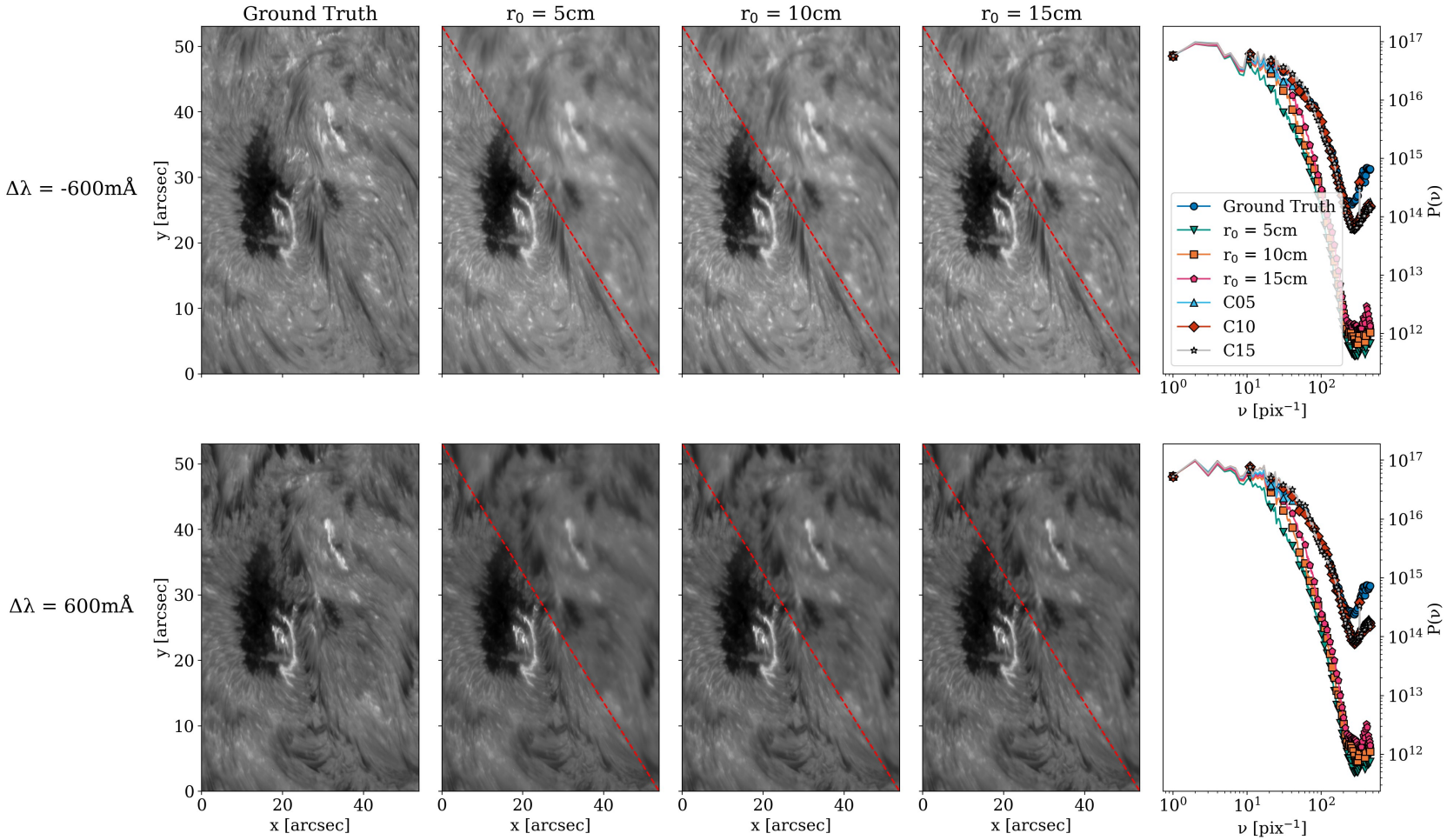

Figure 5.18: Data from the same observation as Fig. 5.1 but this time showing the blue and red line wing observations (top and bottom row, respectively) and their changes due to the synthetic seeing. In this figure, the first column represents the ground truth for both wings, the second shows the wings affected by and corrected for seeing characterised by $r_0 = 5$ cm, the third $r_0 = 10$ cm and the fourth $r_0 = 15$ cm. The columns with data corrected for seeing are split by the diagonal dashed line in the same way as Fig. 5.17. The last column shown the azimuthally-averaged power spectra over the images with the symbols retaining their definition from Fig. 5.17. The power spectra in both wings show a loss of high frequency information consistent with the line centre observations and what is to be expected by the effects of seeing. The blue and red wing observations shown are taken at $\Delta\lambda = \mp 600$mÅ, respectively.

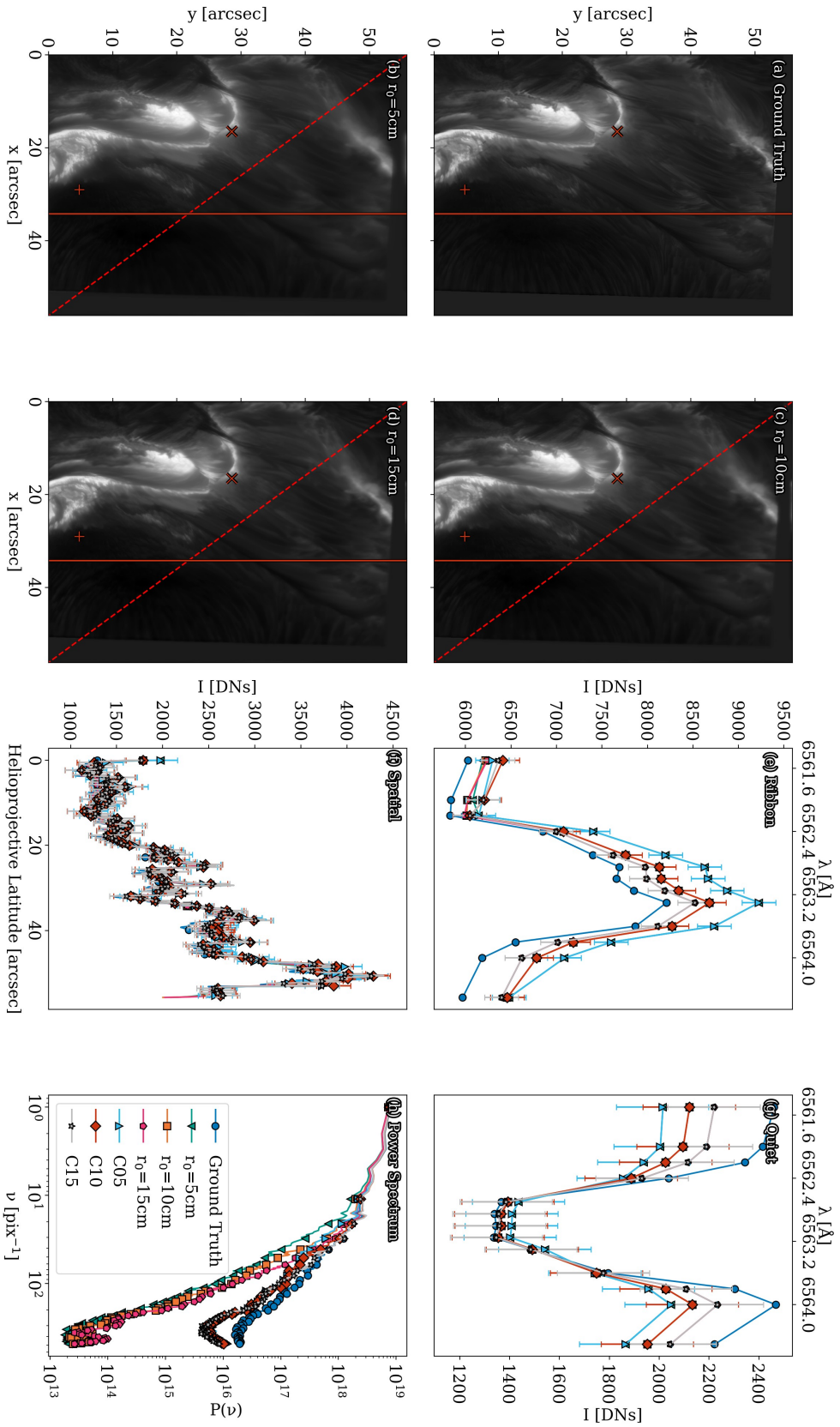

Figure 5.19: An equivalent figure to Fig. 5.17 (everything retains its meaning) but for the observations of Fig. 5.3.

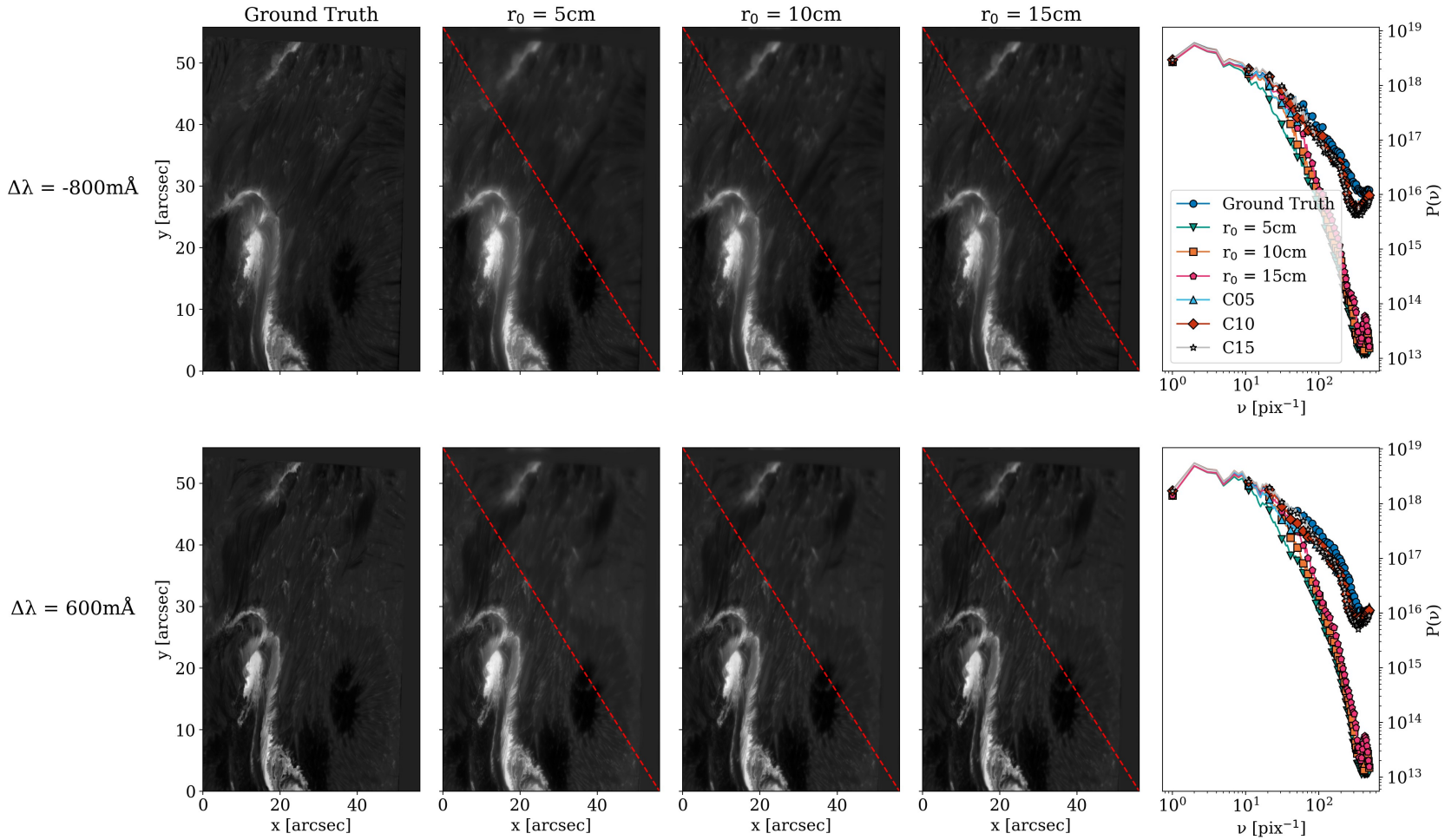

Figure 5.20: An equivalent figure to Fig. 5.18 but for the line wing observations of the flare from Fig. 5.19. The blue wing observation shown here is at $\Delta\lambda = -600$mÅ with the red wing observation being at $\Delta\lambda = 800$mÅ.

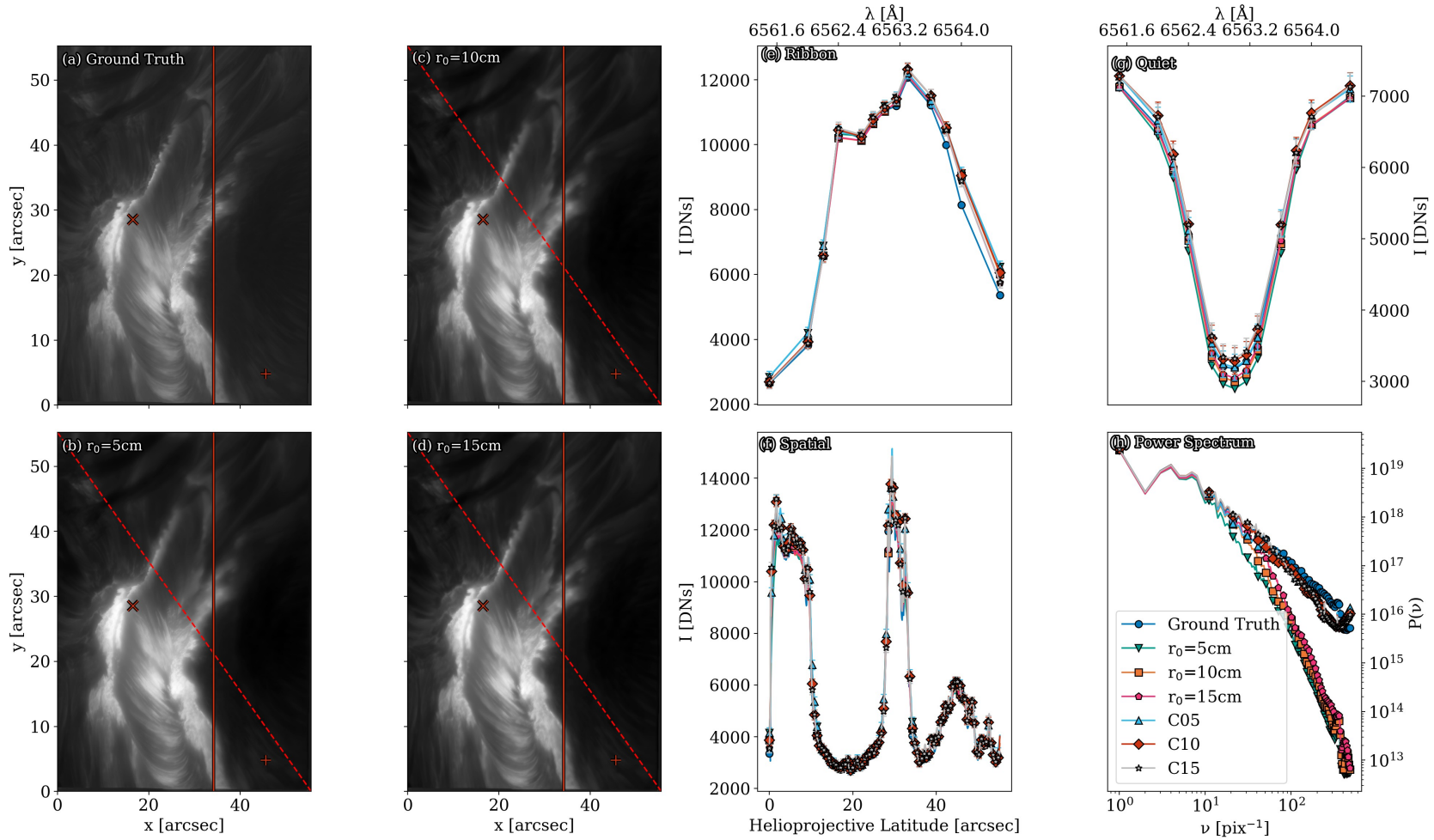

Figure 5.21: An equivalent figure to Figs. 5.17 & 5.19 (everything retains their meaning) but for the observations of Fig. 5.5.

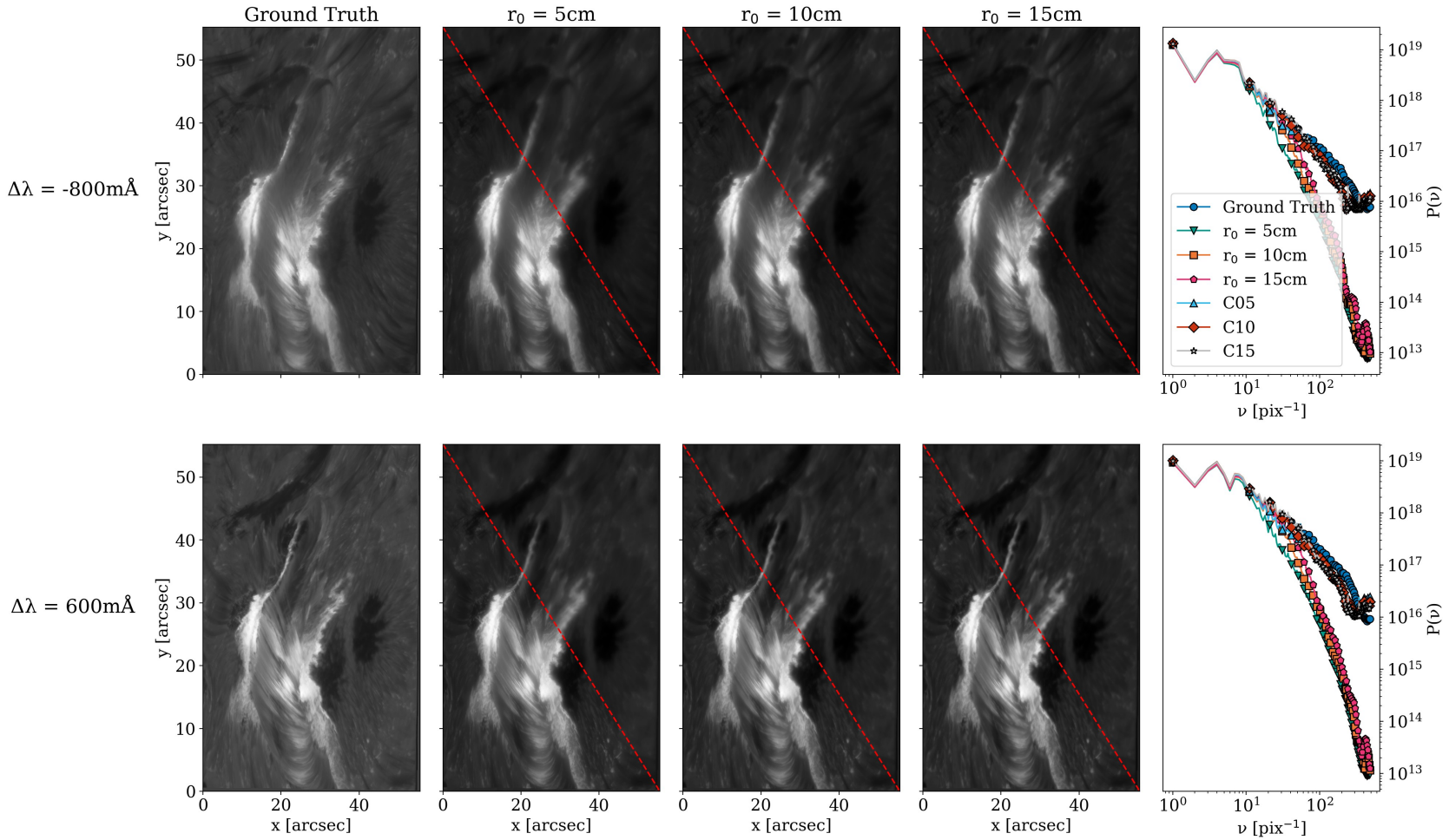

Figure 5.22: An equivalent figure to Figs. 5.18 & 5.20 but for the line wing observations of the flare from Fig. 5.21. The blue wing observation shown here is at $\Delta\lambda = -600$mÅ with the red wing observation being at $\Delta\lambda = 800$mÅ.

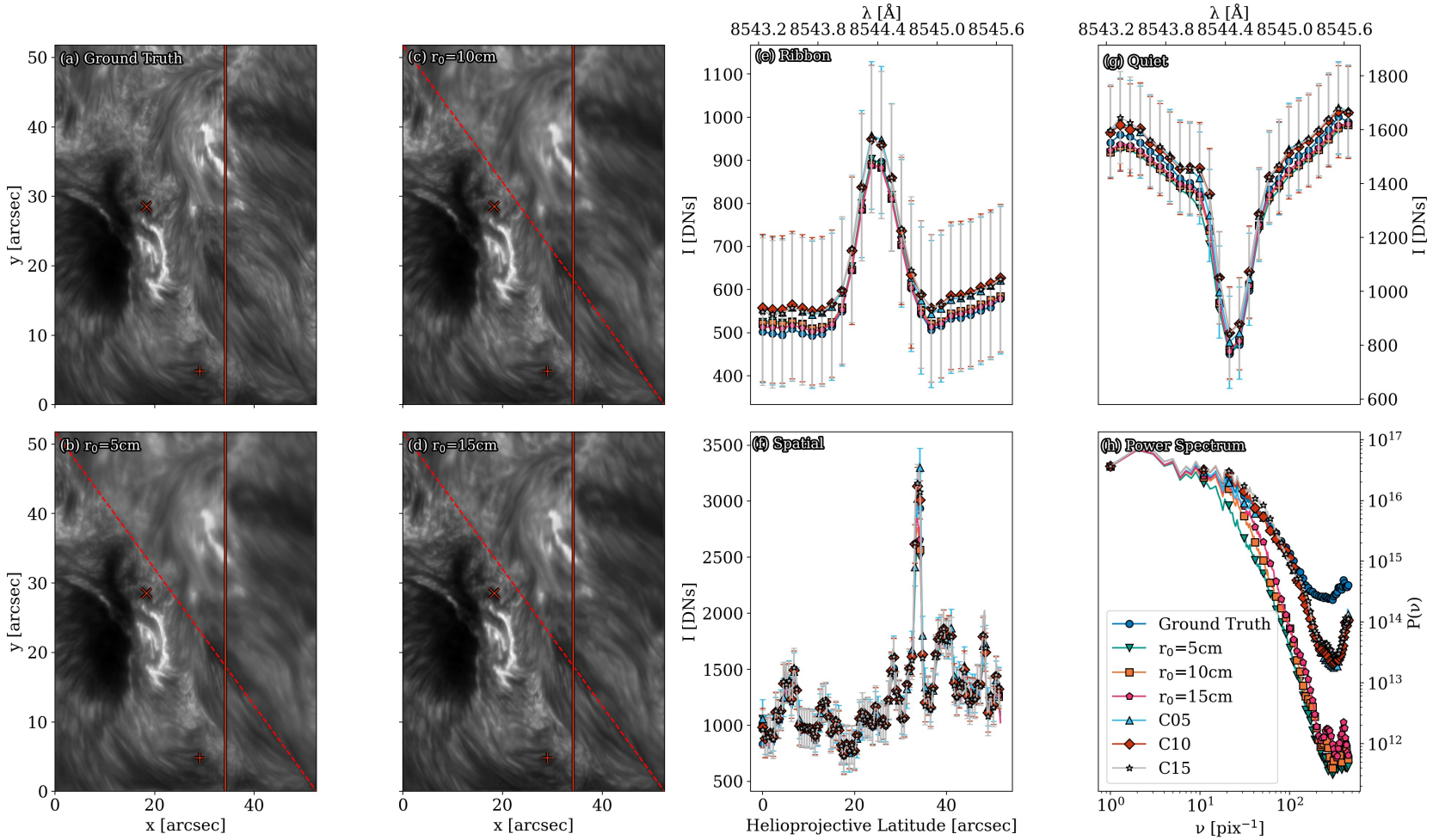

Figure 5.23: Same figure layout as for Figs. 5.17, 5.19 & 5.21 with everything retaining the same meanings. This time, though, the observations shown are the line centre of the Ca II λ8542 line. The observation used here is cotemporal with the Hα observation in Fig. 5.17.

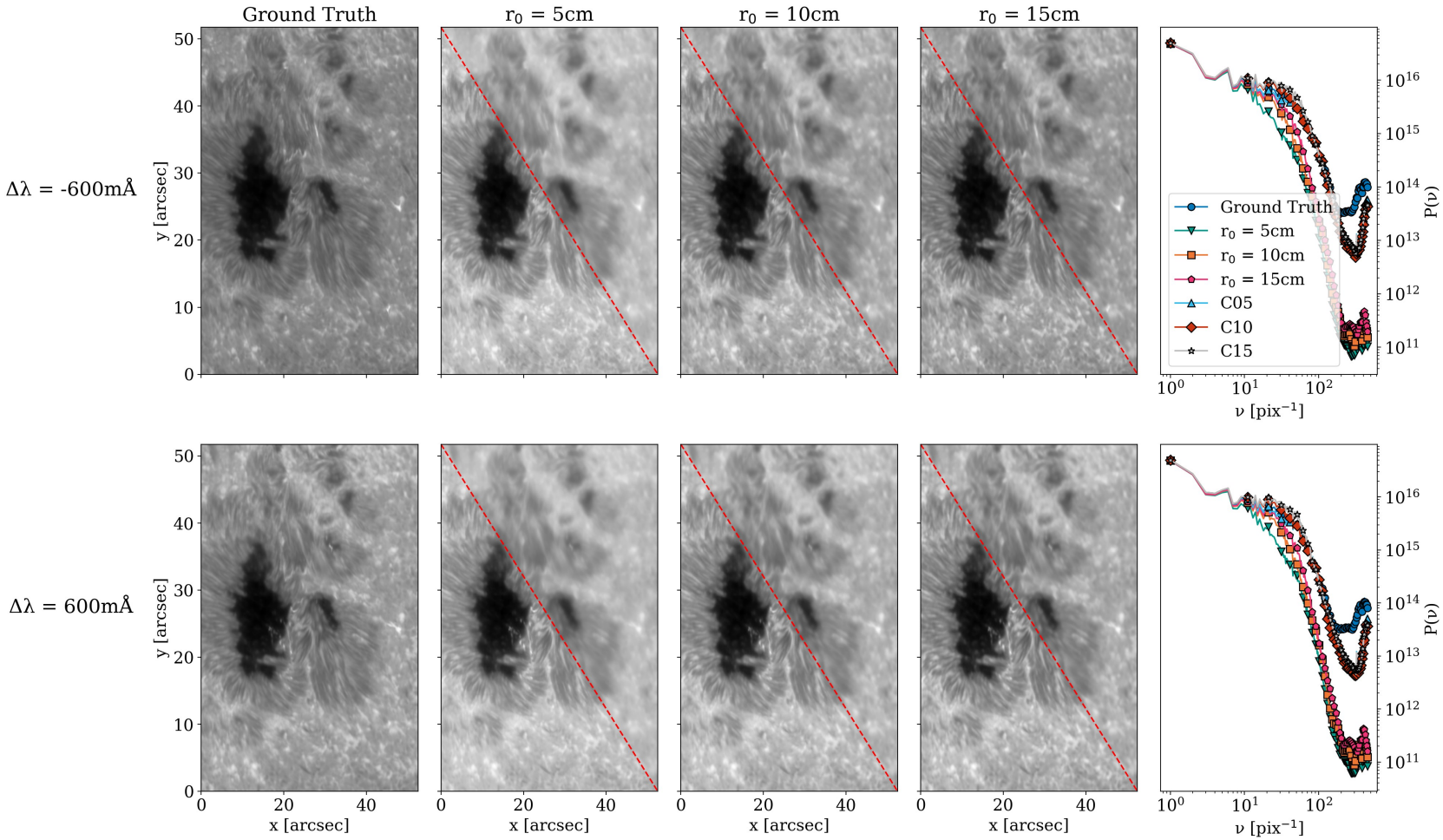

Figure 5.24: Same figure layout as for Figs. 5.18, 5.20 & 5.22 showing the complimentary line wing observations to Fig. 5.23. The blue and red wings are observed at $\Delta\lambda = \mp 600$mÅ, respectively.

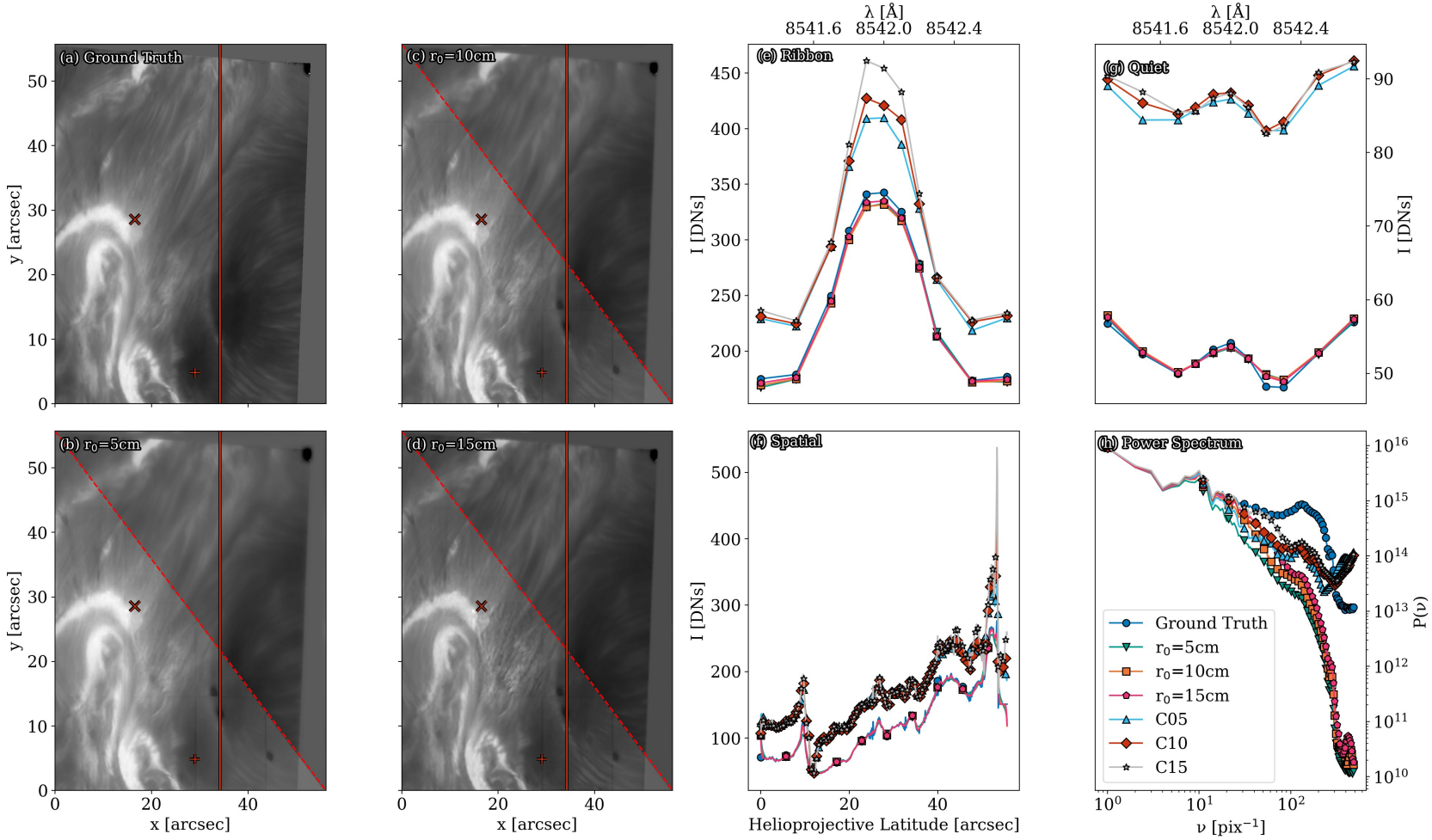

Figure 5.25: An equivalent figure to Fig. 5.23 where the observations shown here are cotemporal with the observations shown in Fig. 5.19.

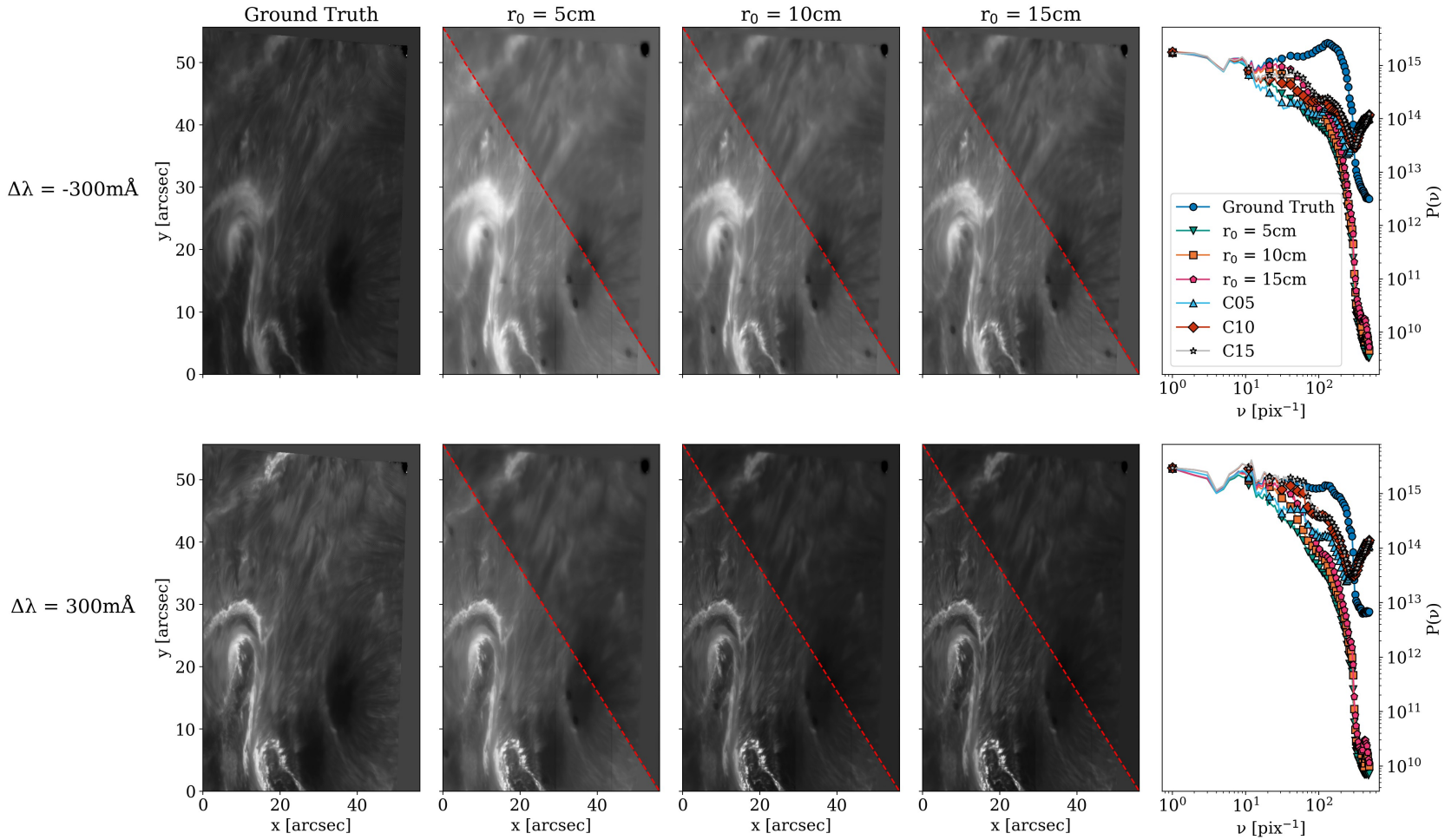

Figure 5.26: An equivalent figure to Fig. 5.24 but in this figure, the complimentary wing images are from the data used in Fig. 5.25.

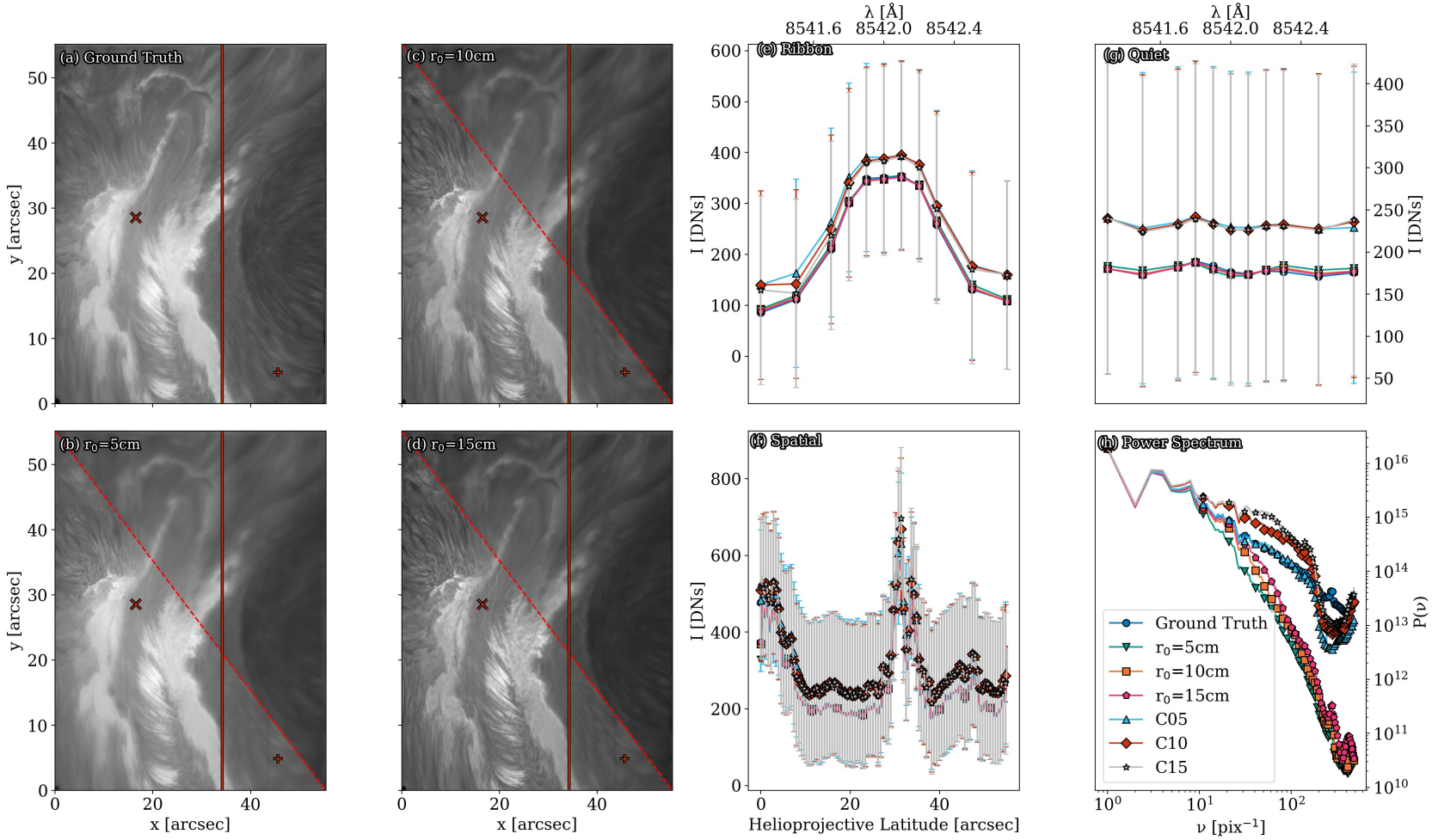

Figure 5.27: An equivalent figure to Figs. 5.23 & 5.25 where the observations shown here are cotemporal with the observations shown in Fig. 5.21.

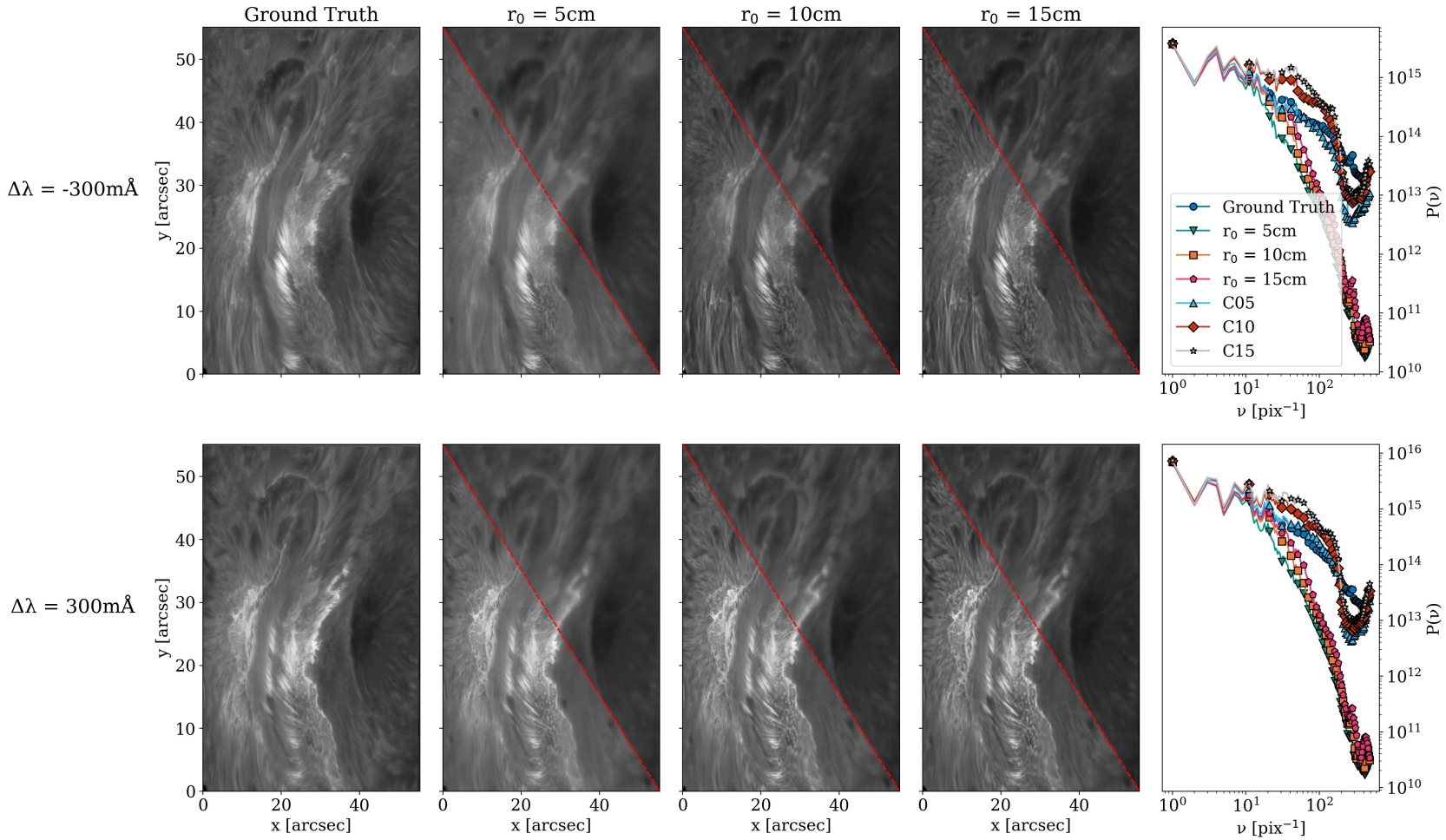

Figure 5.28: An equivalent figure to Figs. 5.24 & 5.26 but in this figure, the complimentary wing images are from the data used in Fig. 5.27.

from Eq. 5.69. This leads to an error for the Hα model of $\sigma_{\mathrm{H}\alpha}$ = 185.14DNs and an error for the Ca II λ8542 model of $\sigma_{\mathrm{Ca\ II}}$ = 170.99DNs. A more robust error analysis will be proposed in Sec. 5.7.

The results of the validation reconstruction for the Hα data are shown in Figs. 5.17–5.22. Figure 5.17 shows the reconstruction of the data presented in Fig. 5.1. Panel (a) again shows the ground truth data before artificial seeing is applied. However, in Fig. 5.17, panels (b)–(d) shows both the data with bad seeing and the reconstruction. The reconstruction is shown below the diagonal dashed line in these panels and the data with bad seeing (that is the data from Fig. 5.1(b)–(d)) is shown above the line. As with Fig. 5.1, the cross in panels (b)–(d) represents the data shown in panel (e), the plus indicates the data shown in panel (g) and the vertical solid line the data in panel (f). Moreover, panel (h) shows the azimuthally-averaged power spectrum as described in Sec. 5.3. Here, however, panels (e)–(h) have three extra curves to show the correction to the data with bad seeing. That is, the data represented by the upward facing triangles are the reconstructed data from the r_0=5 cm model, the rhombi are the reconstructions of the r_0=10 cm model and the five pointed stars are the reconstruction of the r_0=15 cm model. All other symbols retain their meaning from Fig. 5.1. The estimates from the network are plotted with their associated error $\sigma_{\mathrm{H}\alpha}$.

Looking at panel (e), despite the large error bars, each case of different seeing is reconstructed well by the model with the ground truth falling within the error bars. Furthermore, the central reversal and blue asymmetry of the spectral line present in the original data are reconstructed accurately. For instance, the correction to the r_0=5 cm model slightly overestimates the value of the intensity for this particular spectrum but the ratio of intensities between the blue and red peaks is approximately equal for the ground truth and the reconstruction. Panel (g) shows the results for the spectral line from the quieter part of the atmosphere. In this case, all of the reconstructions are close to the ground truth with the worse-seeing models having a less accurate reconstruction. This may be due to the focus of the trained model being subverted by the bright features with the lower contrast features not being as crucial in the reconstruction. Panel (f) shows the slice of constant y. This illustrates on the whole that the brighter and darker features are reconstructed well by the model as there is not much discrepancy between reconstruction and ground truth along the slice. Panel (h) shows the power spectrum for the ground truth, each of the degraded images and the reconstructed images. The reconstructions

show that the large- to medium-scale structure (up to $\nu = 10$ pix^{-1}) within the field of view is almost perfectly reconstructed regardless of seeing conditions. The rest of the spectrum for each reconstruction shows a tendency to reconstruct smaller features but not with the power they are represented by in the ground truth image. This is noticeable towards the highest frequencies where length-scales approach a single pixel; however, when the features are on scales of tens of pixels their power is still restored well for all seeing conditions. For example, when r_0=15 cm, there is still a good reconstruction up to approximately $\nu = 200$ pix^{-1}. The shapes of the reconstructed power spectra are correct compared with the ground truth, which suggests that learning for a better convergence of the L2 loss (Eq. 5.71) may result in the restoration of the lost power.

Figure 5.18 shows the correction to the line wing reconstructions with the blue wing observed at $\Delta\lambda = -600$mÅ shown in the top row and the red wing observed at $\Delta\lambda = 600$mÅ shown in the bottom row. The columns retain their same meaning from Fig. 5.2. However, much like the first four panels in Fig. 5.17, the reconstructions are shown with the ground truth contaminated by bad seeing in the second, third and fourth columns with the former being above the diagonal dashed line and the latter being below. The last column shows the power spectra of each of the cases where the markers of each curve retain their meanings from Fig. 5.17. Qualitatively, the reconstructions in the second, third and fourth columns look accurate and sharp. This is reinforced quantitatively by the last column of this figure as again the power spectra of the images are reconstructed well up to around $\nu = 200$ pix^{-1}.

Figures 5.19 & 5.20 are equivalent figures to Figs. 5.17 & 5.18 but demonstrating the reconstructions of Figs. 5.3 & 5.4. The model struggles more with the reconstruction for this particular data, which gets worse towards the line centre. In the two points selected (the cross and plus in Fig. 5.19 (a)–(d)), there is some discrepancy between the reconstructions and the ground truth. The point on the ribbon is more accurate in the wings of the line compared to the core where the trained DNN seems to overestimate how bright these pixels should be at line centre. Conversely, for the quieter part of the atmosphere, the wing data is estimated as darker while the line core is estimated well. These differences may arise from a lack of diversity in the training dataset for these regions as the network may decide a certain region is familiar to another and correct it in the same way. Moreover, the energy conservation constraint introduced by the L2 loss could be at fault as the total number of photons needs to be redistributed throughout the image leading to some misplaced photons.

Looking at the power spectra (Fig. 5.19 (h)), a discrepancy starts to appear at spatial frequencies of $\nu \approx 70$ pix^{-1}, implying a worse reconstruction of the data. However, the last column in Fig. 5.20, shows that the line wing data for this observation are reconstructed very well. In the blue wing, the images are reconstructed perfectly up until $\nu = 300$ pix^{-1} and, in the red wing, the images are reconstructed nearly perfectly for all spatial frequencies.

Again, Figs. 5.21 & 5.22 are equivalent to Figs. 5.17 & 5.18 but for the reconstruction of the data in Figs. 5.5 & 5.6. The model performs a much better reconstruction than the previous observation. This is demonstrated by looking at panels (e), (f) & (g) of Fig. 5.21 (which take their usual meanings) where the reconstructed light curves are very close to the ground truth. The biggest discrepancy here seems to be in the blue wing of the ribbon spectrum where the trained model seems to perform consistently less well than for other points along the spectrum. Regardless, looking at the reconstructed power spectra for the line core images reveals great reconstruction with the finer details recovered well up to $\nu \approx$ 120pix^{-1}. At the very highest spatial frequencies ($\nu \gtrsim O(400$ pix$^{-1})$), the power in the image increases *above* the ground truth indicating there are small features (on the order of a few pixels) that are made bright in the reconstruction when they weren't originally. This is likely in the region of the flare ribbons as there is a lot of fine structure there. Looking at the power spectra in the last column in Fig. 5.22, the red and blue wing images are reconstructed similarly to the core images, again with a slight overestimation of some of the smaller scale features but with good reconstruction up to scales of $\nu \approx 300$ pix^{-1}.

Figures 5.23–5.28 shows the results of the validation reconstructions for the Ca II λ8542 trained DNN. These are formatted in the same way as the figures for the Hα model with Figs. 5.23 & 5.24 being cotemporal with Figs. 5.17 & 5.18; Figs. 5.25 & 5.26 cotemporal to Figs. 5.19 & 5.20; and Figs. 5.27 & 5.28 cotemporal to Figs. 5.21 & 5.22. The error bars for the Ca II λ8542 model look unflattering in Figs. 5.25 & 5.27(e), (f) & (g) due to the lower number of DNs in these light curves. This is actually an issue that stemmed from the calibration of the data due to the highly energetic nature of these events. The data was originally reduced as 16-bit integers but the flux incident on the detector was so large that the 16-bit integers overflowed meaning the largest values in the flare were negative. To fix this, the data were reprocessed as floats with a (unknown) scaling factor resulting in the DN values being small (Aaron Reid, priv. comm.). This is, unfortunately, not good for

the training of the DNN and the reconstruction of data from this flare. As each wavelength image is reconstructed independently by this model, the ability of accurate reconstruction depends somewhat on the intensity values within an image. Furthermore, as seen in the top right corner of Fig. 5.25 (a)–(d), there are dark artefacts present in this data which occur due to difficulties during data reduction. This causes the appearance of dark patches in other points throughout the restoration of this data indicating that not including this data in the training set might improve the network performance. This is particularly present in the reconstructions in Fig. 5.26. Disregarding this data from the training dataset would also help reduce the size of the error bars.

The reconstruction of the M1.1 flare works well despite the contamination from the other flare data. The light curves are reconstructed to high accuracy in Fig. 5.7(e)–(g) and the power spectra are accurate up to $\nu \approx 300\text{pix}^{-1}$ for the line centre images and similarly for the line wing images. The power spectra for Figs. 5.25, 5.26, 5.27 & 5.28 are only accurate on the larger scales $\nu \approx 10\text{pix}^{-1}$. There is often overestimation on the smaller scales in these power spectra, indicating an overcompensation in the pixels.

5.6.2 Correction to New Data

For the AR12157 dataset, there are three examples in this section where the data contain bad seeing: one from the pre-flare phase, one from the rise of the soft X-ray peak, and one in the decay phase. The results of the trained Hα DNN applied to this data is shown are shown in Figs. 5.29 & 5.30. Figure 5.29 (a) shows the GOES soft X-ray light curves indicating the flare classification and is annotated to show the three different examples used throughout Figs. 5.29 & 5.30. The examples are: the pre-flare of SOL20140906T17:10 at 15:33:14UTC; during the rise of the soft X-ray peak of SOL20140906T17:10 at 16:54:13UTC; and in the decay of SOL20140906T17:10 at 17:15:24UTC. Figure 5.29 (d)–(g) show the pre-flare observation in the Hα red wing $\Delta\lambda$ = +1.0Å, the corrected red wing observation, the Hα line core observation, and the corrected line core observation, respectively. Figure 5.29 (h)–(k) and (l)–(o) follow the same layout for the rise of the soft X-ray peak of SOL20140906T17:10 and the decay of SOL20140906T17:10, accordingly.

Each of Fig. 5.29 (d)–(o) is annotated with a “+” and an “x”. The “+” indicates the spectra shown in Fig. 5.29 (b) and the “x” the spectra shown in Fig. 5.29 (c).

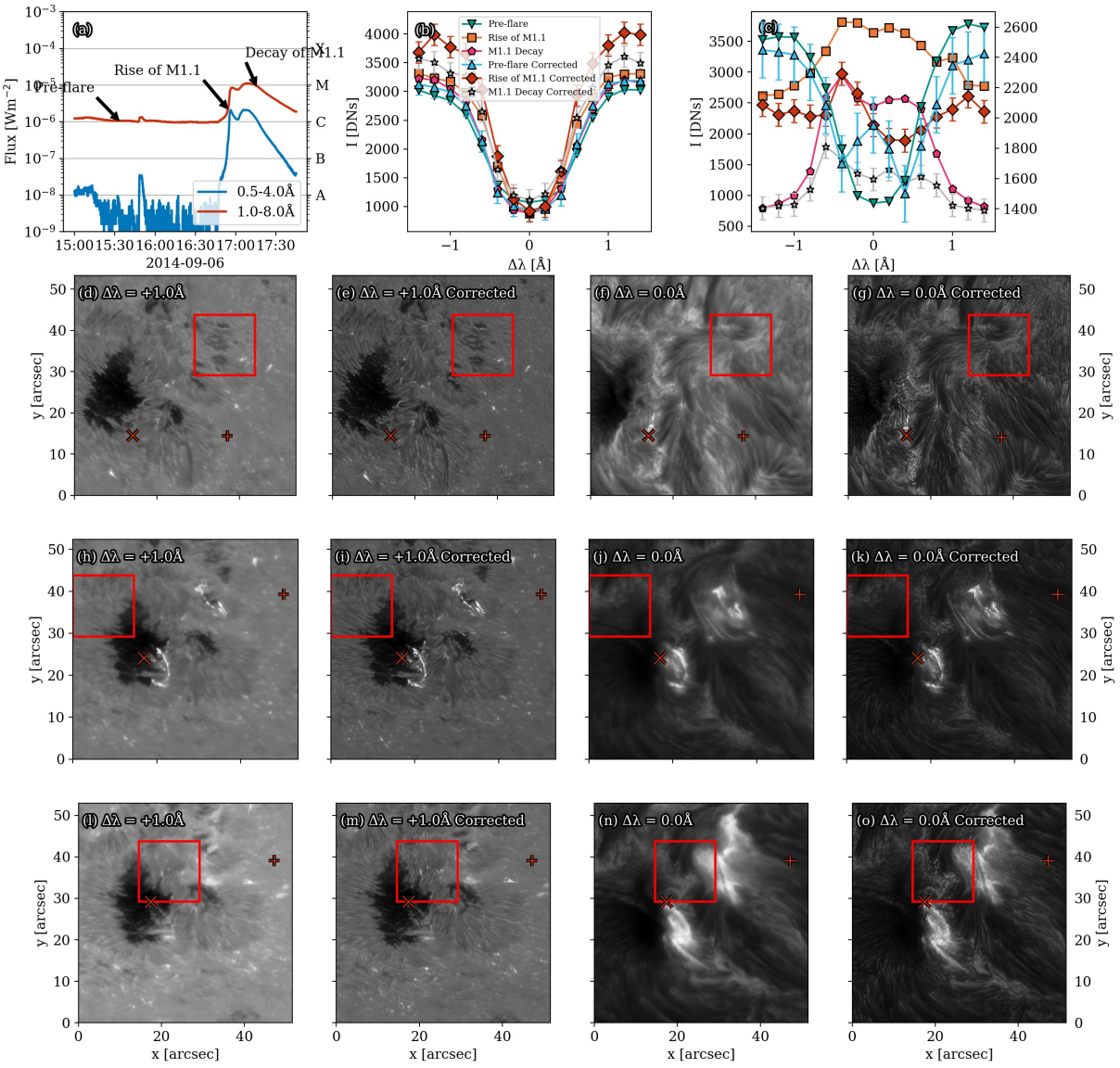

Figure 5.29: Panel(a) shows the GOES soft X-ray curve for the AR12157 data, annotated to show the time of each observation. (b) and (c) show absorption and emission spectra, respectively, from the uncorrected data, and the corrected versions for each of the cases. The *y*-axis labels on the left of panel (c) show the intensity values of the spectra at the rise and decay times of the flare with the right labels showing the intensity of the preflare spectrum. The trained model applied to observations from AR12157 for the pre-flare of SOL20140906T17:10 is shown in (d)–(g); the rise of the soft X-ray peak of SOL20140906T17:10 is shown in (h)–(k); and the decay of SOL20140906T17:10 is shown in (l)–(o). In each row, the first panel is the observation before correction, taken in the red wing of Hα ($\Delta\lambda$ = 1.0Å); the second is the correction to the red wing image; the third panel is the image in the line core before correction; and the last panel is the corrected line core image. The spectra shown are indicated in (d)–(o) using the "+" and "x" for (b) and (c), respectively. The boxes in panels (d)–(o) represent the subfields shown in Fig. 5.30.

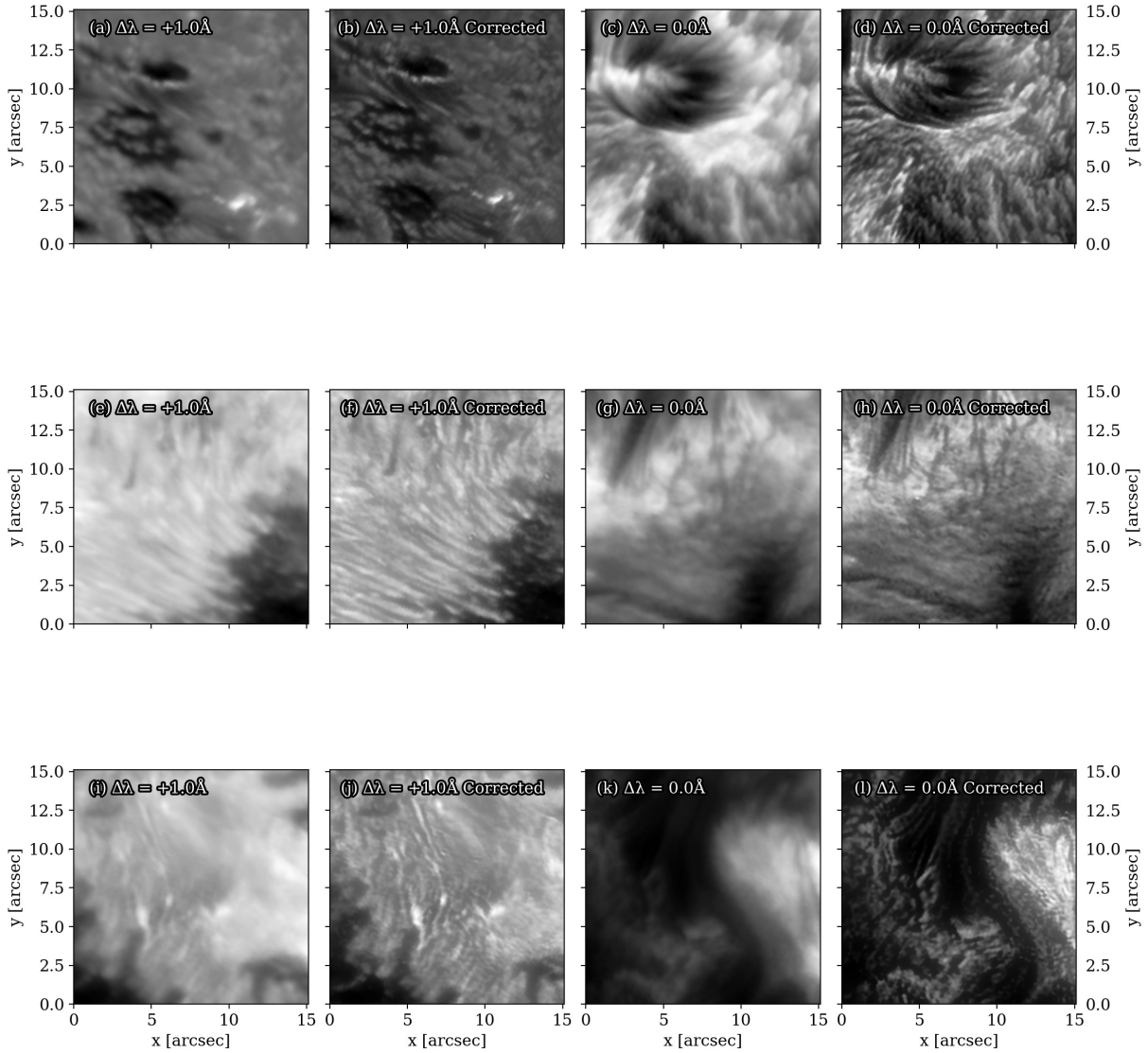

Figure 5.30: The subfields indicated by the boxes in Fig. 5.29. These show the correction on the small-scale features in the image for the three different observations of AR12157 indicated in Fig. 5.29(a). Panels (a)–(d) show the model applied to part of AR12157 north-west of the main sunspot during the pre-flare of SOL20140906T17:10 in both the Hα red wing – (a) and (b) – and Hα line core – (c) and (d). Panels (e)–(h) show the application to part of the sunspot umbra/penumbra during the rise of SOL20140906T17:10 following the same layout as the previous row. Likewise, panels (i)–(l) show the application to a region containing some sunspot penumbra and some of the northern flare ribbon during the decay of SOL20140906T17:10.

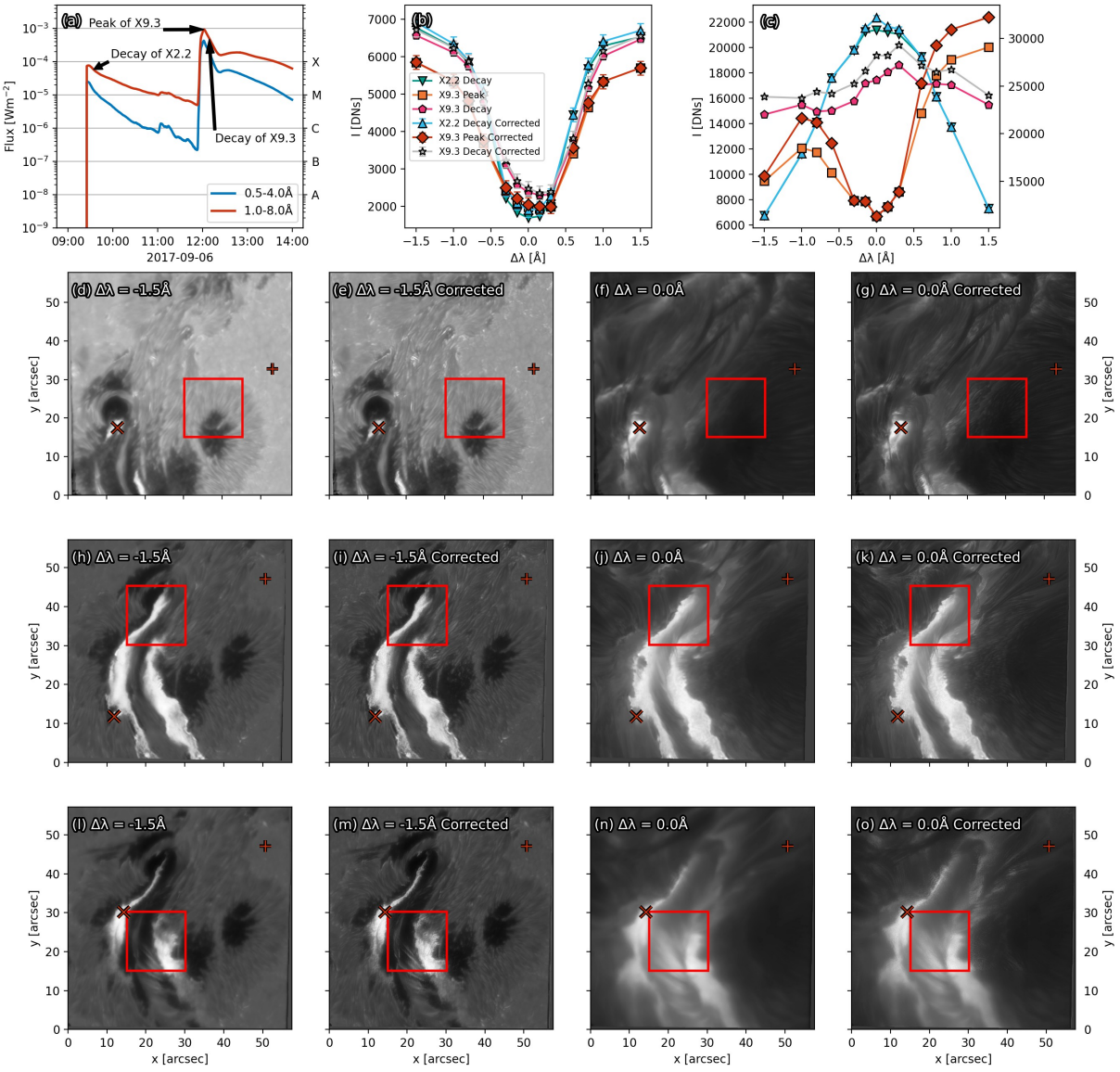

Figure 5.31: Panel (a) shows the GOES soft X-ray curve for the AR 12673 data, annotated to show where each observation corresponds to. (b) and (c) show spectra off and on the flare ribbon from the raw frames, respectively, and the corrected spectra for each of the cases. Trained model applied to observations from AR 12673 for the decay phase of the X2.2 flare SOL20170906T09:10 is shown in panels (d)–(g); the soft X-ray peak of the X9.3 flare SOL20170906T12:02 is shown in panels (h)–(k); and the decay phase of SOL20170906T12:02 is shown in panels (l)–(o). In each row, the first panel is the observation before correction taken in the far blue wing of Hα ($\Delta\lambda = -1.5$Å); the second panel is the correction to the blue wing image; the third panel is the image in the line core before correction; and the last panel is the correction to the line core. The spectra shown are indicated in (d)–(o) using "+" and "x" for (b) and (c), respectively. The boxes in panels (d)–(o) represent the subfields shown in Fig. 5.32.

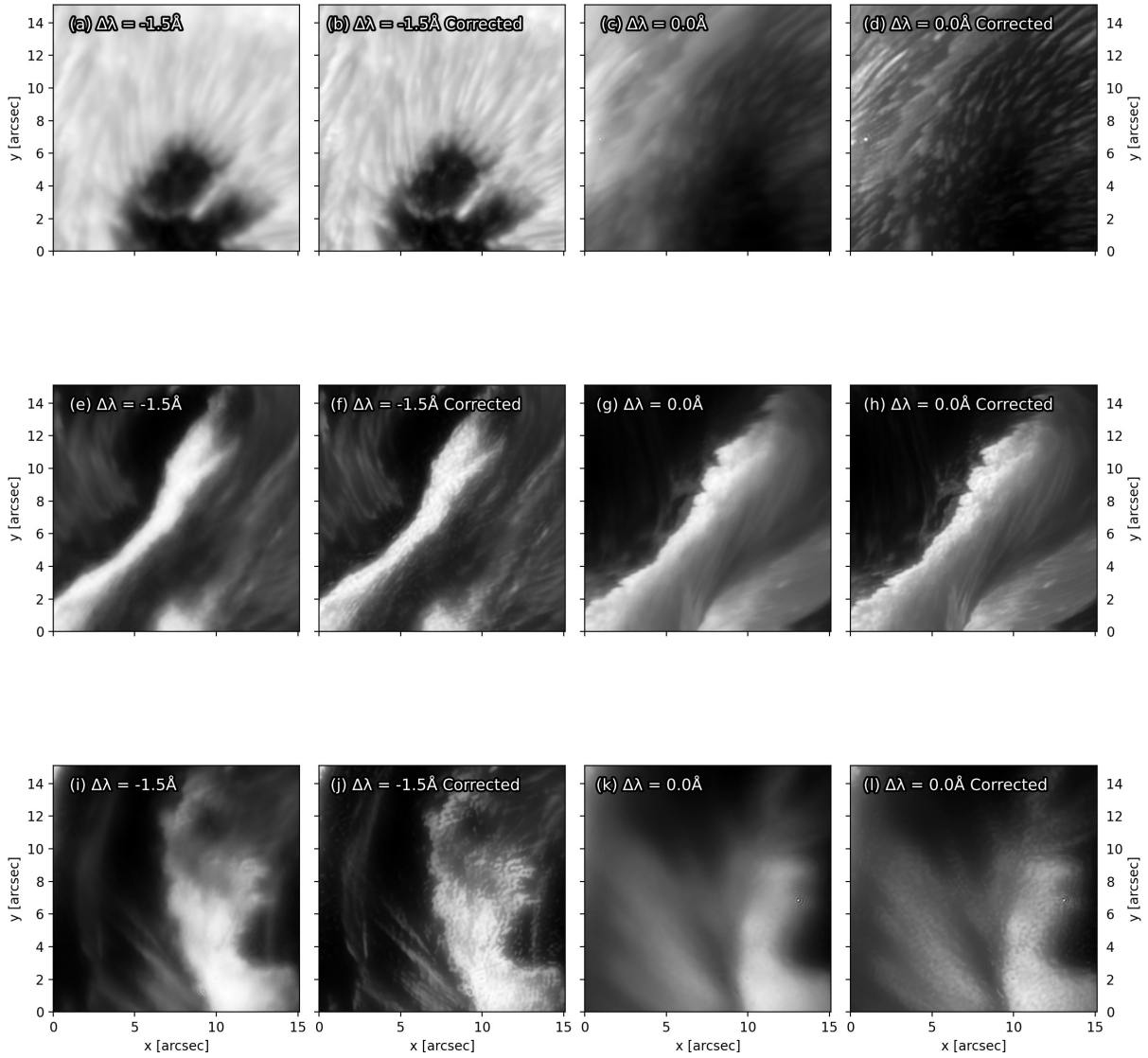

Figure 5.32: The subfields indicated by the boxes in Fig. 5.31. This shows the correction on the small-scale features in the images for the three different observations of AR 12673 indicated in Fig. 5.31(a). Panels (a)–(d) show the model applied to a sunspot umbra/penumbra region in the decay phase of SOL20170906T09:10 in both the Hα blue wing – (a) and (b) – and Hα line core – (c) and (d). Panels (e)–(h) show the application to the eastern flare ribbon at the peak of the SOL20170906T12:02 event following the same convention as the previous row. Similarly, panels (i)–(l) show the application to the western flare ribbon during the decay phase of SOL2017:0906T12:02.

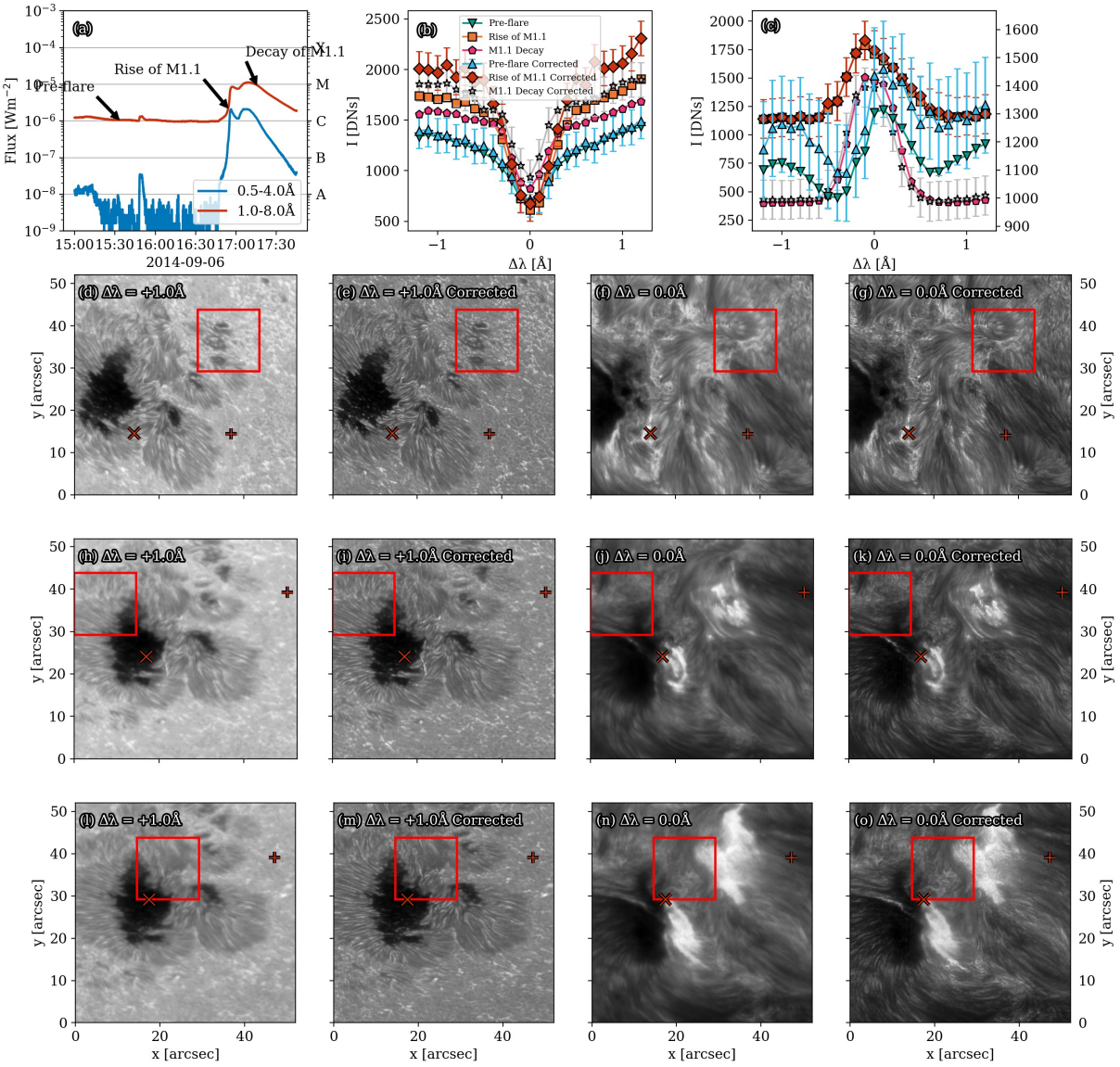

Figure 5.33: Panel(a)shows the GOES soft X-ray curve for the AR12157 data, annotated to show the time of each observation. (b) and(c) show absorption and emission spectra, respectively, from the uncorrected data, and the corrected versions for each of the cases. The trained model applied to observations from AR12157 for the pre-flare of SOL20140906T17:10 is shown in (d)–(g); the rise of the soft X-ray peak of SOL20140906T17:10 is shown in (h)–(k); and the decay of SOL20140906T17:10 is shown in (l)–(o). In each row, the first panel is the observation before correction, taken in the red wing of Ca II λ8542 ($\Delta\lambda$ = 1.0Å); the second is the correction to the red wing image; the third panel is the image in the line core before correction; and the last panel is the corrected line core image. The spectra shown are indicated in (d)–(o) using the "+" and "x" for (b) and (c), respectively. The boxes in panels (d)–(o) represent the subfields shown in Fig. 5.34.

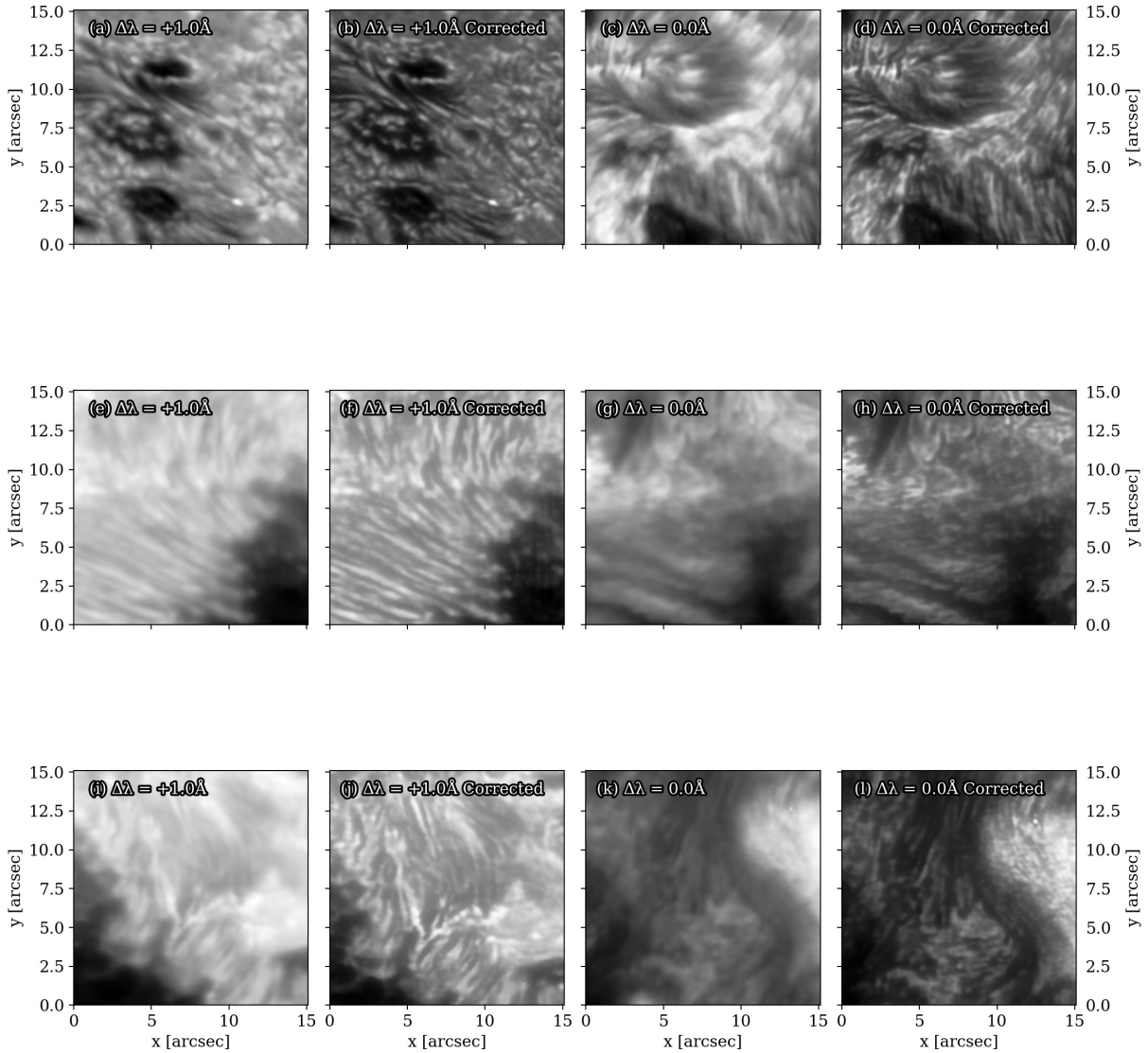

Figure 5.34: The subfields indicated by the boxes in Fig. 5.33. These show the correction on the small-scale features in the image for the three different observations of AR12157 indicated in Fig. 5.33(a). Panels (a)–(d) show the model applied to part of AR12157 north-west of the main sunspot during the pre-flare of SOL20140906T17:10 in both the Hα red wing – (a) and (b) – and Hα line core – (c) and (d). Panels (e)–(h) show the application to part of the sunspot umbra/penumbra during the rise of SOL20140906T17:10 following the same layout as the previous row. Likewise, panels (i)–(l) show the application to a region containing some sunspot penumbra and some of the northern flare ribbon during the decay of SOL20140906T17:10.

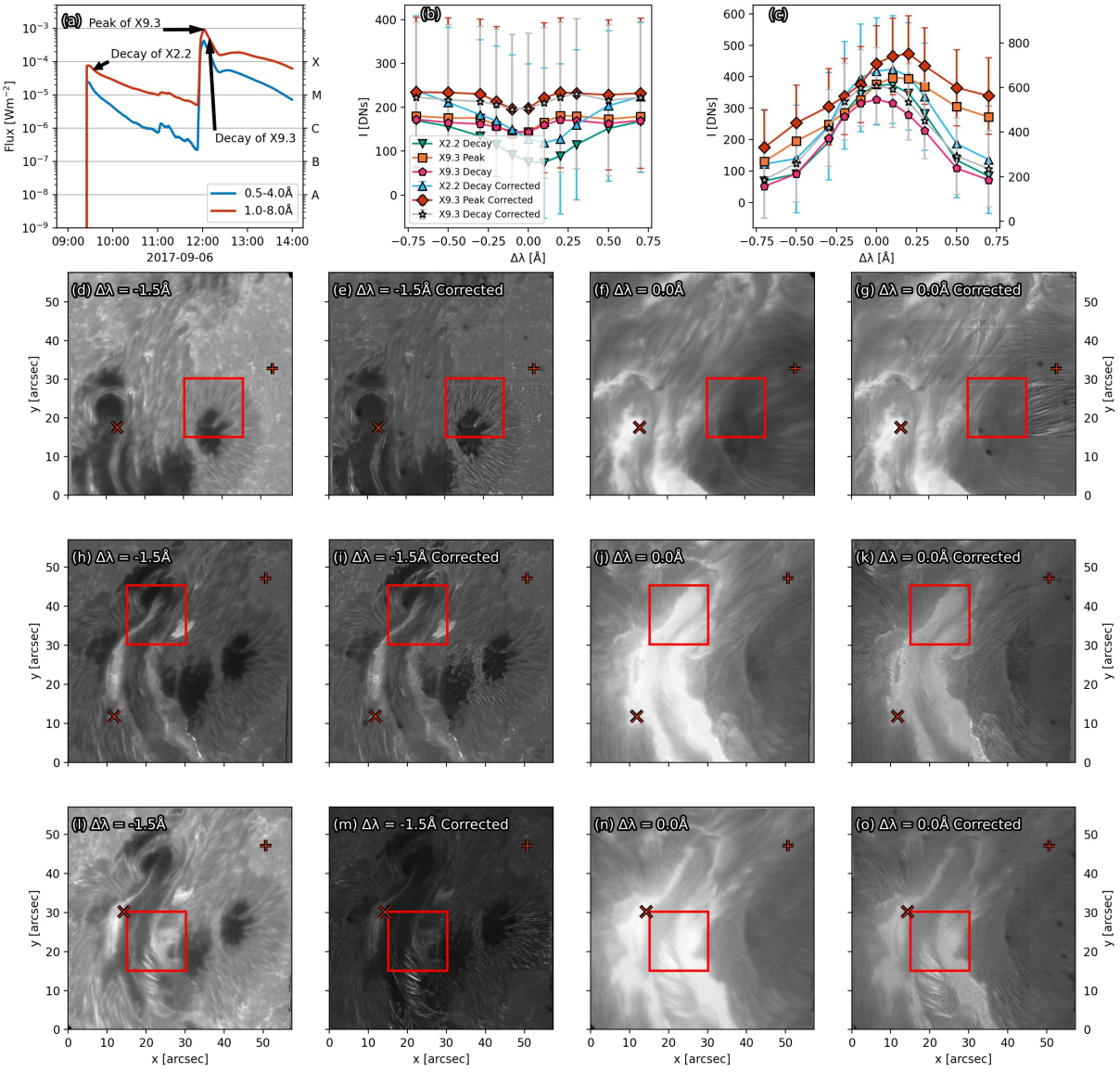

Figure 5.35: Panel (a) shows the GOES soft X-ray curve for the AR 12673 data, annotated to show where each observation corresponds to. (b) and (c) show spectra off and on the flare ribbon from the raw frames, respectively, and the corrected spectra for each of the cases. Trained model applied to observations from AR 12673 for the decay phase of the X2.2 flare SOL20170906T09:10 is shown in panels (d)–(g); the soft X-ray peak of the X9.3 flare SOL20170906T12:02 is shown in panels (h)–(k); and the decay phase of SOL20170906T12:02 is shown in panels (l)–(o). In each row, the first panel is the observation before correction taken in the far blue wing of Ca II λ8542 ($\Delta\lambda = -1.5$Å); the second panel is the correction to the blue wing image; the third panel is the image in the line core before correction; and the last panel is the correction to the line core. The spectra shown are indicated in (d)–(o) using "+" and "x" for (b) and (c), respectively. The boxes in panels (d)–(o) represent the subfields shown in Fig. 5.36.

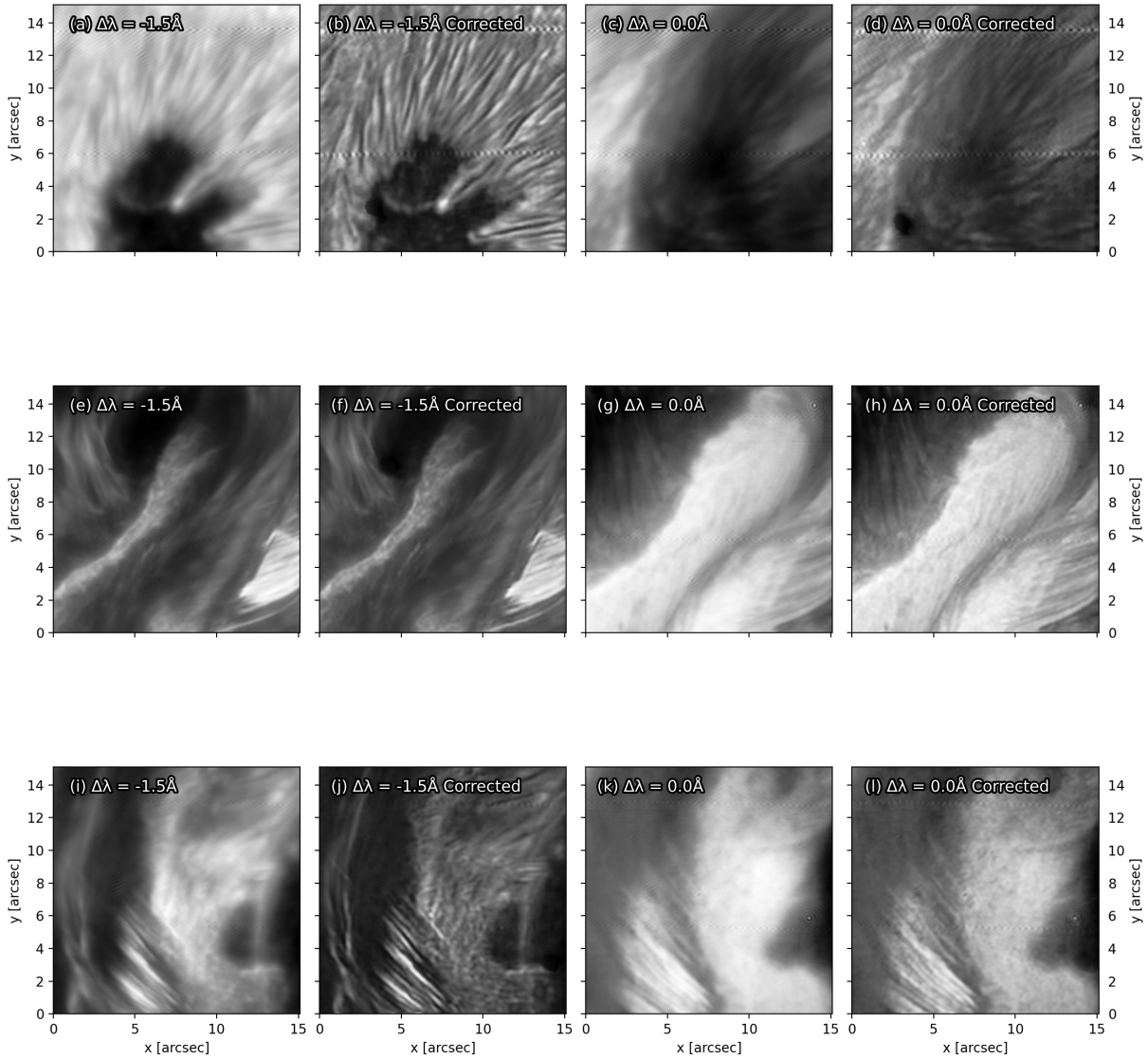

Figure 5.36: The subfields indicated by the boxes in Fig. 5.35. This shows the correction on the small-scale features in the images for the three different observations of AR 12673 indicated in Fig. 5.35(a). Panels (a)–(d) show the model applied to a sunspot umbra/penumbra region in the decay phase of SOL20170906T09:10 in both the Hα blue wing – (a) and (b) – and Hα line core – (c) and (d). Panels (e)–(h) show the application to the eastern flare ribbon at the peak of the SOL20170906T12:02 event following the same convention as the previous row. Similarly, panels (i)–(l) show the application to the western flare ribbon during the decay phase of SOL2017:0906T12:02.

In these panels, the downward-facing triangles represent the spectral line before correction for the pre-flare observation and the upward-facing triangles represent the post-correction spectrum; the square points show the line profile before correction for the rise of the soft X-ray peak of SOL20140906T17:10, with the diamonds showing the line profile post correction; and the pentagons correspond to the profile before correction of the decay observation, with the stars showing the profile post correction. The line profiles in Fig. 5.29 (b) retain their shape and have enhanced intensities across the lines. The pre-flare-corrected spectrum is the one that has changed the most with a noticeable increase in intensity towards the line core. This is a result of the "smearing" of light mentioned in Sec. 5.2. Again, the Doppler shifts and intensity-averaged wavelengths are approximately conserved. The line profiles from the rise-phase observations in Fig. 5.29 (c) show a different story. The intensity values are estimated at around 1.5× higher after reconstruction. This could be in part due to not only the "smearing" of the flare ribbon emission before correction but also due to overestimation of the bright features. Moreover, the shapes of the spectral lines differ significantly. Before the correction, the line has the typical Hα centrally-reversed shape whereas after, the intensity in the blue part of the line is greatly reduced compared to the red part. The change in line shape may arise from the location of the point selected. The bright point examined here is directly above the sunspot umbra which typically have deep absorption profiles. Therefore, when correcting these pixels, their close proximity to the umbra may lead the network to deciding these locations are part of the umbra at the bluer wavelengths but not at the redder wavelengths. This is something that could be rectified through refined training of the network and having more diversity within the dataset. Similarly for the pre-flare spectrum, the corrected profile has a different shape than the pre-flare corrected with the line core intensities being larger than their surrounding points. This could be due to the fibril structure in the line core regions of the line becoming finer through correction with an initially darker fibril becoming brighter post-correction. This correction is outside of the range of the error bars of the estimate and a more robust approach to error calculation for this model may be needed and is discussed further in Sec. 5.7.

The boxes in Fig. 5.29 (d)–(o) reference the subfields shown in Fig. 5.30. This is to illustrate how well the model recovers small-scale features across the varied field of view. Fig. 5.30 (a)–(d) show the north-easterly part of AR12157 for the pre-flare both in the red wing – panels (a) and (b) – and line core – panels (c) and (d) – of Hα.

Fig. 5.30 (e)–(h) show part of the sunspot umbra/penumbra in the previous format during the rise of the soft X-ray curve of the flare, and Fig. 5.30 (i)–(l) part of the sunspot penumbra during the decay of the flare in the same format. This figure is for illustrative purposes and shows the good recovery of small-scale feature particularly in the line core.

In Fig. 5.31, there are three examples of corrections made to Hα observations with the trained Hα DNN. Fig. 5.29(a) shows the GOES soft X-ray light curves indicating the de facto flare classification. This is annotated to show the three different times the examples are from: in the decay phase of SOL20170906T09:10 at 09:34:26UTC; at the peak of SOL20170906T12:02 at 12:02:26UTC; and in the decay phase of SOL20170906T12:02 at 12:09:11UTC. Fig. 5.31 (d)–(g) show the SOL20170906T09:10 decay phase observation in the Hα blue wing $\Delta\lambda = -1.5$Å, the corrected observation in the blue wing, the observation in the Hα line core, and the corrected observation in the line core, respectively. Similarly, Fig. 5.31 (h)–(k) shows the peak of SOL20170906T12:02 and Fig. 5.31 (l)–(o) shows the decay phase of SOL20170906T12:02.

Each of Fig. 5.31 (d)–(o) is annotated with a "+" and an "x". The "+" indicates a point in a quieter part of the atmosphere with "x" indicating a point on the flare ribbons. Correspondingly, the spectra from these points are shown in Fig. 5.31(b) & (c). In these panels, the downward-facing triangles represent the spectral line before correction for the decay of SOL20170906T09:10, with the upward- facing triangles representing the spectrum following correction; the square points show the line profile before correction for the peak of SOL20170906T12:02, with the diamonds showing the line profile post correction; and the pentagons correspond to the profile before correction for the decay phase of SOL20170906T12:02, with the stars showing the profile post correction. The line profiles in Fig. 5.31 (b) retain their shape when corrected with the intensity values in the wings (and, to a lesser extent, the core) increasing, which we would expect as seeing will effectively "smear" light over many pixels causing a reduction in intensity in one pixel. This correction also preserves asymmetry in the line profile and Doppler shifts that can be seen clearly due to the differences in wing intensities between the blue and red wings and the intensity-averaged line core not being equal to the emitted wavelength, respectively. The line profiles in Fig. 5.31 (c) show three very different line profiles on the flare ribbon depending on the time at which it is observed. For the decay phase of SOL20170906T09:10, the line profile has small changes in the wings after correction

but a larger change towards the line core. The peak of SOL20170906T12:02 spectral line before correction appears as the characteristic twin-peaked Hα profile (with a very broad red wing) with the correction implying that the blue wing should be stronger than that in the raw observations. The decay phase of SOL20170906T12:02 spectral line before and after correction maintains a similar shape with the intensities of the corrected profile being larger at every wavelength point. The increases in intensity of each of these line profiles are to be expected by the same spatial "smearing" effect mentioned earlier but not every pixel would have an increased intensity (otherwise, there would be phantom photons in the observations).

The boxes in Fig. 5.31 (d)–(o) reference the subfields shown in Fig. 5.32. This is to illustrate how well the model recovers small-scale features in the flare ribbons and quieter parts of the Sun. Figure 5.32 (a)–(d) show part of the umbra/penumbra of AR 12673 for the decay phase of SOL20170906T09:10 both in the far blue wing – panels (a) and (b) – and line core – panels (c) and (d) – of Hα. Figure 5.32 (e)–(h) show the eastern flare ribbon in the prior format for the peak of SOL20170906T12:02 and Fig. 5.32 (i)–(l) show the western flare ribbon in the same format for the decay phase of SOL20170906T12:02. This figure is for illustrative purposes and shows the good recovery of small-scale features even when the seeing is particularly bad, as is most prominently seen in the observation of the decay phase of SOL20170906T12:02.

Figures 5.33 & 5.34 are cotemporal counterparts to Figs. 5.29 & 5.30 but for Ca II λ8542 observations corrected by the train Ca II λ8542 DNN. Similarly, Figs. 5.35 & 5.36 are cotemporal counterparts to Figs. 5.31 & 5.32.

5.7 Conclusions and Future Work

A new method for seeing correction of intensity (Stokes I) images in ground-based solar flare observations has been presented. This method can be adapted to other problems after generation of the training set. Also, the ability to reconstruct these observations removing the residual seeing allows for more coherent time series analysis of solar flares. Previously, observations would have to be discarded due to the ambiguity in whether motion is due to the flare or due to the bad seeing conditions. Moreover, this method could be implemented in data pipelines for post-processing of data to produce data products with this residual seeing corrected for.

In this method, a neural network was trained to learn to correct for synthetic seeing (Secs. 5.4 & 5.5), generated by a mathematical model (Sec. 5.2), which is applied

to data observed in good seeing (Sec. 5.3). This network is then applied to real data taken in bad seeing. It was found that the network performs best when the effects of seeing are minimal, as expected. When seeing is worse, the network is still good at recovering large- scale features in the images but, on small scales, the reconstruction is perceptibly less accurate. Moreover, when the seeing is worse, the network seems to overcompensate on the small scales introducing features that are not necessarily physical. On the other hand, the overcompensation may not be due to the bad seeing entirely, as a small instrumental artefact can be seen in Fig. 5.30(k) & (l). This takes the form of a Moiré pattern that may be introduced during the observation or the calibration of the data. Further examples of this pattern appearing can be seen on larger scales in Fig. 5.29(h) & (l). These patterns may cause inaccuracies in the reconstruction by the network. Furthermore, Figs. 5.35 & 5.36 show patterns of noise horizontally across the field of view which can also be attributed to the data reduction process. Characterising and adding these kinds of noise to the training set may help the DNNs be able to deal with them.

An estimate of the error of the network was made by taking the final trained model and applying it to the training and validation sets combined and calculating the mean of the calculated losses by Eq. 5.69. This is a rather ad hoc error that can be improved in the future using the method proposed in Lowe and Zapart (1999) of training a network with an additional input that is the variance of the estimate that the network generates. This will add a robustness to the error calculation and deliver a network capable of providing corrections and their confidence intervals. Another potential improvement to the network could come from implementing a variational inference system using the methods of Tonolini et al. (2019). This would provide the network with the means to sample the posterior distribution of the corrected images to gain more confidence in the reconstructions. This would also increase the confidence in producing intermediate data products using machine learning methods. Like all statistical methods, there needs to be some trust when using a method to generate intermediate data products which can be difficult due to the interpretability issues facing neural networks. In this case, since the overall average error is low, the intermediate data products can be trusted but by taking a more probabilistic approach, a stronger trust can be achieved.

All in all, the model that has been trained produces accuirate corrections on spectroscopic images that would otherwise be plagued with bad seeing. This allows for the study of these flare events at higher time resolution more confidently as the

geometry of the ribbons and their intensities have been corrected for bad seeing. This model only performs seeing correction for Stokes I and in the case of having full spectropolarimetric imaging it is hypothesised that the seeing in Stokes Q, U, and V can be corrected for using the method in Díaz Baso et al. (2019).

6 | Solar Flare Atmosphere Determination Using Invertible Neural Networks

The following chapter is based on and builds upon the work in Osborne et al. (2019). In particular, Secs. 6.2 & 6.3 correspond to work published in Osborne et al. (2019) while Sec. 6.4 contains new analysis and unpublished results.

6.1 Introduction to Inverse Problems in Solar Physics

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 system or the complexity of the physics involved. This leads to the determination of the relationship between observables and system becoming mathematically ill-posed: from a parameter estimation point of view, this means that the marginalised posterior distributions of the parameters of interest will be multimodal as demonstrated by Asensio Ramos et al. (2007). This can be shown visually using set theory with the example of the inverse problem being investigated in this chapter (Fig. 6.1). On the left of Fig. 6.1 are sets of solar atmospheric parameters that produce the intensity observables (right of Fig. 6.1). The function f is known as the *forward process* and is the mapping that produces observables and can be characterised, in solar physics (depending on

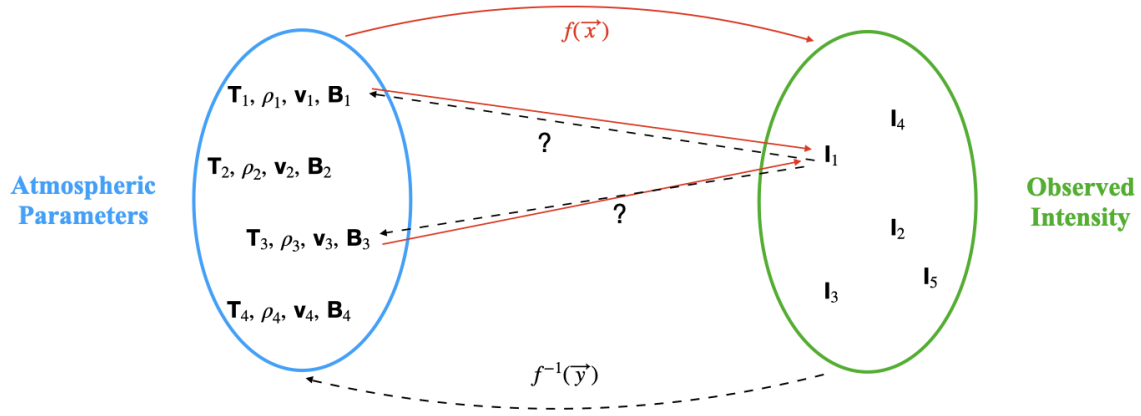

Figure 6.1: This plot shows that while the function f from the physical system x mapping to the observables y i.e. $y = f(x)$ is well-defined for each set of possible parameters within the physical system, the inverse $x = f^{-1}(y)$ is not due to information about the system being lost in producing the observables. Here the example is of the physical parameters describing a system being mapped to the observed data collected by telescopes with the function f describing the process of radiative transfer producing the observations.

the context), by various radiative transfer and (magneto)hydrodynamics processes. The *inverse process* is the mapping f^{-1} that maps the observables back to the atmospheric quantities that produced them. The difficulty in formulating this mapping comes from the inverse mapping not being *bijective* (or one-to-one). A bijective function is one in which for a domain X and codomain Y the function connecting the two maps *exactly one* input to output. Due to the information lost in the forward process, the inverse process cannot be bijective. This is illustrated in Fig. 6.1 as two sets of atmospheric parameters can result in the same observables which causes ambiguity when investigating the inverse.

Currently in solar physics, there are two main ways to infer atmospheric parameters from spectroscopic observations and the method used depends on how the observed spectral line is formed. There is the method of "direct" inversions for optically thin spectral lines (Hannah and Kontar, 2012; Cheung et al., 2015) and "forward modelling" for optically thick spectral lines. This chapter is concerned with observations and parameter estimation from optically thick spectral lines thus a comparison will be drawn with forward modelling techniques[1]. Forward modelling is an iterative process where the equations of (sometimes polarised) radiative transfer are solved for a given set of atmospheric parameters to give emulated observables. These syn-

[1]However, many of the statements henceforth can be applied to direct inversion methods too.

thetic observables are then compared to the true observables that the atmospheric parameters are to be estimated for using a least squares metric to measure "closeness". If the value of this metric is high, i.e. the simulated observables are not a good fit, then the atmospheric parameters are modified before the radiative transfer problem is solved again producing new synthetic observables to be compared to the real observables. Atmospheric parameters are updated by defining "nodes" at specific points along them e.g. if one of the parameters was how the temperature of the plasma changes with height above the surface of the Sun then n nodes would be chosen at different heights and the parameters would be varied at these points depending on the value of the metric. For a bad fit, the nodes are updated using the Levenberg-Marquardt algorithm of least squares minimisation (Levenberg, 1944; Marquardt, 1963, similar to how SGD updates the learnable parameters in neural networks but also including information about the second derivative of the least squares metric) with the values between the nodes (since the atmospheric parameters are continuous variables) estimated by interpolation.

This method has seen great success in a variety of areas of solar physics such as analytic methods employing the Milne–Eddington approximation for frequency-independent opacity in a local thermodynamic equilibrium (LTE) atmosphere (Skumanich and Lites, 1987) allowing for the investigation of the magnetic properties of sunspots and their umbra/penumbra; the Stokes Inversion based on Response functions (SIR) code (Ruiz Cobo and Del Toro Iniesta, 1992) which solves the problem of local thermodynamic equilibrium polarised radiative transfer including the Zeeman effect by making use of the response functions of various spectral lines[2]; the HAZEL code (Asensio Ramos et al., 2008) which solves non-local thermodynamic equilibrium (non-LTE) polarised radiative transfer including the Hanle effect for accurate simulation of neutral helium spectral lines; the NICOLE code (Socas-Navarro et al., 2000, 2015) which solves non-LTE polarised radiative transfer for a variety of spectral lines to investigate the chromosphere; and the newer STockholm inversion Code (STiC; de la Cruz Rodríguez et al., 2019) which builds upon NICOLE and incorporates the RH code (Uitenbroek, 2001) to solve the radiative transfer problem allowing for the analysis of multiple spectral lines with a complicated atmospheric structure. The inner workings of these inversion codes are succinctly described in

[2]A response function is how the outgoing intensity at a particular wavelength reacts to perturbations in the atmosphere and has been shown to be an invaluably useful diagnostic when exploring these problems (Milić and van Noort, 2017; Osborne, 2021a)

the corresponding papers and a curious reader is referred there for further information.

While this method of forward modelling to iterate towards a solution is a sound one, there are three key issues that this chapter aims to address. Firstly, none of these "traditional" inversion codes were designed with solar flares in mind: they all assume hydrostatic equilibrium when solving the radiative transfer problem which is an assumption that breaks down for flares. As such, the use of these methods for solar flare data is discouraged but can give indicative results when used under certain assumptions as demonstrated by Kuridze et al. (2017, 2018) where the NICOLE inversion code is used to estimate atmospheric parameters for flaring observations. However, to understand the full extent of the flaring lower solar atmosphere, the full coupled system of radiation hydrodynamics (RHD) must be solved[3]. RHD codes which model flares have been used in a different definition of "forward modelling" where the simulations will be run for a variety of different flare energy inputs and a best match to observations is performed "by eye" giving the resultant solution (Kuridze et al., 2015; Kerr et al., 2016; Simões et al., 2017; Kowalski et al., 2017; Costa et al., 2016). Efforts to consolidate the two forms of forward modelling have been unsuccessful leading to the need for the proposed solution in this chapter.

Secondly, with ever-increasing resolution in all dimensions – spatial, temporal, spectral – (discussed in Chap. 4), the need for efficient, automated algorithms has never been greater. Traditional inversion techniques can be slow to converge for complicated observations and currently work one pixel at a time. This can lead to weeks of computational power to estimate the parameters over a large field of view. Being able to estimate these parameters quickly given the volume of data from instruments is only increasing is a major bonus to implementing machine learning when doing inversions.

Finally, as discussed earlier, the inversion mapping is ill-defined meaning that there is no way to know if the converged solution is correct. Disambiguation techniques are not natively built-in to traditional inversion codes and are often not considered when estimating parameters. This means that all results obtained through inversions have some inherent uncertainty associated with them[4]. The method proposed in this chapter has ambiguity resolution built-in by construction under the as-

[3]Really it should be RMHD (radiation magnetohydrodynamics) but current codes only simulate RHD for flares.

[4]see https://github.com/ivanzmilic/toy_model_inversion for why this is the case

sumption that the algorithm learns the inverse process sufficiently from the training and validation dataset.

Motivated by the first and second reasons, a new automated technique for learning how to do inversions is constructed and applied to solar flare data which can also deal with the third reason, namely an *invertible neural network* (INN; Ardizzone et al., 2019). Note that, the results of traditional inversion codes have been improved upon using DNNs trained on inversions to create a starting atmosphere for traditional inversion codes providing better convergence (Gafeira et al., 2021). This typically works better as a more realistic atmosphere is used as a starting point for the LM algorithm rather than a semi-empirical model which is the norm. Further note that Asensio Ramos and Diaz Baso (2019) used several traditional DNNs to learn inversions of spectropolarimetric data for spectral lines whose parameters had unimodal posterior distributions as discussed in Asensio Ramos et al. (2007).

6.2 Invertible Neural Networks

Supervised deep learning that has been discussed so far in this thesis is unsuited to the task of learning the inverse mapping due to its ill-defined nature. Learning the mapping from produced observables to atmospheric parameters will more often than not generalise poorly when using traditional DNNs since the function is not deterministic. In fact, Ardizzone et al. (2019) claim that training such a DNN would result in either the DNN picking one of the possible solutions at random or an average of all possible solutions. In either case, this can lead to physically incompatible solutions. On the other hand, DNNs would work very well in learning the forward model from atmospheric parameters to observations, however, inverting a traditional DNN made up of convolutional and fully-connected layers is non-trivial as the sets of learnable parameters are not guaranteed to be non-singular (that is, the learnable parameters for each layer can be thought of as a matrix, Θ, which would be non-singular if $\det\Theta \neq 0$). If any of the trained model's layers have a singular set of learnable parameters then the inverse of that neural network cannot be computed. Both of these reasons lead to the choice of using an invertible neural network (INN) to model the inverse problem.

6.2.1 Constructing RADYNVERSION

RADYNVERSION is the INN trained on flare simulations from the 1D radiation hydrodynamics code RADYN. The training of this INN is discussed in Sec. 6.2.2 while this section focuses on its architecture and inner workings.

INNs circumvent the issue of the inverse function being ill-defined in two ways: by the introduction of a latent space Z learned to encapsulate the information lost in the forward process to provide a one-to-one mapping for the inverse; and by construction of the network allowing both the forward and inverse problem to be learned without having to invert large, possibly singular, matrices of learnable parameters. The latent space Z represents the space of all information lost in the forward process, such that a sample from the latent space, $z \in Z$ combined with the observation $y \in Y$ can be mapped to the correct input parameters $x \in X$. This is shown in the same set theory illustration in Fig. 6.2. Through the Cartesian product of the observation space Y and the latent space Z, there now exists a bijective mapping $g : [y, z] \leftrightarrow x$ such that the inverse process g and the forward process g^{-1} are deterministic. g^{-1}, by design, will track which element of the latent space corresponds to the correct set of observations to be produced. The form of this latent space is 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_{true} with the information lost in the forward process. RADYNVERSION is then trained in such a way (see Sec. 6.2.2) 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. That is, each training sample during the forward process will map to values in the latent space drawn from the true latent distribution. Then when the inverse is performed, latent values are drawn from the unit Gaussian, with these random draws compared to the true latent variables and a loss function minimised between them such that the forward process produces latent variables drawn from the same latent distribution that is sampled by the inverse process. The choice of drawing from the unit multivariate Gaussian is an arbitrary one. It is true that any distribution could be used here, but a Gaussian is chosen because it is smooth and continuous.

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

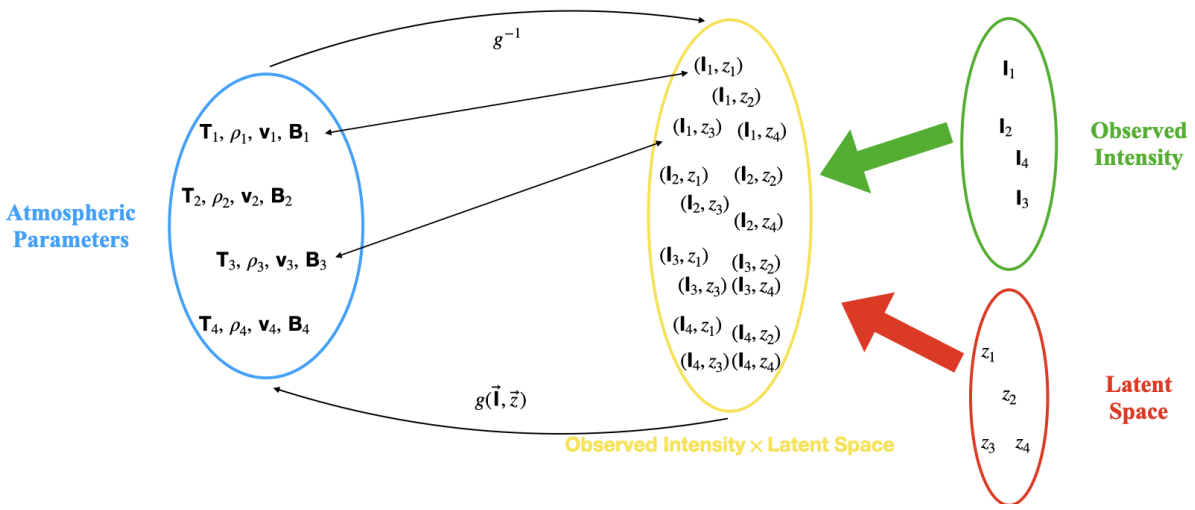

Figure 6.2: Modification of Fig. 6.1 including the latent space introduced by the invertible neural network. This means the system is now defined by a bijective function $g : [y, z] \leftrightarrow x$ where an exact solution to the inverse problem can be found given sufficient training of the INN.

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, the inverse function is learned for free. This is due to the reversibility of the blocks, illustrated in Fig. 6.3. The layers are based on the form first presented in Ardizzone et al. (2019). The input x is split into two equals parts $[x_1, x_2]$ that are propagated through the forward direction of the layer. 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 Fig. 6.3. There is now a simple relation between the input and output for this layer

$$y_1 = x_1 \otimes \exp\left(s_2\left(x_2\right)\right) + t_2\left(x_2\right), \tag{6.1}$$

$$y_2 = x_2 \otimes \exp\left(s_1\left(y_1\right)\right) + t_1\left(y_1\right), \tag{6.2}$$

where $\otimes$ denotes the element-wise multiplication of two matrices, 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 simply expressed as

$$x_2 = (y_2 - t_1(y_1)) \oslash \exp(s_1(y_1)), \tag{6.3}$$

$$x_1 = (y_1 - t_2(x_2)) \oslash \exp(s_2(x_2)), \tag{6.4}$$

where $\oslash$ denotes the element-wise division of two matrices. This defines a setup in which the inverse is easily calculable. This means that the only problem now is learning the mapping from the true latent distribution to the multivariate normal latent distribution to make sure that RADYNVERSION produces the correct inversion. 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. As such FCNs (see Sec. 2.2.1) are used to approximate the optimal affine transformation. The FCNs comprising the functions s_i, t_i are initialised randomly with a fixed seed, contain four hidden layers and every activation besides the last in each FCN is a *leaky* ReLU (the last activation is a normal ReLU described in Sec. 2.1.1). A leaky ReLU differs from a normal ReLU (Eq. 2.4) by

$$\phi(x) = \max(0.01x, x). \tag{6.5}$$

Thus rather than setting all negative values to zero, negative values in a leaky ReLU have a smaller but non-zero effect on the gradients. This can help avoid the dying ReLU problem (Sec. 2.1.1). 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).

The RADYNVERSION INN is comprised of five stacked affine coupling layers. This means that the network is dependent on 20 deep neural networks to approximate the inverse problem. Between each subsequent affine coupling layer, there is what is known as a permutation layer. This introduces channel mixing into RADYNVERSION 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 the INN in a random but fixed way (Dinh et al., 2014, 2016). Each permutation is different from the previous. This

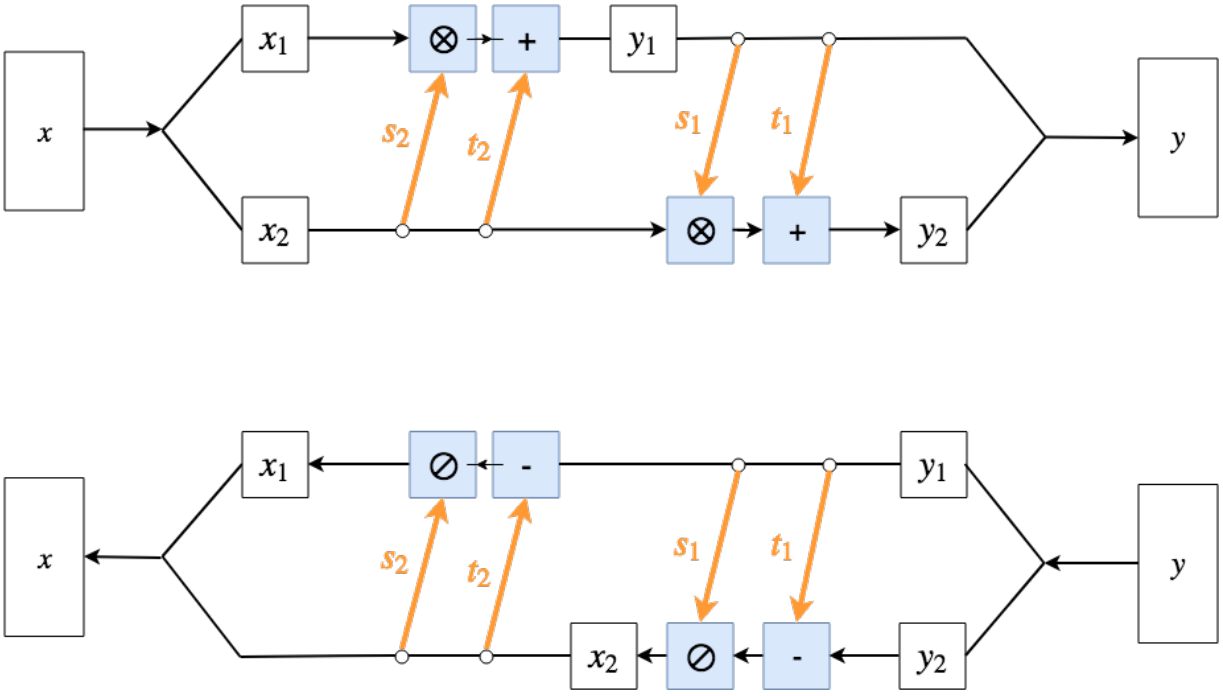

Figure 6.3: Affine coupling layer showing the affine transformation between input and output for the forward process (top) and reverse process (bottom). These form the building blocks of our INN, as they are easily invertible.

will increase the generalisation properties of RADYNVERSION. The architecture of RADYNVERSION is shown in Fig. 6.4. The flow of the forward model is shown by the black arrows, and the flow of the inverse is shown by the cyan arrows.

6.2.2 Training RADYNVERSION

As mentioned in Sec. 6.1, the forward modelling technique for fitting solar flare data consists of simulating a flare with a 1D RHD code and looking for the best fit to a small area of observations. There are several state of the art forward modelling codes for solar flares such as RADYN (Carlsson and Stein, 1992, 1997; Allred et al., 2005, 2015); FLARIX (Varady et al., 2010; Heinzel et al., 2016); and HYDRAD (Bradshaw and Cargill, 2013). Due to its widespread acceptance in analysing optical spectral lines (Kuridze et al., 2015; Kerr et al., 2016; Capparelli et al., 2017), UV spectral lines (Brown et al., 2018; Kerr et al., 2019b,a) and white light and UV continua (Simões et al., 2017) and the existence of a preexisting database of models

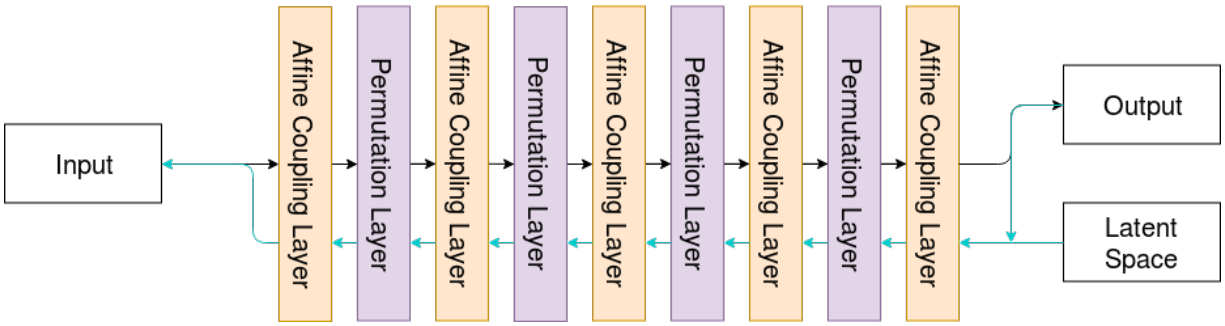

Figure 6.4: RADYNVERSION architecture. There are five affine coupling layers with a permutation layer sandwiched between each pair of affine coupling layers (four in total). The forward process mapping the input to the output is illustrated by the black arrows. The inverse process mapping a combination of the output and the latent space to the input is illustrated by the cyan arrows.

compiled as part of the FCHROMA project[5], RADYN is chosen as the forward model to be learned by the INN shown in Fig. 6.4. Hence, RADYNVERSION is born.

All of the simulations in the FCHROMA RADYN grid start from a modified VAL-C (Vernazza et al., 1981) quiet Sun atmosphere which then has the flare energy deposited by means of an electron beam with varying characteristics. Each simulation lasts for 50s with timesteps saved every 0.1s (resulting in 501 timesteps per simulation) with the energy deposition beginning at t = 0.0s with the beam profile being a symmetric triangular pulse (in all cases) peaking at t = 10.0s and lasting until t = 20.0s. The distribution of electron energies within the beam are modelled as a power law described by three parameters:

1. The total energy flux of the beam which is one of four values $\{3 \times 10^{10}, 1 \times 10^{11}, 3 \times 10^{11}, 1 \times 10^{12}\}$erg cm^{-2}.

2. The low energy cutoff, E_c, which takes one of the values $\{10, 15, 20, 25\}$keV.

3. The spectral index of the distribution, δ, which can be one of six values $\{3, 4, 5, 6, 7, 8\}$.

This results in 96 simulations in the grid. However, two thirds of the simulations with a total energy flux of 1×10^{12}erg cm^{-2} did not converge and thus were left out of the training set, reducing the number of simulations used to 80[6]. Most of these

[5]Available from https://star.pst.qub.ac.uk/wiki/doku.php/public/solarmodels/start.

[6]In fact, one can note an interesting trend in the simulations that did not converge: fewer simulations converge for lower values of E_c regardless of the value of δ with high E_c simulations only not converging when δ is larger.

simulations contain 501 timesteps (some contain 500) and each timestep has its atmospheric parameters and spectra extracted and used to make input, output pairs for training. There are now approximately 40,000 pairs of atmospheric parameters and spectral lines to train the INN on. 20% of the training data chosen at random is used as the validation dataset. The relationship chosen for RADYNVERSION to learn is that of the electron number density of the plasma (n_e), electron plasma temperature (T) and bulk flow velocity (v) mapping to the spectral lines of Hα and Ca II λ8542, to line up with the sets of observations from the SST/CRISP available. More spectral lines could be included in the forward process but would require cotemporal and cospatial observations between all selected lines which is why only Hα and Ca II λ8542 are chosen. Moreover, other atmospheric parameters such as the level populations or the electron beam parameters could be included for estimation but the three chosen were done so as they are quantities typically estimated in traditional inversions so an easy comparison can be drawn.

To prepare the RADYN data for training RADYNVERSION, the atmospheric parameters are interpolated from RADYN's adaptive spatial grid to a fixed spatial grid to avoid the INN learning changes in the adaptive grid as being relevant to the synthesis of the spectral lines. The observations of interest are going to be affected most by the plasma parameters in the (upper) chromosphere, therefore, the static height grid the parameters are interpolated onto has 45 points spaced linearly apart from $z = 0 - 3.5$Mm (where $z = 0$Mm is defined by the point in the adaptive grid where $\log_{10}\tau_{5000} = 0$ before heating and $z = 3.5$Mm is roughly the height of the transition region before heating) with 5 points used to represent the corona spaced exponentially from $z = 3.5$Mm to $z = 10$Mm. Furthermore, to reduce the dynamic range in the plasma parameters (and thus accelerate training of the INN), the parameters undergo the following mappings before training

$$n_e \mapsto \log_{10} n_e, \tag{6.6}$$

$$T \mapsto \log_{10} T, \tag{6.7}$$

$$v \mapsto \text{sign}(v)\log_{10}(|v| + 1). \tag{6.8}$$

The spectral lines are interpolated onto a grid of 30 linearly spaced wavelength points with the half-widths of the lines in RADYN being 1.4Å and 1Å for Hα and Ca II λ8542, respectively. These lines are then normalised to the range [0–1] by dividing by the maximal intensity across both profiles. An example of the training data is

shown in Fig. 6.5. Henceforth, the concatenation of the atmospheric parameters combining to give the input to RADYNVERSION will be called x, with the output being the concatenation of the spectral lines called y, z will refer to the true latent space of the system. The estimate of the spectral lines obtained by the forward process in the INN will be $\tilde{y}$ and $\tilde{x}$ will be the estimate of the atmospheric parameters obtained by the inverse process in the INN.

RADYNVERSION is trained differently to the other neural networks discussed in Chaps. 4 & 5 due to the inclusion of the latent space. Rather than a fully supervised learning approach, an INN is part of the family of semi-supervised learning algorithms (Tab. 2.1) which is reflected by the training set including known mappings (which atmospheric parameters map to which spectral lines) and unknown mappings (the true form of the latent space required to produce unique inverse mappings). To jointly learn the forward and the inverse mappings, bidirectional training is employed. This is when an epoch now consists of the input data being passed forwards through the network to calculate an output with learnable parameter updates estimated by the gradient of the loss function, as well as the output data together with draws from the latent space being passed backwards through the network to calculate the input with its own set of learnable parameter updates estimated through a different loss function. The learnable parameters are then updated by both sets of gradients in each epoch. Both the forward and inverse processes are constrained by the linear combination of two loss functions (but the total forward and inverse losses are considered independent from one another)

$$\mathcal{L}_{\mathrm{f}} = \lambda_1 \mathcal{L}_{\mathrm{MSE},f} + \lambda_2 \mathcal{L}_{\mathrm{MMD},f}, \tag{6.9}$$

$$\mathcal{L}_{\mathrm{i}} = \lambda_3 \mathcal{L}_{\mathrm{MSE},i} + \xi(n) \lambda_4 \mathcal{L}_{\mathrm{MMD},i}, \tag{6.10}$$

where $\mathcal{L}_f$ refers to the total forward loss and $\mathcal{L}_i$ refers to the total inverse loss. Both losses are a combination of an MSE loss (described in Sec. 5.5, Eq. 5.71) and the maximum mean discrepancy (MMD) loss (Gretton et al., 2012). The MMD is a statistic used for computing the similarity 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 mapping (see Appendix of Osborne et al. (2019) and Chap. 7 of Osborne (2021a) for more information). The

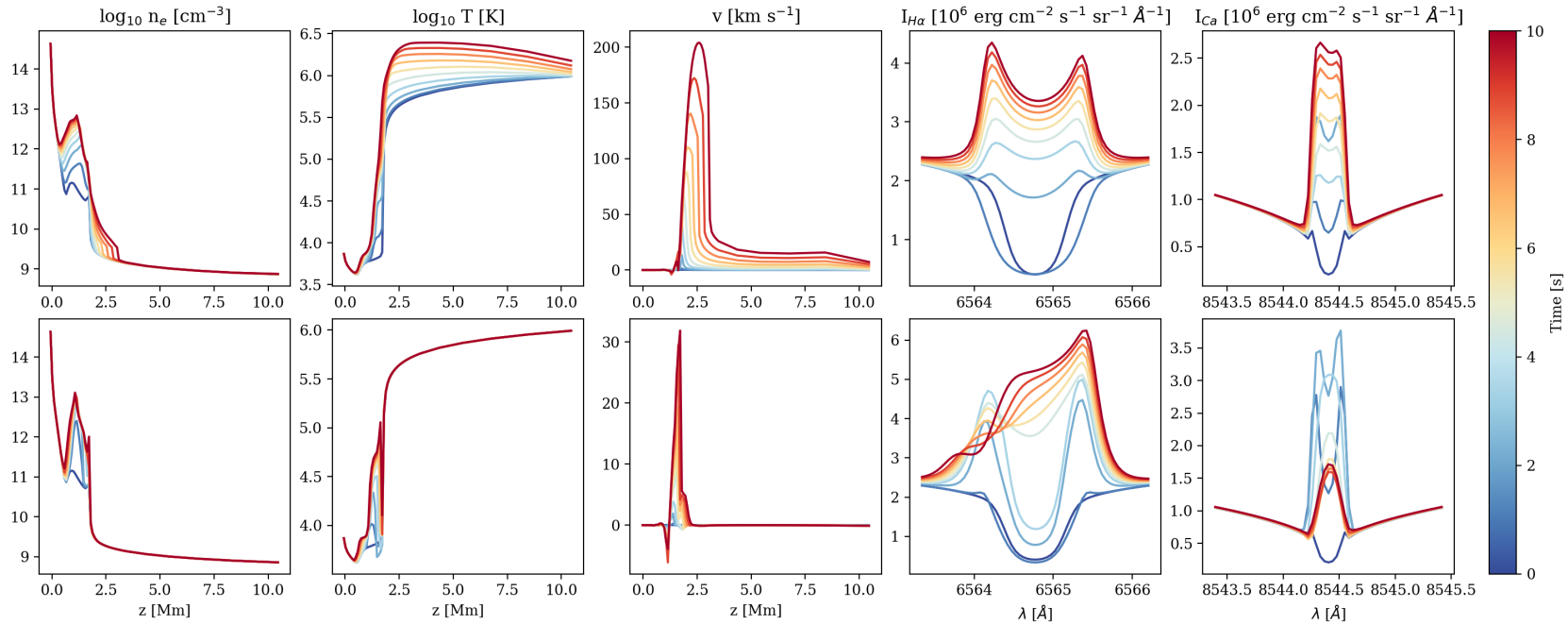

Figure 6.5: Examples of the training data used for RADYNVERSION. The top row shows examples at different times (from left to right) of the logarithm of the electron number density, the logarithm of the electron plasma temperature, the plasma flow velocity as functions of height above the initial position of the $\log_{10}\tau_{5000} = 0$ line; and the Hα and Ca II λ8542 spectral lines for a RADYN simulation whose electron beam used for heating the atmosphere is characterised by total energy flux 1×10^{11}erg cm^{-2}, $E_c = 10$keV and $\delta = 3$. The bottom row shows the same as the top row but for a RADYN simulation with electron beam parameters of total energy flux 3×10^{11}erg cm^{-2}, $E_c = 25$keV and $\delta = 8$.

MMD between two probability distributions P, Q can be written

$$\text{MMD}^2 = ||\mu_P - \mu_Q||^2_{\mathcal{F}} = \langle \mu_P, \mu_P \rangle_{\mathcal{F}} + \langle \mu_Q, \mu_Q \rangle_{\mathcal{F}} - 2\langle \mu_P, \mu_Q \rangle_{\mathcal{F}}, \tag{6.11}$$

where μ_P, μ_Q are the expectation vectors of the distributions P, Q for samples drawn from these distributions evaluated on the reproducing kernel Hilbert space (RKHS) and $\langle \cdot, \cdot \rangle_{\mathcal{F}}$ represents the inner product on the RKHS. For the two probability distributions being compared, it can be assumed that the randomly drawn samples come from the same underlying set of numbers X. Therefore, a function known as a feature map (different from the CNN feature map) can be defined as the mapping $\chi : X \to \mathcal{F}$ which maps probability distribution draws to values in the RKHS. Then for two random draws from the space $z_1, z_2 \in X$, a kernel k can be defined such that

$$k(z_1, z_2) = \langle \chi(z_1), \chi(z_2) \rangle_{\mathcal{F}}, \tag{6.12}$$

which now gives a closed form for which two random draws can be compared on the RKHS. The reason for the choice of comparing the distributions on an RKHS is that the kernel will always be recovered when finding the inner product of two features. This means that the inner product of the distributions can be written as

$$\langle \mu_P, \mu_Q \rangle = \mathbb{E}_{P,Q}[k(z_1, z_2)], \tag{6.13}$$

where $z_1 \sim P$ and $z_2 \sim Q$. This then gives a closed form for the MMD which can be computed given the choice of a kernel function, k,

$$\text{MMD}^2 = \mathbb{E}_P[k(z_1, z_1)] + \mathbb{E}_Q[k(z_2, z_2)] - 2\mathbb{E}_{P,Q}[k(z_1, z_2)], \tag{6.14}$$

The kernel used for calculating the MMD loss is the same as that of Tolstikhin et al. (2017) and Ardizzone et al. (2019), the inverse multiquadric (IMQ) kernel

$$k_\alpha(z_1, z_2) = \frac{\alpha^2}{\alpha^2 + \|z_1 - z_2\|^2}, \tag{6.15}$$

as it has been found most effective for comparing sample quality in inverse problems. The choice of the value for α is then a hyperparameter in the system. Ardizzone et al. (2019) showed that the sum of IMQ kernels with different α (due to the properties of the RKHS over which the MMD is defined, this sum is also a kernel) is the best

kernel for their examples. This did not work when training RADYNVERSION as it was difficult to find a set of values for α that were effective in training the latent distribution to match the expected distribution without dependence on the spectral lines. By plotting the MMD for the same input and output samples but for different values of α, it was found that the biased sample estimate of the MMD between two random variables drawn from similar but perturbed distributions produced a peak for certain values of α. The maximum value of $\mathrm{MMD}^2(\alpha)$ is then computed for a set of α during training and α itself is updated 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.

In learning the forward process, RADYNVERSION is attempting to approximate the function $x \mapsto [y, z]$. The forward MMD loss, $\mathcal{L}_{\mathrm{MMD},f}$, compares $[\tilde{y}, z]$ with $[y, \mathcal{N}(0, \mathcal{I}_N)]$. During backpropagation, the gradients pertaining to $\tilde{y}$ from the MMD loss are ignored so that the learnable parameters can learn the mapping from $z \mapsto \mathcal{N}(0, \mathcal{I}_N)$ without interfering with the learning of the mapping $x \mapsto y$ from the MSE loss term, $\mathcal{L}_{\mathrm{MSE},f}$. The convergence of this forward MMD loss ensures that z is independent of y. The inverse process is trained similarly to learn the mapping $[y, z] \mapsto x$. The vector of y and the latent variables z generated by the forward iteration is propagated through the network in reverse, and an MSE loss is applied between $\tilde{x}$ and x. Another vector of y with latent variables drawn from $\mathcal{N}(0, \mathcal{I}_N)$ is also propagated in reverse, and an MMD loss is computed between $\tilde{x}$ and x. This second MMD loss serves to ensure that sampling the true latent distribution is equivalent to sampling the normal distribution (while taking into account internal variability within the true distribution).

The $\xi(n)$ term in Eq. 6.10 is a function used in the initial stages of training the INN to stop the inverse MMD loss (which initially has large dominating gradients) from diverging the INN from the correct solution. This takes the form

$$\xi(n) = \left(\min\left(\frac{n}{N_{\mathrm{fade}}}, 1\right)\right), \tag{6.16}$$

where n is the current epoch number and N_{fade} is the number of epochs over which the gradients of the inverse MMD should be suppressed. In training RADYNVERSION, $N_{\mathrm{fade}} = 800$ epochs. Over these initial 800 epochs, $\lambda_1 = \lambda_3 = 4000$, $\lambda_2 = 900$, $\lambda_4 = 1000$. Then every 400 epochs, up to 4800 epochs, λ_1 & λ_3 are increased by

1000; which was then repeated every 600 epochs until 12000 epochs has passed. The model with the best performance on the validation data was chosen as the trained RADYNVERSION model – this was obtained after 11400 epochs.

Furthermore, the Adam optimiser was used in training (see Sec. 5.5) with $\beta_1 = \beta_2 = 0.8$ and $\epsilon = 1 \times 10^{-6}$. The learning rate η is initially set to 1.5×10^{-3} and is decayed by a factor of $0.004^{1/1333}$ every 12 epochs resulting in a final learning rate of 3.38×10^{-5} after 11400 epochs. Minibatch training was used here also (as described in Sec. 5.5) with a minibatch size of 500 and 20 minibatches per epoch.

Results of RADYNVERSION on a validation example are shown in Fig. 6.6 for the forward process and Fig. 6.7 for the inverse process. Figure 6.6 shows near perfect synthesis of the spectral lines from the set of atmospheric parameters given to RADYNVERSION (synthesised lines are solid and ground truth are dashed lines in the bottom row). Figure 6.7 shows a two dimensional histogram where the dashed line are the true atmospheric profiles and each entry in the histogram are the estimated parameters for the ground truth spectral lines inverted with random samples from the normal latent distribution (this inversion is performed 20,000 times i.e. 20,000 different sets of latent variables). Given how well the atmospheric parameters are recovered by the inverse process, one can conclude that sampling the normal latent distribution is equivalent to sampling the true latent distribution and thus the inverse process has also been learned.

6.3 Single Pixel Inversions

The next step is to apply RADYNVERSION to real spectroscopic data, with the intention of characterising the atmosphere that produced it, and eventually learning about the physics of a flaring chromosphere. Since RADYNVERSION's training is only constrained by the formation of Hα and Ca II λ8542 which occurs in the chromosphere (line cores) and upper photosphere (line wings), the atmospheric parameters estimated below around 2 Mm are focused on. Since Hα and Ca II λ8542 are not sensitive to changes in the corona[7] not much significance is attributed to the few points there.

The spectral lines chosen to invert come from the M1.1 SOL20140906T17:09 flare

[7]The conditions in the corona *will* influence the conductive flux responsible for secondary heating in the lower atmosphere. Although, due to RADYNVERSION treating each timestep independently, any conclusions drawn about the conductive heating in later timesteps would be speculative.

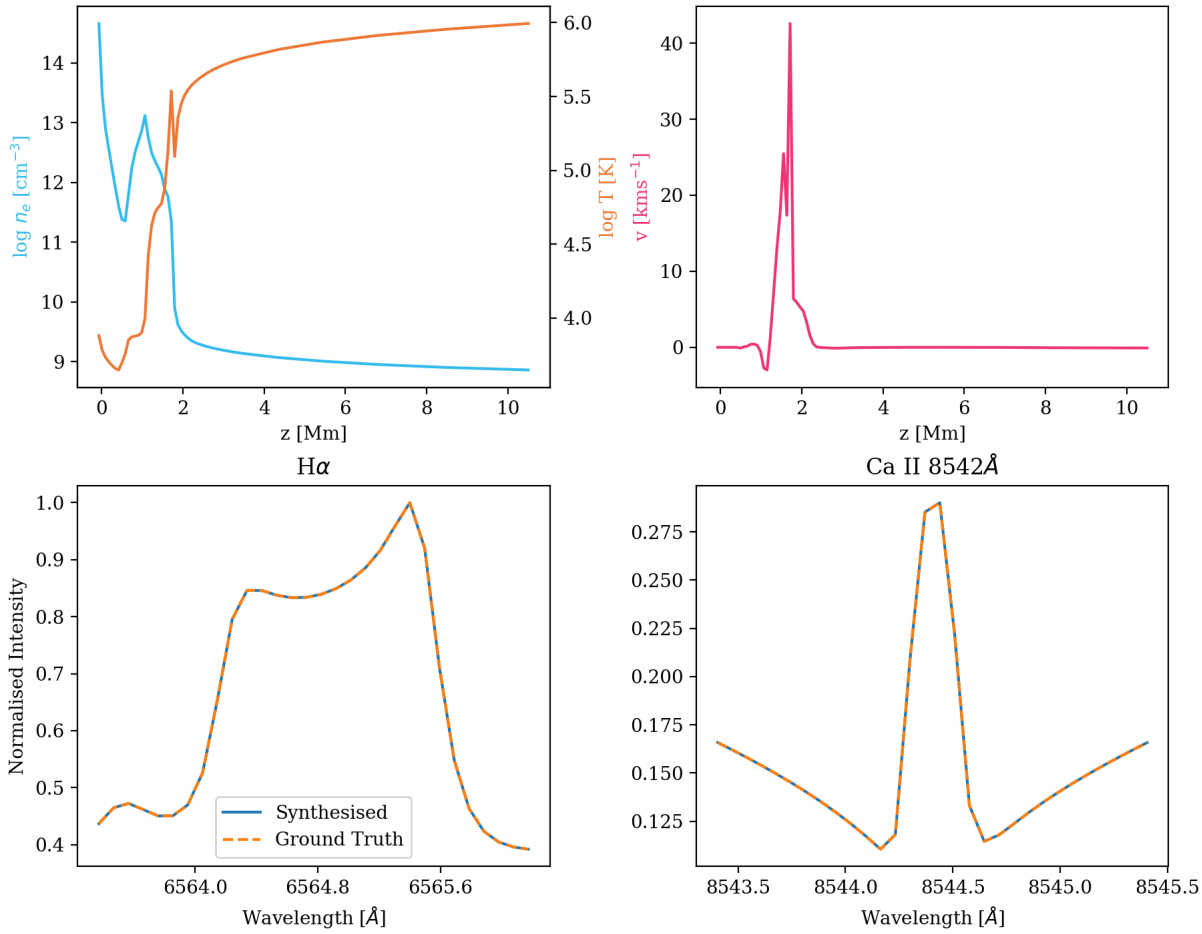

Figure 6.6: An example showing the synthesis of spectral lines by the learned RADYNVERSION model on the validation dataset. The top two panels show the input plasma parameters as a function of height above the photosphere previously unseen during training and the bottom two panels show the comparison of the synthesised lines and the true spectral line profiles. This shows very good agreement between the ground truth observables and synthesised observables by the forward model.

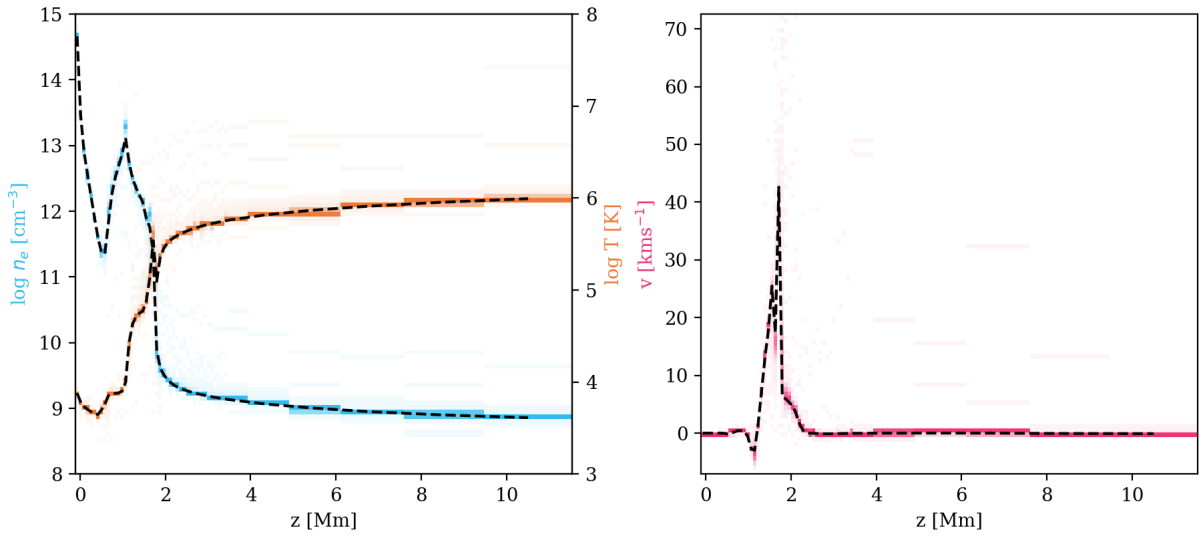

Figure 6.7: The spectral lines shown in the bottom row of Fig. 6.6 are inverted by the learned RADYNVERSION inverse with a variety of different random draws from the latent space. This provides a two-dimensional histogram describing the recovered atmospheric parameters where the dashed lines refer to the ground truth parameters to be recovered. This shows that sampling the normal latent distribution recovers the profiles of the atmospheric parameters very well indicating a learned mapping between the true latent space and the normal latent space.

observed by CRISP described in Sec. 5.3. Due to the joint data reduction via the CRISPRED pipeline, it is assumed that the intercalibration of the two lines is reliable such that the relative intensities between the two spectral lines are physically meaningful which RADYNVERSION assumes for the joint normalisation. The aim of analysing a single pixel is to determine the properties of the flaring velocity field responsible for the asymmetry observed in the spectral lines (since the lines being analysed would be symmetric around the line core for a static atmosphere Canfield and Gunkler, 1984; Fang et al., 1993; Cheng et al., 2006). This is to demonstrate the method that will be used in Sec. 6.4 over an extended area to determine global relationships between line asymmetry and flare velocity field at different times throughout the flare. The complex nature of the flare velocity field is likely linked to chromospheric condensation (Ichimoto and Kurokawa, 1984; Wülser and Marti, 1987) and evaporation (Neupert, 1968; Fisher et al., 1985; Graham and Cauzzi, 2015). 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 could be due to emission from upflowing plasma or absorption by down-

Spectral Line	λ_0 [Å]	σ [Å]	λ_{0B} [Å]	λ_{0R} [Å]	$\delta\lambda$ [Å]	I_B/I_R
Hα	6564.573	0.618	6563.478	6565.662	0.322	1.035
Ca II λ8542	8544.433	0.412	8543.504	8545.340	0.288	0.949

Table 6.1: The results of calculating the intensity-averaged line core and standard deviation from moments analysis and using these values to calculate the asymmetries in the spectra from Fig. 6.8.

flowing plasma. Likewise a red asymmetry could be due to emission from downflowing plasma or absorbing upflowing plasma.

To calculate the asymmetries in the profiles, a technique similar to that described in Mein et al. (1997); De Pontieu et al. (2009); Kuridze et al. (2015) is used. Namely, the following expressions are used to find the integrated blue and red intensities for a spectral line

$$I_B = \int_{\lambda_{0B}-\delta\lambda}^{\lambda_{0B}+\delta\lambda} I(\lambda)\ \mathrm{d}\lambda, \tag{6.17}$$

$$I_R = \int_{\lambda_{0R}-\delta\lambda}^{\lambda_{0R}+\delta\lambda} I(\lambda)\ \mathrm{d}\lambda, \tag{6.18}$$

where λ_{0B} and λ_{0R} are the centre wavelengths of the blue and red wings, respectively. $\delta\lambda$ is the width of the wing from its centre wavelength. The wings are defined as being the area starting one standard deviation, σ, away from the intensity-averaged line core, λ_0. λ_0 and σ can be calculated via (Jeffrey et al., 2016)

$$\lambda_0 = \frac{\int I(\lambda)\lambda\ \mathrm{d}\lambda}{\int I(\lambda)\ \mathrm{d}\lambda}, \tag{6.19}$$

$$\sigma^2 = \frac{\int I(\lambda)(\lambda-\lambda_0)^2\ \mathrm{d}\lambda}{\int I(\lambda)\ \mathrm{d}\lambda}. \tag{6.20}$$

This defines the end of the blue wing and the start of the red wing as $\lambda_0 - \sigma$ and $\lambda_0+\sigma$, respectively. The midpoint between the bluest wavelength sampled and $\lambda_0-\sigma$ defines λ_{0B}. Similarly, λ_{0R} can be found as the midpoint between $\lambda_0 + \sigma$ and the reddest wavelength sampled. The half-width of the wings, $\delta\lambda$, is then the difference between the wing centre and wing edge. A measure of the asymmetry of the spectral line can then be given by the ratio I_B/I_R, i.e. if there is a red asymmetry then this ratio will be less than 1 and vice versa for a blue asymmetry.

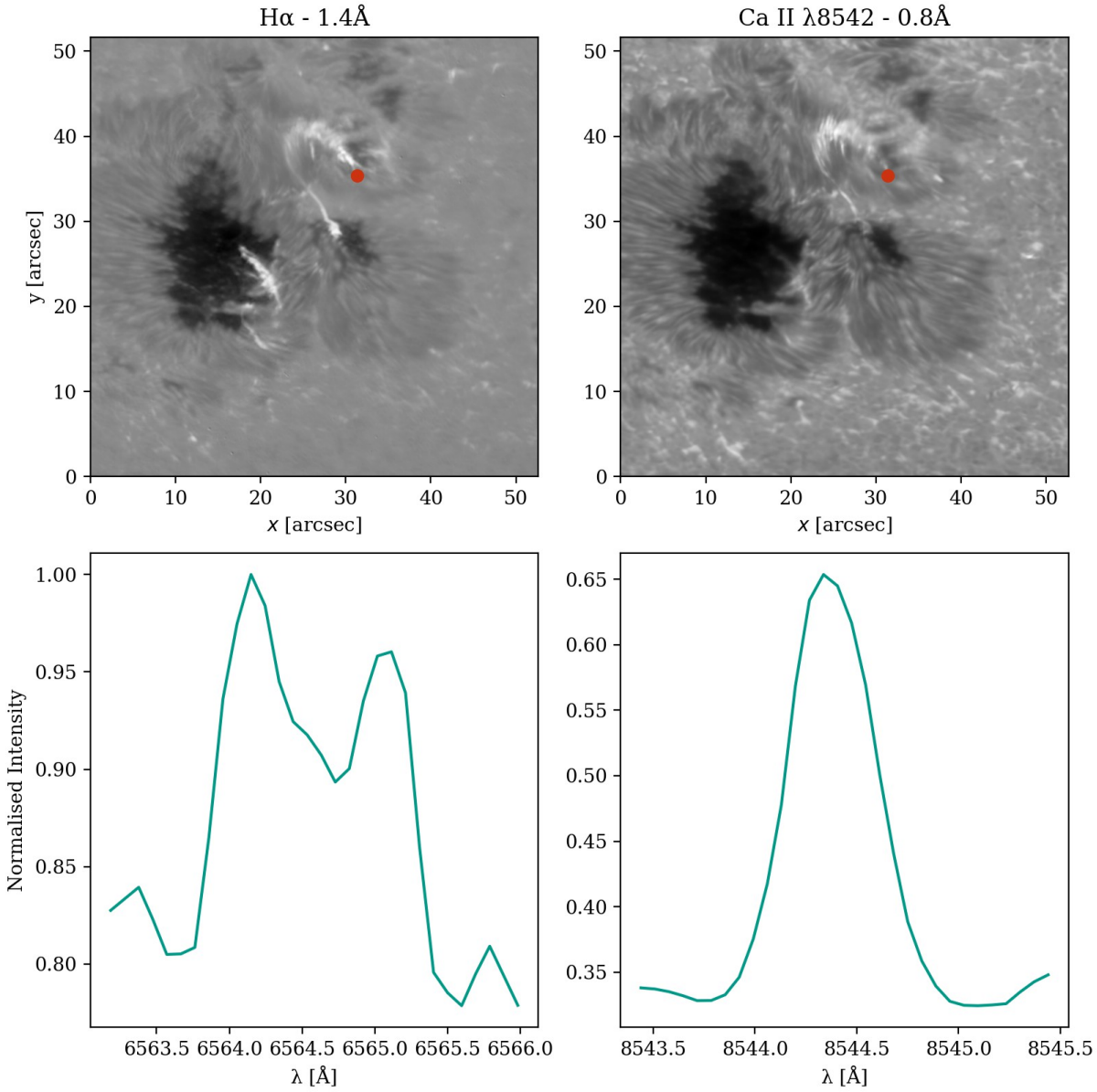

Figure 6.8: The data used for the single pixel inversion example. The top row shows Hα and Ca II λ8542 CRISP blue wing data from 16:56:13UTC of the M1.1 SOL20140906T17:09 solar flare with the circular point indicating the spatial location where the spectra are taken from for inversion. The spectra themselves at these points are shown in the bottom row, normalised to each other for input to RADYNVERSION.

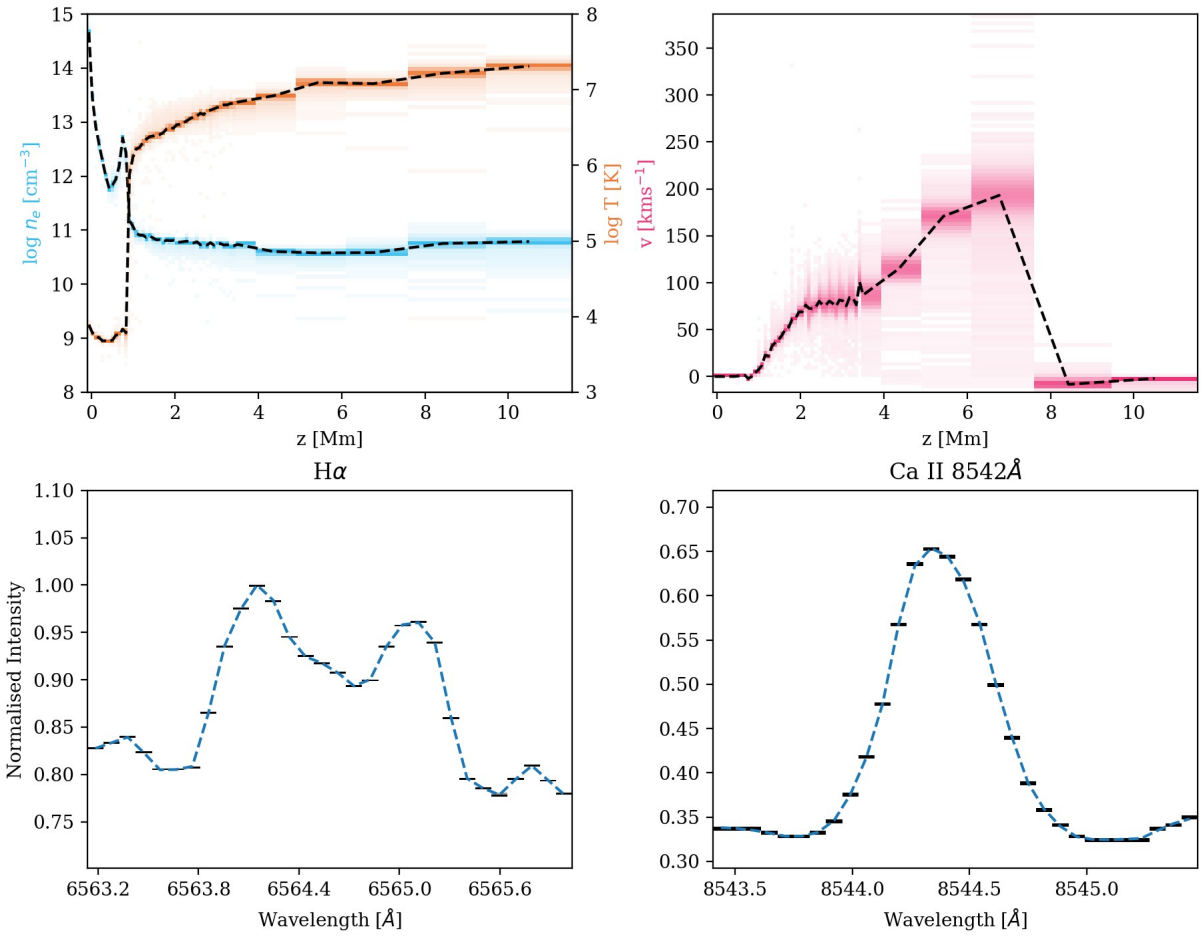

Figure 6.9: The results of inverting the data in Fig. 6.8. The top panels show the atmospheric parameters obtained from the inversion where the latent space is sampled 20,000 times and the results are plotted as two-dimensional histograms. The top left panel shows the electron density and temperature plotted on log scales, and the top right panel shows the flow velocity in our plasma. 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 obtained from the inversions.

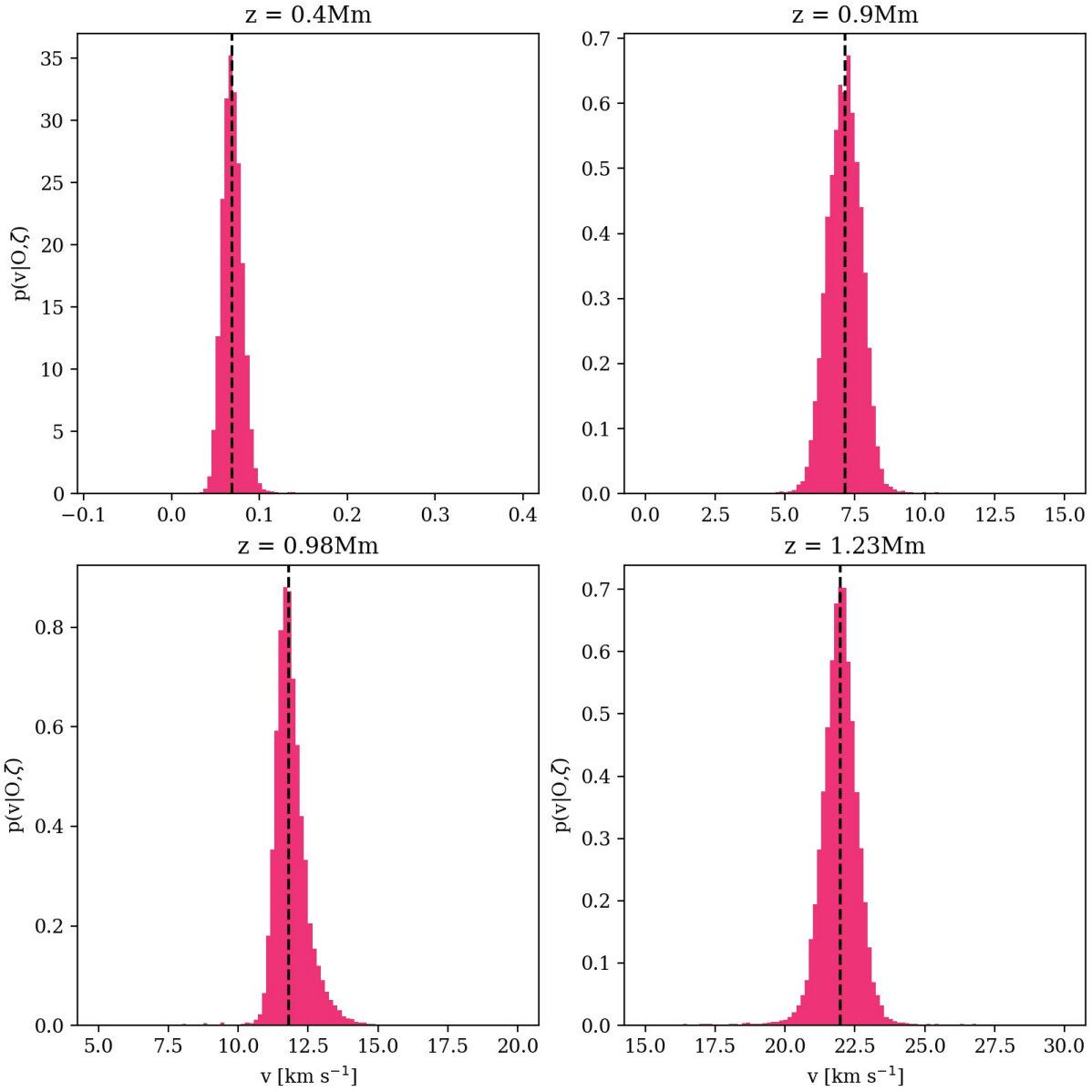

Figure 6.10: The one-dimensional histogram of the flow velocities obtained from applying RADYNVERSION to the data in Fig. 6.8 at specified heights in the atmosphere (that is, these are slices of the top right panel in Fig. 6.9). These histograms give an insight into what properties of the velocity field are responsible from the observed asymmetries in the spectra (calculated in Tab. 6.1) as they are shown at the formation heights of the wings and line cores of Hα and Ca II λ8542 as demonstrated by Kuridze et al. (2015) and Kerr et al. (2016), respectively.

The point inverted in this section is shown by the circular point indicated in the blue wing images of the spectral lines in the top row of Fig. 6.8. The bottom row of Fig. 6.8 shows the spectra to be inverted already normalised with respect to each other to serve as input to RADYNVERSION. The Ca II λ8542 line is characteristic of the profile during a flare as it is fully in emission. The line peak is also slightly blue-shifted with respect to the intensity-averaged line core by 3.23km s^{-1}. Calculation of the asymmetry of this line reveals a red asymmetry of ~5.1% (last column of Tab. 6.1). The Hα line is centrally reversed (typical in some regions in flares) with the blue horn having a larger intensity than the red horn pointing to a blue asymmetry in the Hα which is corroborated by the calculations of the asymmetry shown in Tab. 6.1 (blue asymmetry of ~3.5%). Due to its centrally reversed nature, it is difficult to pinpoint the motion of the core of the Hα line in the same way that can be done for Ca II λ8542.

The results of inverting these lines with 20,000 draws from the latent space are shown in Fig. 6.9. The results of the inversions are plotted as two-dimensional histograms (top panels of Fig. 6.9). The dashed lines show the median profile for the parameters. This gives an approximation to the true solution. The bottom panels of Fig. 6.9 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 produced 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).

According to Kerr et al. (2016), the Ca II λ8542 line during solar flares is formed between 0.2 Mm and 1.0 Mm above the base of the height grid used here. In particular, the wings of the line are formed between 0.2–0.4 Mm, with the core formed between 0.9–1.0 Mm (with the core region defined to be within ±0.3Å of the vacuum wavelength). Exploring these regions in the inverted atmospheres can lead to understanding of the physics involved in producing the blue-shifted line core and the red asymmetry observed in the Ca II λ8542 spectral line. To do so, the posterior distributions of the velocity specific heights in the atmosphere is studied and shown at four different heights ($z = \{0.4, 0.9, 0.98, 1.23\}$ Mm) in Fig. 6.10. In Fig. 6.10, the black dotted lines correspond to the median velocity value for that height. Each of these distributions points to upflows being responsible for the behaviour in the observed spectral line. This would indicate that in the region of the line wing formation since

the asymmetry is red there would be absorbing material being evaporated, while in the region of line core formation there would be some emitting material being evaporated. This is reasonable on the timescales of the flare as these observations are taken just after the flare onset, and before the flare SXR peak, where it would be expected for evaporation to be the dominant process as the material in the lower atmosphere is beginning to be heated by the flare energy.

Similarly for Hα, Kuridze et al. (2015) estimate that during this flare in particular, the line is formed below heights of 1.2 Mm with the wings forming below 0.95 Mm and the core forming above this. Since there is a strong blue asymmetry in this line, Figs. 6.9 & 6.10 point to upflowing emitting material being responsible for this asymmetry. A further confirmation of upflowing emitting material in the formation heights of Hα and the Ca II λ8542 line core is the temperature. The temperature at these heights is around the temperature required to give the species enough energy to emit these photons ($\sim 1 - 1.5 \times 10^4$ K).

Similar analysis will now be applied to entire flare ribbon structures to identify the physics occurring in these regions.

6.4 Flare Ribbon Identification and Asymmetries

6.4.1 Finding Flare Ribbons Using Unsupervised Machine Learning

Before the comparison of the flare velocity field and the asymmetries in the flare ribbon spectral lines can be made, a method for identifying the flare ribbons must be formulated. Fletcher et al. (2004); Fletcher (2009) previously employed manual identification of flare ribbon footpoints in UV Transition Region And Coronal Explorer (TRACE) observations. This method involved selecting a bright point in one observation and tracking its evolution in time by fitting a two-dimensional Gaussian in the initial time frame and drawing a small "tracking box" centred on the Gaussian centroid. At the next time step, the same Gaussian fit is reapplied and a centroid within the distance of the tracking box is taken to be the same feature having undergone motion. A new tracking box is then centred on the new centroid. This requires the tracking box to be large enough such that the same source will still be within the bounds of the old box for identification but small enough to only contain one source. It is these conditions on the tracking box which makes this an

ill-suited method for the SST/CRISP data. The CRISP data has $\sim 10\times$ higher spatial resolution than TRACE (Handy et al., 1999) thus leading to sources taken to be single sources in the TRACE data likely to be identified as multiple sources in the CRISP data. This makes it harder to use the tracking box method as it must be smaller and there are also many more sources to track. Moreover, for the purposes of this study, the identification of the ribbon structures as a whole is more important than the identification of individual flare bright points. Therefore, a new flare ribbon identification technique based on using clustering algorithms is proposed.

Clustering algorithms are a class of *unsupervised* machine learning techniques. Unsupervised machine learning differs from supervised machine learning in that the task to be learned is not necessarily known. While with supervised learning there is a defined input and output to learn from, unsupervised learning provides only the input and hyperparameters to an algorithm with the result analysed to see if it fits the user's specification. As such, hyperparameter tuning in unsupervised learning techniques is very important. There are two main families of unsupervised machine learning algorithms: clustering and dimensionality reduction. Dimensionality reduction algorithms aim to use statistical properties of the training data to find optimal representations of the data to then make further data analysis simpler, these include techniques such as principal component analysis. Clustering algorithms sort the training data into groups based on the properties of the data. Clustering algorithms have seen a rise in popularity in solar physics with an algorithm called k-means proving particularly useful in analysing singly-ionised magnesium (Mg II) spectral line profiles in solar flares (Panos et al., 2018; Panos et al., 2021; Panos and Kleint, 2021) and in the compression of data for use in future space-based solar missions (Ivanov et al., 2021). Below, a generalisation of the k-means clustering algorithm known as a *Gaussian mixture model* (GMM) is used in conjunction with the *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN) algorithm to identify and separate flare ribbons in CRISP data (this workflow is inspired by Sisti et al., 2021, who use k-means clustering followed by DBSCAN to identify reconnecting current sheets in 2D MHD simulations.).

A Gaussian mixture is a function comprising of a set of K Gaussians where each Gaussian $k \in \{1, \ldots, K\}$ is described by a mean $\vec{\mu}_k$, a covariance matrix Σ_k and a so-called mixing coefficient π_k (where $\sum_{k=1}^{K} \pi_k = 1$ to ensure the normalisation of the total probability distribution). A GMM uses each Gaussian in the mixture as a cluster and assigns each of the data to one of the clusters with a certain probabil-

ity that it belongs to each cluster. These probabilities are used to update the means and covariance matrices of the Gaussians using the expectation-maximisation (EM) algorithm to find the maximum likelihood estimate that the Gaussian mixtures describe the clusters required. Mathematically, the quantity of interest is the probability that a data point $\vec{x}$ belongs to cluster k which can be written as $p(z_k \mid \vec{x}_i)$ where z_k is a latent variable equal to 1 when $\vec{x}_i$ comes from Gaussian k and zero otherwise. There are then a set of K latent variables $\vec{z} = \{z_1, \ldots, z_K\}$ one for each cluster. Via Bayes' theorem, $p(z_k \mid \vec{x}_i)$ can be written

$$p(z_k \mid \vec{x}_i) = \frac{p(\vec{x}_i \mid z_k)p(z_k)}{p(\vec{x}_i)}. \tag{6.21}$$

The prior probability of observing any point from one of the clusters k is equal to the mixing coefficient $p(z_k) = \pi_k$. Moreover, the probability of observing any point from any of the clusters can be written

$$p(\vec{z} = \{z_1, \ldots, z_K\}) = \prod_{k=1}^{K} p(z_k) = \prod_{k=1}^{K} \pi_k, \tag{6.22}$$

since the probabilities of any point belonging to any Gaussian are statistically independent. $p(\vec{x}_i)$ is found by marginalising over the joint distribution $p(\vec{x}_i, \vec{z})$

$$p(\vec{x}_i) = \int p(\vec{x}_i, \vec{z})\mathrm{d}\vec{z} = \int p(\vec{x}_i \mid \vec{z})p(\vec{z})\mathrm{d}\vec{z} = \sum_{k=1}^{K} p(\vec{x}_i \mid z_k)p(z_k), \tag{6.23}$$

since the probabilities of observing a point and observing something from a cluster are statistically independent (i.e. one can be done without the other). $p(\vec{x}_i|z_k)$ is just the probability that $\vec{x}_i$ was drawn from Gaussian k meaning that Eq. 6.23 can be written as

$$p(\vec{x}_i) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\vec{x}_i \mid \vec{\mu}_k, \Sigma_k). \tag{6.24}$$

Now Eq. 6.21 becomes

$$p(z_k \mid \vec{x}_i) = \frac{\pi_k \mathcal{N}(\vec{x}_i \mid \vec{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\vec{x}_i \mid \vec{\mu}_j, \Sigma_j)}. \tag{6.25}$$

The log likelihood for the point $\vec{x}_i$ to come from Gaussian k can then be maximised

by finding the optimal solutions for $\vec{\mu}_k$, Σ_k and π_k. However, $\vec{x}_i$ is just one of the points to assign to a cluster ($X = \{\vec{x}_1, \ldots, \vec{x}_i, \ldots, \vec{x}_N\}$) so in reality, the likelihood function becomes

$$p(X \mid z_k) = \prod_{i=1}^{N} p(\vec{x}_i|z_k) = \prod_{i=1}^{N} \pi_i \mathcal{N}(x_i \mid \vec{\mu}_k, \Sigma_k), \tag{6.26}$$

and maximising the log of Eq. 6.26 becomes infeasible from an analytical standpoint. Therefore a numerical method, the EM algorithm, is used to estimate the properties of the Gaussians to cluster each data point where the likelihood of each data point will be maximal for exactly one Gaussian cluster.

There are four steps to the EM algorithm:

1. Each Gaussian representing a cluster is described by the mixing coefficient, mean vector and covariance matrix, therefore, the set of these parameters represent the learnable parameters in the system. There parameters must be initialised accordingly and this is typically achieved by initialising them by fitting an initial k-means to the data and using the k-means result as a starting point for the GMM (this is possible as k-means is actually a specific case of a GMM where the covariance matrices are diagonal and equal for each Gaussian).

2. Expectation step: Compute the expected value of the log likelihood of the optimal parameters θ, log of Eq. 6.26, with respect to the current set of posterior distributions for the latent variables.

$$Q(\theta = \{M, S, \Pi\}, \theta_t = \{M_t, S_t, \Pi_t\}) = \mathbb{E}_{p(\vec{z} \mid X;\theta_t)} \left[\ln p(X \mid z_k; \theta)\right], \tag{6.27}$$

   where $M = \{\vec{\mu}_1, \ldots, \vec{\mu}_K\}$ is the set of mean vectors, $S = \{\Sigma_1, \ldots, \Sigma_K\}$ is the set of covariance matrices and $\Pi = \{\pi_1, \ldots, \pi_K\}$ is the set of mixing coefficients. The subscript t variables refer to the estimates of the parameters at iteration t with the non-subscript referring to the optimal values of said parameters.

3. Maximisation step: An update for the learnable parameters is computed by finding the maximum value for θ such that

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta, \theta_t), \tag{6.28}$$

4. Steps 2 and 3 are iterated until a convergence criterion is met.

The estimated optimal clustering for the data can now be found. GMM is preferred to k-means here due to its ability to model more complex distributions of data which was found to be important when identifying flare ribbons.

The data the GMM is trained on is from the M1.1 SOL20140906T17:09 CRISP flare dataset described in Sec. 5.3 – particularly, the observations at 16:54:18UTC. A separate GMM is trained for Hα and Ca II λ8542. The spectral lines are interpolated to 15 points using the Weno interpolation method (Janett et al., 2019) implemented in the `Weno4Interpolation` Python package (Osborne, 2021b) to allow for use on other datasets. It was found that the clustering worked better when the spectral line profiles had their preflare profiles subtracted and so the observations from 16:38:09UTC were subtracted before clustering (examples are shown in Fig. 6.13). Only points that always remain within the field of view can have their subtracted profiles clustered so the spatial mask for this dataset from Millar et al. (2021) is used to extract only omnipresent pixels. Examples of the data used to train the GMMs are shown in Figs. 6.11 & 6.12 – that is the spectrum of each point that is omnipresent in the field of view is used to train the GMM for these observations.

The number of clusters to be used can be estimated by evaluating a statistic known as the *Bayesian information criterion* (BIC).

$$\mathrm{BIC}(K) = K \ln(N) - 2 \ln \hat{L}, \tag{6.29}$$

where $\hat{L}$ is the estimate of the maximum likelihood from the converged model for a number of clusters K. The BIC gives a metric for detecting overfitting in the number of clusters. Increasing the number of clusters can increase the maximum likelihood but will lead to overfitting if the value of K is too large. Overall, a smaller value of the BIC indicates a better fit, however with real data being noisy, there is not always necessarily an increase in the BIC after overfitting but rather a plateauing of the BIC value. In this case, it is prudent to look at the gradient of the BIC with respect to K to get an idea of what the optimal number of clusters are. This is the case for clustering the data here with the optimal number of clusters found to be K = 9 for Hα and K = 11 for Ca II λ8542. These numbers are decided upon as after this number of clusters, the gradient of the BIC remains approximately constant (Fig. 6.14).

The cluster means can be plotted to give an idea of the types of profiles contained

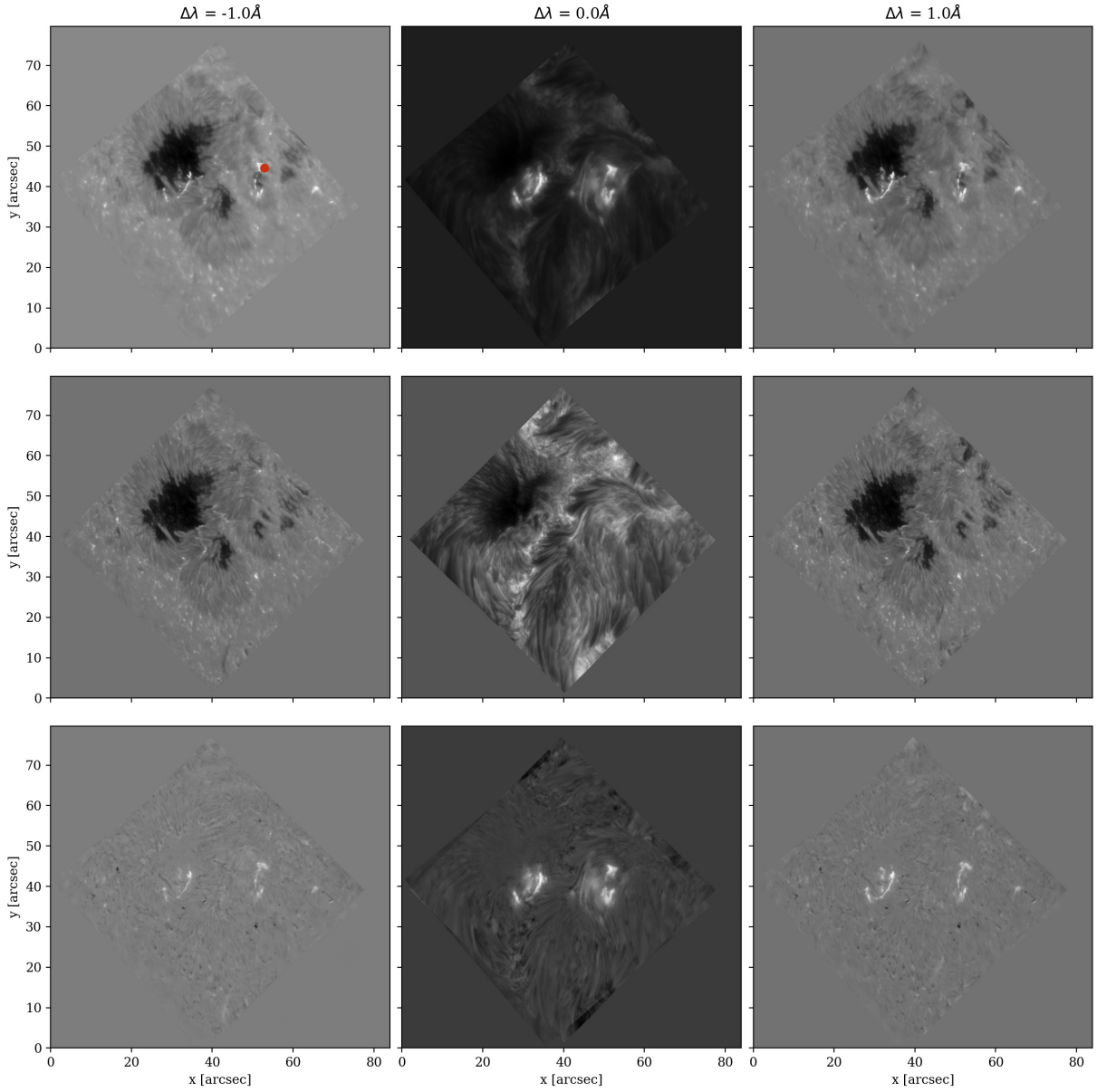

Figure 6.11: Examples of preflare subtracted Hα images from the M1.1 SOL20140906T17:09 used to train the GMM for Hα. The top row shows observations from during the flare at 16:54:13UTC. The middle row shows the preflare observation that is subtracted before clustering is performed, namely the observation from 16:38:04UTC. The bottom row shows the preflare subtracted 16:54:13UTC observation. The left column shows the blue wing of Hα, the middle column shows the line centre and the right column shows the red wing.

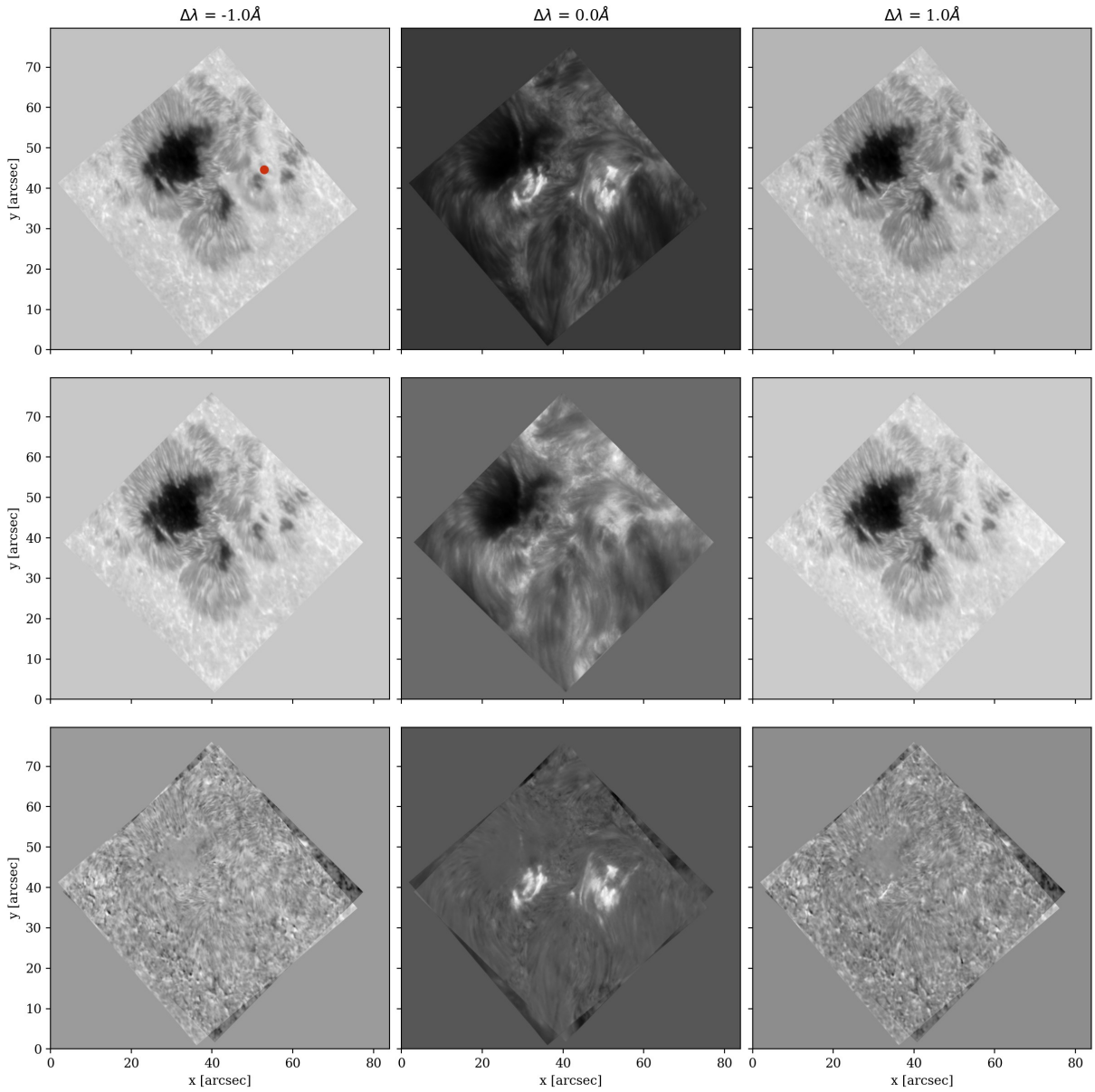

Figure 6.12: Equivalent figure to Fig. 6.11 but for Ca II λ8542.

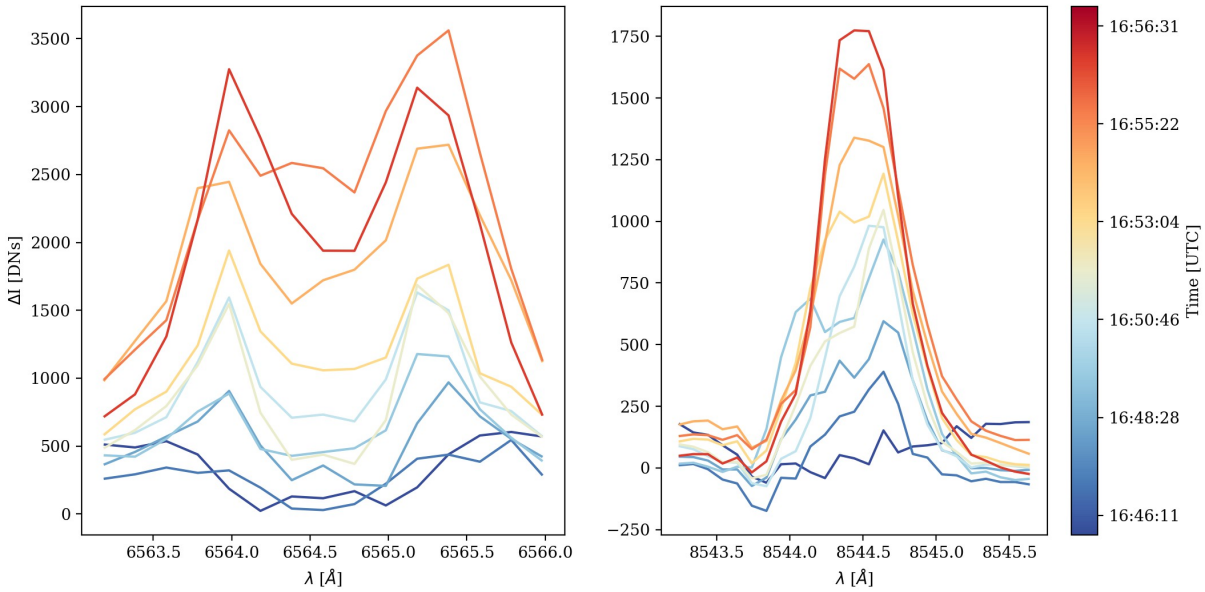

Figure 6.13: The preflare subtracted Hα and Ca II λ8542 spectra for the point indicated by the circle in the top left panel of Figs. 6.11 & 6.12. This is shown for certain points in time that are all clustered by the trained GMM. Here, $\Delta I = I - I_{\text{pf}}$ where I_{pf} is the preflare intensities at each wavelength point.

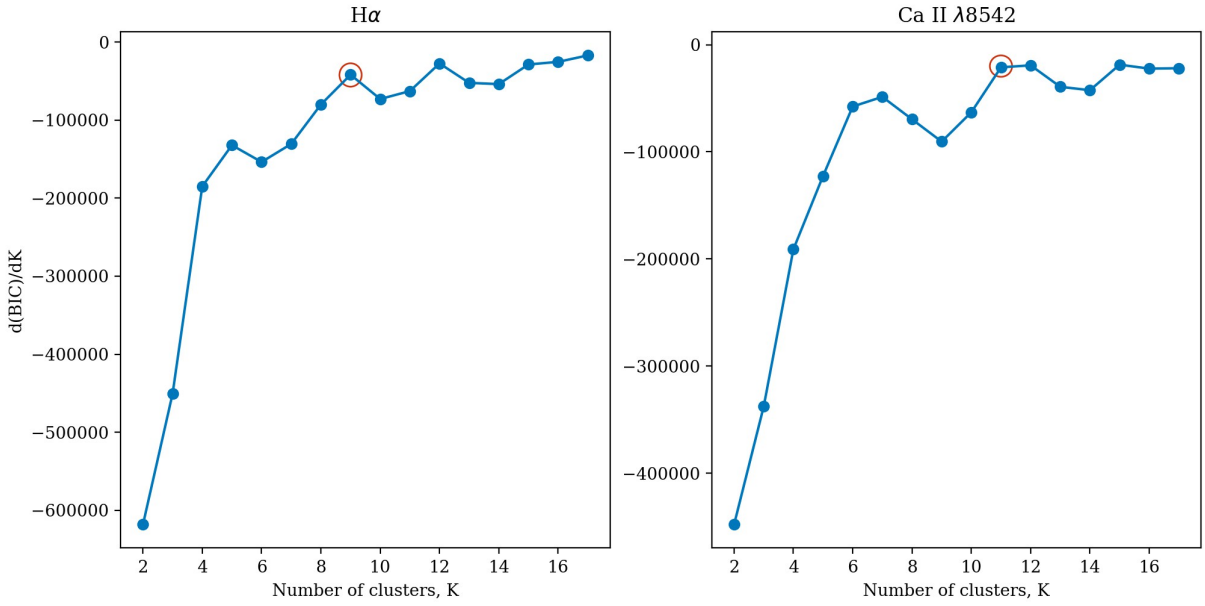

Figure 6.14: The gradient of the Bayesian Information Criterion (BIC) with respect to the number of clusters in a GMM. Where the gradient of the BIC plateaus indicates the optimal number of clusters for a GMM before overfitting. It is shown here that the optimal number of clusters is 9 for Hα and 11 for Ca II λ8542.

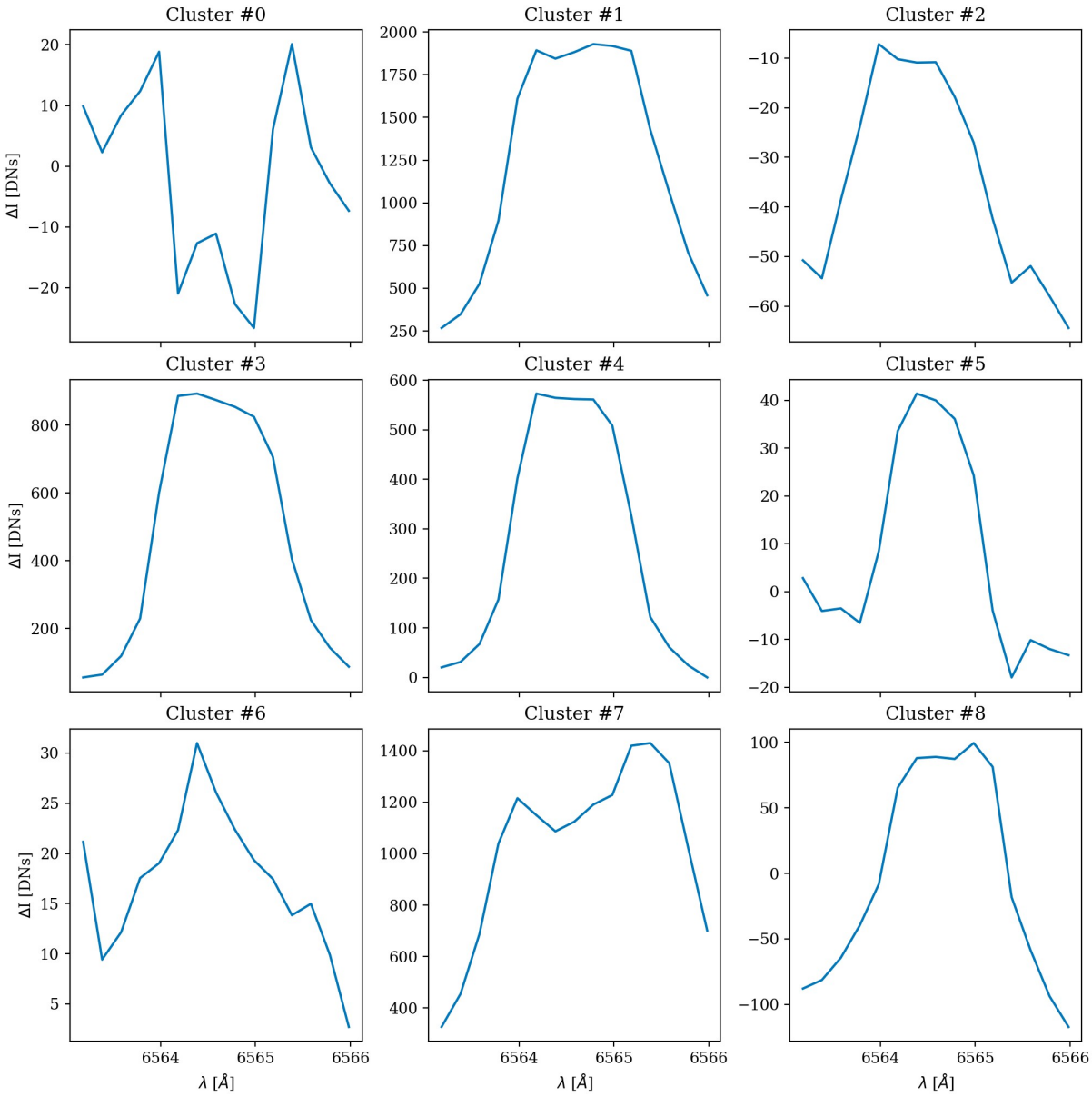

Figure 6.15: The Hα cluster means for the trained GMM. This indicates the clusters of interest for flare ribbons. Cluster #1 clearly contains the brightest points in the flare ribbons and this represents the group where the Hα line is fully in emission and without central reversal. Clusters #3 & #4 represent similar profiles to cluster #1 but with less of an excess in intensity which are taken to be the outer regions of the flare ribbon structures. Cluster #7 represents Hα emission lines with a central reversal, indicating a complex velocity field in the region where these lines are formed.

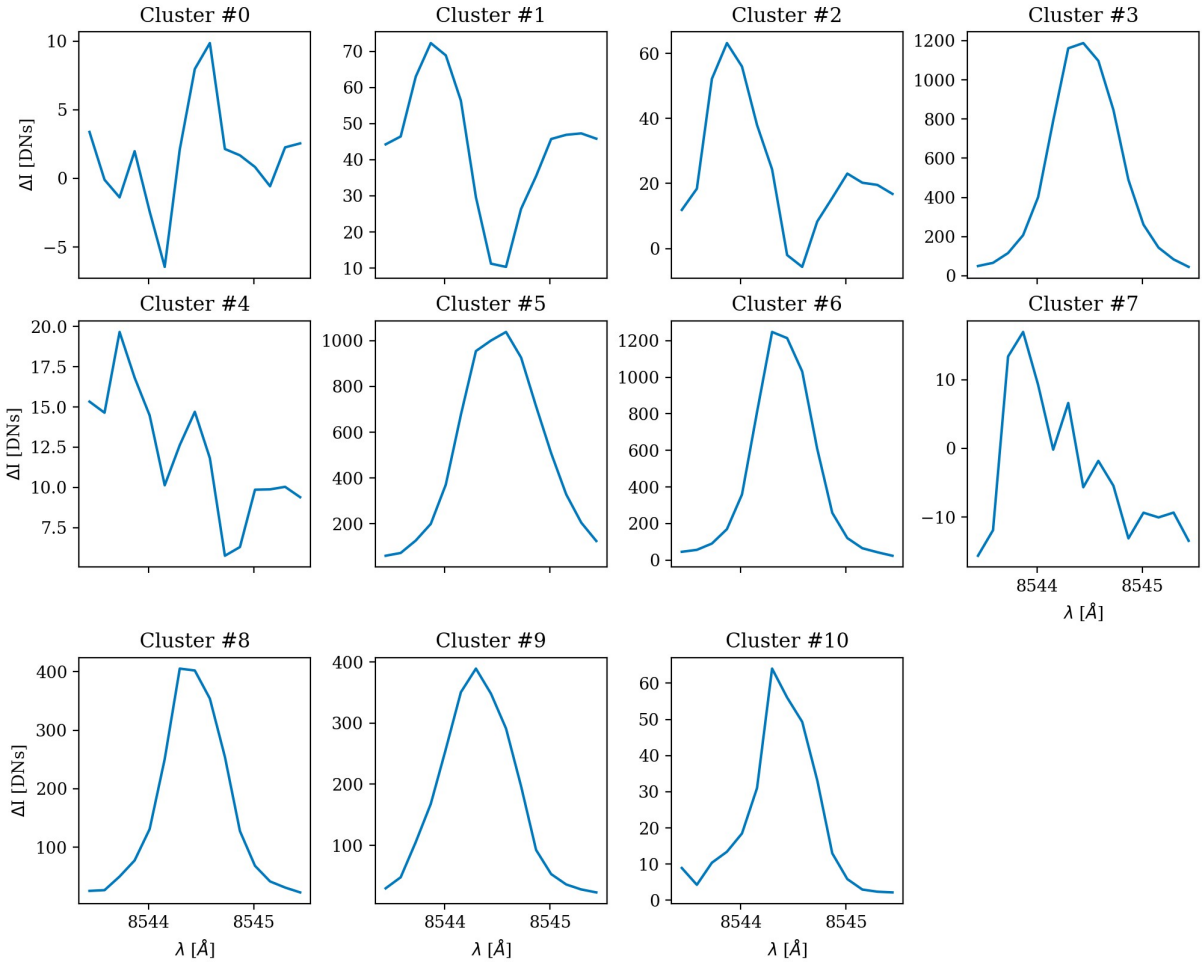

Figure 6.16: The Ca II λ8542 cluster means for the trained GMM. This indicates the clusters of interest for flare ribbons. In the Ca II λ8542 model, clusters #3, #5 and #6 seem to best represent the flare ribbons with clusters #8 & #9 representing the points near the flare ribbons that are heated.

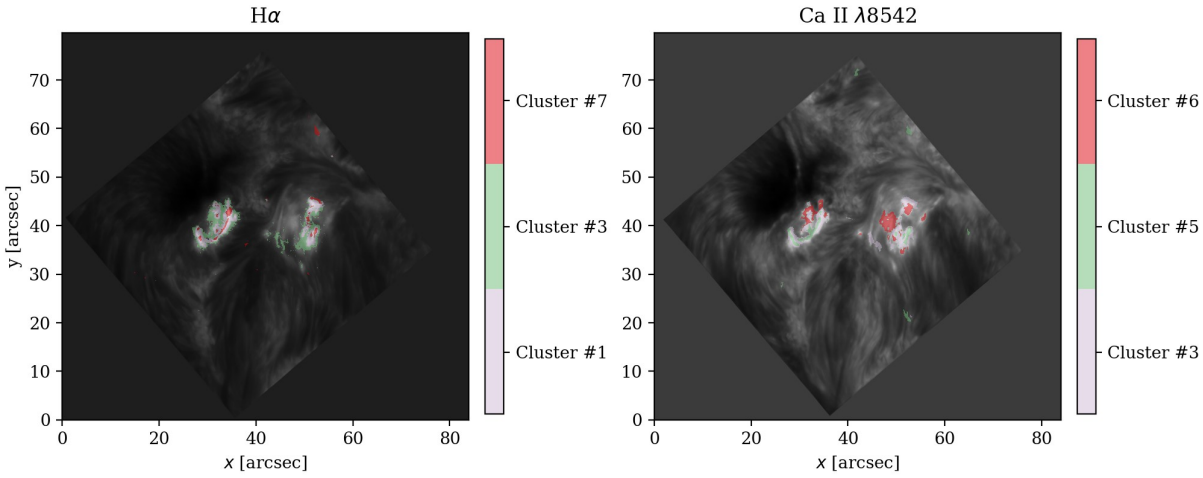

Figure 6.17: The clusters pertaining to flare ribbons overplotted on the images where the spectra were extracted to train the GMM. This shows that the combinations of clusters #1, #3 & #7 for Hα and clusters #3, #5 & #6 for Ca II λ8542 are optimal for covering the flare ribbon regions of interest.

within each cluster. This is shown in Fig. 6.15 for Hα and Fig. 6.16 for Ca II λ8542. Both the shape and the magnitude of the cluster means point to the spectra encompassed by those clusters. For instance, in the Hα cluster means, cluster #1 contains the brightest points since the intensity is the largest and also the line profiles are fully in emission. Cluster #7 shows line profiles with the characteristic centrally-reversed Hα profiles and clusters #3 & #4 show similar profiles to cluster #1 but with lower intensities. The spatial distribution of all the clusters was examined and it was determined that clusters #1, #3 and #7 are the representative clusters of the flare ribbons in the Hα data. This is shown in the left panel of Fig. 6.17. Similarly for Ca II λ8542, the cluster means are shown in Fig. 6.16. Clusters #3, #5 and #6 are found to describe the flare ribbons and overplotted to demonstrate this in the right panel of Fig. 6.17. The choice of these clusters is sound as they demonstrate lines in emissions with large intensities as is expected of Ca II λ8542 spectral lines in flare ribbons.

Now that flare ribbons can be identified in the CRISP observations, a way to separate the ribbons in two ribbon flares (and possibly to segment each ribbon into smaller sources depending on their GMM cluster) is needed. The tool used for this is a deterministic unsupervised machine learning method, DBSCAN (Ester et al., 1996; Schubert et al., 2017). DBSCAN is not like other ML algorithms in that there is not a model that is trained to be applied to other observations. Instead, DB-

SCAN uses the spatial locations of points along with hyperparameters to segment the structures given to the algorithm. This is done by classifying each point to be clustered as one of three: core points, density-reachable points and outliers. The points are classified by the two hyperparameters in the system: `minPts` and ϵ. A point is considered a core point if there are `minPts` points within a distance ϵ where the distance is defined by a user-chosen metric (here it is Euclidean distance in the image plane). A density-reachable point is one that is within a distance ϵ of a core point but is not itself a core point. All other points are considered outliers. A starting point is chosen at random and if it is assigned as a core point, a cluster is formed. All of the other points are then classified with respect to this core point until they all have a designation. The cluster is then populated by anything density-reachable from the initial core point and the chain created by the points in its vicinity. A random point classified as an outlier from the initial cluster is then chosen as a new core point to follow the same method for determining the next cluster from the remaining set of points. This process is repeated until all points are part of a cluster or designated as noise. This is applied to the identified flare ribbon areas from the trained GMM with promising results. Note that the fact that DBSCAN does not work from a trained model is actually beneficial in the context of robustness. If a user would like to study larger or smaller scale phenomena they can simply modify the `minPts` and ϵ hyperparameters without having to change anything else about the model.

There are now two roads which can be travelled to study the atmospheric conditions that lead to these spectral line profiles: either each GMM cluster is treated individually to see if the same underlying physics contributes to all line profiles in a cluster or all three flare ribbon GMM clusters are used and the west and east ribbons of this flare are separated to study the physics in each of them individually. The former will be explored followed by the latter.

6.4.2 Cluster Asymmetries and Correlations to the Flare Velocity Field

In this section, three different times from around the impulsive phase of the M1.1 solar flare SOL20140906T17:09 are analysed using the clustering methods and RADYNVERSION described above to investigate the relationship between the asymmetry of the spectral lines and the flaring velocity field. Kurokawa et al. (1986) found

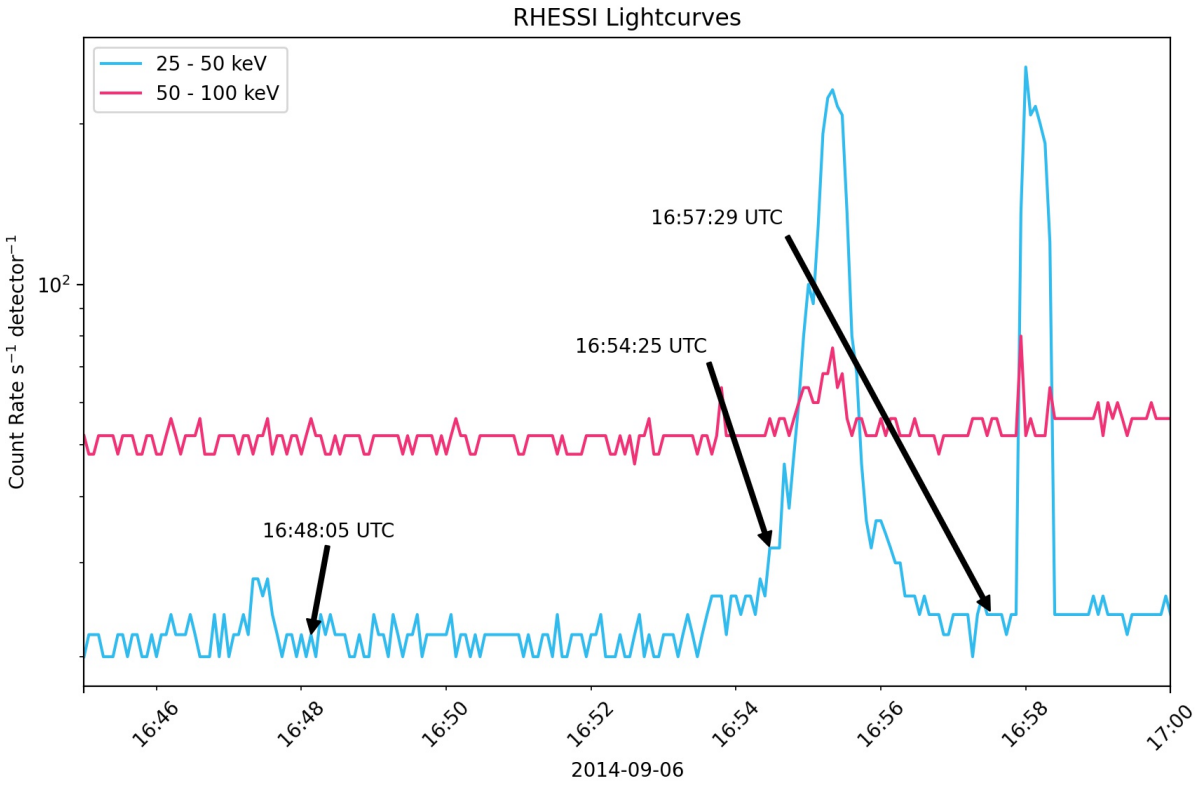

Figure 6.18: RHESSI lightcurves for the SOL20140906T17:09 M1.1 solar flare with arrows indicating the time that are to be analysed in the following section: 16:48:05 UTC before non-thermal electrons are injected by the flare; 16:54:25 UTC as the electrons are being injected and 16:57:29 UTC after the initial population of electrons impact the lower atmosphere.

that there is a subsecond temporal correlation between Hα brightenings and hard X-ray (HXR) spikes during the impulsive phase of solar flares. As such, the times of interest analysed in this section are selected based on the HXR lightcurves taken by the Reuven Ramaty High Energy Solar Spectroscopic Imager (RHESSI; Lin, 2000). These lightcurves for energy ranges 25-50 keV and 50-100 keV are shown in Fig. 6.18. In this figure, the three different times are indicated by arrows showing the three times to be analysed as being: 16:48:05 UTC just after a small event in the 25-50keV range; 16:54:25 UTC during the first big event which shows clear signals in both energy ranges; and 16:57:29 UTC after the first big HXR event but before the second. These observations then have their flare ribbons identified using the the Hα GMM described in Sec. 6.4.1. The points for Hα clusters #7 and #1 are extracted as the points directly excited by the injection of energy on the flare ribbons. Once the points for these clusters are subtracted, DBSCAN is performed on the clusters

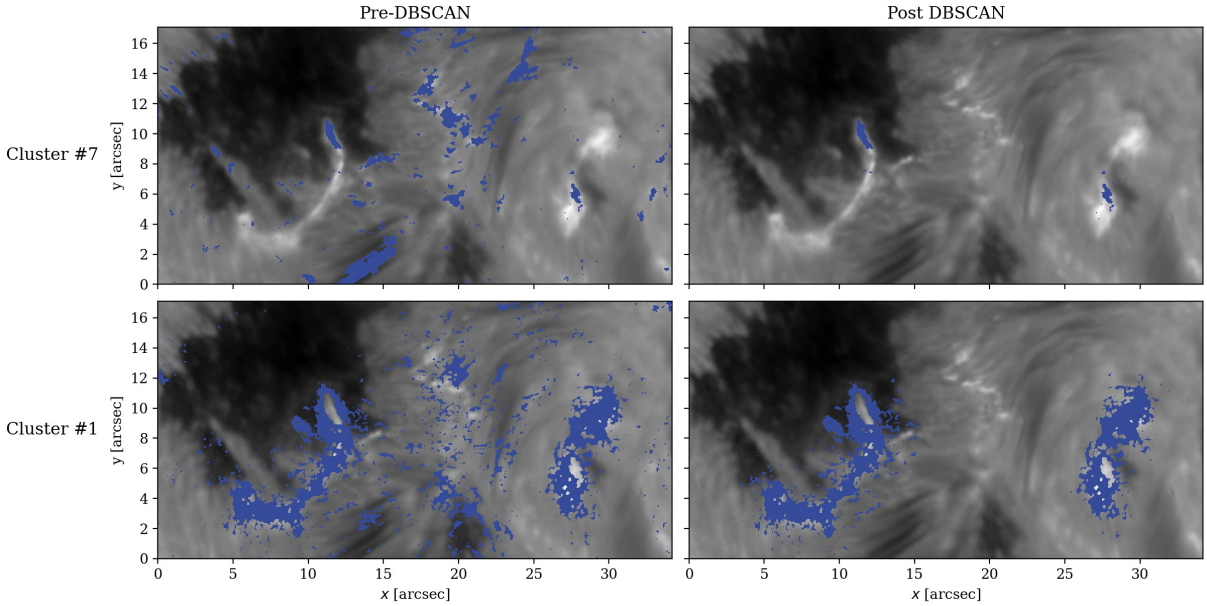

Figure 6.19: Demonstrating how DBSCAN refines the areas around the flare ribbons for further analysis. The top row demonstrates the reduction in numbers of points when using DBSCAN on Hα GMM cluster #7 (see Fig. 6.15) while the bottom row shows the same for cluster #1. The left column shows the points selected for each cluster by the GMM with the right column showing the post-DBSCAN results.

in turn to identify the flare ribbon structures. This is shown in Fig. 6.19 for 16:48:05 UTC, Fig. 6.24 for 16:54:25 UTC and Fig. 6.29 for 16:57:29 UTC. The specific hyperparameters used for DBSCAN at each time for each cluster is given in Tab. 6.2. The asymmetries in these locations in the Hα and Ca II λ8542 spectral lines are then studied.

16:48:05 UTC

Initially focusing on the observations from 16:48:05 UTC, Fig. 6.20 shows the asymmetries of the spectra (calculated using the method in Sec. 6.3) for the points identified by DBSCAN as flare ribbons for GMM clusters #7 and #1. The top row of Fig. 6.20 shows the asymmetries in cluster #7 and interestingly, every spectrum in the eastern ribbon in cluster #7 is red asymmetric. As for the western ribbon, the spectra are more blue asymmetric with some red asymmetric patches occurring in the same spatial locations. The bottom row shows cluster #1 asymmetries for Hα (left) and Ca II λ8542 (right). The western ribbon in both lines is mostly red asymmetric with some blue asymmetries cropping up in concentrated spatial locales. The

		ϵ	minPts
16:48:05	Cluster #7	0.1	150
	Cluster #1	0.1	300
16:54:25	Cluster #7	0.25	100
	Cluster #1	0.25	100
16:57:29	Cluster #7	0.25	200
	Cluster #1	0.25	100

Table 6.2: The DBSCAN hyperparameters used at the three different times producing the results in Figs. 6.19, 6.24 & 6.29.

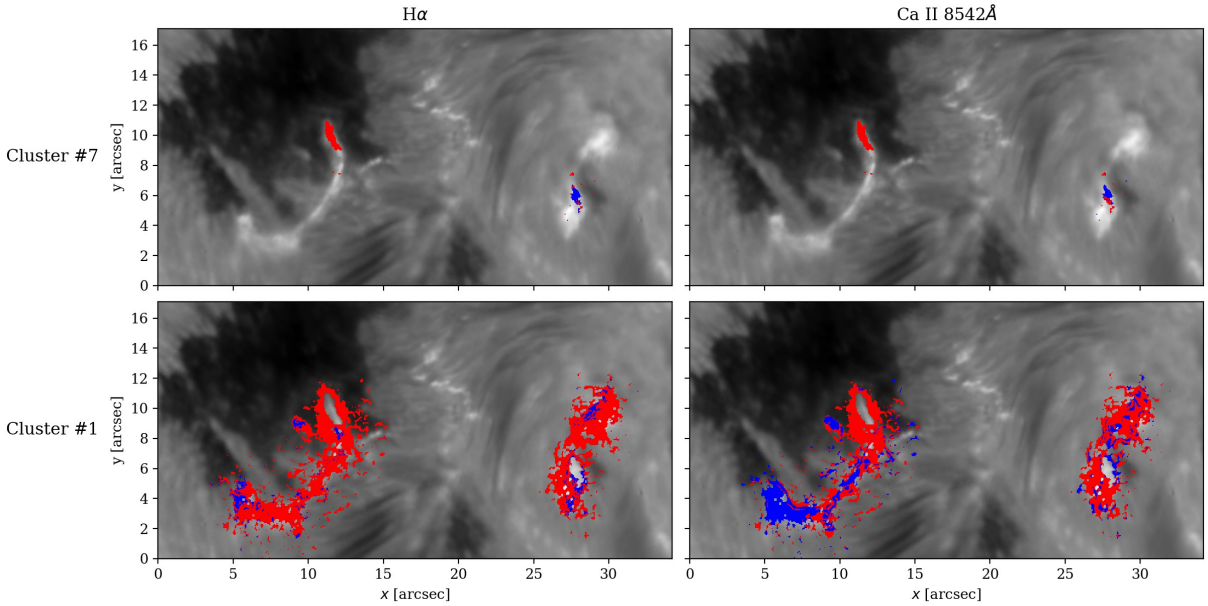

Figure 6.20: The calculated line asymmetries for Hα (left column) and Ca II λ8542 (right column). The colours are indicative of the asymmetry in the pixels as calculated by the method described in Sec. 6.3. The top row shows the asymmetries from GMM cluster #7 and the bottom row shows the asymmetries from GMM cluster #1 (both post-DBSCAN).

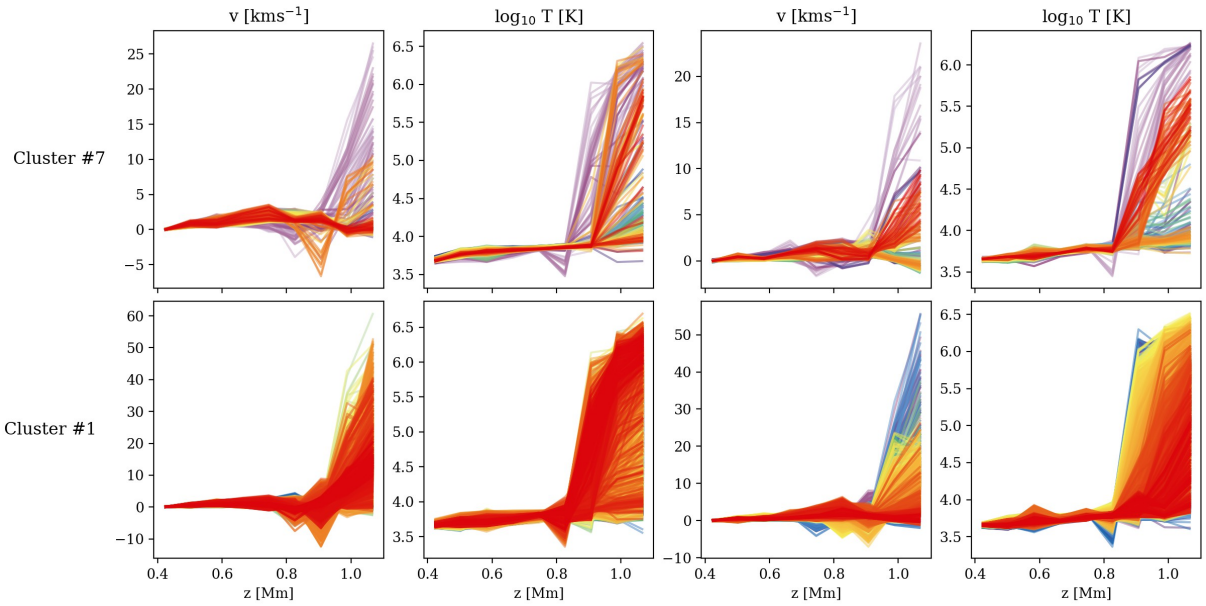

Figure 6.21: The flow velocity and temperatures estimated by RADYNVERSION for the eastern flare ribbon (left columns) and western (right columns). These are also split by GMM cluster with the top row showing cluster #7 and the bottom cluster #1. These parameters are plotted over the ranges in height of line formation of Hα and Ca II λ8542. The gradation in colour indicates different spatial locations within the specified ribbons.

eastern ribbon is again interesting with most of the Hα profiles being red asymmetric but nearly half of the Ca II λ8542 profiles being blue asymmetric. In fact, there is a large portion of this eastern ribbon where Hα is red asymmetric and Ca II λ8542 is blue asymmetric. This points to some potentially complex motions in the lower solar atmosphere in the early phases of this flare and impacts how the data is analysed.

The points identified by DBSCAN for each cluster are inverted by RADYNVERSION 20,000 times as with the example in Sec. 6.3. The inverted atmospheres are used to investigate the causes of the asymmetries observed in the spectra, particularly the gradient of the flow velocity and the temperature. Furthermore, the ribbons were analysed based on their GMM cluster and per ribbon structure (east and west). The ribbons were then split into red and blue components to investigate the differences in the atmospheres producing each type of asymmetry[8].

Figure 6.21 shows the inverted atmospheres for the flare ribbons as whole structures by GMM cluster for 16:48:05 UTC. The left two columns are the atmospheres for the eastern flare ribbon and the right two columns are for the western flare rib-

[8]These splits were done depending on the asymmetry of the Ca II λ8542 line.

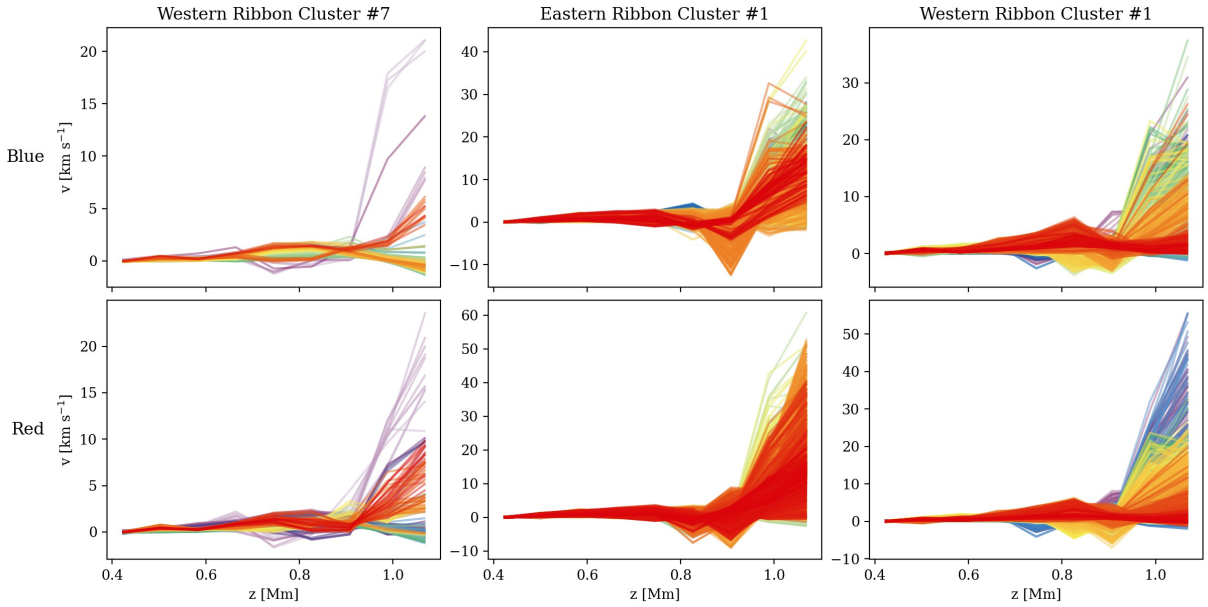

Figure 6.22: The flow velocity for the identified flare ribbons at 16:48:05 UTC split by asymmetry and GMM cluster. The left column shows the western ribbon's cluster #7 points split into blue (top row) and red (bottom row) asymmetry. The middle and right columns show the same for the eastern and western ribbons' cluster #1 points, respectively. Note that the eastern ribbon at this time has all red asymmetries hence it is excluded from this figure.

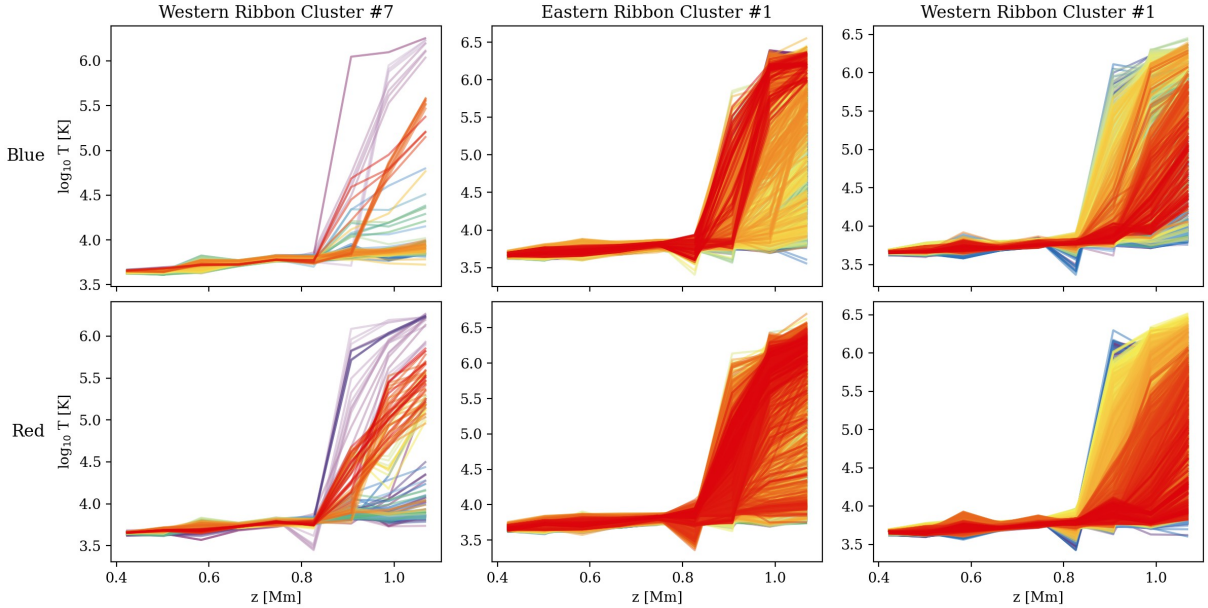

Figure 6.23: An identical figure to Fig. 6.22 but this time showing the temperatures of the flare ribbons split by asymmetry.

bon with the top row being GMM cluster #7 and the bottom #1. The first and third columns are the flow velocity and the second and fourth columns are the logarithm of the temperature. For the eastern flare ribbon, the red asymmetric cluster #7 points typically have profiles with upflows in the lower part before sharp downflows around the steep increase in temperature followed by more upflows when temperatures reach coronal values. Looking at the regions where the wings of these spectra typically form during flares leads to the conclusion that the sharp downflows are responsible for the Hα red asymmetry while the upflows are responsible for the Ca II λ8542 red asymmetry. This seems typical in comparison with Ichimoto and Kurokawa (1984) who concluded it was the downward condensation of plasma responsible for red Hα asymmetries during flares. The red asymmetry in Ca II λ8542 can be thought of as upward motion of absorbing material which ties into the reciprocal chromospheric evaporation that occurs during flares. The small magnitudes of the velocities are appropriate for the early time in the flare as there has not been a lot of HXR emission observed at this time.

The remaining panels describe structures which have both red and blue asymmetric components. While the profiles for all points in these ribbon structures are shown in Fig. 6.21, it is more constructive to split these structures by asymmetry to consider the physics causing each asymmetry within a structure. This is shown for velocity in Fig. 6.22 and temperature in Fig. 6.23. N.B. in Figs. 6.22 & 6.23, the top row shows the blue asymmetric atmospheres and the bottom row shows the red asymmetric atmospheres. The western ribbon's cluster #7 points (left column of Figs. 6.22 & 6.23) are mostly blue asymmetric with some small regions of red asymmetry. This shows that blue asymmetric Ca II λ8542 profiles seem to be affected by upward moving emitting material while red asymmetric profiles seem to be affected by upward moving absorbing material. In the region of formation of Hα wings, both the blue and red asymmetric profiles have velocities both positive and negative indicating a lot of complex motion around this region with small clouds of emitting and absorbing plasma rising and falling in the excited atmosphere (similar to what was reported in Graham and Cauzzi, 2015).

Switching focus back to the eastern ribbon, now considering the points identified as being cluster #1, most of the Hα profiles are red asymmetric but there is a large proportion of Ca II λ8542 profiles with a blue asymmetry. Particularly, the southern portion of the ribbon is mostly blue asymmetric and moving north there appears a mix before a large area where all profiles are red asymmetric. Also, the profiles

surrounding the points in cluster #7 are all red asymmetric. Examining the inverted atmospheres split by asymmetry (middle columns of Figs. 6.22 & 6.23, respectively), there are large negative velocities ($\mathcal{O}(10\text{km s}^{-1})$) in the regions where the Hα wings form. This places the cause of the red asymmetry as the same as for the cluster #7 points: emitting material flowing downwards. The jointly red asymmetric Ca II λ8542 profiles are formed in regions with small upflow velocities again indicating evaporation processes occurring here.

For the locations where a blue asymmetric Ca II λ8542 is cospatial with red asymmetric Hα, the top panels of the middle column of Figs. 6.22 & 6.23 are examined. There are larger downflows in the Hα forming region with upflows in the Ca II λ8542 forming region. This implies that the Ca II λ8542 profiles are produced by upward moving emitting material – potentially evaporations that have not reached a high enough temperature yet to ionise all of their Ca II. The Hα lines are again fueled by the downward moving condensations.

Lastly, considering the cluster #1 points for the western flare ribbon it is seen in the bottom right panel of Fig. 6.20 that the ribbon is mostly red asymmetric in both lines with some small regions of blue asymmetry sprinkled throughout. Regardless, the asymmetries in this ribbon seem to be cospatial (i.e. red asymmetric Hα is cospatial with red asymmetric Ca II λ8542). Investigating the right columns of Figs. 6.22 & 6.23 gives insights into the causes behind these asymmetries. The atmospheres responsible for the red asymmetric profiles in this case do not provide a definitive answer for the Hα asymmetries: some inverted atmospheres boast downflows in Hα forming regions while others indicate upflows. This means that in some regions the evaporating absorbing material is more prevalent while in others it is the condensation emitting material. A further study of the spatial locations where these differing atmospheres are estimated could highlight some explanations for this dichotomy. In the Ca II λ8542-forming regions for all red asymmetric inverted atmospheres, there is again an upflow of material pointing to hot evaporation material. The story for the blue asymmetric profiles is the opposite as the inverted atmospheres have similar velocity and temperature profiles. This means that for Hα there is emitting material moving upwards with absorbing material moving downwards. For Ca II λ8542, there is some emitting material moving upward as the cause for the blue asymmetric profiles.

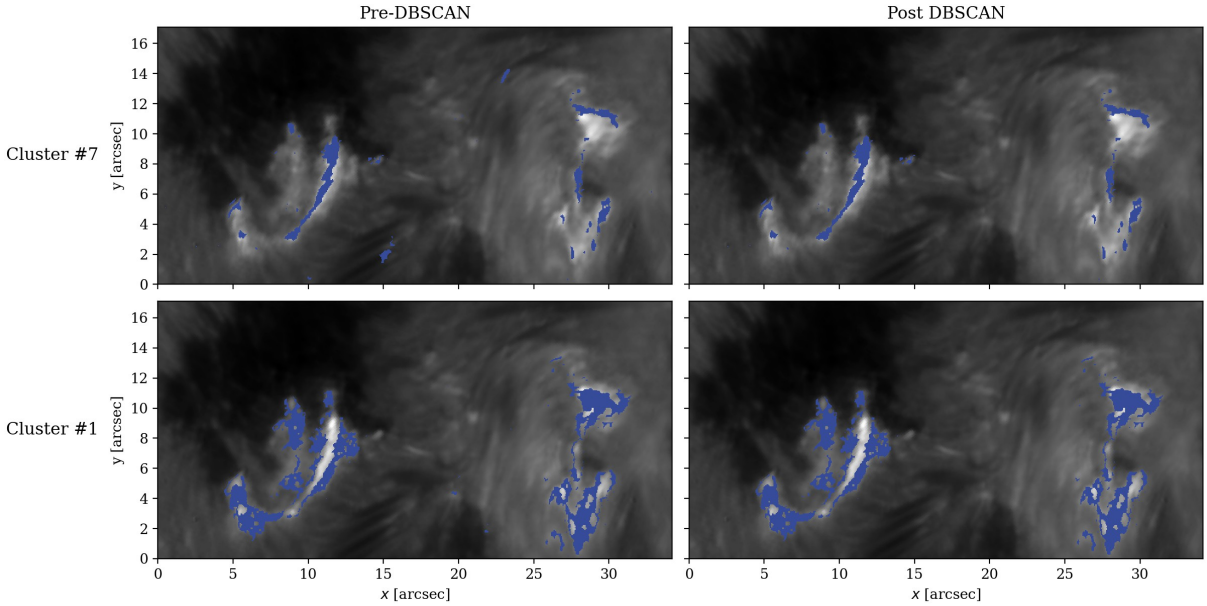

Figure 6.24: An equivalent to Fig. 6.19 but for 16:54:25 UTC – when energetic electrons are being injected into the lower solar atmosphere.

16:54:25 UTC

Moving on to the observation from 16:54:25 UTC, the asymmetries of the spectral lines for the two clusters of interest are shown in Fig. 6.25. The dynamics of the ribbons is more complicated at this time as more and more HXRs are observed (Fig. 6.18). The same approach is taken with the atmospheric parameters for each ribbon and cluster shown in Fig. 6.26 with the same layout as Fig. 6.21 (eastern ribbon represented in left two columns and western ribbon in right two columns). Similarly, Figs. 6.27 & 6.28 show the velocities and temperatures of the ribbon locations split by their asymmetry with the blue asymmetric profiles shown in the top row and the red in the bottom as in Figs. 6.22 & 6.23. The difference at this time is that all ribbon structures have a mix of asymmetries and so can all be split in this way. Consquently, in Figs. 6.27 & 6.28, the left column is the western ribbon's cluster #7 points, the second column is the eastern ribbon's cluster #7 points, the third column is the eastern ribbon's cluster #1 points and the last column is the western ribbon's cluster #1 points.

Following the previous observation, the first structure discussed will be the eastern ribbon's cluster #7 points. Compared to the earlier time, there is now an elongated, curved structure of cluster #7 points. They are mostly red asymmetric in both

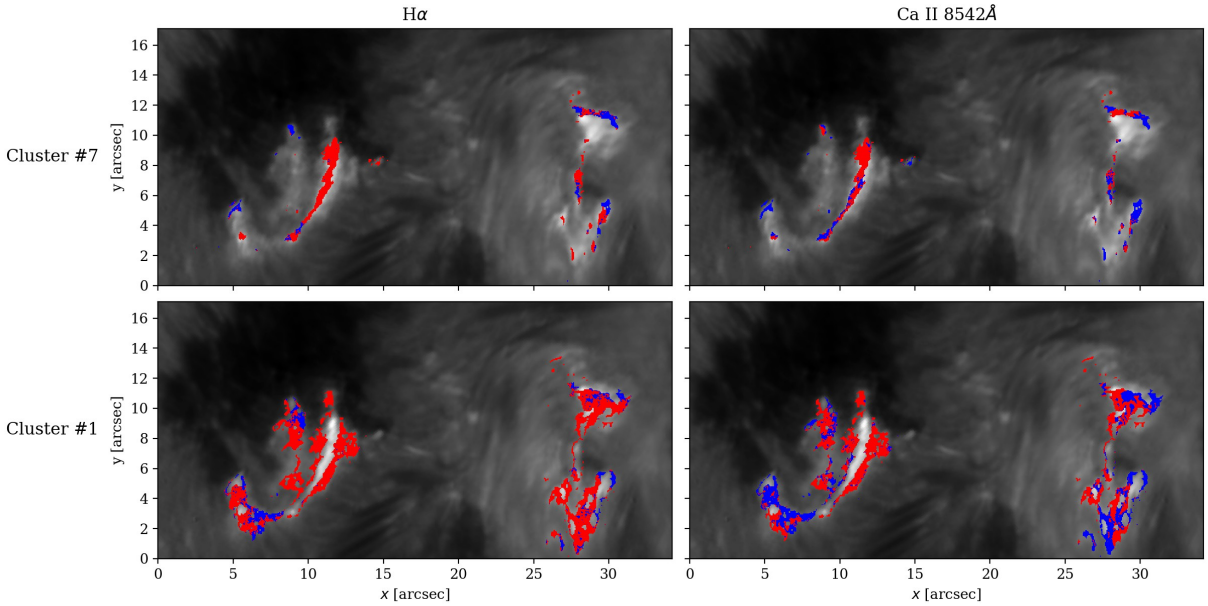

Figure 6.25: An equivalent to Fig. 6.20 but for 16:54:25 UTC.

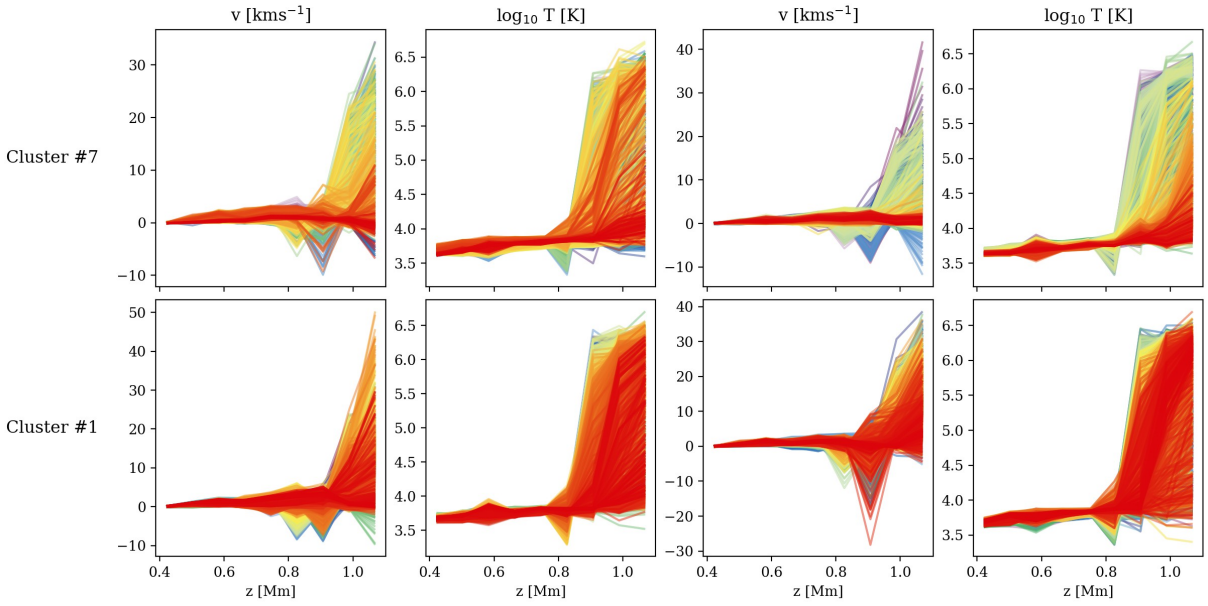

Figure 6.26: An equivalent to Fig. 6.21 but for 16:54:25 UTC.

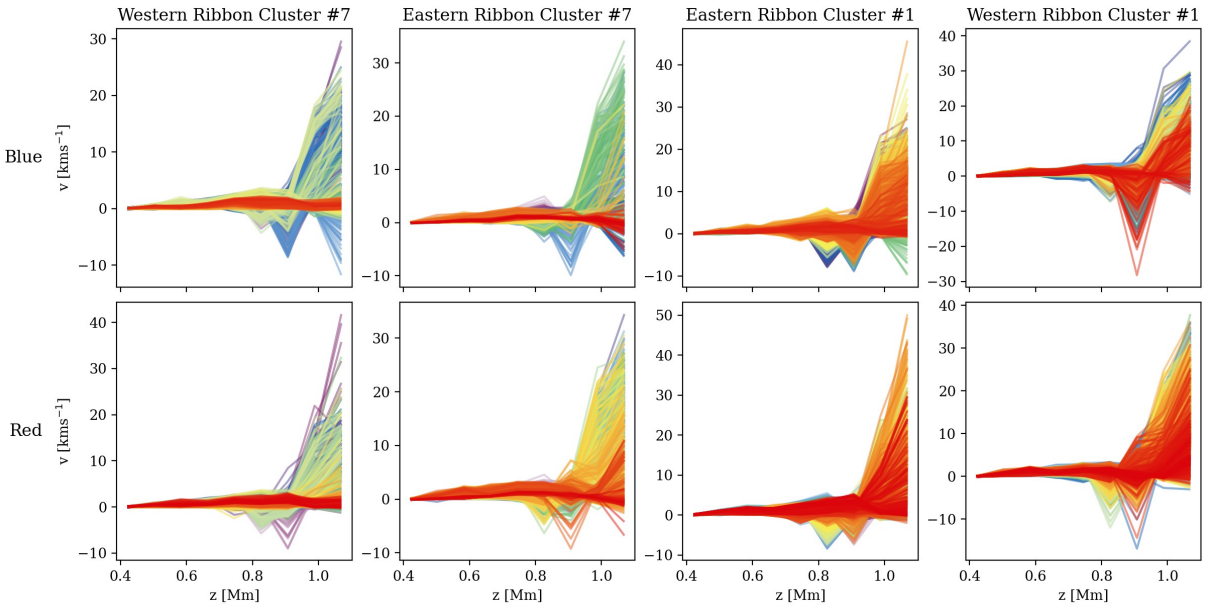

Figure 6.27: Similar to Fig. 6.22 but this time at 16:54:25 UTC, the eastern ribbon can also be split by asymmetry for the cluster #7 points (second column). The western ribbon's cluster #7 points are represented in the left column with the ribbons' cluster #1 points shown in the third and fourth columns.

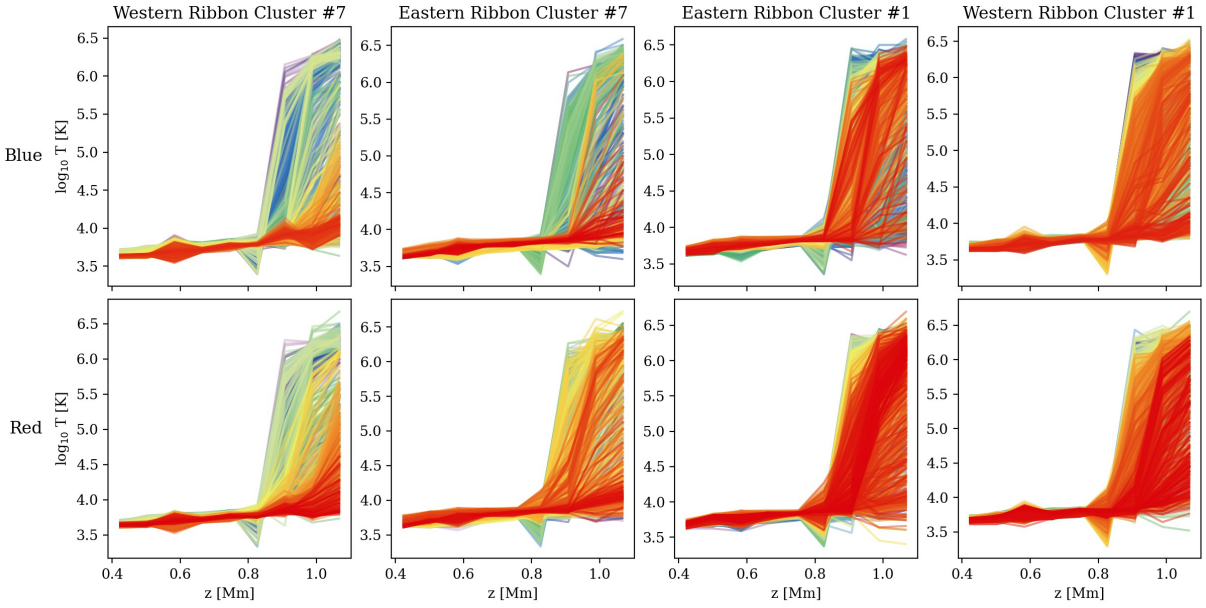

Figure 6.28: An identical figure to Fig. 6.27 but this time showing the temperatures of the flare ribbons split by asymmetry.

Hα and Ca II λ8542, including the area in which there existed cluster #7 profiles in the earlier observation, with some blue asymmetric profiles in small concentrations. Looking at the velocities and temperatures split by asymmetry for this structure (second column of Figs. 6.27 & 6.28) gives an insight into the dynamics of the flaring atmosphere at this time. The velocity of the plasma is positive in the region of Ca II λ8542 regardless of the asymmetry being red or blue implying regions of blue asymmetry result from upflowing emitting material with upflowing absorbing material being responsible for the red asymmetry. For Hα blue asymmetric profiles, the prevailing velocity signature seems to be downflows in the region of formation. This combined with the large gradient in temperature at these heights points to this blue asymmetry being caused by the heating of the lower solar atmosphere and its compression by the impact of nonthermal electrons. The red asymmetrical profiles are formed via both upflows and downflows in the Hα formation region indicating that there is evaporation and condensation processes at work in these regions.

Considering the western ribbon's cluster #7 points, Fig. 6.25 shows that there is both red and blue asymmetric profiles for both spectral lines with the majority of same asymmetries occurring in the same spatial locations. Looking at Fig. 6.27, both red and blue asymmetries of Hα can be associated with downflowing plasma implying an interplay of downflowing emission and downflowing absorption. The asymmetries of Ca II λ8542 are both due to upflowing material with the red asymmetry due to upflowing absorbing plasma (evaporation) of all similar velocity magnitudes and the blue asymmetry due to upflowing emitting plasma with a varying magnitude of velocity. In fact, looking at Fig. 6.28, there is a noticeable spread in the temperature of the plasma around the region of Ca II λ8542 formation implying that each of the different regions with blue asymmetry are in different stages of heating.

Returning to the east ribbon and considering its cluster #1 points, it is noted that a lot of the points selected by cluster #1 that exhibit a red asymmetry in both spectral lines do not appear to be part of the brightest ribbon structure (Fig. 6.25). These locations are postulated to be heated either by some form of radiative heating owing to their adjacency with the brightest flare ribbons or via weaker direct flare heating due to the sweeping motion of the flare ribbons in the chromosphere. It is likely a combination of both effects that leads to these points being candidates for cluster #1. For the Hα and Ca II λ8542 red asymmetric points upflows are mostly responsible. This indicates evaporating hot plasma in these regions. Considering the blue asymmetries, the Ca II λ8542 formation region experiences upflows indi-

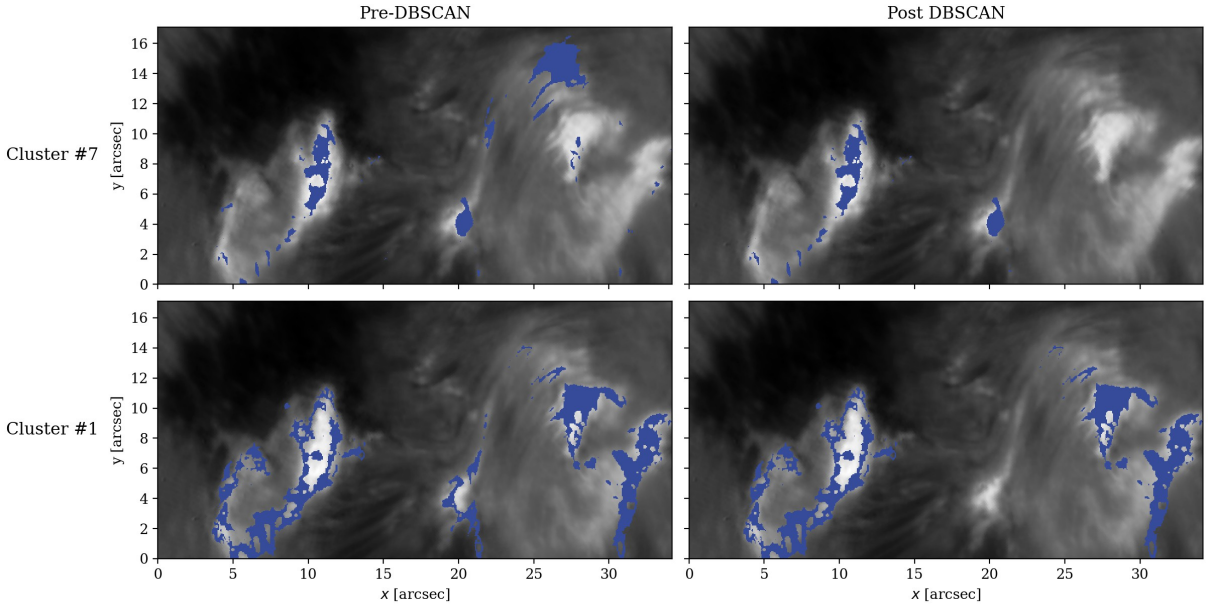

Figure 6.29: Equivalent to Fig. 6.19 & 6.24 but for 16:57:29 UTC – after energetic electrons have been injected into the lower solar atmosphere.

cating emission in upflowing plasma with most of the profiles in the Hα formation region presenting downflows similar to the eastern ribbon's cluster #7 points.

For the western ribbon's cluster #1 points, there is a mix of red and blue asymmetric profiles. The Hα blue asymmetric profiles are mainly caused by downflows of up to 30km s^{-1} implying fast moving absorbing material. This is again roughly at the same height where the transition region occurs in the temperature profiles implying that these points with blue Hα asymmetry are caused by material being strongly heated in the lower atmosphere. The Hα red asymmetric profiles form due to downflows of cooler emitting material. Both Ca II λ8542 asymmetries occur as a result of upflows as has been seen in the other structures.

16:57:29 UTC

The asymmetries of the spectral line profiles for the 16:57:29 UTC observation are shown in Fig. 6.30 (which is equivalent to Figs. 6.20 & 6.25 for this observation). The identified flare ribbons are then analysed in the same way as for the previous two observations. This time represents one after which there has been energy injected into the lower atmosphere (after a HXR peak).

The eastern ribbon's cluster #7 points are mainly red asymmetric in both Hα

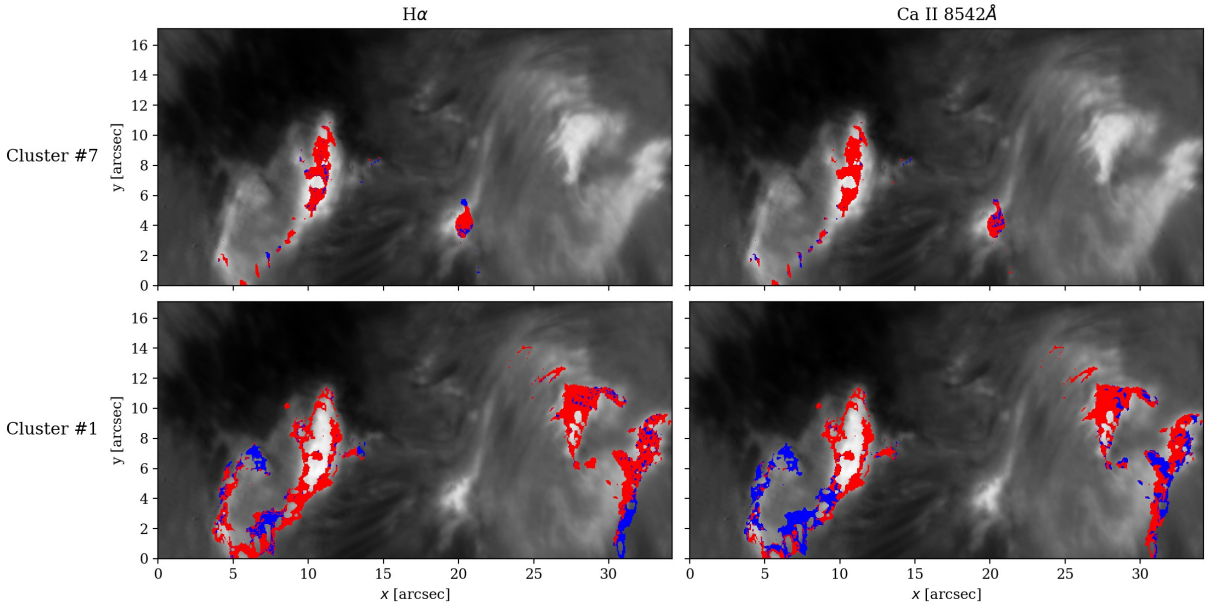

Figure 6.30: Equivalent to Fig. 6.20 & 6.25 but for 16:57:29 UTC.

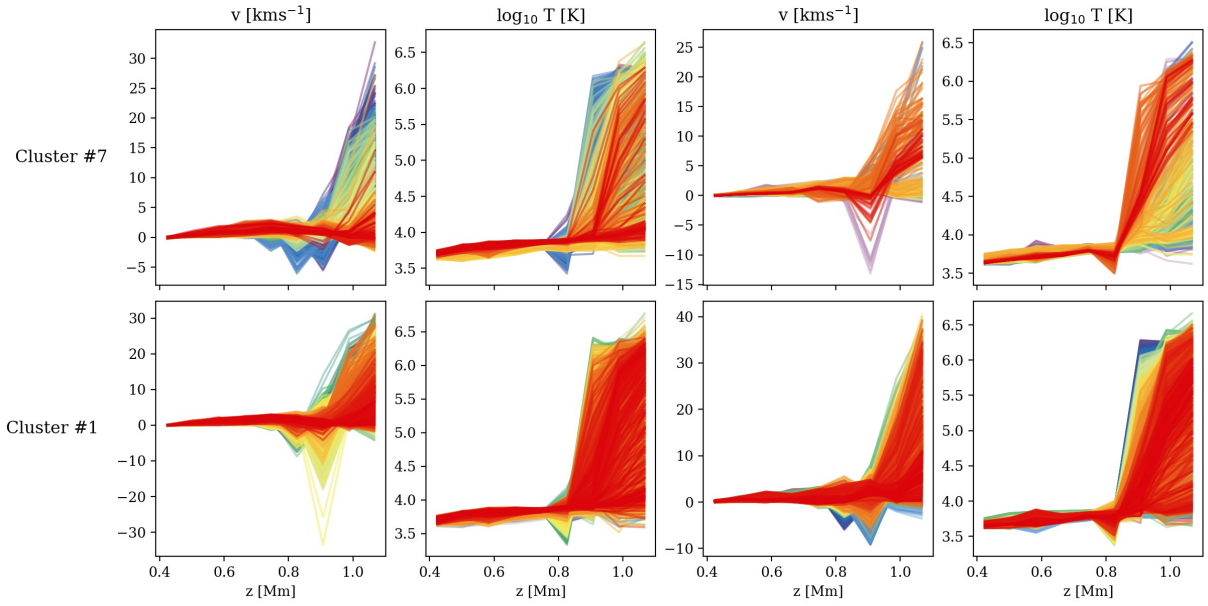

Figure 6.31: An equivalent to Fig. 6.21 & 6.26 but for 16:57:29 UTC.

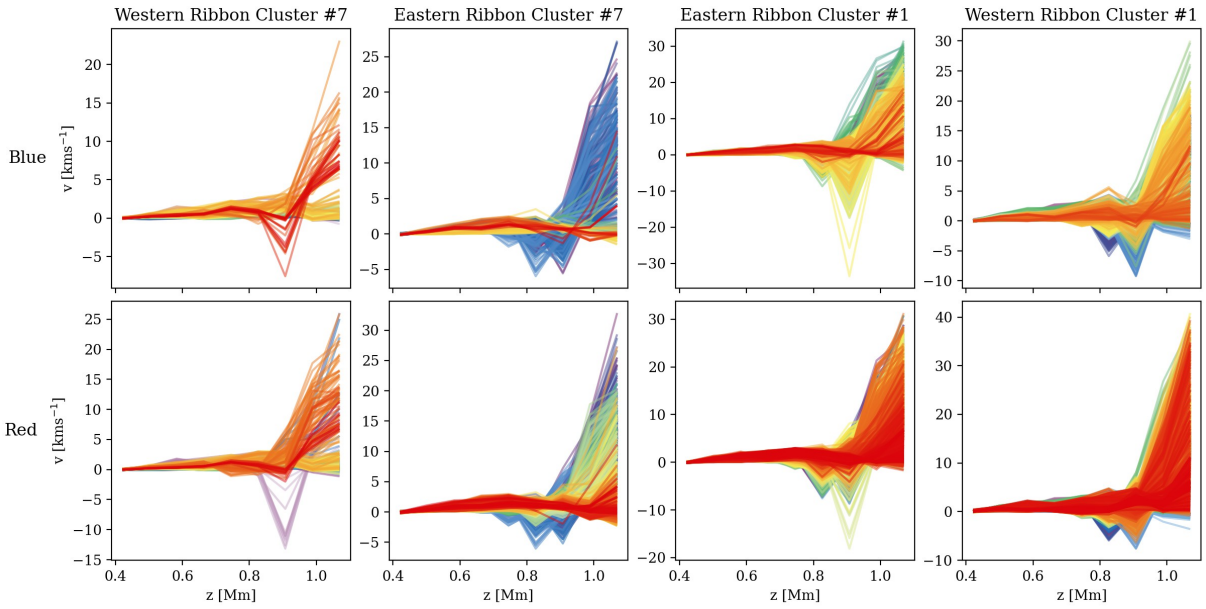

Figure 6.32: Similar to Fig. 6.27 but this time at 16:57:29 UTC.

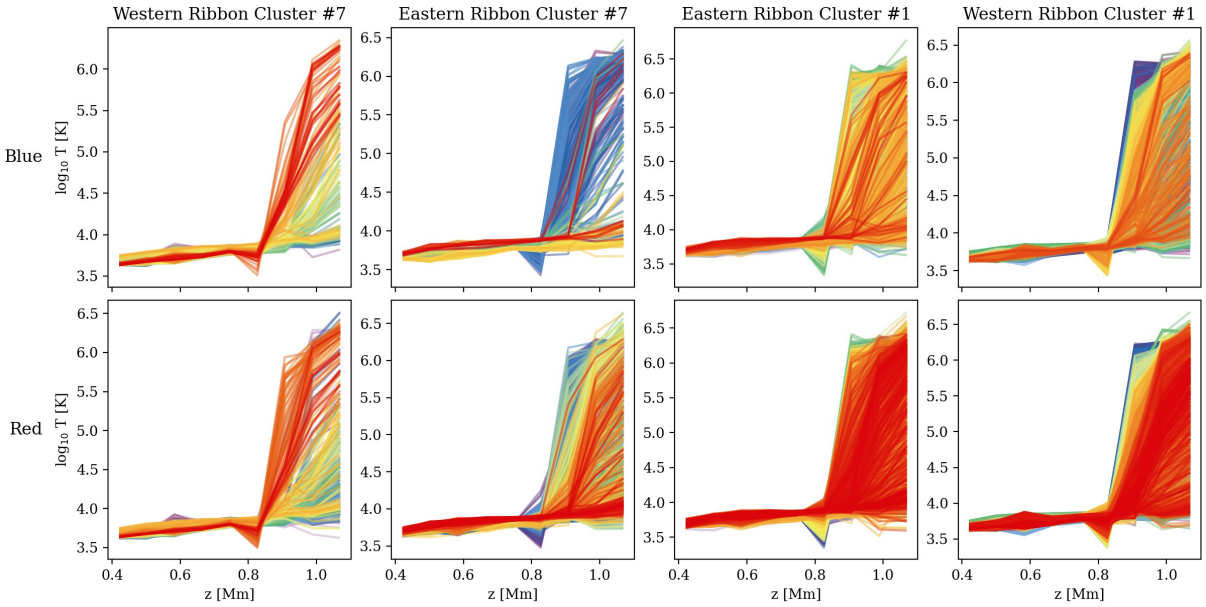

Figure 6.33: An identical figure to Fig. 6.32 but this time showing the temperatures of the flare ribbons split by asymmetry.

and Ca II λ8542. Looking at Figs. 6.31 & 6.32, it can be seen the red asymmetric Hα profiles are influenced by downflowing plasma implying this downflowing plasma is emitting. Moreover, the Ca II λ8542 profiles are red asymmetric due to evaporating absorbing plasma. In a few locations, the spectra are blue asymmetric implying in these locations that there is downflowing material absorbing Hα and upflowing material emitting Ca II λ8542.

For the western ribbon's cluster #7 points, both spectral lines are mostly red asymmetric too. Similarly to the eastern ribbon, these asymmetries are due to downflowing emitting plasma for Hα and upflowing absorbing plasma for Ca II λ8542. Also, for the few regions showing a blue asymmetry in the line profiles, the same explanation as for the eastern ribbon's cluster #7 points apply.

For the eastern ribbon's cluster #1 points, there is again a split between red and blue asymmetric profiles in Ca II λ8542 with nearly all Hα points red asymmetric. As such when both are red asymmetric, the Hα can be described by downflowing emitting material and the Ca II λ8542 by upflowing absorbing material. However, when the Ca II λ8542 is blue asymmetric and the Hα is still red asymmetric, the Ca II λ8542 can be described by upflowing emitting material with the Hα still described by downflowing emitting material.

Lastly, for the western ribbon's cluster #1 points, most of the points are red asymmetric in both spectral lines with some blue asymmetric profiles present in the western edges of the ribbons for both spectral lines. The red asymmetric profiles for Hα correspond to downflowing emitting material with the blue asymmetric profiles corresponding to downflowing absorbing material. For Ca II λ8542, the red asymmetric profiles correspond to upflowing absorbing material and the blue asymmetric profiles to upflowing emitting material.

6.5 Discussion

This chapter has presented a novel deep learning algorithm for analysing inverse problems in solar physics and the wider astronomy community: an invertible neural network. A specific INN trained on 1D RHD flare simulations from RADYN, RADYNVERSION, was trained to produce the first determinations of the flaring atmosphere from observations considering the full RHD treatment of the solar plasma. This was then applied to real data from SST/CRISP to learn about the chromospheric flaring velocity field and its relation to the observed spectral line asymme-

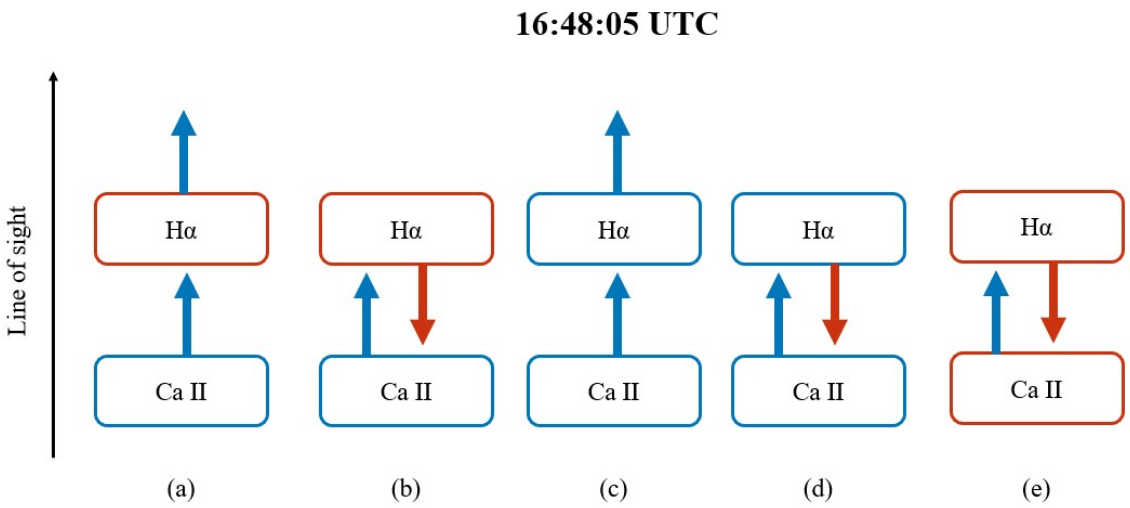

Figure 6.34: An illustration of the asymmetries observed and the motion causing them at 16:48:05 UTC. The colour of the box of material responsible for a spectral line indicates which type of asymmetry the line has in such a case with the arrow indicating the motion of the material. When the colours of the box and arrow match, there is emitting material responsible in the corresponding direction for the asymmetry and when they differ there is absorbing material responsible for the asymmetry. (a) and (b) both correspond to areas where Ca II λ8542 has a blue asymmetry and Hα has a red asymmetry. The difference is in (a), the Hα red asymmetry is caused by upflowing absorbing material whereas in (b) it is caused by downflowing emitting material. Similarly, (c) and (d) both correspond to regions where Ca II λ8542 and Hα have blue asymmetries. (e) shows the case where both lines have red asymmetries.

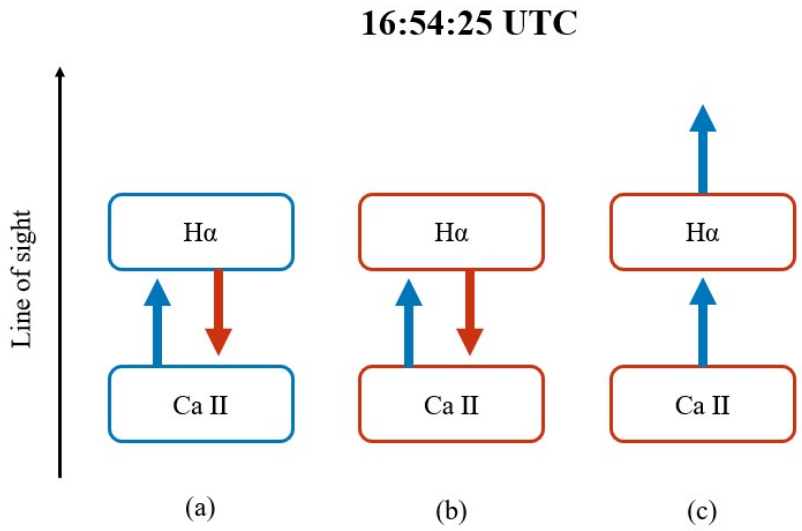

Figure 6.35: An illustration of the asymmetries observed and the motion causing them at 16:54:25 UTC. (a) shows the case where both lines have blue asymmetries and (b) and (c) shows the two different cases where both lines have red asymmetries.

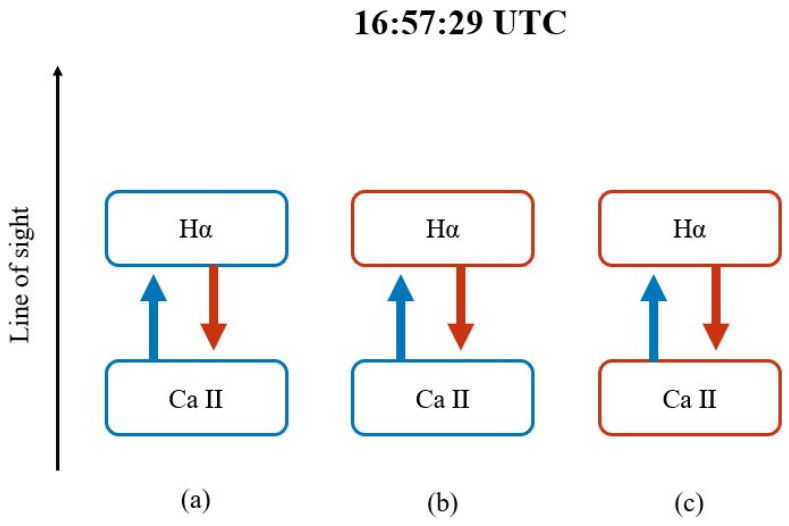

Figure 6.36: An illustration of the asymmetries observed and the motion causing them at 16:57:29 UTC. (a) shows the case where both lines have blue asymmetries, (b) shows Ca II λ8542 having a blue asymmetry and Hα having a red asymmetry and (c) shows the case where both lines have red asymmetries.

tries.

While this is a good first step, improvements can be made in the training of RADYNVERSION to provide more physically reliable results. For instance, the grid of models RADYNVERSION is trained on are all heated by an electron beam with a triangular pulse time profile. This is not necessarily the time profile of an electron beam and so running more RADYN simulations with different beam heating profiles and including them in the training of RADYNVERSION could improve results. In fact, at times where the HXR flux peaks in the SOL20140906T17:09 M1.1 flare, the Hα line profile is centrally reversed with its horns broadened so much that it appears like an absorption profile. The current RADYNVERSION has not been trained on profiles this broad hence the inverted atmosphere seems quieter than one would expect (lower velocities and temperatures in the lower atmopshere). Including other beam heating profiles in the training of RADYNVERSION allows for a more predictable atmosphere to be recovered indicating that these profiles may be broadened to such an extent due to the beam heating profile. Moreover, including other heating mechanisms of the lower solar atmosphere, such as Alfvénic wave heating (Fletcher and Hudson, 2008; Hudson and Fletcher, 2009; Kerr et al., 2016), may also lead to more accurate atmospheric determinations. Also, Osborne et al. (2021) showed that proper non-local thermodynamic equilibrium (NLTE) treatment of the hydrogen Lyman lines has a non-negligible effect on the emergent Ca II λ8542 radiation. Resimulating the RADYN models with a proper treatment of the Lyman lines can then therefore change which atmospheric parameters map to what observations. This could potentially alter any of the conclusions previously drawn from the analysis in this chapter. Lastly, all of the simulations that RADYNVERSION was trained on have the same viewing angle ($\mu = 0.95$) which corresponds to a flare occurring at approximately disk centre on the Sun. As a result, any flares analysed that are not near disk centre will experience projection effects in their spectra that the recovered atmospheres will not take into account. This can be improved upon by including simulations from a multitude of viewing angles in the training dataset.

The RADYNVERSION model was used to determine the properties of the flaring atmosphere in the vicinity of the flare ribbons – points directly heated by flare energy – at three different times determined by their coincidence with the observed HXR from RHESSI. At each time the flare ribbons were identified using a GMM and DBSCAN with the asymmetry of the Hα and Ca II λ8542 spectra in each point calculated. Each location was then inverted by RADYNVERSION with 20,000 samples of

the latent distribution with the median profile of the atmospheric parameters used in the final analysis.

Throughout the times analysed, both red and blue asymmetric profiles were observed for both Hα and Ca II λ8542. In line with the works of Svestka (1976) and Canfield et al. (1990), this study finds that Hα profiles are mostly red asymmetric throughout the flare with small confined patches of blue asymmetry occurring during flare energy input (according to the HXR lightcurves) and after the first HXR spike. Canfield et al. (1990) remarked that the patches of blue asymmetry they observed occur in patches with diameter <10″, while the largest Hα blue asymmetric patches observed here are around half that size (such is the nature of having higher spatial resolution observations). Moreover, the red asymmetry of Hα is consistently present in agreement with Ichimoto and Kurokawa (1984).

Additionally, this study provides the first analysis of the asymmetries in Ca II λ8542 during a solar flare. Heinzel et al. (1994) previously studied the asymmetry during flares of the Ca II H spectral line and the Balmer lines of hydrogen concluding that the blue asymmetry in these lines are driven by downward plasma motions caused by the nonthermal electrons with a return current. Larger areas of blue asymmetry of Ca II λ8542 were observed during the aforementioned observations often occurring cospatially with the Hα blue asymmetry but in certains cases, Ca II λ8542 would be blue asymmetric while Hα would be red. Similarly, there are plenty of areas where Ca II λ8542 was red asymmetric and this always coincided with red asymmetric Hα.

The inverted atmospheres were studied to determine the flow velocity and temperature structure in the regions of line wing formation of Hα and Ca II λ8542 according to Kuridze et al. (2015) and Kerr et al. (2016), respectively. It was found at each of the three times that the explanation for the Ca II λ8542 asymmetries was always the same: the regions where the wings of Ca II λ8542 are formed at the times studied always had upflow velocities indicating the material responsible for the asymmetry was moving towards the observer. This means that when Ca II λ8542 is red asymmetric the upflowing material can be classed as absorbing with blue asymmetry being caused by emitting upflowing material. Overall, this can be interpreted as the plasma in the region of formation evaporating due to the injection of energy from the nonthermal electrons. Looking at the temperature profiles for each type of asymmetry shows an overall increase in temperature in the region of formation of Ca II λ8542 for the red asymmetric profiles. This solidifies the interpre-

tation of absorbing material as the hotter material moving upwards can have more of its Ca II λ8542 population ionised leading to a reduction in Ca II λ8542 emission.

The Hα asymmetries were more varied in their origin. During the early time, 16:48:05 UTC, the red asymmetric Hα was caused by downward flowing emitting material in some regions and upward flowing absorbing material in others. In particular, when Ca II λ8542 was blue asymmetric and Hα was red asymmetric, the cause of the red asymmetry in Hα in many locations was due to the upflow of absorbing material. This upflow of material could be tied to the same upflowing material responsible for the blue asymmetric Ca II λ8542 profiles. In this case, it is postulated that the plasma has enough energy for the Ca II λ8542 transition but not for Hα. Other locations with overlapping blue Ca II λ8542 and red Hα have downward emission responsible for the red asymmetric Hα. This is the result of downward condensation material being pushed into the lower atmosphere by energy deposition as described by Ichimoto and Kurokawa (1984). Regions of blue asymmetric Hα at this time coincided with blue Ca II λ8542 with upflows mostly being responsible for these asymmetries. These upflows of emitting material are thought to be the initial evaporation stages as the plasma is heated but before it is too hot to produce Hα and Ca II λ8542. The fact that the overlapping blue asymmetries in both lines do not occur very frequently points to this being a transient consequence of the efficient flare heating. There are also regions where the blue Hα asymmetry is caused by downflows interpreted to be of the same cause as given by Heinzel et al. (1994).

During both the 16:54:25 and 16:57:29 UTC observations, there is much more occurrence of blue asymmetric Hα. These points are much more spatially correlated with the Ca II λ8542 blue asymmetries and studying the velocity profiles in these regions points to downward moving plasma absorbing Hα responsible for these asymmetries. Given that the Ca II λ8542 blue asymmetry is caused by emitting upwards moving plasma and the Hα blue asymmetry is caused by absorbing downwards moving plasma indicates chromospheric condensation producing the Hα observations while chromospheric evaporation of the material producing the Ca II λ8542 observables takes place. This makes some semblance of sense when considering the RHESSI observations show that both of these times occur after high energy HXRs have been observed.

The red asymmetric Hα at time 16:54:25 UTC boast some locations being caused by downward moving emission and others by upward moving absorption. The locations correlate with the locations of red asymmetric Ca II λ8542 which is always

caused by upward moving absorption. In the regions where both lines' asymmetries are caused by upwards moving absorption, it is concluded that the evaporating material has reached such a temperature that it is too hot to produce either line and is opaque enough to obscure the plasma condensation below that may be emitting Hα. The converse was assumed about the opacity of the evaporating material when the red asymmetry of Hα was caused by downflowing emission: the material was still opaque to Ca II λ8542 but not to Hα where the emission of condensation material was observed.

At 16:57:29 UTC, the red asymmetric Hα was caused purely by downflowing emission. This can be seen as the condensation material cooling and emitting Hα after the original influx of nonthermal electrons.

An illustration of the different combinations and causes of asymmetries at this time is presented in Fig. 6.34. Similar demonstrations are given in Fig. 6.35 for 16:54:25 UTC and Fig. 6.36 for 16:57:29 UTC.

7 | Conclusions

In this thesis, the applications of machine learning techniques in solar flare data analysis pipelines from identifying flare ribbons observed in optical spectral lines to correcting residual seeing effects in such observations and estimating the underlying atmospheric conditions responsible for the observations. This has involved a mix of deep learning – the field of using deep neural networks to automate tasks – and clustering methods – a family of unsupervised machine learning algorithms used to categorise data based in their intrinsic properties. This thesis serves as a good proof of concept for how automation via machine learning can be beneficial in optimising workflows in solar physics as well as uncovering new answers to old problems. All of the code used in this thesis is publicly available under the MIT license with the code for Chap. 4 available at https://github.com/bionictoucan/slic/, Chap 5 https://github.com/bionictoucan/shaun/ and Chap. 6 https://github.com/bionictoucan/Radynversion/[1] & https://github.com/bionictoucan/ribbon_asymmetries.

In Chap. 4, the first deep convolutional neural network (CNN) trained on images of the Sun for the classification of different solar features is presented. This CNN, referred to as Slic, is trained on ~13,000 Hα images taken by Hinode/SOT split into five different categories: filaments, flare ribbons, prominences, sunspots and the lack of the other four features (the quiet Sun). The trained Slic model, benefitting from good initialisation via He initialisation (Sec. 2.3.1), took only four epochs to reach 99.2% classification accuracy on the validation dataset (corresponding to one misclassified image in the validation dataset). After further investigation via the confusion matrix, it was found that the misclassified image was an image of a fila-

[1]This is a fork of the original repo at https://github.com/Goobley/Radynversion/ and may not be up to date.

ment misclassified as flare ribbons indicating that more examples of filaments and flare ribbons in training could resolve this ambiguity in the knowledge of Slic. Slic was then presented with adversarial examples (examples whose answer is obvious to the user but can confuse the network) in the form of sunspot observations taken with SDO/AIA in 1600 & 1700Å UV continua and prominence observations taken with AIA in 304Å EUV band. This test showed that the trained network was not able to identify sunspots in 1600Å or prominences in 304Å at all and would only sparingly correctly identify a sunspot in 1700Å. Despite the perceptual similarity between Hα and 1600/1700Å sunspots, Slic does not have the capability to be applied to AIA UV data. This is potentially due to bright plage regions around the sunspots that are present in UV and not in Hα or other optical wavelengths. The bright plage confuses Slic into thinking the images contain flare ribbons as that is typically the class that dominates these misclassifications. For prominences in 304Å, the common misconception is again towards flare ribbons. This is thought to be due to the presence of noisy coronal emission in the background at the height of the prominence. In these examples, this acts like the quiet Sun background in flare ribbon images hence the misclassification. Furthermore, the background emission is not present in Hα prominence images at all. Finally, the use of Slic in transfer learning is realised in Chap. 5 as it is used to quantify the perceptual loss between reconstructed and ground truth images with good seeing.

Chapter 5 focuses on the creation and implementation of the Seeing AUtoeNcoder (Shaun). Shaun was designed to correct for residual atmospheric seeing in optical solar flare observations taken with SST/CRISP (Sec. 3.1). Shaun was developed as current postprocessing techniques for seeing correction are not suited for flares as they either need 100s of frames without much solar variation (Speckle methods) or use wideband images to aid in the restoration which do not always contain flare ribbons (MOMFBD). Firstly, a model was derived to imprint synthetic atmospheric seeing on images taken during good seeing conditions (and thus considered approximately diffraction-limited). This was done by starting from the statistical description of the Earth's atmosphere as a medium with smoothly varying turbulence. This led to the derivation of the Kolmogorov structure function (Eq. 5.63) which describes how the turbulence of the atmosphere varies from a defined centre point. A seeing disc was then defined based on the value of the Fried parameter wishing to be emulated and the observed wavelength of the light. This seeing disc was populated by Eq. 5.63 and convolved with good seeing images according to Eq. 5.4. Seeing discs

are then created for a range of Fried parameters ($r_0 = \{1, 2.5, 5, 10, 12.5, 15\}$ cm) and convolved with CRISP data to create a training dataset.

The two spectral lines of focus in the CRISP datasets of flares used are Hα and Ca II λ8542. Due to the difference in wavelength between their vacuum values, seeing will have different effects on them for the same value of the Fried parameter (as $r_0 \propto \lambda^{6/5}$) therefore two different Shauns were trained: one for Hα and one for Ca II λ8542. Both have the same fully convolutional autoencoder architecture and are trained identically besides different numbers of epochs for convergence. The performance on the validation dataset is good providing good reconstruction of small and large scale features within the field of view. The Hα model seems to perform more reliably than the Ca II λ8542 but this was due to artifacts in the Ca II λ8542 training dataset being learned by the network causing poor reconstructions. This could be fixed by a more careful construction process for the training data. The data is then applied to CRISP observations with real bad seeing to promising results. The trained Shauns are subject to an adhoc error estimate on the reconstructed intensities taken to be the average of the total loss of the system calculated over the whole training and validation dataset at the epoch of convergence. This is an incredibly fast method for correction of seeing taking ~500 ms for a 1k×1k image with 15 wavelength channels. Due to the advent of Pytorch's `torchscript` allowing for the compilation of deep learning models, this can be reduced to ~80 ms. Moreover, this compilation could be used to create a Shaun module that could be run on any system with sufficient hardware e.g. telescope data collection pipelines.

Chapter 6 introduced a novel deep learning approach to estimating the parameters of the solar atmosphere that lead to the observations: RADYNVERSION. This was done through the application of an invertible neural network (INN). Inverse problems are ill-defined due to information loss in the forward process meaning it is difficult to disambiguate the solution. INNs avoid the need for disambiguation via their architecture being mathematically invertible. This invertibility allows for a bijective solution to the inverse problem to be given.

RADYNVERSION learned from 1D RHD simulations of flares from the RADYN code the connection between the atmospheric parameters (electron density, temperature and bulk flow velocity) to the observable Hα and Ca II λ8542 spectral lines. In doing so, RADYNVERSION is capable of synthesising these spectral lines from a set of RADYN-like atmospheric parameters and, more importantly, can estimate the atmospheric parameters from a set of observations. This was an important step in

understanding the flaring chromosphere as traditional parameter estimation techniques used in solar physics were not as applicable to flares due to the radiative transfer occurring under hydrostatic equilibrium. This is the first method to produce atmospheric parameters as they are discussed elsewhere in solar physics literature but using the full RHD treatment of a flare.

The trained RADYNVERSION was then used over areas of entire flare ribbons at certain points indicated by the RHESSI 25-50keV light curve. The flare ribbons were identified using a combination of two classical unsupervised machine learning techniques: Gaussian Mixture Models (GMM) and Density-Based Spatial Clustering on Applications with Noise (DBSCAN). The GMM identified the different types of spectra in the data wishing to be analysed indicating points that are more likely to occur in a flare ribbon. DBSCAN was then used to filter out the points clustered into a flare ribbon spectral cluster but were far away from the flare ribbon structures in space. The estimated velocities and temperatures in the regions where the Hα and Ca II λ8542 spectra form was used to infer the atmospheric dynamics which leads to the asymmetric profiles observed.

These asymmetries were studied at three different times during the M1.1 flare based on when the observations were taken with regard to the RHESSI spectra. In particular, one time was selected before the initial HXR spike in 25-50keV, another during this spike and the last after this spike but before any subsequent spikes. This spike in 25-50keV also showed a healthy signal in the 50-100keV range indicating a good presence of nonthermal electrons. For blue asymmetric Ca II λ8542 profiles, it was found at all times that this asymmetry was caused by upflowing emitting plasma. An explanation for this is chromospheric evaporation which results from chromospheric plasma being heated to coronal temperatures by flare energy mechanisms causing the material to rise. At early stages of energy release, the evaporating material will be cooler allowing it to expand upwards while still producing Ca II λ8542 emission. This is supported by cospatial blue asymmetric Hα which is also caused by upflowing emitting plasma at early times indicating that the plasma is rising as it is heated. For the two later times, the cospatial blue Hα was caused by downward absorbing plasma showing that while there is evaporation, there is also hot material that is opaque to Hα.

For red symmetric Ca II λ8542 profiles, it was found that upflowing absorbing material was responsible. The temperature profiles of red asymmetric Ca II λ8542 had higher temperatures in the region of Ca II λ8542 formation pointing to the up-

ward moving material being opaque to Ca II λ8542. The Hα blue asymmetries were found to be caused by both upward emitting material and downward absorbing material while the red asymmetries were found to be caused by both upward absorbing material and downward emitting material depending on the spatial location and the time observed.

Future Work

While the work presented in this thesis is a good jumping off point in demonstrating the use of machine learning tools in data processing and analysis pipelines in solar physics, these works can be improved upon.

Firstly, the Slic network introduced in Chap. 4 shows the feasibility and robustness of deep convolutional neural networks when classifying images of the Sun. Currently, this model is trained only on Hα observations from Hinode/SOT but the expansion of the training set to include other wavelengths such as UV/EUV observations from SDO/AIA will increase its merit. Moreover, models such as these can be compressed and run on-board new solar space missions to aid in identification and pointing of the telescopes. Having an accurate model trained in many wavelengths would also benefit the tagging of data after downlink from satellite or after post-processing from a ground-based facility. There would be an increase in efficiency releasing these datasets to the public leading to more efficient science and less researcher time spent trawling through image databases.

Moving onto Shaun, introduced in Chap. 5. Shaun demonstrated the use of deep neural networks for data post-processing by learning to reconstruct imaging spectroscopic data of solar flares without any residual seeing left behind by any post-processing done up until that point. This was implemented as current methods for atmospheric seeing correction in solar observations are ill-suited to solar flares due to their subsecond evolution. Further development into Shaun and other similar methods would be twofold.

The model used to generate the synthetic seeing point-spread functions and thus the training data assumes that these PSFs are azimuthally-symmetric. As discussed in Sec. 5.1, for the applications of Shaun to time-integrated data this is a valid assumption but for the application to instantaneous frames of data being images by a telescope, the azimuthally-symmetric assumption is too simplistic – as shown in Asensio Ramos and Olspert (2021) and references therein, these PSFs are far from

azimuthally-symmetric. As such, the generation of the PSFs for instantaneous data correction would need to include a more accurate depiction of the PSFs such as using Zernike polynomials to generate them (Van Noort et al., 2005, and references therein).

Furthermore, the estimation of robust errors when removing the atmospheric seeing would increase the trustworthiness of Shaun. The author thinks that the best way to do this would be to apply a setup similar to Tonolini et al. (2019) who employ conditional variational autoencoders to sample the posterior distribution of images without applied noise. This could be easily applied to sample the posterior distribution of solar flare data corrected for seeing. This would also lead to an estimate of the confidence in the reconstructions.

Overall, Shaun can be applied today out-of-the-box to give accurate reconstructions of solar flare data after correcting for residual seeing leftover by post-processing techniques such as MOMFBD or Speckle interferometry. On the other hand, there are some critical improvements needed to both the synthetic seeing model and the learned seeing correction before it can be applied to raw frames at a telescope.

Lastly, the RADYNVERSION technique for estimating the atmospheric parameters of a solar flare was presented in Chap. 6. RADYNVERSION is a great first step in developing inferential tools to study the optical spectra of a solar flare. RADYNVERSION presents the first purpose-built inversion code to examine flares as it is trained specifically using flare simulations. This is an important step in studying the evolution of the flaring velocity field as this seems to be the most important when considering optical emissions during these events. However, as with all ML, RADYNVERSION is only as good as its training dataset meaning there is an asterisk attached to the results obtained using this method: the assumption that the simulations comprising the training dataset are accurate representations of solar flares. Increasing the training dataset to include more varied simulations would go a long way in improving the applicability of RADYNVERSION.

Finally, the RADYNVERSION model shows the applicability of INNs in physical inverse problems. It has shown that a mathematical inverse can be formulated for the data it is trained on and this may have implications in other areas of astrophysics where the scientists are always trying to make inferences about systems they have no control over. As with other ML algorithms used for parameter estimation, the INN has its drawbacks but its ability to learn a mathematically invertible function

has its appeal for anyone looking to study astrophysical phenomena.

The work presented in this thesis has presented the usefulness of exploiting machine learning tools for data processing and analysis purposes in solar flare physics. As the field of applied machine learning in solar physics continues to grow, it is prudent to remember that these "black box" methods are not so. There is a certain amount of mathematical rigour underpinning these methods that should not be understated. The uniqueness of data driven modelling lies in the unquantifiable. Many of the tasks set out to be modelled by machine learning do not have a nice formulation that can be simply written down. As such, the process of learning any of these functions is often seen as opaque and can be difficult to comprehend in the beginning[2]. But without any suspension of disbelief, it can be seen logically how these methods learn what they do. A system is being defined with $N >> 1$ degrees of freedom meaning there are innumerable possibilities of combinations of these degrees of freedom that can result in different answers. The user then gives the system a guide in the right direction and given that these are all conceived by human ingenuity, it should not be surprising that the conclusion the system arrives at is one that a person might given the same information. At the end of the day, everything is data and given a large enough parameter space anything can be learned.

[2]It took me about two years of this actually being my job to where I was comfortable saying I "get" what a neural network is doing.

Bibliography

Alfvén, H. (1942). Existence of Electromagnetic-Hydrodynamic Waves. *Nature*, 150(3805):405–406.

Allred, J. C., Hawley, S. L., Abbett, W. P., and Carlsson, M. (2005). Radiative Hydrodynamic Models of the Optical and Ultraviolet Emission from Solar Flares. *The Astrophysical Journal*, 630(1):573–586.

Allred, J. C., Kowalski, A. F., and Carlsson, M. (2015). A Unified Computational Model for Solar and Stellar Flares. *The Astrophysical Journal*, 809(1):104.

Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E. W., Klessen, R. S., Maier-Hein, L., Rother, C., and Köthe, U. (2019). Analyzing Inverse Problems with Invertible Neural Networks. *arXiv:1808.04730 [cs, stat]*. arXiv: 1808.04730.

Armstrong, J. A. and Fletcher, L. (2019). Fast Solar Image Classification Using Deep Learning and Its Importance for Automation in Solar Physics. *Solar Physics*, 294(6):80.

Armstrong, J. A. and Fletcher, L. (2021). A machine-learning approach to correcting atmospheric seeing in solar flare observations. *Monthly Notices of the Royal Astronomical Society*, 501(2):2647–2658.

Asensio Ramos, A. and Diaz Baso, C. (2019). Stokes Inversion based on Convolutional Neural Networks. *Astronomy & Astrophysics*, 626:A102. arXiv: 1904.03714.

Asensio Ramos, A., Martínez González, M. J., and Rubiño-Martín, J. A. (2007). Bayesian inversion of Stokes profiles. *Astronomy & Astrophysics*, 476(2):959–970.

Asensio Ramos, A. and Olspert, N. (2021). Learning to do multiframe wavefront sensing unsupervised: Applications to blind deconvolution. *Astronomy & Astrophysics*, 646:A100.

Asensio Ramos, A., Trujillo Bueno, J., and Landi Degl'Innocenti, E. (2008). Advanced Forward Modeling and Inversion of Stokes Profiles Resulting from the Joint Action of the Hanle and Zeeman Effects. *The Astrophysical Journal*, 683(1):542–565.

Bradshaw, S. J. and Cargill, P. J. (2013). The Influence of Numerical Resolution on Coronal Density in Hydrodynamic Models of Impulsive Heating. *The Astrophysical Journal*, 770(1):12.

Brown, J. C. (1971). The Deduction of Energy Spectra of Non-thermal Electrons in Flares from the Observed Dynamic Spectra of Hard X-ray Bursts. *Solar Physics*, 18.

Brown, S. A., Fletcher, L., Kerr, G. S., Labrosse, N., Kowalski, A. F., and Rodríguez, J. D. L. C. (2018). Modeling of the Hydrogen Lyman Lines in Solar Flares. *The Astrophysical Journal*, 862(1):59.

Bui, H. M., Lech, M., Cheng, E., Neville, K., and Burnett, I. S. (2016). Using grayscale images for object recognition with convolutional-recursive neural network. In *2016 IEEE Sixth International Conference on Communications and Electronics (ICCE)*, pages 321–325, Ha-Long City, Quang Ninh Province, Vietnam. IEEE.

Canfield, R. C. and Gunkler, T. A. (1984). The Hα Spectral Signatures of Solar Flare Nonthermal Electrons, Conductive Flux, and Coronal Pressure. *The Astrophysical Journal*, 282.

Canfield, R. C., Penn, M. J., Wulser, J.-P., and Kiplinger, A. L. (1990). H alpha Spectra of Dynamic Chromospheric Processes in Five Well-observed X-Ray Flares. *The Astrophysical Journal*, 363:318.

Capparelli, V., Zuccarello, F., Romano, P., Simões, P. J. A., Fletcher, L., Kuridze, D., Mathioudakis, M., Keys, P. H., Cauzzi, G., and Carlsson, M. (2017). H α and H β Emission in a C3.3 Solar Flare: Comparison between Observations and Simulations. *The Astrophysical Journal*, 850(1):36.

Carlsson, M. and Stein, R. F. (1992). Non-LTE Radiating Acoustic Shocks and Ca II K2V Bright Points. *The Astrophysical Journal*, 397:L59–L62.

Carlsson, M. and Stein, R. F. (1997). Formation of Solar Calcium H and K Bright Grains. *The Astrophysical Journal*, 481(1):500–514.

Carmichael, H. (1964). *A Process for Flares*, volume 50, page 451.

Carrington, R. C. (1859). Description of a Singular Appearance seen in the Sun on September 1, 1859. *Monthly Notices of the Royal Astronomical Society*, 20:13–15.

Cheng, J. X., Ding, M. D., and Li, J. P. (2006). Diagnostics of the Heating Processes in Solar Flares Using Chromospheric Spectral Lines. *The Astrophysical Journal*, 653(1):733–738.

Cheung, M. C. M., Boerner, P., Schrijver, C. J., Testa, P., Chen, F., Peter, H., and Malanushenko, A. (2015). Thermal Diagnostics with the Atmospheric Imaging Assembly on Board the Solar Dynamics Observatory: A Validated Method for Differential Emission Measure Inversions. *The Astrophysical Journal*, 807(2):143.

Choudhuri, A. R. (1998). *The physics of fluids and plasmas : an introduction for astrophysicists* /.

Costa, F. R. d., Kleint, L., Petrosian, V., Liu, W., and Allred, J. C. (2016). Data-driven Radiative Hydrodynamic Modeling of the 2014 March 29 X1.0 Solar Flare. *The Astrophysical Journal*, 827(1):38.

Culhane, J. L., Harra, L. K., James, A. M., Al-Janabi, K., Bradley, L. J., Chaudry, R. A., Rees, K., Tandy, J. A., Thomas, P., Whillock, M. C. R., Winter, B., Doschek, G. A., Korendyke, C. M., Brown, C. M., Myers, S., Mariska, J., Seely, J., Lang, J., Kent, B. J., Shaughnessy, B. M., Young, P. R., Simnett, G. M., Castelli, C. M., Mahmoud, S., Mapson-Menard, H., Probyn, B. J., Thomas, R. J., Davila, J., Dere, K., Windt, D., Shea, J., Hagood, R., Moye, R., Hara, H., Watanabe, T., Matsuzaki, K., Kosugi, T., Hansteen, V., and Wikstol, Ø. (2007). The EUV Imaging Spectrometer for Hinode. *Solar Physics*, 243(1):19–61.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314.

de la Cruz Rodríguez, J., Leenaarts, J., Danilovic, S., and Uitenbroek, H. (2019). STiC: A multiatom non-LTE PRD inversion code for full-Stokes solar observations. *Astronomy & Astrophysics*, 623:A74.

de la Cruz Rodríguez, J., Löfdahl, M. G., Sütterlin, P., Hillberg, T., and Rouppe van der Voort, L. (2015). CRISPRED: A data pipeline for the CRISP imaging spectropolarimeter. *Astronomy & Astrophysics*, 573:A40.

De Pontieu, B., McIntosh, S. W., Hansteen, V. H., and Schrijver, C. J. (2009). Observing the Roots of Solar Coronal Heating—in the Chromosphere. *The Astrophysical Journal*, 701(1):L1–L6.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. page 8.

Dinh, L., Krueger, D., and Bengio, Y. (2014). NICE: Non-linear Independent Components Estimation. *arXiv:1410.8516 [cs]*. arXiv: 1410.8516.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2016). Density estimation using Real NVP. *arXiv:1605.08803 [cs, stat]*. arXiv: 1605.08803.

Díaz Baso, C. J., de la Cruz Rodríguez, J., and Danilovic, S. (2019). Solar image denoising with convolutional neural networks. *Astronomy & Astrophysics*, 629:A99. arXiv: 1908.02815.

Elmore, D. F., Rimmele, T., Casini, R., Hegwer, S., Kuhn, J., Lin, H., McMullin, J. P., Reardon, K., Schmidt, W., Tritschler, A., and Wöger, F. (2014). The Daniel K. Inouye Solar Telescope first light instruments and critical science plan. In Ramsay, S. K., McLean, I. S., and Takami, H., editors, *Ground-based and Airborne Instrumentation for Astronomy V*, volume 9147 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 914707.

Emslie, A. G. and Smith, D. F. (1984). Microwave signature of thick-target electron beams in solar flares. *The Astrophysical Journal*, 279:882–895.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press.

Fang, C., Henoux, J., and Gan, W. (1993). Diagnostics of Non-thermal Processes in Chromospheric Flares I. Hα and CaII K Line Profiles of an Atmosphere Bombarded by hectra keV Electrons. *Astronomy & Astrophysics*, 274.

Feurer, M. and Hutter, F. (2019). *Hyperparameter Optimization*, pages 3–33. Springer International Publishing, Cham.

Fisher, G. H., Canfield, R. C., and McClymont, A. N. (1985). Flare Loop Radiative Hydrodynamics. V. Reponse to Thick-target Heating. *The Astrophysical Journal*, page 28.

Fletcher, L. (2009). Ultra-violet footpoints as tracers of coronal magnetic connectivity and restructuring during a solar flare. *Astronomy & Astrophysics*, 493(1):241–250.

Fletcher, L., Dennis, B. R., Hudson, H. S., Krucker, S., Phillips, K., Veronig, A., Battaglia, M., Bone, L., Caspi, A., Chen, Q., Gallagher, P., Grigis, P. C., Ji, H., Milligan, R. O., and Temmer, M. (2011). An Observational Overview of Solar Flares. *Space Science Reviews*, 159(1-4):19–106. arXiv: 1109.5932.

Fletcher, L. and Hudson, H. S. (2008). Impulsive Phase Flare Energy Transport by Large-Scale Alfvén Waves and the Electron Acceleration Problem. *The Astrophysical Journal*, 675(2):1645–1655.

Fletcher, L., Pollock, J. A., and Potts, H. E. (2004). Tracking of TRACE Ultraviolet Flare Footpoints. *Solar Physics*, 222(2):279–298.

Foukal, P. V. (2004). *Solar Astrophysics, 2nd, Revised Edition*.

Fried, D. L. (1966). Optical Resolution Through a Randomly Inhomogeneous Medium for Very Long and Very Short Exposures. *Journal of the Optical Society of America*, 56(10):1372–1379.

Fárník, F., Hudson, H., and Watanabe, T. (1996). Spatial Relations Between Preflares and Flares. *Solar Physics*, 165.

Gafeira, R., Orozco Suárez, D., Milić, I., Quintero Noda, C., Ruiz Cobo, B., and Uitenbroek, H. (2021). Machine learning initialization to accelerate Stokes profile inversions. *Astronomy & Astrophysics*, 651:A31.

Golub, L., Deluca, E., Austin, G., Bookbinder, J., Caldwell, D., Cheimets, P., Cirtain, J., Cosmo, M., Reid, P., Sette, A., Weber, M., Sakao, T., Kano, R., Shibasaki, K., Hara, H., Tsuneta, S., Kumagai, K., Tamura, T., Shimojo, M., McCracken, J., Carpenter, J., Haight, H., Siler, R., Wright, E., Tucker, J., Rutledge, H., Barbera, M., Peres, G., and Varisco, S. (2007). The X-Ray Telescope (XRT) for the Hinode Mission. *Solar Physics*, 243(1):63–86.

Graham, D. R. and Cauzzi, G. (2015). Temporal Evolution of Multiple Evaporating Ribbon Sources in a Solar Flare. *The Astrophysical Journal*, 807(2):L22.

Green, J. L., Boardsen, S., Odenwald, S., Humble, J., and Pazamickas, K. A. (2006). Eyewitness reports of the great auroral storm of 1859. *Advances in Space Research*, 38(2):145–154.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A Kernel Two-sample Test. *Journal of Machine Learning Research*, 13:723–773.

Handy, B. N., Acton, L. W., Kankelborg, C. C., Wolfson, C. J., Akin, D. J., Bruner, M. E., Caravalho, R., Catura, R. C., Chevalier, R., Duncan, D. W., Edwards, C. G., Feinstein, C. N., Freeland, S. L., Friedlaender, F. M., Hoffmann, C. H., Hurlburt, N. E., Jurcevich, B. K., Katz, N. L., Kelly, G. A., Lemen, J. R., Levay, M., Lindgren, R. W., Mathur, D. P., Meyer, S. B., Morrison, S. J., Morrison, M. D., Nightingale, R. W., Pope, T. P., Rehse,

R. A., Schrijver, C. J., Shine, R. A., Shing, L., Strong, K. T., Tarbell, T. D., Title, A. M., Torgerson, D. D., Golub, L., Bookbinder, J. A., Caldwell, D., Cheimets, P. N., Davis, W. N., Deluca, E. E., McMullen, R. A., Warren, H. P., Amato, D., Fisher, R., Maldonado, H., and Parkinson, C. (1999). The transition region and coronal explorer. *Solar Physics*, 187(2):229–260.

Hannah, I. G. and Kontar, E. P. (2012). Differential emission measures from the regularized inversion of Hinode and SDO data. *Astronomy & Astrophysics*, 539:A146.

Harra, L. K., Matthews, S. A., and Culhane, J. L. (2001). Nonthermal Velocity Evolution in the Precursor Phase of a Solar Flare. *The Astrophysical Journal*, 549(2):L245–L248.

Harra, L. K., Williams, D. R., Wallace, A. J., Magara, T., Hara, H., Tsuneta, S., Sterling, A. C., and Doschek, G. A. (2009). Coronal Nonthermal Velocity Following Helicity Injection Before an X-class Flare. *The Astrophysical Journal*, 691(2):L99–L102.

He, K., Zhang, X., Ren, S., and Sun, J. (2015a). Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*. arXiv: 1512.03385.

He, K., Zhang, X., Ren, S., and Sun, J. (2015b). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv:1502.01852 [cs]*. arXiv: 1502.01852.

Heinzel, P., Karlicky, M., Kotrc, P., and Svestka, Z. (1994). On the Occurrence of Blue Asymmetry in Chromospheric Flare Spectra. *Solar Physics*, 152(2):393–408.

Heinzel, P., Kašparová, J., Varady, M., Karlický, M., and Moravec, Z. (2016). Numerical RHD simulations of flaring chromosphere with Flarix. In Kosovichev, A. G., Hawley, S. L., and Heinzel, P., editors, *Solar and Stellar Flares and their Effects on Planets*, volume 320, pages 233–238.

Hirayama, T. (1974). Theoretical Model of Flares and Prominences. I: Evaporating Flare Model. *Solar Physics*, 34(2):323–338.

Hochreiter, S., Bengio, Y., Frasconi, P., and Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In Kremer, S. C. and Kolen, J. F., editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press.

Holman, G. D. (2012). Solar eruptive events. *Physics Today*, 65(4):56.

Hornik, K. (1991). Approximation Capabilities of Muitilayer Feedforward Networks. *Neural Networks*, 4:7.

Hua, X. M. and Lingenfelter, R. E. (1987). Solar Flare Neutron Production and the Angular Dependence of the Capture Gamma-Ray Emission. *Solar Physics*, 107(2):351–383.

Hudson, H. S. (1972). Thick-target Processes and White-light Flares. *Solar Physics*, 24.

Hudson, H. S. and Fletcher, L. (2009). Flares and the chromosphere. *Earth, Planets and Space*, 61(5):577–580.

Ichimoto, K. and Kurokawa, H. (1984). Hα Red Asymmetry of Solar Flares. *Solar Physics*, 93.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv:1502.03167 [cs]*. arXiv: 1502.03167.

Ivanov, S., Tsizh, M., Ullmann, D., Panos, B., and Voloshynovskiy, S. (2021). Solar activity classification based on Mg II spectra: Towards classification on compressed data. *Astronomy and Computing*, 36:100473.

Janett, G., Steiner, O., Alsina Ballester, E., Belluzzi, L., and Mishra, S. (2019). A novel fourth-order WENO interpolation technique. A possible new tool designed for radiative transfer. *Astronomy & Astrophysics*, 624:A104.

Jeffrey, N. L. S., Fletcher, L., and Labrosse, N. (2016). First evidence of non-Gaussian solar flare EUV spectral line profiles and accelerated non-thermal ion motion. *Astronomy & Astrophysics*, 590:A99.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution. *arXiv:1603.08155 [cs]*. arXiv: 1603.08155.

Jones, L. (1990). Constructive approximations for neural networks by sigmoidal functions. *Proceedings of the IEEE*, 78(10):1586–1589.

Kerr, G. S., Carlsson, M., and Allred, J. C. (2019a). Modeling Mg II during Solar Flares. II. Nonequilibrium Effects. *The Astrophysical Journal*, 885(2):119.

Kerr, G. S., Carlsson, M., Allred, J. C., Young, P. R., and Daw, A. N. (2019b). SI IV Resonance Line Emission during Solar Flares: Non-LTE, Nonequilibrium, Radiation Transfer Simulations. *The Astrophysical Journal*, 871(1):23.

Kerr, G. S., Fletcher, L., Russell, A. J. B., and Allred, J. C. (2016). Simulations of the Mg ii K and Ca ii 8542 Lines From an Alfvén Wave-heated Flare Chromosphere. *The Astrophysical Journal*, 827(2):101.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*. arXiv: 1412.6980.

Kiselman, D., Pereira, T. M. D., Gustafsson, B., Asplund, M., Meléndez, J., and Langhans, K. (2011). Is the solar spectrum latitude-dependent?. An investigation with SST/TRIPPEL. *Astronomy & Astrophysics*, 535:A14.

Kolmogorov, A. (1941). The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds' Numbers. *Akademiia Nauk SSSR Doklady*, 30:301–305.

Kopp, R. A. and Pneuman, G. W. (1976). Magnetic reconnection in the corona and the loop prominence phenomenon. *Solar Physics*, 50(1):85–98.

Kosugi, T., Matsuzaki, K., Sakao, T., Shimizu, T., Sone, Y., Tachikawa, S., Hashimoto, T., Minesugi, K., Ohnishi, A., Yamada, T., Tsuneta, S., Hara, H., Ichimoto, K., Suematsu, Y., Shimojo, M., Watanabe, T., Shimada, S., Davis, J. M., Hill, L. D., Owens, J. K., Title, A. M., Culhane, J. L., Harra, L. K., Doschek, G. A., and Golub, L. (2007). The Hinode (Solar-B) Mission: An Overview. *Solar Physics*, 243(1):3–17.

Kowalski, A. F., Allred, J. C., Daw, A., Cauzzi, G., and Carlsson, M. (2017). The Atmospheric Response to High Nonthermal Electron Beam Fluxes in Solar Flares. I. Modeling the Brightest NUV Footpoints in the X1 Solar Flare of 2014 March 29. *The Astrophysical Journal*, 836(1):12.

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J. (2017). DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks. *arXiv:1711.07064 [cs]*. arXiv: 1711.07064.

Kuridze, D., Henriques, V., Mathioudakis, M., Koza, J., Zaqarashvili, T. V., Rybák, J., Hanslmeier, A., and Keenan, F. P. (2017). Spectroscopic Inversions of the Ca ii 8542 å Line in a C-class Solar Flare. *The Astrophysical Journal*, 846(1):9.

Kuridze, D., Henriques, V. M. J., Mathioudakis, M., van der Voort, L. R., Cruz Rodríguez, J. d. l., and Carlsson, M. (2018). Spectropolarimetric Inversions of the Ca ii 8542 å Line in an M-class Solar Flare. *The Astrophysical Journal*, 860(1):10.

Kuridze, D., Mathioudakis, M., Simões, P. J. A., Voort, L. R. v. d., Carlsson, M., Jafarzadeh, S., Allred, J. C., Kowalski, A. F., Kennedy, M., Fletcher, L., Graham, D., and Keenan, F. P. (2015). H α Line Profile Asymmetries and the Chromospheric Flare Velocity Field. *The Astrophysical Journal*, 813(2):125.

Kurokawa, H., Kitahara, T., Nakai, Y., Funakoshi, Y., and Ichimoto, K. (1986). High resolution observation of Hα solar flares and temporal relation between Hα and X-ray, microwave emission. *Astrophysics & Space Science*, 118(1-2):149–152.

Labrosse, N., Heinzel, P., Vial, J. C., Kucera, T., Parenti, S., Gunár, S., Schmieder, B., and Kilper, G. (2010). Physics of Solar Prominences: I—Spectral Diagnostics and Non-LTE Modelling. *Solar System Reviews*, 151(4):243–332.

LeCun, Y., Bottou, L., Bengio, Y., and Ha, P. (1998). Gradient-Based Learning Applied to Document Recognition. In *Proc. of IEEE*, page 46.

Lemen, J. R., Title, A. M., Akin, D. J., Boerner, P. F., Chou, C., Drake, J. F., Duncan, D. W., Edwards, C. G., Friedlaender, F. M., Heyman, G. F., Hurlburt, N. E., Katz, N. L., Kushner, G. D., Levay, M., Lindgren, R. W., Mathur, D. P., McFeaters, E. L., Mitchell, S., Rehse, R. A., Schrijver, C. J., Springer, L. A., Stern, R. A., Tarbell, T. D., Wuelser, J.-P., Wolfson, C. J., Yanari, C., Bookbinder, J. A., Cheimets, P. N., Caldwell, D., Deluca, E. E., Gates, R., Golub, L., Park, S., Podgorski, W. A., Bush, R. I., Scherrer, P. H., Gummin, M. A., Smith, P., Auker, G., Jerram, P., Pool, P., Soufli, R., Windt, D. L., Beardsley, S., Clapp, M., Lang, J., and Waltham, N. (2012). The Atmospheric Imaging Assembly (AIA) on the Solar Dynamics Observatory (SDO). *Solar Physics*, 275(1-2):17–40.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 2(2):164–168.

Lin, R. P. (2000). The High Energy Solar Spectroscopic Imager (HESSI) Mission. In *High Energy Solar Physics: Anticipating HESSI*, volume 206 of *ASP Conference Series*.

Lingenfelter, R. E. and Ramaty, R. (1967). High Energy Nuclear Reactions in Solar Flares. In *High-Energy Nuclear Reactions in Astrophysics*, page 99.

Löfdahl, M. G., Hillberg, T., de la Cruz Rodríguez, J., Vissers, G., Andriienko, O., Scharmer, G. B., Haugan, S. V. H., and Fredvik, T. (2021). SSTRED: Data- and metadata-processing pipeline for CHROMIS and CRISP. *Astronomy & Astrophysics*, 653:A68.

Lowe, D. and Zapart, K. (1999). Point-wise Confidence Interval Estimation by Neural Networks: A comparative study based on automotive engine calibration. *Neural Computing and Applications*, 8:77–85.

Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017). The Expressive Power of Neural Networks: A View from the Width. *arXiv:1709.02540 [cs]*. arXiv: 1709.02540.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441.

Mein, P., Mein, N., Malherbe, J., Heinzel, P., Kneer, F., von Uexkull, M., and Staiger, J. (1997). Flare Multi-line 2D-spectroscopy. *Solar Physics*, 172:10.

Milić, I. and van Noort, M. (2017). Line response functions in nonlocal thermodynamic equilibrium. Isotropic case. *Astronomy & Astrophysics*, 601:A100.

Millar, D. C. L., Fletcher, L., and Milligan, R. O. (2021). The effect of a solar flare on chromospheric oscillations. *Monthly Notices of the Royal Astronomical Society*, 503(2):2444–2456.

Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning*, ICML'10, pages 807–814.

Neupert, W. M. (1968). Comparison of Solar X-Ray Line Emission with Microwave Emission during Flares. *The Astrophysical Journal*, 153:L59.

Obukhov, A. M. (1970). Structure of the temperature field in turbulent flow. *Defense Technical Information Center*.

Osborne, C. M. J. (2021a). *Casting a new light on radiative transfer in solar flare models: synthesis and inversion*. PhD thesis, University of Glasgow.

Osborne, C. M. J. (2021b). Weno4interpolation.

Osborne, C. M. J., Armstrong, J. A., and Fletcher, L. (2019). RADYNVERSION: Learning to Invert a Solar Flare Atmosphere with Invertible Neural Networks. *The Astrophysical Journal*, 873(2):128.

Osborne, C. M. J., Heinzel, P., Kašparová, J., and Fletcher, L. (2021). On the importance of Ca II photoionization by the hydrogen lyman transitions in solar flare models. *Monthly Notices of the Royal Astronomical Society*, 507(2):1972–1978.

Panos, B. and Kleint, L. (2021). Exploring Mutual Information between IRIS Spectral Lines. II. Calculating the Most Probable Response in all Spectral Windows. *The Astrophysical Journal*, 915(2):77.

Panos, B., Kleint, L., Huwyler, C., Krucker, S., Melchior, M., Ullmann, D., and Voloshynovskiy, S. (2018). Identifying Typical Mg ii Flare Spectra Using Machine Learning. *The Astrophysical Journal*, 861(1):62.

Panos, B., Kleint, L., and Voloshynovskiy, S. (2021). Exploring Mutual Information between IRIS Spectral Lines. I. Correlations between Spectral Lines during Solar Flares and within the Quiet Sun. *The Astrophysical Journal*, 912(2):121.

Park, S. and Kwak, N. (2017). Analysis on the Dropout Effect in Convolutional Neural Networks. In Lai, S.-H., Lepetit, V., Nishino, K., and Sato, Y., editors, *Computer Vision – ACCV 2016*, volume 10112, pages 189–204. Springer International Publishing, Cham.

Pesnell, W. D., Thompson, B. J., and Chamberlin, P. C. (2012). The Solar Dynamics Observatory (SDO). *Solar Physics*, 275(1-2):3–15.

Racine, R. (1996). The Telescopic Point Spread Function. *Publications of the Astronomical Society of the Pacific*, 108:699–705.

Read, J., Pfahringer, B., Holmes, G., and Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359.

Rokach, L., Schclar, A., and Itach, E. (2014). Ensemble methods for multi-label classification. *Expert Systems with Applications*, 41(16):7507–7523.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Ruiz Cobo, B. and Del Toro Iniesta, J. (1992). Inversion of Stokes Profiles. *The Astrophysical Journal*, 398:375–385.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986a). 8. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing : Explorations in the Microstructure of Cognition*, volume 1. MIT Press.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986b). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Scharmer, G. B. (2006). Comments on the optimization of high resolution Fabry-Pérot filtergraphs. *Astronomy & Astrophysics*, 447(3):1111–1120.

Scharmer, G. B., Bjelksjo, K., Korhonen, T. K., Lindberg, B., and Petterson, B. (2003). The 1-meter Swedish solar telescope. In Keil, S. L. and Avakyan, S. V., editors, *Proc. SPIE 4853,*

Innovative Telescopes and Instrumentation for Solar Astrophysics, page 341, Waikoloa, Hawai'i, United States.

Scharmer, G. B., Narayan, G., Hillberg, T., de la Cruz Rodriguez, J., Löfdahl, M. G., Kiselman, D., Sütterlin, P., van Noort, M., and Lagg, A. (2008). CRISP Spectropolarimetric Imaging of Penumbral Fine Structure. *The Astrophysical Journal*, 689(1):L69–L72.

Scherrer, P. H., Schou, J., Bush, R. I., Kosovichev, A. G., Bogart, R. S., Hoeksema, J. T., Liu, Y., Duvall, T. L., Zhao, J., Title, A. M., Schrijver, C. J., Tarbell, T. D., and Tomczyk, S. (2012). The Helioseismic and Magnetic Imager (HMI) Investigation for the Solar Dynamics Observatory (SDO). *Solar Physics*, 275(1-2):207–227.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3).

Shea, M. A. and Smart, D. F. (2006). Compendium of the eight articles on the "Carrington Event" attributed to or written by Elias Loomis in the American Journal of Science, 1859 1861. *Advances in Space Research*, 38(2):313–385.

Simard, P., Steinkraus, D., and Platt, J. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, volume 1, pages 958–963, Edinburgh, UK. IEEE Comput. Soc.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs]*. arXiv: 1409.1556.

Simões, P. J. A., Kerr, G. S., Fletcher, L., Hudson, H. S., Giménez de Castro, C. G., and Penn, M. (2017). Formation of the thermal infrared continuum in solar flares. *Astronomy & Astrophysics*, 605:A125.

Sisti, M., Finelli, F., Pedrazzi, G., Faganello, M., Califano, F., and Delli Ponti, F. (2021). Detecting Reconnection Events in Kinetic Vlasov Hybrid Simulations Using Clustering Techniques. *The Astrophysical Journal*, 908(1):107.

Skumanich, A. and Lites, B. W. (1987). Stokes Profile Analysis and Vector Magnetic Fields I. Inversion of Photospheric Lines. *The Astrophysical Journal*, 322:473–482.

Socas-Navarro, H., de la Cruz Rodríguez, J., Asensio Ramos, A., Trujillo Bueno, J., and Ruiz Cobo, B. (2015). An open-source, massively parallel code for non-LTE synthesis and inversion of spectral lines and Zeeman-induced Stokes profiles. *Astronomy & Astrophysics*, 577:A7.

Socas-Navarro, H., Trujillo Bueno, J., and Ruiz Cobo, B. (2000). Non-LTE Inversion of Stokes Profiles Induced by the Zeeman Effect. *The Astrophysical Journal*, 530:977–993.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Lanckriet, G. R., and Schölkopf, B. (2009). Kernel Choice and Classifiability for RKHS Embeddings of Probability Distributions. page 9.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. page 30.

Sturrock, P. A. (1966). Model of the High-Energy Phase of Solar Flares. *Nature*, 211(5050):695–697.

Sutskever, I., Martens, J., and Dahl, G. (2013). On the importance of initialization and momentum in deep learning. page 9.

Svestka, Z. (1976). *Solar Flares*.

Tatarski, V. I. (2016). *Wave Propagation in a Turbulent Medium*. Dover Publications Inc., Mineola, New York, reissue edition.

Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2017). Wasserstein Auto-Encoders. *arXiv:1711.01558 [cs, stat]*. arXiv: 1711.01558.

Tonolini, F., Radford, J., Turpin, A., Faccio, D., and Murray-Smith, R. (2019). Variational Inference for Computational Imaging Inverse Problems. *arXiv e-prints*, page arXiv:1904.06264.

Tsuneta, S., Ichimoto, K., Katsukawa, Y., Nagata, S., Otsubo, M., Shimizu, T., Suematsu, Y., Nakagiri, M., Noguchi, M., Tarbell, T., Title, A., Shine, R., Rosenberg, W., Hoffmann, C., Jurcevich, B., Kushner, G., Levay, M., Lites, B., Elmore, D., Matsushita, T., Kawaguchi, N., Saito, H., Mikami, I., Hill, L. D., and Owens, J. K. (2008). The Solar Optical Telescope for the Hinode Mission: An Overview. *Solar Physics*, 249(2):167–196.

Tsurutani, B. T., Gonzalez, W. D., Lakhina, G. S., and Alex, S. (2003). The extreme magnetic storm of 1-2 September 1859. *Journal of Geophysical Research (Space Physics)*, 108(A7):1268.

Uitenbroek, H. (2001). Multilevel Radiative Transfer with Partial Frequency Redistribution. *The Astrophysical Journal*, 557(1):389–398.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv:1607.08022 [cs]*. arXiv: 1607.08022.

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., Yu, T., and the scikit-image contributors (2014). scikit-image: image processing in Python. *PeerJ*, 2:e453.

van Driel-Gesztelyi, L. and Green, L. M. (2015). Evolution of Active Regions. *Living Reviews in Solar Physics*, 12(1):1.

Van Noort, M., Der Voort, L. R. V., and Löfdahl, M. G. (2005). Solar Image Restoration By Use Of Multi-frame Blind De-convolution With Multiple Objects And Phase Diversity. *Solar Physics*, 228(1-2):191–215.

Varady, M., Kasparova, J., Moravec, Z., Heinzel, P., and Karlicky, M. (2010). Modeling of Solar Flare Plasma and Its Radiation. *IEEE Transactions on Plasma Science*, 38(9):2249–2253.

Vernazza, J. E., Avrett, E. H., and Loeser, R. (1981). Structure of the solar chromosphere. III - Models of the EUV brightness components of the quiet-sun. *The Astrophysical Journal Supplement Series*, 45:635.

von der Lühe, O. (1993). Speckle Imaging of Solar Small Scale Structure I. Methods. *Astronomy & Astrophysics*, 268:374–390.

von der Lühe, O. and Dunn, R. (1987). Solar Granulation Power Spectra from Speckle Interferometry. *Astronomy & Astrophysics*, 177:265–276.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. (2016). CNN-RNN: A Unified Framework for Multi-label Image Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2285–2294, Las Vegas, NV, USA. IEEE.

Woods, T. N., Eparvier, F. G., Hock, R., Jones, A. R., Woodraska, D., Judge, D., Didkovsky, L., Lean, J., Mariska, J., Warren, H., McMullin, D., Chamberlin, P., Berthiaume, G., Bailey, S., Fuller-Rowell, T., Sojka, J., Tobiska, W. K., and Viereck, R. (2012). Extreme Ultraviolet Variability Experiment (EVE) on the Solar Dynamics Observatory (SDO): Overview of Science Objectives, Instrument Design, Data Products, and Model Developments. *Solar Physics*, 275(1-2):115–143.

Wülser, J.-P. and Marti, H. (1987). High Time Resolution Observations of Hα Line Profiles During the Impulsive Phase of a Solar Flare. *The Astrophysical Journal*, 341:9.