The Astronomical Journal, 164:49 (21pp), 2022 August https://doi.org/10.3847/1538-3881/ac738e

Identifying Exoplanets with Deep Learning. IV. Removing Stellar Activity Signals from Radial Velocity Measurements Using Neural Networks

Zoe. L. de Beurs[1,2,3,4], Andrew Vanderburg[2,3,5,24], Christopher J. Shallue[6], Xavier Dumusque[7], Andrew Collier Cameron[8], Christopher Leet[9], Lars A. Buchhave[10], Rosario Cosentino[11], Adriano Ghedina[11], Raphaëlle D. Haywood[12,25], Nicholas Langellier[6,13], David W. Latham[6], Mercedes López-Morales[6], Michel Mayor[7], Giusi Micela[14], Timothy W. Milbourne[6,13], Annelies Mortier[15,16], Emilio Molinari[17], Francesco Pepe[7], David F. Phillips[6], Matteo Pinamonti[18], Giampaolo Piotto[19,20], Ken Rice[21,22], Dimitar Sasselov[6], Alessandro Sozzetti[18], Stéphane Udry[7], and Christopher A. Watson[23]

[1] Department of Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139, USA; zdebeurs@mit.edu
[2] Department of Astronomy, University of Texas at Austin, 2515 Speedway, Austin, TX 78712, USA
[3] Department of Astronomy, University of Wisconsin−Madison, Madison, WI 53706, USA
[4] NSF Graduate Research Fellow
[5] Department of Physics and Kavli Institute for Astrophysics and Space Research, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[6] Center for Astrophysics| Harvard & Smithsonian, 60 Garden Street, Cambridge, MA 02138, USA
[7] Observatoire de Genève, Université de Genève, 51 chemin des Maillettes, 1290 Versoix, Switzerland
[8] Centre for Exoplanet Science, SUPA, School of Physics and Astronomy, University of St Andrews, St Andrews KY16 9SS, UK
[9] Facebook, 181 Fremont St., San Francisco, CA 94105, USA
[10] DTU Space, National Space Institute, Technical University of Denmark, Elektrovej 328, DK-2800 Kgs. Lyngby, Denmark
[11] Fundacion Galileo Galilei-INAF, Rambla J. A. F. Perez, 7, E-38712, S.C. Tenerife, Spain
[12] Astrophysics Group, University of Exeter, Exeter EX4 2QL, UK
[13] Department of Physics, Harvard University, 17 Oxford Street, Cambridge, MA 02138, USA
[14] INAF-Osservatorio Astronomico di Palermo, Piazza del Parlamento 1, I-90134 Palermo, Italy
[15] Astrophysics Group, Cavendish Laboratory, J.J. Thomson Avenue, Cambridge CB3 0HE, UK
[16] Kavli Institute for Cosmology, University of Cambridge, Madingley Road, Cambridge CB3 0HA, UK
[17] INAF-Osservatorio Astronomico di Cagliari, Via della Scienza 5, I-09047 Selargius CA, Italy
[18] INAF-Osservatorio Astrofisico di Torino, via Osservatorio 20, I-10025 Pino Torinese, Italy
[19] Dipartimento di Fisica e Astronomia "Galileo Galilei," Università di Padova, Vicolo dell'Osservatorio 3, I-35122 Padova, Italy
[20] INAF-Osservatorio Astronomico di Padova, Vicolo dell'Osservatorio 5, I-35122 Padova, Italy
[21] SUPA, Institute for Astronomy, Royal Observatory, University of Edinburgh, Blackford Hill, Edinburgh EH93HJ, UK
[22] Centre for Exoplanet Science, University of Edinburgh, Edinburgh, UK
[23] Astrophysics Research Centre, School of Mathematics and Physics, Queen's University Belfast, BT7 1NN, Belfast, UK
Received 2020 October 13; revised 2022 April 21; accepted 2022 May 19; published 2022 July 13

Abstract

Exoplanet detection with precise radial velocity (RV) observations is currently limited by spurious RV signals introduced by stellar activity. We show that machine-learning techniques such as linear regression and neural networks can effectively remove the activity signals (due to starspots/faculae) from RV observations. Previous efforts focused on carefully filtering out activity signals in time using modeling techniques like Gaussian process regression. Instead, we systematically remove activity signals using only changes to the average shape of spectral lines, and use no timing information. We trained our machine-learning models on both simulated data (generated with the SOAP 2.0 software) and observations of the Sun from the HARPS-N Solar Telescope. We find that these techniques can predict and remove stellar activity both from simulated data (improving RV scatter from 82 to 3 cm s^{-1}) and from more than 600 real observations taken nearly daily over 3 yr with the HARPS-N Solar Telescope (improving the RV scatter from 1.753 to 1.039 m s^{-1}, a factor of ∼1.7 improvement). In the future, these or similar techniques could remove activity signals from observations of stars outside our solar system and eventually help detect habitable-zone Earth-mass exoplanets around Sun-like stars.

Unified Astronomy Thesaurus concepts: Exoplanet astronomy (486); Radial velocity (1332); Convolutional neural networks (1938)

Supporting material: data behind figure, animation

1. Introduction

The radial velocity (RV) method has seen tremendous improvements since the first detections of exoplanets around Sun-like stars between 1988 and 1995 (Campbell et al. 1988; Latham et al. 1989; Mayor & Queloz 1995). Currently, the primary challenge in measuring extremely precise RVs (EPRVs) is overcoming noise from stellar variability (National Academies of Sciences, Engineering, and Medicine and others 2018; Haywood et al. 2020). The surfaces of Sun-like stars are affected by numerous phenomena from convective

[24] NASA Sagan Fellow.
[25] STFC Ernest Rutherford Fellow.

granulation to magnetic activity in the form of spots, plages, and faculae. Due to the time-evolving and sometimes periodic nature of these features, they have been mistaken for planets on several occasions (e.g., Queloz et al. 2001; Huélamo et al. 2008; Setiawan et al. 2008) and can severely complicate the interpretation of RV measurements. Currently, these forms of stellar variability commonly limit RV measurement precision to $\gtrsim$1 m s^{-1} (Haywood et al. 2016; Dumusque 2018). To detect the 10 cm s^{-1} signals induced by Earth-mass exoplanets in the habitable zones of Sun-like stars, our limiting RV precision must improve by an order of magnitude.

Characterizing and removing these stellar activity[26] signals is especially crucial and timely, as current and future high-resolution spectrographs (including HARPS, Mayor et al. 2003; Wilken et al. 2012; HARPS-N, Cosentino et al. 2012; ESPRESSO, Pepe et al. 2020 (accepted); G-CLEF, Szentgyorgyi et al. 2014; EXPRES, Jurgenson et al. 2016) already have (Anglada-Escudé et al. 2016; Suárez Mascareño et al. 2020) or are expected to reach the long-term instrumental RV precision necessary to detect Earth-mass habitable-zone exoplanets.

The signals that limit RV precision on stars like the Sun are caused by four main physical processes:

1. *Solar-type oscillations*.—Produced by pressure waves propagating at the surface, pressure-mode (*p*-mode) oscillations result in a contraction and expansion of the external envelope of the star on timescales of a few minutes (Leighton et al. 1962; Ulrich 1970; Kjeldsen & Bedding 1995; Butler et al. 2003; Arentoft et al. 2008). These oscillations can produce RV signals ranging from 10 cm s^{-1} to 1 m s^{-1} for solar-like stars (Arentoft et al. 2008). The period and amplitude vary depending on the stellar type and evolutionary stage. For our Sun, this RV variation is at the 0.5 m s^{-1} level at a ∼5-minute period (Strassmeier et al. 2018; Cegla 2019).
2. *Granulation phenomena*.—Originating from convection in the outer layers of solar-type stars, granulation and supergranulation can induce RV signals at the m s^{-1} level (Lefebvre et al. 2008; Dumusque et al. 2011b) on timescales from a few minutes up to 48 hr. These granulation phenomena are found throughout the photosphere, except in active regions, where convection is suppressed by magnetic fields (Dravins et al. 1981; Livingston 1982; Brandt & Solanki 1990).
3. *Short-term stellar activity*.—Induced by stellar rotation paired with dark spots and bright faculae on the surface of the Sun, short-term stellar activity is caused by two different physical effects. In the first effect, the presence of strong magnetic fields in active regions suppresses the convection and thereby the convective blueshift effect (Dravins 1982; Livingston 1982; Cavallini et al. 1985; Brandt & Solanki 1990). Relative to the quiet photosphere, the active regions then seem redshifted (Cavallini et al. 1985). As the active regions come in and out of view during the rotation, they produce RV signals of ∼0.4–1.4 m s^{-1} for the Sun (Meunier et al. 2010). In the second effect, the temperature difference between these active regions and the quiet photosphere results in flux differences. For example, dark sunspots are ∼700 K cooler and thus have much lower flux than the rest of the star (Meunier et al. 2010). In this way, spots break the balance between the blueshifted approaching limb and the redshifted receding limb as they pass across the stellar disk and induce RV variations that can reach 0.4 m s^{-1} on the Sun at high activity (Saar & Donahue 1997; Meunier et al. 2010). In other cases, like young stars and M dwarfs, dark spots can dominate the activity signals. Both the suppression of convective blueshift effect and the flux effect produce RV variations on the timescale of the rotation period.
4. *Long-term stellar activity*.—Generated by solar-like magnetic activity cycles, long-term stellar activity variations influence RV measurements on the timescale of several years. In solar-type magnetic cycles, the filling factor of active regions increases during high-activity phases. Since the increase in magnetic field in active regions suppresses the convection (and thereby convective blueshift), these areas will be relatively redshifted (positive velocity) as the activity level rises. Thus, the activity level and RVs are positively correlated (Lindegren & Dravins 2003; Meunier et al. 2010). Dumusque et al. (2011a) found that stars other than the Sun can also have these solar-like magnetic cycles and their corresponding long-term RV variations. For our Sun, we observe an 11 yr magnetic cycle during which the sunspot number varies from zero to ∼150–200 on the visible hemisphere (Hathaway 2015) and large, bright magnetic regions can dominate solar RVs (Milbourne et al. 2019).

On the Sun, all four of these phenomena contribute activity signals with comparable amplitudes. In RV analysis, these signals are often aggregated into a single measurement of the stellar activity. When approximating each source of RV variations as Gaussian noise, the total scatter in an RV observation, $\sigma_{\rm tot}$, can be summarized as

$$\sigma_{\rm tot} \approx \sqrt{\sigma^2_{\rm phot} + \sigma^2_{\rm ins} + \sigma^2_{\rm magn} + \sigma^2_{\rm gran} + \sigma^2_{p-{\rm mode}}}, \qquad (1)$$

where $\sigma_{\rm phot}$ is photon noise, $\sigma_{\rm ins}$ is instrumental noise, $\sigma_{\rm magn}$ originates from both short and long-term activity, $\sigma_{\rm gran}$ is scatter from granulation phenomena, and $\sigma_{p-{\rm mode}}$ is scatter from *p*-modes.

To mitigate some of these forms of stellar variability, observing strategies have been developed to average out noise from granulation phenomena and *p*-mode oscillations. Dumusque et al. (2011b) showed that RV signals caused by these two stellar noise sources can be averaged out with longer integration times and a higher frequency of observations throughout the night (or day in the case of observing the Sun). Later, Medina et al. (2018) extended this strategy to evolved stars. For *p*-mode oscillations specifically, Chaplin et al. (2019) demonstrated that fine-tuning exposure times to stellar parameters (e.g., 5.4 minutes for the Sun) can also efficiently average out *p*-modes down to ∼10 cm s^{-1}.

Other methods of distinguishing planetary systems from stellar variability include tracing activity indicators such as log $R'_{\rm HK}$ (Noyes et al. 1984), the Bisector Inverse Slope Span (Queloz et al. 2001), and Hα (Bonfils et al. 2007; Robertson et al. 2014); using statistical methods (like Gaussian processes (GPs), Haywood et al. 2014; Rajpaul et al. 2015; Jones et al. 2017; Delisle et al. 2018; moving average, Tuomi et al. 2013; tomography modeling, Donati et al. 2014); or measuring the

[26] We note that throughout the paper we will at times refer to stellar variability and stellar activity interchangeably because the main variability contribution in our data set is magnetic activity.

RV from stellar lines that are the least affected by stellar activity (Dumusque 2018; Cretignier et al. 2020) to reduce the impact of stellar activity in RV data sets. Other methods of capturing stellar activity variations include using photometry (the FF' method, Aigrain et al. 2012; a GP framework that extends the FF' method, Rajpaul et al. 2015); using simultaneous spectroscopy and photometry to disentangle the contribution of spots, plagues, and network regions to the RV signal (Milbourne et al. 2021); and combining RV metrics with solar photometry to predict rotation periods (Kosiarek & Crossfield 2020). More recently, several studies have investigated how individual spectral lines are affected differently by stellar activity (Dumusque 2018; Cretignier et al. 2021; Wise et al. 2022).

Although methods such as the FF' method and the GP frameworks have been successfully applied to numerous data sets and detected low-amplitude planetary signals, they often require and rely on high-cadence and carefully timed observations. It is often difficult to obtain such timely observations given the myriad scheduling constraints involved in running astronomical observatories. Ideally, we would employ a method that successfully addresses short-term stellar activity but does not require high sampling or timing information that is necessary for GPs and moving averages. In this paper, we illustrate that machine-learning (ML) algorithms have the potential to resolve both these challenges by identifying changes to the average shape of spectral lines. While this technique does require a substantial set of observations for training, densely sampled observations are not necessary.

Neural networks, a form of ML, have solved many complex problems in many other fields, ranging from natural language processing (Collobert & Weston 2008) to medicine (Ramesh et al. 2004). Neural networks are also gaining ground in solving astrophysical problems (e.g., Bloom et al. 2012; Domínguez Sánchez et al. 2018), including in the field of exoplanets. Specifically, neural networks have successfully identified exoplanet transits in simulated data (Zucker & Giryes 2018; Pearson et al. 2018), as well as classified planet candidates and false positives detected by Kepler (Ansdell et al. 2018; Shallue & Vanderburg 2018), K2 (Dattilo et al. 2019), TESS (Yu et al. 2019; Osborn et al. 2020), NGTS (Chaushev et al. 2019), and WASP (Schanche et al. 2019).

Our strategy is to use ML to identify and interpret the subtle changes to stellar spectra that are caused by stellar activity. Previously, Davis et al. (2017) used principal component analysis (PCA) to show that photospheric activity signals are clearly distinct from Keplerian RV shifts in simulated data. Beyond distinguishing the two phenomena, we want to be able to predict the RV signals induced by stellar activity such that we can remove these signals and reveal smaller Keplerian signals that were previously hidden. Some preliminary methods are now emerging to separate stellar activity signals using these spectral changes (Collier Cameron et al. 2021; Zhao & Ford 2022). In this work, we attack the problem with ML and train multiple models to predict and remove stellar activity RV signals from observations of the HARPS-N Solar Telescope (Dumusque et al. 2015; Phillips et al. 2016; Collier Cameron et al. 2019).

Our paper is organized as follows. In Section 2, we describe the simulated data and real observations that serve as training sets. In Section 3, we describe how we process and prepare the data to be input to our ML models. In Section 4, we describe how the observations were divided into training, (cross-) validation, and test sets. In Sections 5 and 6, we describe the ML architectures we used, including several different neural networks and our training procedure. In Section 7, we present our results. In Sections 8 and 9, we discuss the implications of these results and conclude.

2. Data

ML methods require data on which to learn. We trained ML models on two data sets: one set of simulated stellar spectra from the SOAP 2.0 software (Dumusque et al. 2014), and one set of real RV observations of the Sun from the HARPS-N solar telescope (Dumusque et al. 2015; Phillips et al. 2016; Collier Cameron et al. 2019).

2.1. SOAP Simulations

We first explored the problem of predicting stellar activity variations with a simulated data set. We produced this data set using SOAP 2.0, a software package that estimates photometric and RV variations induced by sunspots and faculae (Boisse et al. 2012; Dumusque et al. 2014). SOAP simulates a star by dividing the visible hemisphere into a grid and injecting in each created cell an observed solar cross-correlation function (CCF; obtained by cross-correlating a solar spectrum by a binary mask, as is done when reducing HARPS-N high-resolution spectra). The CCFs are shifted to account for the star's rotation velocity at each point on the star's surface. In certain user-specified locations, SOAP 2.0 modifies the local CCFs to mimic active regions, like dark spots or faculae. Finally, SOAP 2.0 sums the CCFs over the entire visible hemisphere of the star, convolves the resulting CCF with a simulated instrumental line profile, and fits the result with a Gaussian function to derive the RV due to the activity signal.

We modified SOAP 2.0 to produce a large set of simulated observations with varying parameters chosen with a Monte Carlo technique. We generated 20,000 random starspot/faculae configurations and used SOAP 2.0 to produce a simulated CCF and activity RV measurement for each configuration. Table 1 lists the range of values that each of the stellar parameters spans.

2.2. HARPS-N Solar Telescope

The HARPS-N solar data set consists of 528 days of solar observations (see Section 3.2 for details) from the HARPS-N Solar Telescope spanning 3 yr (2015 July–2018 July). Commissioned at the Telescopio Nazionale Galileo (TNG), the HARPS-N spectrograph is a vacuum-enclosed cross-dispersed echelle spectrograph that has temperature and pressure stabilization (Cosentino et al. 2012). HARPS-N spans the wavelength range from 383 to 693 nm and has a resolving power of $\lambda/\Delta\lambda = 115{,}000$. During the daytime, a custom-built solar telescope connected to HARPS-N continuously observes the Sun with 5-minute integration times designed to mitigate the short-term variability caused by solar 5-minute p-mode oscillations. The solar telescope and control system are further described by Phillips et al. (2016).

The solar data are reduced using the same HARPS-N Data Reduction System (DRS) as used for nighttime stellar observations. By taking calibration exposures at the end of each day of solar observations, we acquire order-by-order information on the locations of the echelle orders and the

Table 1
Monte Carlo Parameters for Simulated Data

Fixed Parameters		Notes
Grid	300	Grid resolution power (N×N)
nrho	20	Resolution for spot's circumference
Instrument resolving power	115,000	Resolving power of the spectrograph (115,000 for HARPS-N)
Radius Sun	696,000	Radius of the Sun [km] [1]
Radius	1	Simulated stellar radius [RSun]
$P_{\rm rot}$	25.05	Rotation period [day] 25.05 for the Sun [1]
I	90	Stellar inclination angle [degree]; 0°: pole-on (north); 90°: equator-on
Psi	0	Initial phase
$T_{\rm star}$	5778	Effective temperature of star, 5778 for the Sun [1]
$T_{\rm diff\ spot}$	663	Temperature difference between the star effective temperature and the spot [2]
Limb1	0.29	Linear limb-darkening coefficient (can be obtained from [3]); 0.29 for the Sun ([4],[3])
Limb2	0.34	Quadratic limb-darkening coefficient (can be obtained from [3]); 0.34 for the Sun ([4],[3])
Random Parameters		**Notes**
Number of active regions	0–4	Follows a Poisson distribution with most probable value of 1
Active region type	Spot or faculae	Equal probability of being assigned as a spot or a plage
Active region longitude	0 to 360	Random uniform distribution between 0° and 360°
Active region latitude	−90 to 90	Random uniform distribution between −90° and 90°
Active region size	0.0067 to 0.090	[In units of the stellar radius] Log uniform distribution

References. [1] From Emilio et al (2012). [2] From Meunier et al. (2010). [3] From Claret & Bloemen (2011). [4] From Oshagh et al. (2013).

wavelength calibration scale. The instrumental drift is monitored by taking exposures of light passed through a stabilized Fabry–Perot cavity concurrently with the solar exposures. After applying optimal extraction procedures (Horne 1986; Marsh 1989), the data are calibrated in wavelength such that we can obtain a 1D background-subtracted spectrum in each echelle order. Lastly, the data are cross-correlated with a digital mask (Baranne et al. 1996; Pepe et al. 2002) derived from solar observations and corrected for instrumental drift based on the Fabry–Perot exposures. The resulting CCF is used for our input representation to the ML method. Finally, the DRS extracts the RV of each observation by fitting the CCF with a Gaussian function. A Gaussian function is simple and symmetric. Therefore, it is unable to model the small perturbations to CCF shapes induced by stellar activity. So these RVs include both center-of-mass RV shifts and stellar activity signals.

We note that the full 3 yr data set of HARPS-N used in our paper was recently released to the public, as described by Dumusque et al. (2021). The data products released by Dumusque et al. (2021)[27] were reduced with a new extraction pipeline originally built for the newly commissioned ESPRESSO instrument. This new ESPRESSO pipeline analysis resulted in more precise RV measurements than the original HARPS-N DRS reductions (Collier Cameron et al. 2019) and were thus used in this analysis.

3. Preparing the Input Representations

For our ML models, we cannot use the data directly from the telescope or simulations. Instead, we have to preprocess these data products into a uniform format that makes capturing the features in the data easier for ML models. We outline the steps we took to prepare the input representation for the ML models for both the simulated (Section 3.1) and HARPS-N Solar Telescope data (Section 3.2).

[27] https://dace.unige.ch/Sun/?

We design the input representations to pose the problem to our ML models of predicting the activity signal, not the actual center-of-mass velocity of the star. Essentially, we want the ML model to predict *the difference between a Gaussian fit to the CCF and the true velocity shift*. With these predictions in hand, we can easily subtract them from the input RVs to produce a corrected RV time series.

Intuitively, RV signals due to planets cause translational shifts on the CCF but do not result in shape changes of the CCF. In contrast, stellar activity does not result in translational changes and only causes shape changes. Thus, we want the measured RV of our simulated CCF to be shifted to 0 so that the ML methods become primarily sensitive to detecting these shape changes, not translations.

3.1. SOAP Input Representation

Given the CCFs generated by SOAP 2.0 and the measured RV signals (due to the simulated active regions on the star), we apply the following preprocessing steps before sending the CCFs (without any timing information) into our ML models:

1. First, we take the simulated CCF from SOAP 2.0 and shift it by the velocity measured by the SOAP's Gaussian fit to the CCF. We do this by creating an x'-axis that is shifted by $-\Delta$RV, where ΔRV is the stellar activity shift measured by SOAP and interpolating the CCF values from the x'-axis to the original x-axis by using `scipy.interpolate.interp1d()` to perform a cubic interpolation. We tested multiple interpolation methods (linear, nearest, cubic) and found that the cubic method was optimal. We also confirmed that any systematics introduced by interpolation were smaller than the changes due to stellar activity. (*Note:* In the presence of Keplerian shifts this procedure would need to be modified as described in Section 8.4). Shifting the CCF to the velocity measured by the SOAP Gaussian fit purposefully leaves a

small translational shift between the true stellar RV and 0; this shift is exactly what we wish to train the model to predict.
2. We then calculate a differential ΔCCF by subtracting a template CCF generated by SOAP with no active regions (also shifted as described in the previous step). We note that choosing a template CCF with no active regions or another random template CCF does not affect the overall analysis results, but this particular choice of ΔCCF highlights the changes to the shape of the CCF introduced by the active regions.
3. Lastly, we normalize the inputs to the ML methods. In particular, since each input is an array composing the ΔCCF, we calculate the median and standard deviation of each point in the CCF over the entire simulated data set and normalize by subtracting the median and dividing by the standard deviation. This helps the optimization process by making the scale of variations of each input parameter roughly equal.
4. Each input into the ML methods is only this normalized ΔCCF without any timing information. The neural network is then trained to predict the stellar activity signals only based on shape differences between normalized ΔCCFs from different observations.

3.2. HARPS-N Input Representation

Our preprocessing for the HARPS-N data is nearly identical to our preprocessing for the SOAP data, with only two additional steps. Our procedure is as follows:

1. Create a daily average of CCFs. During the day, HARPS-N takes repeated 5-minute-long exposures of the Sun. We average all of these exposures together to obtain a daily averaged CCF and RV measurement. To do this, we follow Collier Cameron et al. (2021). In short, we perform a signal-to-noise ratio weighted average of the CCFs and RVs. We exclude individual observations where the probability of being cloud-free (as calculated by Collier Cameron et al. 2019) is greater than 99% and where the expected differential extinction[28] correction is less than 10 cm s^{-1}.
2. Remove signals from solar system planets. The raw RVs measured by the DRS consist of both the radial motion induced by the solar system planets and stellar activity signals. The planetary signal is dominated by a sinusoidal signal with a semiamplitude of 12 m s^{-1} and a period of ∼13 months, which is the synodic period of Jupiter observed from Earth. To remove the planetary signals, we transform both the RVs and the CCFs from the barycentric to the heliocentric reference frame using the JPL Horizons ephemeris (Giorgini et al. 1996). For the RVs, we simply subtract the Sun's barycentric motion in the direction of the TNG to derive the heliocentric RV. For the CCFs, we perform the shift with the same method as we used to shift the SOAP CCFs, by creating a shifted x'-axis and interpolating back onto the x-axis. The resulting velocities and CCFs contain only stellar activity shifts (and instrumental systematics), in analogy to the simulated CCFs produced by SOAP 2.0.
3. After removing the solar system planet signals, we shift the CCF so that the velocity measured from the HARPS-N DRS Gaussian fit is 0. This step is identical to Step 1 from our SOAP preprocessing in Section 3.1. As a reminder, we take this step because we know that RV signals from planets cause translational shifts, not shape changes. In contrast, stellar activity results in shape changes and no translational shifts. Thus, we shift the CCF and RV to 0 so that the ML methods become primarily sensitive to detecting these shape changes, not translations.
4. We calculate the differential ΔCCF by subtracting a reference HARPS-N observation taken when the Sun had few magnetic features on its visible hemisphere, as determined by visual inspection of images from the Solar Dynamics Observatory (SDO) Helioseismic and Magnetic Imager (HMI). The observation we used as a quiet reference is from 2016 March 29 (see Figure 1). Although choosing a random other template CCF yields the same overall results, choosing a template with few magnetic features on its visible hemisphere allows us to visualize CCF shape changes as a function of activity more clearly. This step is analogous to Step 2 from our SOAP preprocessing in Section 3.1.
5. We normalize the input features in exactly the same way as described in step 3 from our SOAP preprocessing in Section 3.1.
6. Each input into the ML methods is only this normalized ΔCCF without any timing information. The neural network is then trained to predict the stellar activity signals only based on shape differences between normalized ΔCCFs from different observations.

3.3. Visualizing the Inputs

The result of these processing steps is a residual ΔCCF for each observation in our data set. Here we hope to give an intuitive understanding of what these residual ΔCCFs represent and how they convey information about stellar activity. In Figure 1, we illustrate how we subtract a quiet observation (panels (b) and (d)) from the observation of interest (panels (a) and (c)) to calculate the residual CCF (right column). We note that the residual ΔCCFs shown here have not yet been scaled by subtracting the median and dividing by the standard deviation. Figure 2 shows several example ΔCCFs taken on different dates, illustrating how different activity patterns change the ΔCCFs.

In Figures 3 and 4 (animated version available online[29]), we illustrate the differences in shape of the ΔCCFs for different RVs induced by sunspots and faculae over our entire data set. Each ΔCCF is color-coded to show the measured RV induced by stellar activity. Clear patterns emerge in these plots, where similar ΔCCF shapes tend to have similar measured RVs. These are the patterns our ML methods will use to predict

[28] For observations of an extended source such as the Sun, we must consider how the gradient in atmospheric extinction across the star's disk results in asymmetries in the CCF. This phenomenon is often referred to as differential extinction, and this gradient has different effects on blue versus red components (Rušin 1972). As the solar disk rotates, the CCF is rotationally broadened and this systematic signal results in asymmetries of the CCF that can even be mistaken for solar oscillations in some cases (Grec & Fossat 1979; Severnyi et al. 1980).

[29] https://github.com/zdebeurs/exoplanet-ml/tree/master/exoplanet-ml/rv_net

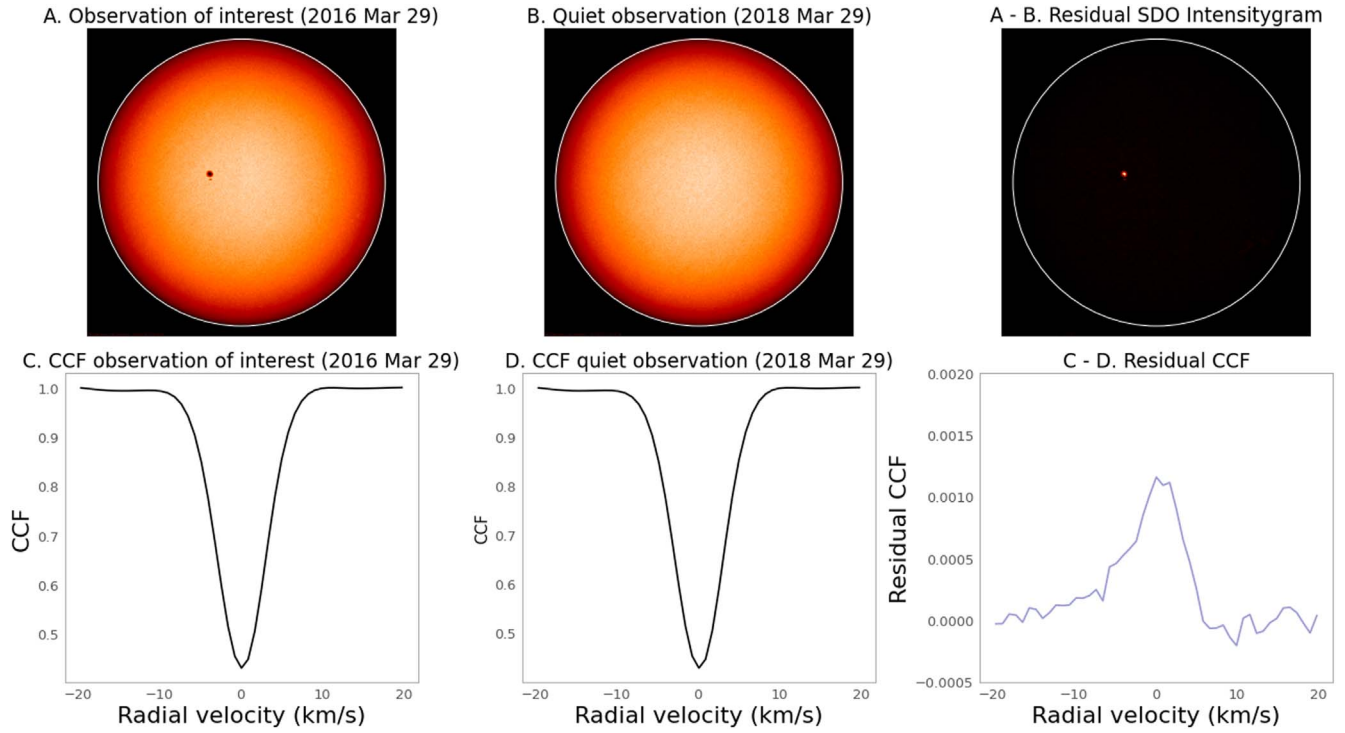

Figure 1. Residual CCF (ΔCCFs) construction. Top row: SDO/HMI intensitygrams; bottom row: CCFs from HARPS-N Solar Telescope observations. Panels (a) and (c) are from a period of relatively large solar activity, while panels (b) and (d) are from observations of the quiet Sun. To highlight differences in shape between CCFs, in the right column we subtract a quiet observation (B/D) from the observation of interest (A/C).

stellar activity signals from the ΔCCFs, which we can use to correct stellar activity.

4. Creating Training, Validation, and Test Sets

In ML, data sets are commonly randomly separated into a training, validation, and testing set. The model is initially fit on the training set, a set of examples used to fit the parameters of the model. Next, the validation set provides a measure of predictive accuracy and model fit. The validation set consists of examples that the model has not seen in the training set and allows for optimization of the architecture and hyperparameters. Lastly, after the model architecture and hyperparameters are finalized, the test set is used as one last objective test of the model accuracy and fit.

In our work, we divided our two data sets (from SOAP 2.0 and the HARPS-N Solar Telescope) into separate groups for training, validation, and testing. Since the simulated SOAP 2.0 training set is sufficiently large, we divided the data set into training (80% of the data), validation (10%), and testing sets (10%).

However, our smaller data set from the HARPS-N Solar Telescope required a different approach. Instead, we created a cross-validation set (80% of the data set), a validation set (10%; which was trained on the full cross-validation set), and a testing set (10%). We then use k-fold cross-validation to provide as many tests with the available data to optimize the architecture and hyperparameters. In k-fold cross-validation, the cross-validation data set is divided into k subsets. For each round of validation, one of the k subsets is treated as a holdout sample, and the model is trained on the other $k-1$ subsets. In this way, k-fold cross-validation significantly decreases bias (i.e., overestimate of model performance), as we are using the majority of the data for fitting. The exact choice of k represents a trade-off between bias and variance (i.e., performance changes significantly based on data chosen to train the model). A higher value of k may decrease the variance but can also increase the bias. We divided our cross-validation set into 10 folds, as $k=10$ has been shown empirically to yield test error estimates that minimize both bias and variance (James et al. 2013). We optimized the architectures by assessing the performance on both the k-fold cross-validation set and the held-out 10% validation set (which we trained on the full cross-validation set). To estimate how well the final model generalizes, we evaluated our best models' performances on the test set (10% of the data set), which consists only of examples that were not used in the cross-validation or validation.

5. Neural Network Architectures

To learn to predict stellar activity RV signals from differences in shape of the ΔCCF, we trained three different ML architectures: a linear regression model, a fully connected neural network (FC NN), and a convolutional neural network (CNN).

5.1. Linear Architecture

The most basic model we trained is a linear regression model, which is equivalent to a zero hidden layer neural network. As illustrated in Figure 5(a), the linear architecture takes a vector $\boldsymbol{x} \in \mathbb{R}^n$ as an input, where $\mathbb{R}^n$ is a real coordinate space of dimension n (where $n=$ dimension of the input data). The input vector $\boldsymbol{x}$ represents the rescaled CCF residuals. After taking the vector $\boldsymbol{x}$, the linear architecture predicts the value of a scalar y as the output, which is the predicted stellar activity

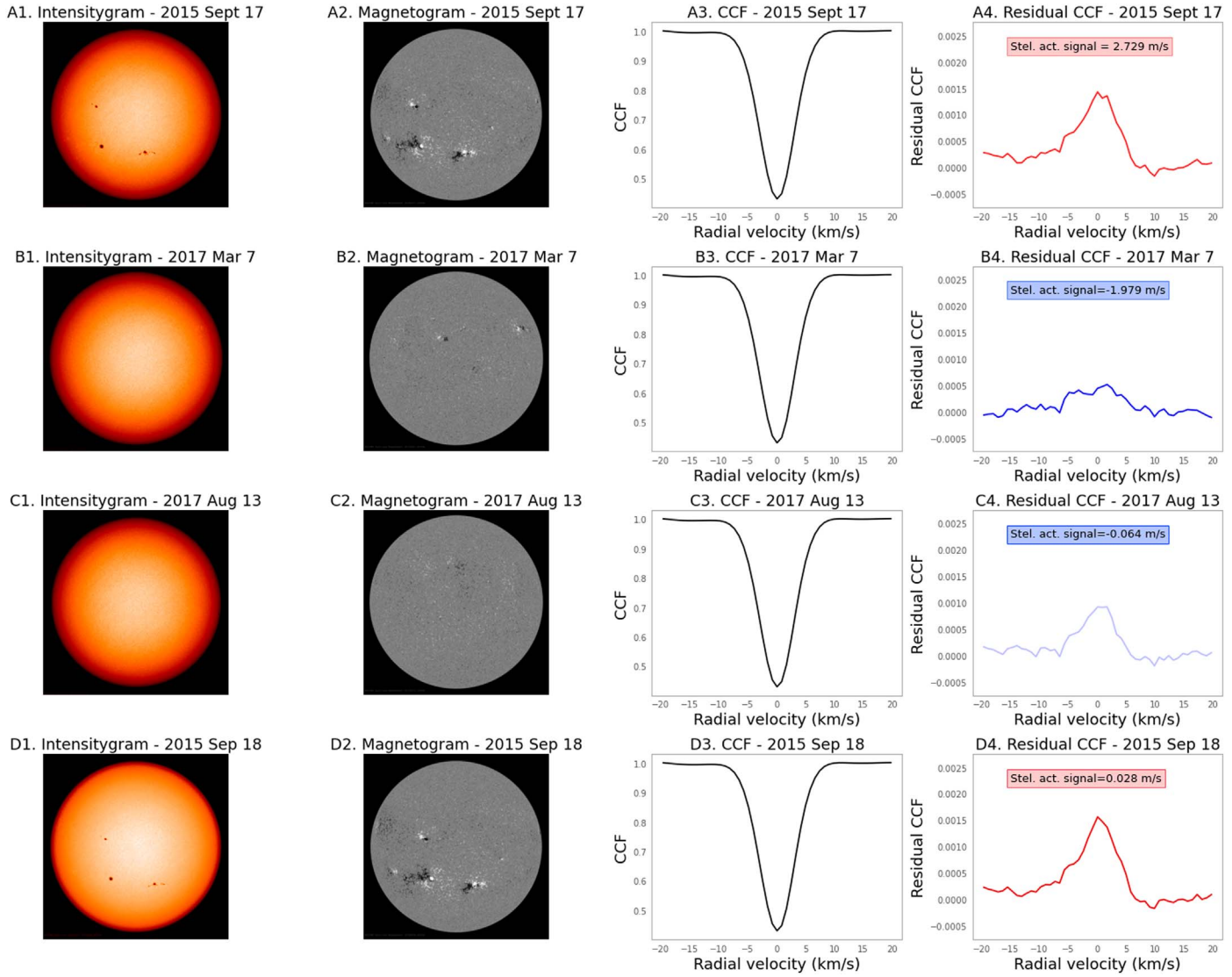

Figure 2. Comparison of residual CCF (ΔCCFs) and SDO observations. The first column shows SDO/HMI intensitygrams, the second column shows SDO/HMI magnetograms, the third column shows CCFs from HARPS-N Solar Telescope observations, and the fourth column shows residual CCFs from HARPS-N where the corresponding RVs are indicated by their color (red = redshifted; blue = blueshifted). Each row is from a different day of observations and subsequently displays distinct surface inhomogeneities that correspond to the residual CCF line shape (panels (a4), (b4), (c4), (d4)).

RVs. Then, the predicted value of y will be

$$\hat{y} = \boldsymbol{w}^\top \boldsymbol{x} + b, \tag{2}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ are the weights that determine how each feature affects the prediction, and b is a bias term. $\boldsymbol{w}$ and b are the trainable parameters of the model.

Although a convenient choice due to its simplicity, a linear architecture makes the strong assumption that a linear relationship exists between the points in the CCF and the RV activity signal. For simulated ideal data this assumption may not pose a problem, but for more complex real data this assumption can break down (see Section 7).

5.2. Fully Connected Architecture

Figure 5(b) shows an FC NN (also sometimes referred to as a multilayer perceptron or feed-forward neural network). Each layer consists of scalar-valued units called *neurons* where the

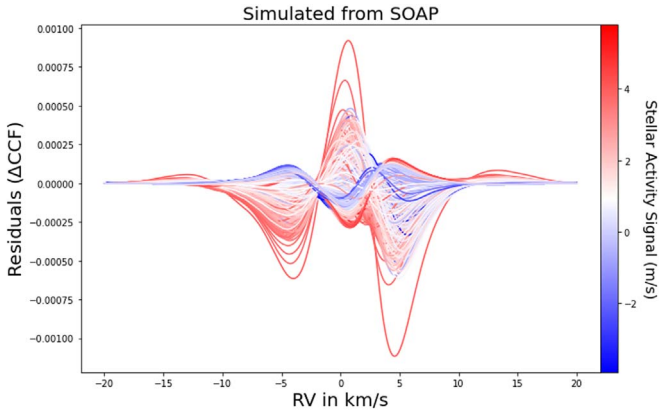

Figure 3. Simulated ΔCCFs from our Monte Carlo training set generated by SOAP 2.0 before normalization. The different shapes are the result of different activity configurations on the star. The measured RV activity signal for each ΔCCF is indicated by its color (red = redshifted; blue = blueshifted).

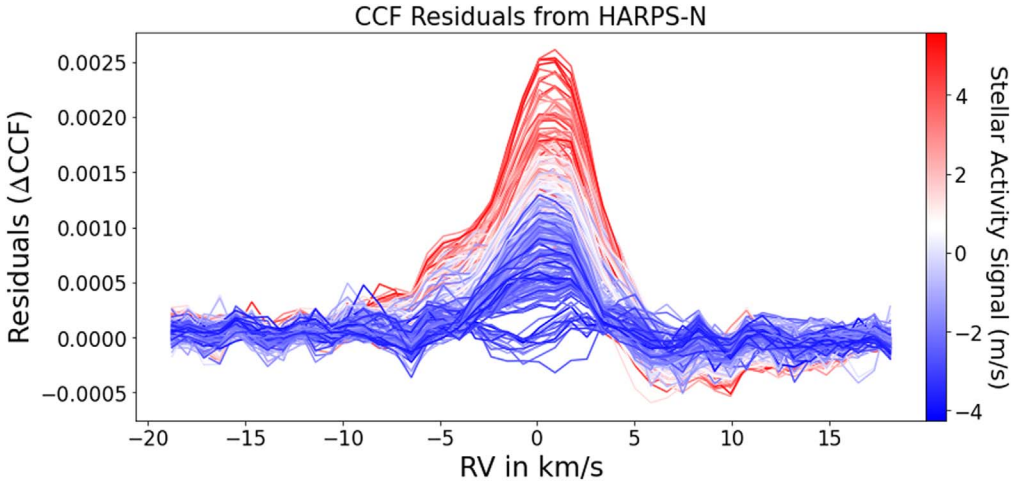

Figure 4. HARPS-N ΔCCFs before normalization. Residual CCFs (ΔCCFs) are computed by subtracting the mean CCF, highlighting differences in features between CCFs. For training the model, ΔCCF is the input and the RV from stellar activity is the output. The RV is indicated by its color (red = redshifted; blue = blueshifted). An animated version of this figure is available online (https://github.com/zdebeurs/exoplanet-ml/tree/master/exoplanet-ml/rv_net) and has a duration of 53 s. In this animated version, the CCF residuals are layered on top of each other in the order with which they were observed. At the end of the animation, we reach the same final frame as is displayed in this static version of the figure.

(An animation of this figure is available.)

outputs from one layer of neurons are the inputs for the next layer. The function that produces outputs based on the inputs is called an activation function. This activation function ϕ produces a new representation of $\boldsymbol{w}^\top \boldsymbol{x} + b$ through a nonlinear transformation; its output $\phi(\boldsymbol{w}^\top \boldsymbol{x} + b)$ can be thought of as a set of features describing $\boldsymbol{x}$.

The values of the first and last layers comprise the inputs and outputs of the network. However, the values (activations) of the intermediate layers are not directly observed and are therefore referred to as hidden layers. The hidden and output layer activations are defined by

$$\boldsymbol{a}_n = \phi(\boldsymbol{W}_n \boldsymbol{a}_{n-1} + \boldsymbol{b}_n), \tag{3}$$

where n is the layer number, $\boldsymbol{a}_n$ is an i_n-long vector of activations in layer n, $\boldsymbol{W}_n$ is an $i_n \times i_{n-1}$ matrix of learned weights, $\boldsymbol{b}_n$ is an i_n-long vector of learned bias parameters, and ϕ specifies an activation function that computes the hidden layer values.

In FC NNs, the most common activation function is the rectified linear unit (ReLu; Jarrett et al. 2009; Nair & Hinton 2010; Glorot et al. 2011), defined by the element-wise activation $\phi(x) = \max\{0, x\}$. The element-wise activation $\phi(x)$ applies a nonlinear transformation where values of $x < 0$ are mapped to zero and others remain equal to x. In this way, rectified linear units are nearly linear and preserve many of the properties that make linear models easy to optimize with gradient-based methods (Goodfellow et al. 2016). In our neural network layers, we used ReLu as our activation functions.

5.3. Convolutional Architecture

FC NNs use matrix multiplication where the matrix has a separate parameter for the interaction between each input unit and every output unit. Every input interacts with every output, causing FC NNs to be "agnostic" to spatial structure present in the data. For example, they treat adjacent data points exactly the same as data points that are far apart, making it inefficient to learn local features (e.g., edges and shapes) that may appear in different locations. In contrast, CNNs have only local (sparse) interactions, which force them to learn local features across the entire input and exploit the spatial structure (Figure 5(c)). Rather than learning local features for every single input−output interaction, they only have to learn these features once. This reduces the number of parameters that the model needs to learn and decreases the number of computational operations required to predict the output.

The 1D convolutional layers depicted in Figure 5(c) require an input stack of K vectors $\boldsymbol{a}_{n-1}^{(k)}(k = 1, 2, \ldots K)$ of length i_{n-1}. Each convolutional layer outputs a stack of L vectors $\boldsymbol{a}_n^{(l)}(l = 1, 2, \ldots L)$ of length i_n. The transformation that takes the stack of K input vectors and produces the lth output vector is called a feature map defined by the operation

$$\boldsymbol{a}_n^{(l)} = \phi\left(\sum_{k=1}^{K} \boldsymbol{w}_n^{(k,l)} * \boldsymbol{a}_{n-1}^{(k)} + \boldsymbol{b}_n^{(l)}\right), \tag{4}$$

where $*$ is the discrete cross-correlation function (often referred to as a "convolution" in the ML literature), $\boldsymbol{w}_n^{(k,l)}$ is an m_n-long vector of learned parameters called the convolution kernel or filter, $\boldsymbol{b}_n$ is an i_n-long vector of learned bias parameters, and ϕ specifies the activation function. By making the kernel size small ($m_n \sim 3$–7), the feature map becomes sensitive to local features along its input.

Intuitively, we would expect the CNN to perform best because we are trying to identify shapes and relationships between adjacent points in our ΔCCF to produce the final stellar activity output.

Commonly, CNNs use both convolutional layers and pooling layers. Pooling layers typically replace the output of the neural net at a certain location with a summary metric (e.g., maximum) of the nearby outputs. Pooling generally helps to make the representation translationally invariant. However, in early iterations of our models, we found that adding pooling layers negatively affected our performance, so we do not use any pooling layers in our CNN models.

After the convolutional layers in our CNN model, we finally include one (or more) fully connected layer(s). The last fully connected layer then produces the final output.

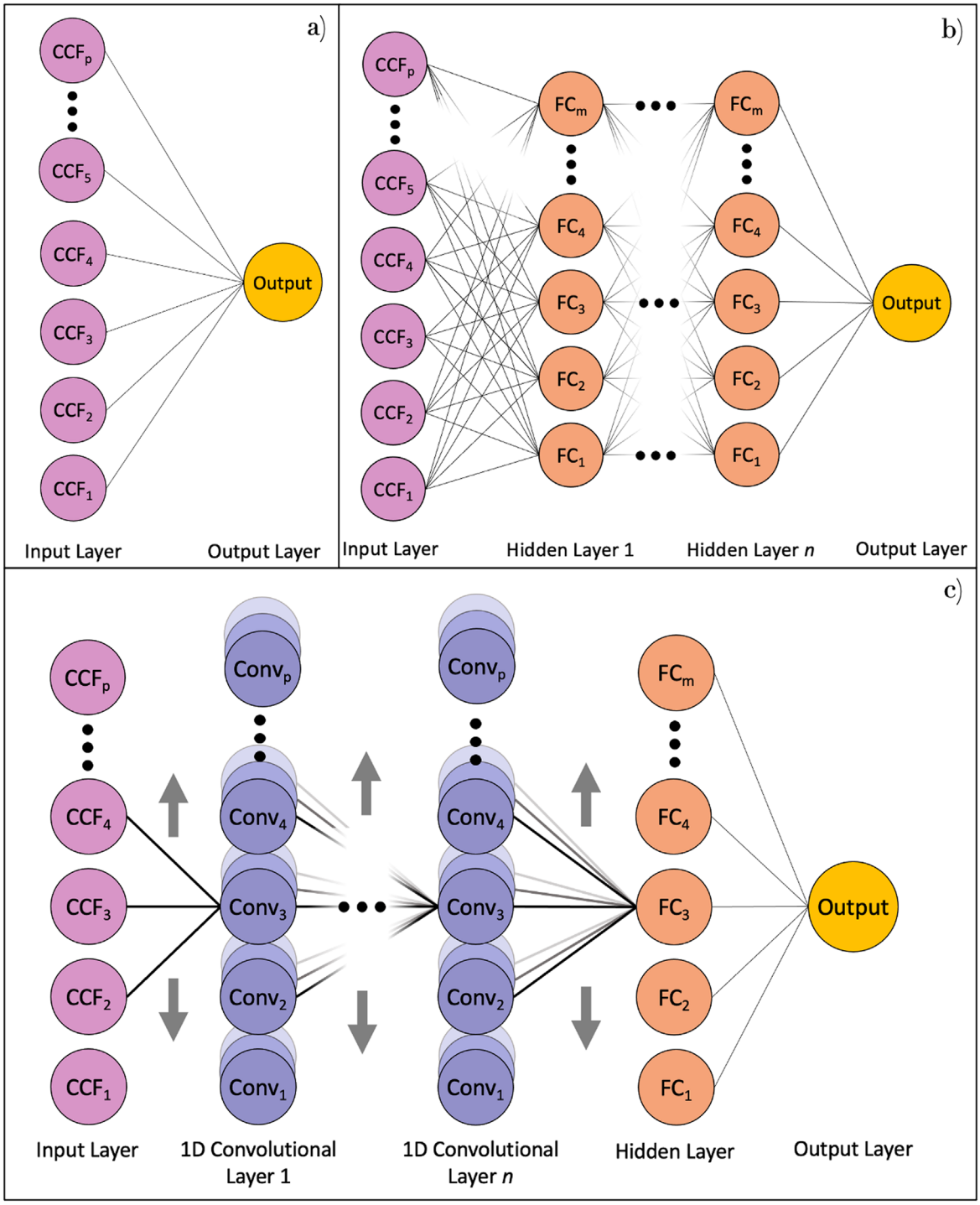

Figure 5. Three ML architectures visualized. In all three architectures, the vector of all parameters in the model $\boldsymbol{p}$ matches the input dimensions, and $\boldsymbol{p} = 401$ for SOAP 2.0 data and $\boldsymbol{p} = 46$ for HARPS-N data. (a) Linear architecture. This architecture is equivalent to a linear regression model and has zero hidden layers. (b) FC NN. Every connection corresponds to a multiplicative weight parameter learned by the model. The CCF inputs are fed into the first layer, the hidden layers represent a hierarchy of learned features, and the output layer generates predictions. For the final FC model, the number of dense units $m = 100$ for SOAP 2.0 data and $m = 200$ HARPS-N data. The number of hidden layers $n = 1$ for SOAP 2.0 data and $n = 8$ for HARPS-N data. (c) CNN. The convolutional layer takes the discrete cross-correlation of the vector in its input layer with the kernel vectors that the model learns. The sparse connections compared to the FC model allow CNNs to learn local features and exploit spatial structure in the data. The stack of nodes going into the page within each convolutional layer represents the different filters. For the final CNN model, the number of 1D convolutional layers $n = 6$ and $m = 1000$ for the SOAP 2.0 data and $n = 1$ and $m = 500$ for the HARPS-N data. In all three models, the ellipses (… and ⋮) represent additional nodes and layers that we have omitted for visual clarity.

6. Neural Network Training

6.1. Training Algorithm

Neural networks are trained to minimize a loss function, which quantifies the difference between the predictions and the true labels in the training set. For regression problems, the mean squared error (MSE) is the standard loss function and is defined by the equation

$$L(\hat{y}_i, y_i|\boldsymbol{p}) = \mathrm{MSE} = \frac{1}{M}\sum_{i=1}^{M}(\hat{y}_i - y_i)^2, \quad (5)$$

where $\boldsymbol{p}$ is the vector of all parameters in the model, $y_1, y_2,...,y_M$ are the true labels of all examples in the training set, and $\hat{y}_1, \hat{y}_2, ..., \hat{y}_M$ are the model's predicted outputs given $\boldsymbol{p}$. The vector $\boldsymbol{p}$ consists of all the free parameters of the given architecture to be learned during training. For the linear architecture, this is simply the vector of weights $\boldsymbol{\omega}^\top$ and the bias term b. For the FC NN, this corresponds to the elements of all weight matrices $\boldsymbol{W}_n$ and the bias vectors $\boldsymbol{b}_n$. For the CNN, the parameters are the elements of all convolutional kernels (or filters) $\boldsymbol{w}_n^{(k,l)}$, bias vectors $\boldsymbol{b}_n$, and the weight matrix and bias matrix of the final fully connected layer.

The most popular neural network training algorithms use gradient descent to find the parameters $\boldsymbol{p}$ that minimize the loss function. These algorithms calculate how the parameters of the model can be changed to decrease the loss function by computing the gradient of the loss function with respect to the parameters. The model's parameters start at random values and are iteratively updated by descending along the gradient until the desirable minimum of the loss function is achieved. The step size is set by the learning rate, which is a hyperparameter that requires tuning during (cross-)validation to achieve optimal performance.

Computing the exact gradient of the loss function is unnecessary and computationally inefficient, as it requires iterating over the entire training set. Rather, each gradient step approximates the true gradient by taking a random batch (i.e., subset) of the training set. The algorithm is then called a stochastic gradient descent (SGD) algorithm. The batch size is typically determined by the available computational resources. We kept the batch size constant at 300—Shallue et al. (2019) demonstrated that the performance should be the same at any batch size, provided that the other hyperparameters are well tuned.

In practice, neural networks are often trained using variants of the basic SGD algorithm. For our FC and CNN neural networks, we used SGD with momentum (Polyak 1964), with the momentum parameter fixed at 0.9.

6.2. Overfitting and Regularization

The fundamental challenge in ML is that algorithms have to perform well on *novel, previously unseen* inputs—not just the data on which the model was trained (Goodfellow et al. 2016). The ability of a model to perform well on previously unseen inputs is called generalization and can be estimated from its performance on a test set composed of data not used during training.

Overall, we can summarize the performance of an ML method by its ability to minimize both (1) the training error and (2) the gap between the training and test error. These two abilities correspond to the problems of underfitting and overfitting on training data, respectively. Techniques that seek to reduce overfitting, and therefore improve generalization, are known as regularization methods.

A common approach to regularization is to limit the complexity of the model by constraining the values of its parameters $\boldsymbol{p}$. Two such methods are L_2 regularization and weight decay regularization. L_2 regularization adds a penalty term to the loss function proportional to the squared L_2 norm of the parameter vector $\boldsymbol{p}$,

$$L_{\rm reg} = \mathrm{MSE} + \frac{\alpha}{2}||\boldsymbol{p}||_2^2, \tag{6}$$

where α is the strength of the regularization[30] and the L_2 norm $||\boldsymbol{p}||_2$ is defined as

$$||\boldsymbol{p}||_2 = \left(\sum_{i=1}^{N}|\boldsymbol{p}_i|^2\right)^{1/2} = \sqrt{\boldsymbol{p}_1^2 + \boldsymbol{p}_2^2 + ... + \boldsymbol{p}_N^2}\,. \tag{7}$$

Weight decay regularization shrinks the parameter vector by a constant factor on each iteration,

$$\boldsymbol{p}_{i+1} = (1 - \alpha)\boldsymbol{p}_i + (\Delta\boldsymbol{p}_i)_{\rm opt}, \tag{8}$$

where $(\Delta\boldsymbol{p}_i)_{\rm opt}$ is the change to the parameter vector computed by the optimization algorithm at iteration i. Both of these techniques encourage smaller values of the parameters of $\boldsymbol{p}$. In fact, L_2 regularization and weight decay are equivalent when using the basic SGD training algorithm. However, they are not equivalent for all variants of SGD, in particular for SGD with momentum, which we used for our FC and CNN neural networks. In those cases, we chose weight decay because it empirically performs better for neural networks (Loshchilov & Hutter 2017).

Larger values of the weight decay parameter α discourage overfitting but can also cause underfitting. We optimized α by exploring the parameter space during validation for the SOAP 2.0 simulated data and during cross-validation for the HARPS-N data for both our FC and CNN models.

6.3. Implementation and Training Procedure

For each of the model architectures, we tune the hyperparameters unique to the architecture. In contrast with parameters that are learned during the training process, hyperparameters are parameters whose value is used to control the learning process. The hyperparameters can significantly affect model performance.

6.3.1. Linear Model Implementation

We implemented our linear model in `scikit-learn`, an open-source library for ML in Python (Pedregosa et al. 2011). Specifically, we performed a ridge regression that is equivalent to a linear regression with L2 regularization.

We performed random searches across the parameter space for α that determines the regularization strength. The values of α spanned 0–800 and significantly affect the model performance as illustrated in Figure 7. In Table 2, the best model's performance across the validation and cross-validation sets is listed. Each model was trained on a single CPU, and the training time took ∼1 minute for the 10 runs over which we average the predictions to compute our final stellar activity predictions.

For our linear model, we extracted the vector weights and plot these alongside the input CCFs in Figure 8. In the bottom panel of Figure 8, neighboring weights appear correlated and the largest weights are concentrated in the center, which is

[30] Sometimes, this regularization strength parameter is referred to as λ in the literature. We choose α here for consistency with our description of our linear model regularization procedure in Section 6.3.1.

Table 2
HARPS-N Validation and Cross-Validation

Linear Model Best Hyperparameters		
Linear Hyperparameters		Best Model
α		3.609
Validation Set Results		
Scatter Metric	Raw Data	Corrected Data Using Best Model
σ_{SD} (m s^{-1})	1.923	1.346
$\sigma_{k\text{-MAD}}$ (m s^{-1})	1.751	1.250
$\sigma_{Percentile}$ (m s^{-1})	2.083	1.430
Cross-Validation Set Results		
Scatter Metric	Raw Data	Corrected Data Using Best Model
σ_{SD} (m s^{-1})	1.828	1.085
$\sigma_{k\text{-MAD}}$ (m s^{-1})	1.744	1.085
$\sigma_{Percentile}$ (m s^{-1})	1.745	1.097

Note. To find the best linear model architecture for the HARPS-N Observations, we performed random searches across the parameter space. The best linear model configuration and its corresponding reductions in RV scatter are listed here. This model was chosen as the final model based on its better performance across the validation and cross-validation sets. This model was thus used on the test set.

expected from visually examining the input CCFs in the top panel of Figure 8.

6.3.2. FC NN and CNN Model Implementation

We implemented our FC NN and CNN models in TensorFlow, an open-source software library for ML algorithms (Abadi et al. 2016).

We used SGD with momentum to minimize the loss function over the training set. We performed random searches across the parameter space for the learning rate, weight decay, kernel size, filters, number of layers, and number of epochs over the (cross-) validation set as listed in Table 3. Each model was trained on a single CPU, and the training time ranged from 5 to 20 minutes depending on the complexity of the model. Across the 10 runs whose results we average, our best models took ∼5 and 8 minutes to train for the fully connected and convolutional architectures, respectively. In Tables 4 and 5, the three best-performing model hyperparameters are listed for the FC NN and CNN architectures, respectively. Further, the final model architectures used for both the simulated SOAP 2.0 and HARPS-N Solar Data are listed in Figures 5 and 6.

6.4. Model Ensembling by Averaging

After optimizing our hyperparameters for a specific architecture, we train 10 independent copies with different random parameter initializations. We then average the 10 outputs for all predictions to compute our results in Section 7. This method of model averaging often improves performance. Across the input space, different versions of the same configuration may perform better or worse, and this process averages out this difference in performance. This is especially important when the training set is small and we are at higher risk for overfitting. In addition, model averaging reduces the

Table 3
Random Search Hyperparameter Space

Linear Hyperparameters	Hyperparameter Distribution	Random Search Space
α	Logarithmic	0–1000
FC NN Hyperparameters	Hyperparameter Distribution	Random Search Space
Learning rate	10^{-x} (x is uniform)	$10^{-4} - 10^{5}$
No. dense units	Discrete	50, 100, 200, 500, 1000, 2000
No. dense layers	Discrete	1, 2, 4, 8, 12, 16, 32
Weight decay	Logarithmic	0.00001 − 0.1
Epochs	Discrete	25, 30, 35, 40, 45, 50, 55, 60
CNN Hyperparameters	Hyperparameter Distribution	Random Search Space
Learning rate	10^{-x} (x is uniform)	$10^{-4} - 10^{5}$
Conv kernel size	Discrete	3, 5, 7
No. conv filters	Discrete	8, 16, 32
No. conv layers	Discrete	2,4,6
No. dense units	Discrete	100, 200, 500, 1000
No. dense layers	Discrete	1, 2, 4, 6, 8
Weight decay	Logarithmic	0.0005 − 0.05
Epochs	Discrete	50, 55, 65, 70, 80, 90, 100

Note. To find the best hyperparameters across model architectures for both the SOAP 2.0 and HARPS-N observations, we performed random searches across the parameter space. The ranges of the parameter space explored are the same for both data sets. Convolutional layer parameters are denoted conv ⟨parameter⟩. Fully connected layer parameters are denoted dense ⟨parameter⟩. For FC NN and CNN models, we kept momentum = 0.9, which is a generally accepted value.

Table 4
HARPS-N Validation and Cross-Validation

FC NN Best Hyperparameters			
FC NN Hyperparameters	Model 1	Model 2	Model 3
Learning rate	0.00161	0.00244	0.00135
No. dense units	200	200	1000
No. dense layers	4	8	4
Weight decay	0.000100	0.000577	0.00010
Validation Set Results			
Scatter Metric	Raw Data	Corrected Data Using	
		Model 1	Model 2	Model 3
σ_{SD} (m s^{-1})	1.923	1.382	1.377	1.417
$\sigma_{k\text{-MAD}}$ (m s^{-1})	1.751	1.353	1.336	1.333
$\sigma_{Percentile}$ (m s^{-1})	2.083	1.425	1.335	1.447
Cross-validation Set Results				
Scatter Metric	Raw Data	Corrected Data Using		
		Model 1	Model 2	Model 3
σ_{SD} (m s^{-1})	1.828	1.085	1.089	1.101
$\sigma_{k\text{-MAD}}$ (m s^{-1})	1.744	1.039	1.036	1.044
$\sigma_{Percentile}$ (m s^{-1})	1.745	1.064	1.038	1.101

Note. To find the best FC NN model architecture for the HARPS-N Observations, we performed random searches across the parameter space. The three best FC NN model configurations and their corresponding reductions in RV scatter are listed here. Although Models 1 and 2 perform similarly, Model 2 was chosen as the final model based on its marginally better performance across the validation set. Model 2 was thus used on the test set.

Table 5
HARPS-N Validation and Cross-Validation

CNN Best Hyperparameters			
CNN Hyperparameters	Model 1	Model 2	Model 3
Learning rate	0.016463	0.011615	0.0038415
Conv kernel size	11	11	9
No. conv filters	8	8	64
No. conv layers	1	1	1
No. dense units	100	2000	500
No. dense layers	1	1	4
Weight decay	0.033076	0.004515	0.000012362
No. epochs	75	100	80

Validation Set Results				
Scatter	Raw	Corrected Data Using		
Metric	Data	Model 1	Model 2	Model 3
σ_{SD} (m s^{-1})	1.923	1.431	1.392	1.399
$\sigma_{k\text{-}MAD}$ (m s^{-1})	1.751	1.367	1.548	1.383
$\sigma_{Percentile}$ (m s^{-1})	2.083	1.376	1.354	1.267

Cross-validation Set Results				
Scatter	Raw	Corrected Data Using		
Metric	Data	Model 1	Model 2	Model 3
σ_{SD} (m s^{-1})	1.828	1.118	1.089	1.083
$\sigma_{k\text{-}MAD}$ (m s^{-1})	1.744	1.047	1.129	1.068
$\sigma_{Percentile}$(m s^{-1})	1.745	1.016	1.123	1.027

Note. To find the best CNN model architecture for the HARPS-N observations, we performed random searches across the parameter space. The three best CNN model configurations and their corresponding reductions in RV scatter are listed here. Model 3 was chosen as the best final model and used on the test set. Convolutional layer parameters are denoted conv⟨parameter⟩.

variance arising from randomness in parameter initialization and data ordering during training, making it easier to compare different architectures.

7. Results

Here we report the results of our ML activity predictions. First, we discuss the metrics we used to evaluate the performance, and then we summarize how the different models performed on each data set.

7.1. Performance Metrics: σ_{SD}, $\sigma_{k.MAD}$, and $\sigma_{Percentile}$

After we finish optimizing our model parameters and hyperparameters on the training data, we evaluate our models' performance by characterizing the scatter of the "corrected" RVs, which we define as

$$RV_{corrected} = RV_{raw} - RV_{predicted}, \quad (9)$$

where RV_{raw} are the input RVs[31] without any activity corrections, and $RV_{predicted}$ are the predictions from our ML models. We introduce three metrics to characterize the scatter in the corrected RVs: σ_{SD}, $\sigma_{k\text{-}MAD}$, and $\sigma_{Percentile}$.

[31] These are the RVs in the heliocentric frame as calculated in step 2 of Section 3.2.

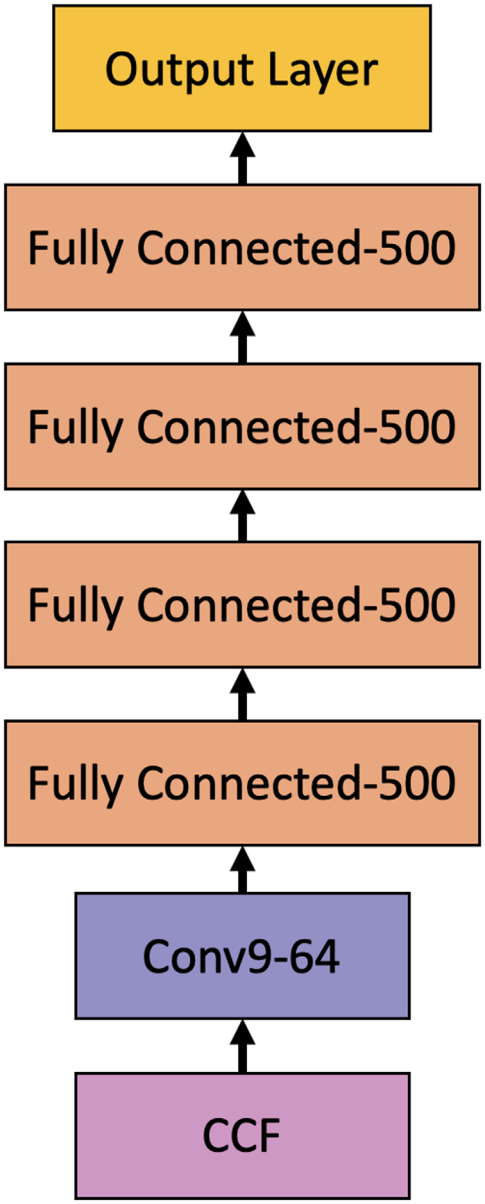

Figure 6. The architecture of our best-performing neural network model. Convolutional layers are denoted conv ⟨kernel size⟩ – ⟨number of feature maps⟩, and fully connected layers are denoted Fully Connected – ⟨number of units⟩.

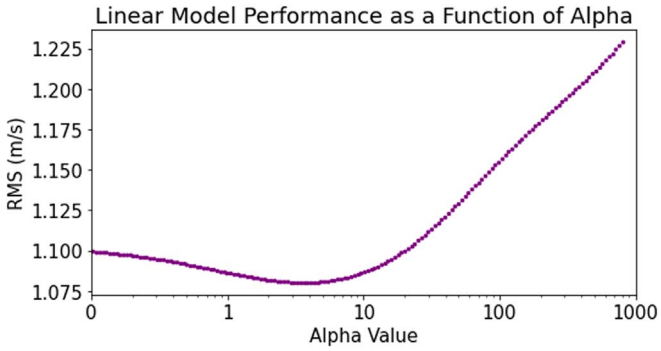

Figure 7. Linear model hyperparameter α optimization. The rms error for the cross-validation set as a function of the value of α is listed. The value of α with the lowest corresponding rms is listed in Table 2.

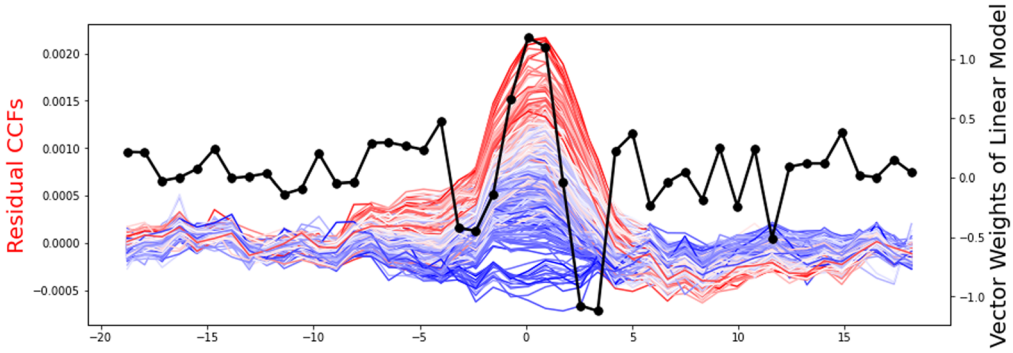

Figure 8. Linear model input CCFs (before normalization) and corresponding vector weights. The residual CCFs that were included in training set for our best linear model ($\alpha = 3.609$) are plotted in red, white, and blue where the color corresponds to the RV signal (red = redshifted; blue = blueshifted). The model input CCFs are normalized, but we plot them before normalization here for visualization purposes. In black, the learned vector weights are plotted. The weights show correlations with their neighbors (reflecting the ≈3 km s^{-1} HARPS-N instrumental profile) and show that most information comes from points within 5 km s^{-1} of the line center.

The first metric we calculate is the standard deviation of the corrected RVs, $\sigma_{\rm SD}$. The standard deviation is given by

$$\sigma_{\rm SD} = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(\mathrm{RV}_{\mathrm{corrected},i} - \mathrm{mean}(\mathrm{RV}_{\mathrm{corrected}}))^2}\,, \quad (10)$$

where M is the number of corrected RV observations. For well-behaved data sets like the SOAP 2.0 simulated data, using just the $\sigma_{\rm SD}$ metric is sufficient to characterize the scatter. On the other hand, the HARPS-N data set is more complex, so we introduce two new metrics in addition to $\sigma_{\rm SD}$: $\sigma_{k\cdot\rm MAD}$, based on the median absolute deviation (MAD), and $\sigma_{\rm Percentile}$. We introduce these additional two metrics for HARPS-N data because our data do not follow a normal distribution perfectly, and the presence of a few outlier data points in the HARPS-N data set made it difficult to assess the model performance using only the $\sigma_{\rm SD}$ metric. These new metrics are less sensitive to outliers than $\sigma_{\rm SD}$. The MAD metric is defined for a set of corrected RVs as

$$\mathrm{MAD} = \mathrm{Median}(|\mathrm{RV}_{\mathrm{corrected},i} - \mathrm{Median}(\mathrm{RV}_{\mathrm{corrected}})|). \quad (11)$$

In other words, MAD takes the median of the data's absolute deviations around the data's median. To ease comparison with our other metrics like the standard deviation, we scale MAD by a factor k such that

$$\sigma_{k\cdot\rm MAD} = k \cdot \mathrm{MAD}, \quad (12)$$

where k depends on the type of distribution. We approximate our distribution as normal and thus k ≈ 1.4826.

We call our final scatter metric $\sigma_{\rm Percentile}$. We calculate this metric by computing

$$\sigma_{\rm Percentile} = \frac{1}{2}(\mathrm{RV}_{84\mathrm{th}\%} - \mathrm{RV}_{16\mathrm{th}\%}), \quad (13)$$

where $\mathrm{RV}_{84\mathrm{th}\%}$ is the 84th percentile of the corrected RVs, and $\mathrm{RV}_{16\mathrm{th}\%}$ is the 16th percentile of the corrected RVs. For a normal distribution, this is equivalent to calculating the standard deviation. However, our distribution of stellar activity signals is not perfectly normal and skewed by some of the outliers. Thus, computing $\sigma_{\rm Percentile}$ serves as a proxy for the standard deviation that is less sensitive to outliers.

7.2. SOAP 2.0 Results

For the simulated data using SOAP 2.0, our best-performing models were the linear model and FC architecture, which reduce the RV scatter, $\sigma_{\rm SD}$, from 82.0 to 3.7 and 3.1 cm s^{-1} across the test set, respectively. These results are summarized in Figure 9 and Table 6. Thus, for the idealized case of simulated data, we can predict the stellar activity signal nearly exactly based on the shape changes in the normalized ΔCCF.

7.3. HARPS-N Results

Our results across all three architectures are summarized in Figure 9 and Table 7. Our best-performing models were the FC NN and CNN (Figure 6), which reduced the RV scatter, $\sigma_{\rm Percentile}$, from 175.3 to 106.2 and 103.9 cm s^{-1}, respectively, across the full data set. This remaining scatter is likely dominated by instrumental noise, not photon statistics. Closely following the performance of the FC NN and CNN, our linear model reached a minimum scatter of 108 cm s^{-1} across the full data set. From Table 7, we note that the overall reduction in scatter varies slightly across scatter metrics and methods. Overall, these results suggest that all three model architectures match the structure of the HARPS-N solar data well, but the CNN model is potentially marginally more suitable. The raw RVs, CNN predicted RVs, and CNN stellar-activity-corrected RVs are plotted over time in Figure 10.

7.3.1. Periodogram: Activity Signal Peaks Disappear

We investigated the behavior of the raw and corrected HARPS-N RVs in the Fourier domain to see which signals are being removed to achieve this reduction in rms scatter. Figure 11 shows the Lomb−Scargle periodograms (Lomb 1976; Scargle 1982) of the RVs before and after applying the activity correction. To implement a generalized Lomb−Scargle periodogram, we used the periodogram functions in `astropy.timeseries` (VanderPlas et al. 2012; VanderPlas & Ivezić 2015), where the periodograms are normalized according to the formalism in Zechmeister & Kürster (2009).

In the top panel of Figure 11, the peaks at ∼25 and ∼12 days correspond to the Sun's rotation period at the equator and half the rotation period, respectively. The signal beyond >900 days is the long-term magnetic cycle. Lastly, the peaks at ∼1 day correspond to aliases from both the rotation period signals and the long-term magnetic cycle. After applying the corrections

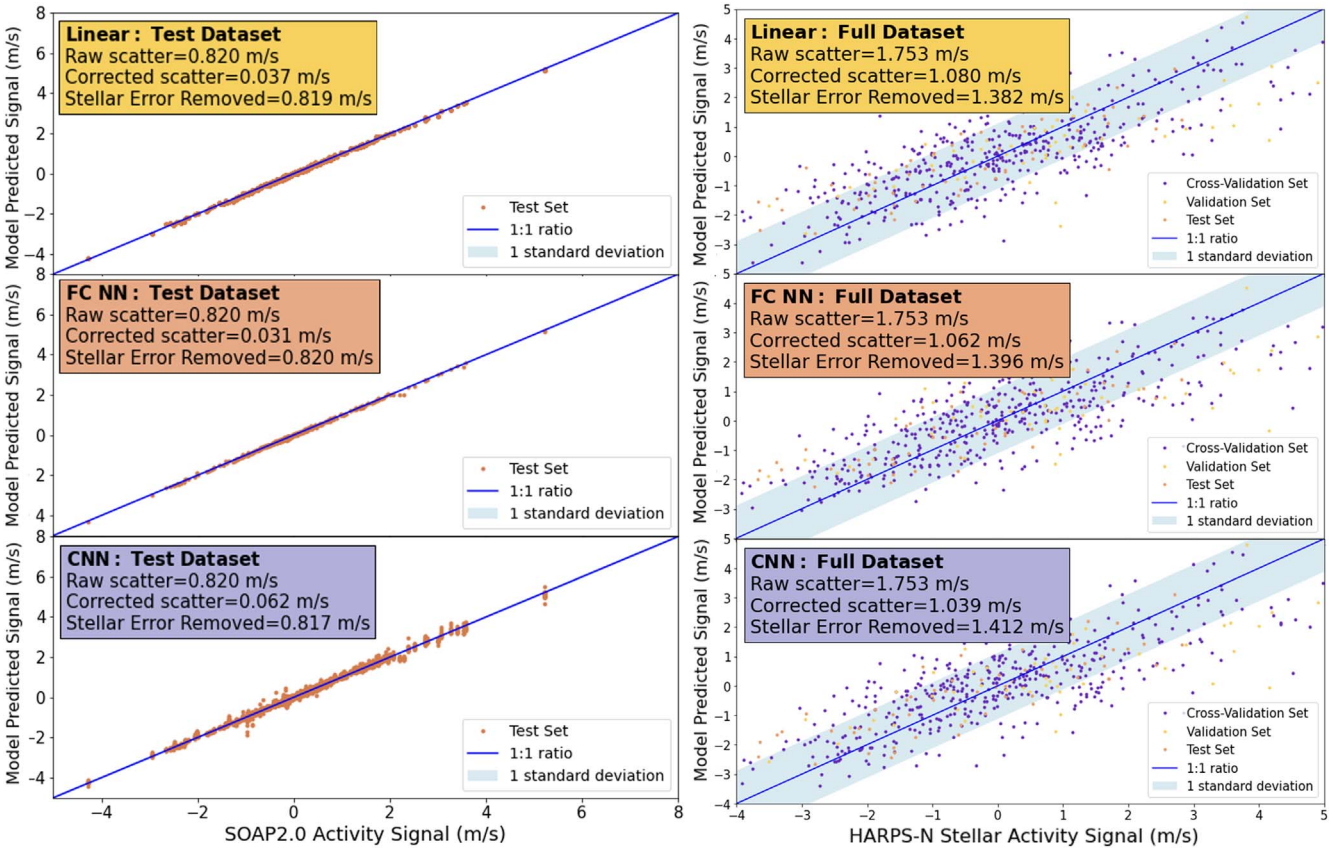

Figure 9. Linear, FC NN, and CNN results for SOAP 2.0 (left column) and HARPS-N data (right column). The scatter metric is standard σ_{SD} for the SOAP 2.0 simulated data and $\sigma_{Percentile}$ for the non-Gaussian HARPS-N observations. The stellar error removed is the difference between the raw scatter and corrected scatter in quadrature. For the SOAP 2.0 simulated data, the FC NN model performs best across the test set, reducing the raw scatter from 82.0 to 3.1 cm s^{-1}. For the HARPS-N solar data, the CNN model marginally outperforms the FC NN architecture by reducing the RV scatter from 175.3 to 103.9 cm s^{-1}, compared to 106.2 cm s^{-1} for the FC NN across the full data set.

from our CNN, the periodogram of the corrected RVs no longer has peaks corresponding to these activity signals. Thus, the CNN is able to identify and remove the quasi-periodic variability at the stellar rotation period based only on shape changes in the ΔCCF and does not use any timing information.

8. Discussion

8.1. What Is Limiting Our Precision?

Using ML, we were able to predict and remove stellar activity signals from HARPS-N Solar Telescope observations and reduce the scatter in the measured RVs by about a factor of two from 175.3 to 103.9 cm s^{-1}. While this improvement in RV precision is impressive and could increase our sensitivity to small planets if applied to observations of stars other than the Sun, our final scatter is still far greater than the roughly 10 cm s^{-1} precision necessary to detect habitable-zone Earth analogs around Sun-like stars.

What is limiting the precision of our activity-corrected HARPS-N RVs? One possibility is that our stellar activity corrections are not perfect, and the scatter in our corrected velocities is dominated by residual stellar activity signals. However, we think that this is unlikely. We see no evidence for any quasi-periodic stellar activity signals in the periodogram of our corrected RVs (see Figure 11), and our experiments with the SOAP 2.0 simulated data indicate that it is possible to achieve precision of few centimeters per second after modeling and removing stellar activity signals in a similar configuration.

Table 6
SOAP 2.0 Results

Test Set Results				
Scatter Metric	Raw Data	Corrected Data Using		
		Linear Model	FC NN	CNN
σ_{SD} (m s^{-1})	0.820	0.037	0.031	0.062

Note. We computed the standard deviation across the simulated stellar activity signals before applying any corrections (raw data) and then applied stellar activity corrections using all three model architectures. Their resulting reductions in scatter are listed across the test set.

Certainly real data will have complications and subtleties that make stellar activity harder to correct than in our idealized SOAP 2.0 simulations, but it seems unlikely that these differences would cause our limiting precision to be 20 times greater. Several other analyses of the HARPS-N solar data seem to agree that, even when we successfully model activity at the rotation period using a variety of different techniques, there is still some other process limiting our RV precision (Dumusque 2018; Milbourne et al. 2019; Miklos et al. 2020).

It is likely that the remaining scatter in our corrected HARPS-N data is dominated by instrumental noise. While HARPS-N is highly stabilized, the instrument does experience slow drifts and requires frequent calibrations to ensure the accuracy of its wavelength solution. The quality of these

Table 7
HARPS-N Results

Test Set Results				
Scatter	Raw	Corrected Data Using		
Metric	Data	Linear Model	FC NN	CNN
σ_{SD} (m s^{-1})	1.736	0.984	0.968	0.967
$\sigma_{k\text{-MAD}}$ (m s^{-1})	1.635	1.200	1.053	1.087
$\sigma_{Percentile}$ (m s^{-1})	1.878	0.871	1.008	0.928
Full Data Set Results				
Scatter	Raw	Corrected Data Using		
Metric	Data	Linear Model	FC NN	CNN
σ_{SD} (m s^{-1})	1.846	1.108	1.113	1.121
$\sigma_{k\text{-MAD}}$ (m s^{-1})	1.772	1.074	1.051	1.078
$\sigma_{Percentile}$ (m s^{-1})	1.753	1.080	1.062	1.039

Note. We computed three different scatter metrics due to the slightly non-Gaussian nature of our data. We list these across the cross-validation results, the test set, and finally combine these corrected data sets (Full Data Set Results).

wavelength solutions limits the precision of velocities measured by HARPS-N. Dumusque et al. (2021) report that wavelength solutions generated by the version of the DRS we use in this paper tend to change by about 74 cm s^{-1} on day-to-day timescales, which could explain almost all of the scatter we see in our corrected HARPS-N solar velocities. If this is the case, then this technique could in principle yield more precise velocities when applied to data from newer stabilized spectrographs like ESPRESSO (Pepe et al. 2021).

8.2. Comparison to Other Methods

Other common methods of reducing the RV scatter by characterizing and removing stellar activity signals include GPs. In a recent paper by Langellier et al. (2021), GPs reduce the rms scatter of the HARPS-N data set to a similar reduction in RV scatter as our ML methods. One notable difference is that we achieve this reduction in rms scatter without using any information about the timing of the observations, potentially eliminating the need for high-cadence sampling.[32]

Another promising method for predicting stellar activity signals is to track the unsigned (unpolarized) magnetic flux as a proxy (Haywood et al. 2020). By estimating rotationally modulated RV variations and the unsigned magnetic flux daily over 8 yr using spatially resolved SDO/HMI images, Haywood et al. (2020) showed that a simple fit with unsigned magnetic flux reduces rotationally modulated RV scatter by 62% (a factor of 2.6 improvement). They successfully recovered planet semiamplitudes of 0.3 m s^{-1} at orbital periods of ∼300 days. These numbers are not directly comparable to the work presented here because of different instrumental systematics and observational baselines; however, the improvement is of similar order. The authors note, however, that the unsigned magnetic flux is not yet measurable at high precision in slowly rotating, relatively inactive stars like the Sun. While this measurement is readily available for the Sun, making similar measurements for other stars will require pushing beyond the current state of the art in measuring Zeeman broadening from stellar spectra.

Recently, Collier Cameron et al. (2021) also explored the shape changes introduced in CCFs by computing an auto-correlation function that is invariant to translation (and thereby not sensitive to planetary reflex motion) but focused on stellar activity shape changes. In this analysis, the full 5 yr of the HARPS-N data set were used and injected planet signals of $K = 0.4$ m s^{-1}, and periods ranging from 7 to 200 days were recovered. Since this analysis used the full 5 yr and an older version of the DRS, these numbers are not directly comparable to our work. However, the improvement in rms is similar, which further supports that these stellar-activity-driven shape changes can serve as a useful indicator for the stellar RV contribution.

8.3. Implications for Planet Detection

To estimate the implications this ML method could have for planet detection, we injected a synthetic planet signal into both the full data set of raw RVs and raw CCFs (528 days of observations). We then ran our full pipeline as described in detail in Sections 3, 4, and 6. We attempted to detect the signals in a Lomb−Scargle periodogram and assess their significance using Markov Chain Monte Carlo (MCMC).

In Figure 12, we show a periodogram of the HARPS-N RVs before and after activity corrections with a planet signal injected with a semiamplitude of 0.4 m s^{-1} at a 1 yr period (corresponding to a planet of 4.53 $M_{\oplus}$). In the periodogram of the uncorrected HARPS-N RVs (top panel), the injected signal is visible but difficult to distinguish from other stronger peaks in the periodogram caused by stellar activity. However, in the periodogram of the corrected RVs (bottom panel), this planet signal is clearly the most prominent after we corrected for stellar activity signals.

Using the periodograms to initialize our MCMC, we derived the phase-folded fit in Figure 13 with $K = 0.53 \pm 0.07$ m s^{-1}, $P = 362.69^{+9.26}_{-8.03}$ days (Figure 14). The MCMC uncertainties in this fit seem to be slightly underestimated, likely due to the non-Gaussian noise properties in the RVs.

The sensitivity of our method appears to compare favorably with the sensitivity of GP regression. Langellier et al. (2021) found that they would need 10–15 yr of HARPS-N solar data to detect an 0.5 m s^{-1} RV signal at a 225-day period with 5σ confidence. We found that we can recover a 0.4 m s^{-1} injected signal with ∼5σ confidence using only 3 yr of HARPS-N solar data.

8.4. Future Work and Prospects for This Technique on Other Stars

To extend this method to other stars, we could take two different approaches. One approach would be to focus on one star at a time where we fit a simple (linear) ML model simultaneously with planet signals. This method has the advantage that it does not require the removal of astrophysical signals before fitting and that we would only need data for one star at a time. Some of the disadvantages would be that it could limit the complexity of the ML model and be more likely to overfit or have degeneracies where it fails to properly distinguish activity from planet signals, which is a common problem for other techniques like GPs. Preliminary

[32] Our method does, of course, require a rich data set for training, but in principle the training observations could be taken with any cadence, as opposed to GPs, which often requires multiple observations per stellar rotation period to effectively model activity.

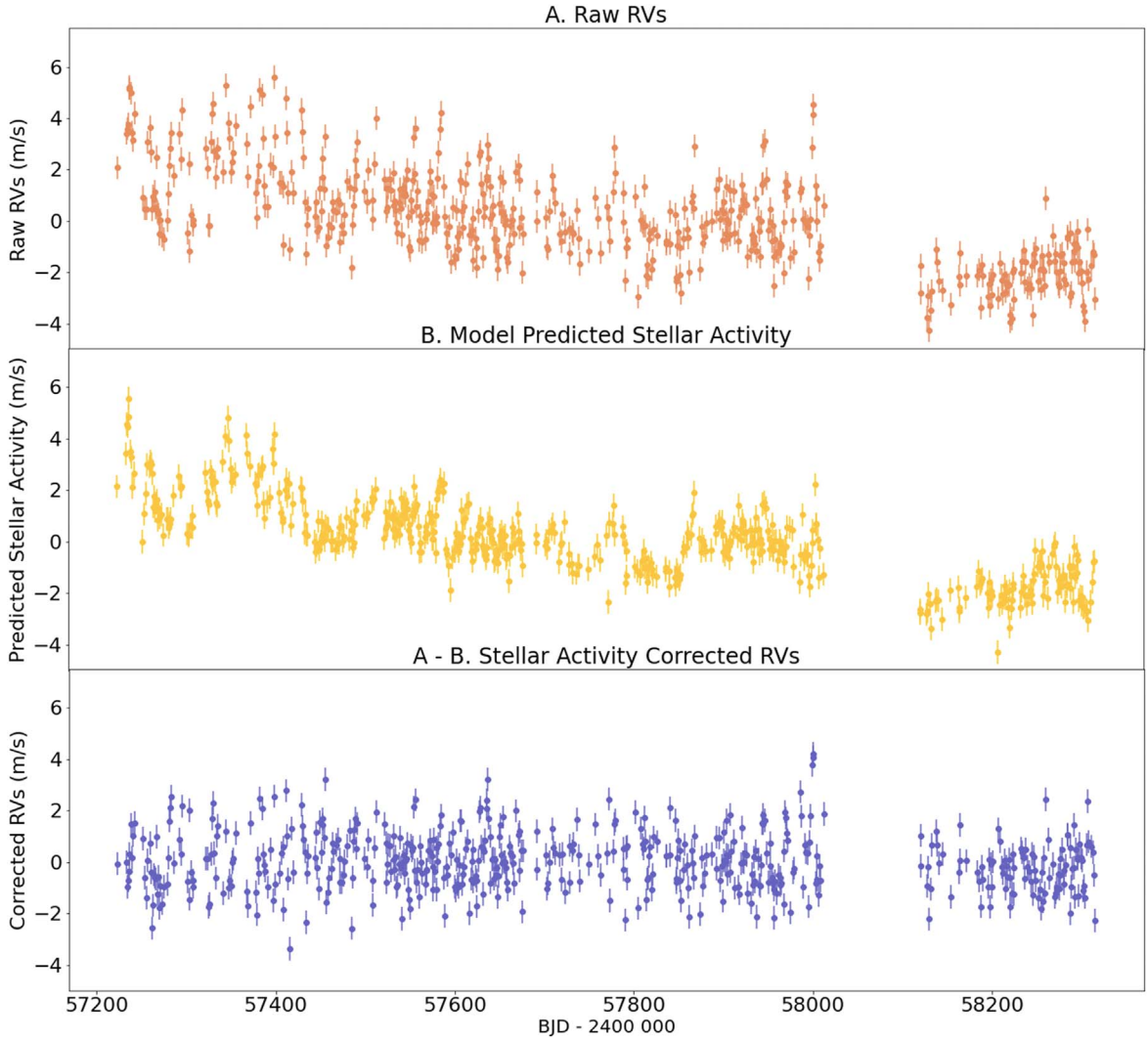

Figure 10. HARPS-N Solar Telescope raw (top), model-predicted (middle), and corrected (bottom) RVs over time. The stellar-activity-corrected RVs in the third panel are obtained by subtracting the predicted RVs (middle panel) from the raw RVs (top panel). The gaps in the observations of ∼58,100 days correspond to hardware downtime.

(The data used to create this figure are available.)

explorations of this type of simplified activity model have shown promising results on data from both HARPS-N (Z. L. de Beurs et al. 2022, in preparation) and EXPRES (Zhao et al. 2022) on stars with between 25 and 100 observations.

Another approach would be to train a more complex model on all stars observed by a given spectrograph simultaneously and predict stellar activity corrections for new stars (not included in the training set) based on the entire ensemble. The advantages would be that the larger training set would allow for more complex ML models that can predict the activity signals more accurately. However, this method has the disadvantage that the planet signals will need to be removed ahead of time. Some undetected planet signals will always remain in the data, meaning that we would lack a perfect "ground truth" on which to train our models. Instead, we would have to hope that undetected signals would average out across the training set. In addition, the different rotation rates, spectral types, and inclination angles may be challenging to solve and require significantly more model complexity. Adding input features to our models, such as the log $R'_{\rm HK}$ or Hα time series, stellar parameters like effective temperature and stellar radius, or stellar inclination angles derived from measurements of the projected rotational velocity and rotation period, may help our models make more accurate predictions.

In some ways, observations of stars at nighttime may be simpler to use as inputs to ML models. Unlike solar observations, nighttime observations have the following properties:

1. Differential extinction is significantly less of a concern for observations of other stars at nighttime. Unlike the Sun, the other stars that we observe are essentially point sources. Thus, differential extinction across the disk would not be resolved and would induce significantly less systematic signals.
2. There are some yearly effects on the CCFs of the Sun that will not appear in stars observed at nighttime. In solar observations, the FWHM of the CCF is modulated with 6-month and 1 yr timescales. This phenomenon is due to the eccentricity of Earth's orbit, which causes Earth's

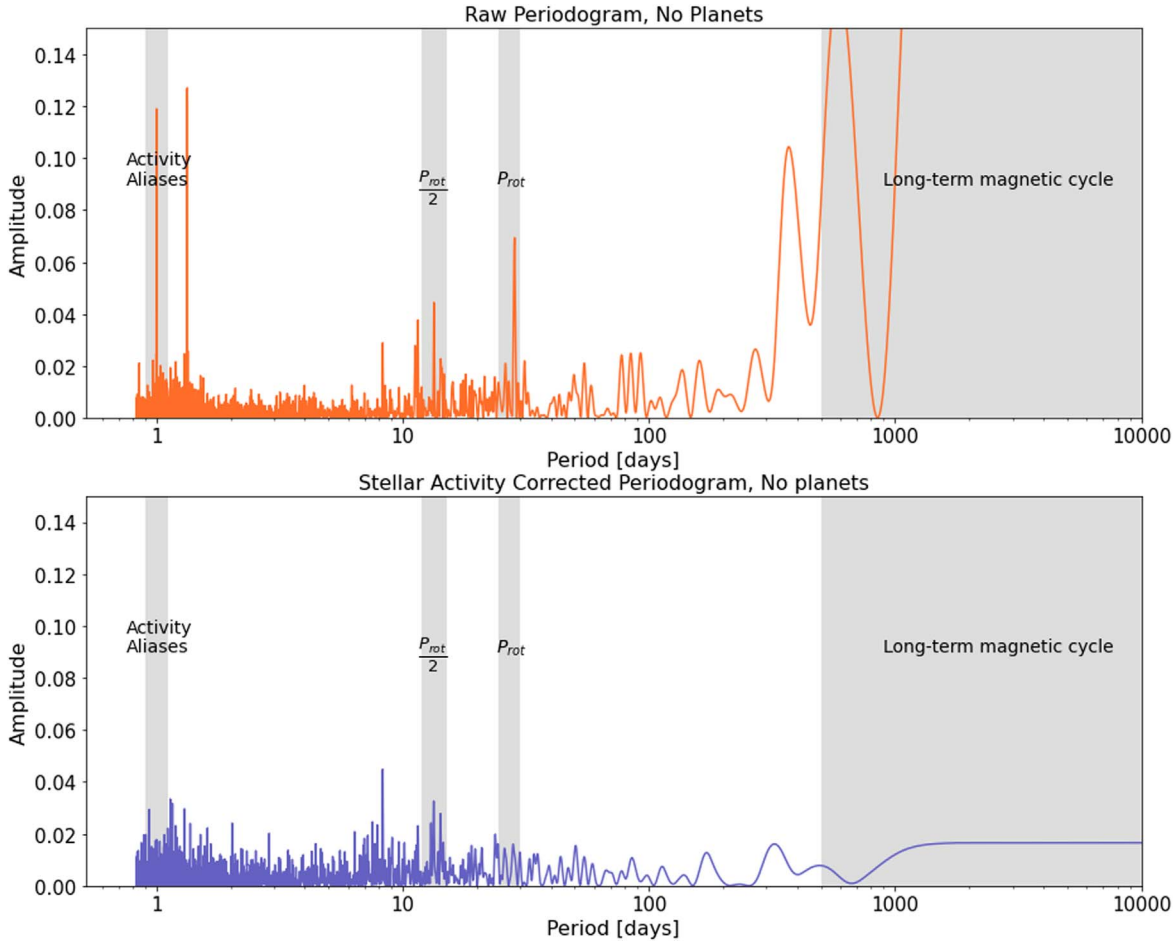

Figure 11. Periodogram: HARPS-N Solar Telescope raw (top) and corrected (bottom) RVs in Fourier space. The peaks in the top panel that correspond to stellar activity signals disappear in the bottom panel after applying the CNN model's stellar activity corrections.

angular velocity about the Sun to vary annually and changes the relative angular velocity of the Sun's rotation that we observe. The changing relative rotational velocity affects our measurement of the rotational broadening of solar spectral features and therefore causes variations in the FWHM of the CCF (see Figure 8(a) in Collier Cameron et al. 2019). The 6-month oscillation in the FWHM arises from the obliquity of the ecliptic plane relative to the solar equator.

On the other hand, nighttime observations will introduce new challenges of their own. For nighttime observations, we will not be able to average out granulation as well as for the Sun owing to lower-cadence observations. For other stars, the spectral lines also move across the detector owing to barycentric velocity changes. In addition, observations of stars other than the Sun will often be photon limited, unlike our solar observations, in which photon noise is negligible. Noisier observations will make separating activity signals from true RV shifts more difficult. Nighttime RV observations are strongly heteroskedastic, unlike solar observations, due to factors like the observing conditions and different exposure times. Training a model may require more sophisticated weighting than we used in this work. In the future, this deep-learning method could be applied to spectrographs like ESPRESSO (Pepe et al. 2020 accepted) and EXPRES (Jurgenson et al. 2016) and might perform especially well on the extremely large data sets expected to be collected by HARPS-3 (Thompson et al. 2016; Hall et al. 2018).

In future NN architectures, it may be advantageous to add pooling layers. Pooling layers take advantage of translational invariance in the input of CNNs. On the one hand, adding pooling layers could help for other stars where there may be slight shifts in the CFF due to undetected planets. On the other hand, pooling layers may prevent the detection of precise shifts in RV due to magnetic features. While our initial tests with pooling layers on solar data did not seem to help network performance, they may be helpful on more complex data sets.

Other possible future applications of our ML stellar activity model include detecting young planets orbiting young stars. Young stars' high rotation rates and high levels of stellar activity make them especially complex but simultaneously open the door to studying planetary formation and migration mechanisms. An ML technique to remove stellar activity signals could open the door to measuring more masses and densities for transiting planets orbiting bright young stars (Mann et al. 2018; Vanderburg et al. 2018; David et al. 2019; Newton et al. 2019). In this way, studying young planets around young stars is crucial to exoplanet demographic studies (Damasso et al. 2020).

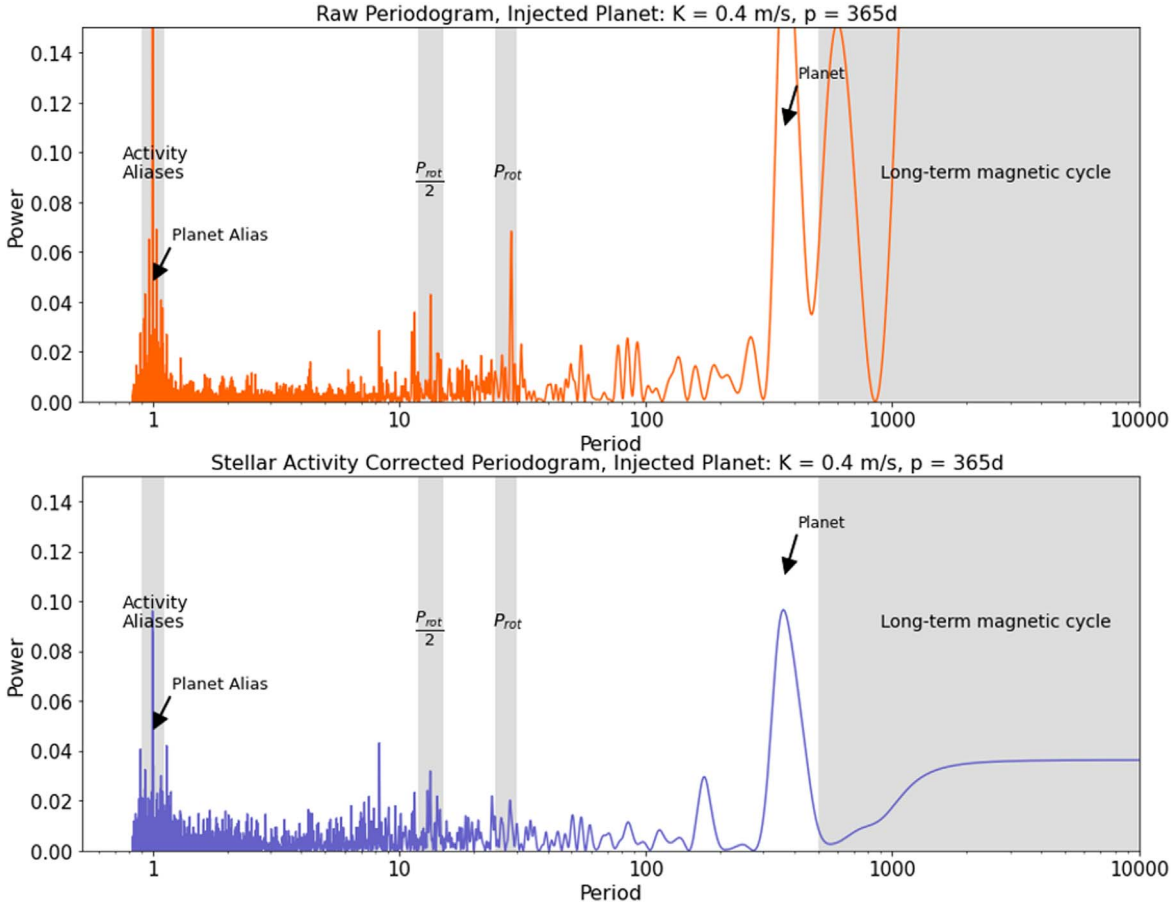

Figure 12. Periodogram, planet injection: $K = 0.3$ m s^{-1}, $P = 365.24$ days. In the top raw periodogram, the injected planet signal is distinct, but not the most prominent signal. Once these prominent stellar activities are corrected by the CNN model, this planet (and its alias) becomes the most dominant signal (bottom panel).

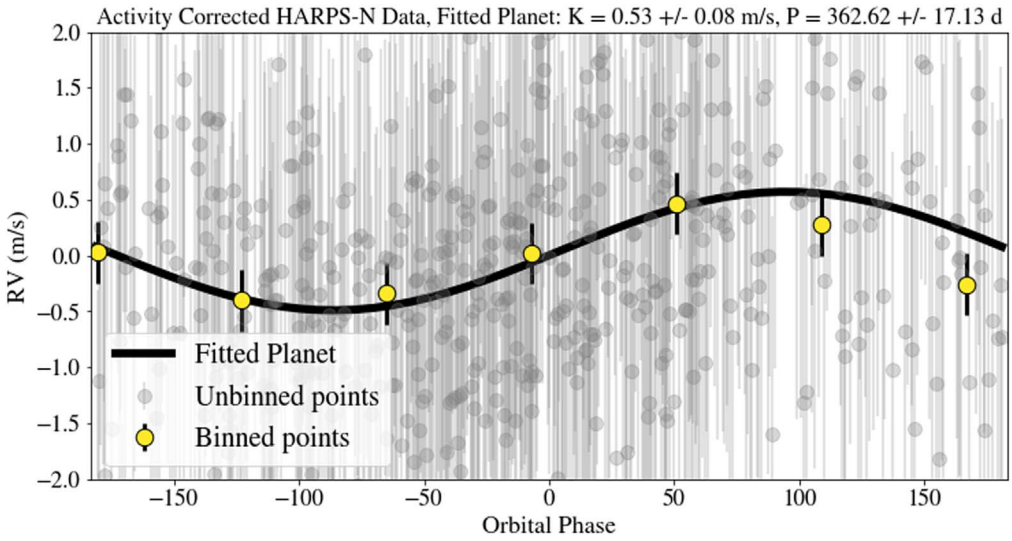

Figure 13. MCMC fitted planet: $K = 0.53 \pm 0.07$ m s^{-1}, $P = 362.69^{+9.26}_{-8.03}$ days. The MCMC corner plot can be found in Figure 14.

9. Conclusion

Achieving the extreme RV precision necessary to detect long-period Earth-mass exoplanets requires mitigating stellar activity signals. These dominant stellar activity signals hide the $\sim$10 cm s^{-1} signatures of Earth analog exoplanets and are difficult to remove owing to their unpredictable time evolution and quasi-periodic nature. Current methods for mitigating stellar activity signals often rely on high-cadence and carefully timed observations, but even with these methods, the detection of an Earth analog around a Sun-like will be very difficult (Langellier et al. 2021).

We have demonstrated an ML approach to removing stellar activity signals that does not require the frequent sampling and

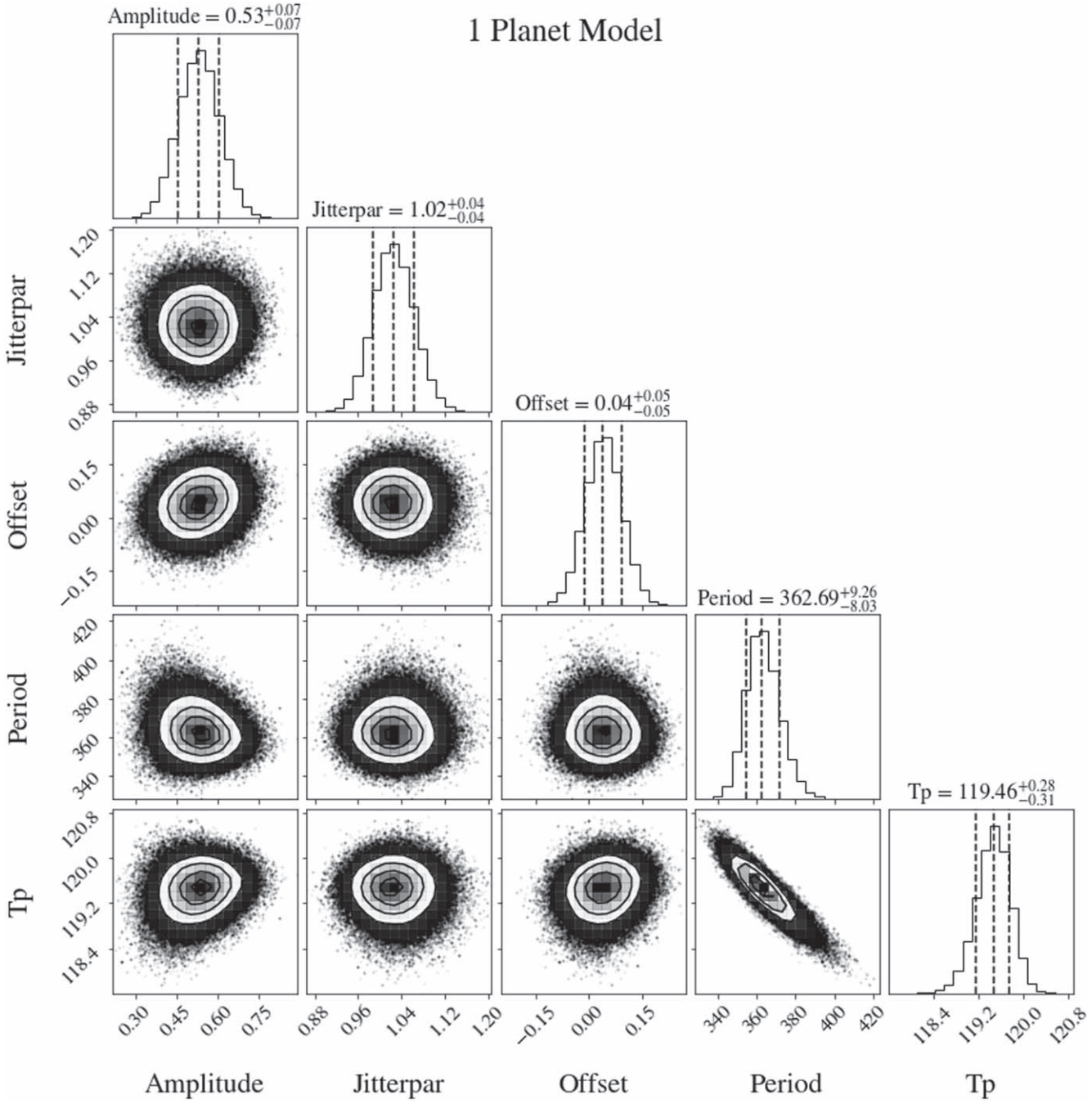

Figure 14. Corner plot for MCMC fitted planet: $K = 0.5 \pm 0.07$ m s^{-1}, $P = 362.69^{+9.26}_{-8.03}$ days.

timing information that other methods (like GP regression) depend on. By interpreting small shape changes in the stellar spectra induced by stellar activity, our ML models can predict and remove these dominant signals. So far, we have trained and tested our methods on simulated data and observations of the Sun and plan to apply these or similar methods to observations of other stars in the future. For stars other than our Sun, we will not have as many spectra that can serve as a training set for a single star of interest, but we may be able to train on an ensemble of other stars instead. We will make our code publicly available online to the exoplanet community.[33]

We developed and tested our methods on both simulated data (Monte Carlo generated using SOAP 2.0; Dumusque et al. 2014) and solar observations from the HARPS-N Solar Telescope (Dumusque et al. 2015). Our best-performing model for simulated data was an FC NN, which successfully reduced the scatter in simulated stellar activity signals from 82.0 to 3.1 cm s^{-1} across the test set. For the HARPS-N solar observations, our best-performing models, an FC NN and a CNN, both reduced the RV scatter from 175.3 to 103.9 cm s^{-1} across 3 yr of observations. When comparing our result to works that use GPs for HARPS-N observations (Langellier et al. 2021), our FC NN and CNN models achieve similar (slightly better) precision to GP regression on the same data, *without the need for high-cadence sampling and timing information*.

[33] https://github.com/zdebeurs/exoplanet-ml/tree/master/exoplanet-ml/rv_net

We explored how much an activity correction like the one we have demonstrated on the Sun could improve the detectability of planets in similar data sets around other stars. We injected planet signals into our activity-corrected HARPS-N observations and were able to recover signals with semiamplitudes down to 30 cm s^{-1}, improving on our detection limits in uncorrected observations by more than a factor of 2. However, these are not end-to-end injection/recovery tests and represent a best-case improvement to detection sensitivity. Future work will focus on investigating how to robustly prevent the algorithms from potentially confusing planetary and stellar activity signals. Nonetheless, these tests demonstrate that these advanced techniques could potentially pave the way to revealing previously hidden planets around our closest stellar neighbors.

We thank the anonymous referee for a constructive and detailed report, which helped us clarify key points and improve this manuscript.

We thank Ellen Price for invaluable assistance with Python environments. We acknowledge helpful conversations and feedback from George Dahl and members of Dave Latham's Coffee Club. The HARPS-N project has been funded by the Prodex Program of the Swiss Space Office (SSO), the Harvard University Origins of Life Initiative (HUOLI), the Scottish Universities Physics Alliance (SUPA), the University of Geneva, the Smithsonian Astrophysical Observatory (SAO), the Italian National Astrophysical Institute (INAF), the University of St Andrews, Queen's University Belfast, and the University of Edinburgh.

Z.L.D. acknowledges the generous support from the UT Office of Undergraduate Research Fellowship, the TIDES Advanced Research Fellowship, Deans Scholars, and the Junior Fellows Honors Program. Z.L.D. and A.V. acknowledge support from the TESS Guest Investigator Program under NASA grant 80NSSC19K0388. A.V.'s work was partially performed under contract with the California Institute of Technology (Caltech)/Jet Propulsion Laboratory (JPL) funded by NASA through the Sagan Fellowship Program executed by the NASA Exoplanet Science Institute. X.D. is grateful to the Branco-Weiss Fellowship for continuous support. This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (SCORE grant agreement No. 851555). A. C.C. acknowledges support from the Science and Technology Facilities Council (STFC) consolidated grant No. ST/R000824/1 and UKSA grant ST/R003203/1. This work was performed under contract with the California Institute of Technology (Caltech)/Jet Propulsion Laboratory (JPL) funded by NASA through the Sagan Fellowship Program executed by the NASA Exoplanet Science Institute (R.D.H.). R.D.H. is funded by the UK Science and Technology Facilities Council (STFC)'s Ernest Rutherford Fellowship (grant number ST/V004735/1). M.P. acknowledges financial support from the ASI-INAF agreement No. 2018-16-HH.0. A.M. acknowledges support from the senior Kavli Institute Fellowships.

Facilities: HARPS-N Solar Telescope, SDO.

Software. numpy (Oliphant 2006), matplotlib (Hunter 2007). Tensorflow, SOAP 2.0, Astropy (Price-Whelan et al. 2018), scipy (Virtanen et al. 2020).

ORCID iDs

Zoe. L. de Beurs https://orcid.org/0000-0002-7564-6047
Andrew Vanderburg https://orcid.org/0000-0001-7246-5438
Christopher J. Shallue https://orcid.org/0000-0002-7585-9974
Xavier Dumusque https://orcid.org/0000-0002-9332-2011
Andrew Collier Cameron https://orcid.org/0000-0002-8863-7828
Christopher Leet https://orcid.org/0000-0003-2369-0481
Lars A. Buchhave https://orcid.org/0000-0003-1605-5666
Rosario Cosentino https://orcid.org/0000-0003-1784-1431
Adriano Ghedina https://orcid.org/0000-0003-4702-5152
Raphaëlle D. Haywood https://orcid.org/0000-0001-9140-3574
Nicholas Langellier https://orcid.org/0000-0003-2107-3308
David W. Latham https://orcid.org/0000-0001-9911-7388
Mercedes López-Morales https://orcid.org/0000-0003-3204-8183
Michel Mayor https://orcid.org/0000-0002-9352-5935
Giusi Micela https://orcid.org/0000-0002-9900-4751
Timothy W. Milbourne https://orcid.org/0000-0001-5446-7712
Annelies Mortier https://orcid.org/0000-0001-7254-4363
Emilio Molinari https://orcid.org/0000-0002-1742-7735
David F. Phillips https://orcid.org/0000-0001-5132-1339
Matteo Pinamonti https://orcid.org/0000-0002-4445-1845
Giampaolo Piotto https://orcid.org/0000-0002-9937-6387
Ken Rice https://orcid.org/0000-0002-6379-9185
Dimitar Sasselov https://orcid.org/0000-0001-7014-1771
Alessandro Sozzetti https://orcid.org/0000-0002-7504-365X
Stéphane Udry https://orcid.org/0000-0001-7576-6236

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467
Aigrain, S., Pont, F., & Zucker, S. 2012, MNRAS, 419, 3147
Anglada-Escudé, G., Amado, P. J., Barnes, J., et al. 2016, Natur, 536, 437
Ansdell, M., Ioannou, Y., Osborn, H. P., et al. 2018, ApJL, 869, L7
Arentoft, T., Kjeldsen, H., Bedding, T. R., et al. 2008, ApJ, 687, 1180
Baranne, A., Queloz, D., Mayor, M., et al. 1996, A&AS, 119, 373
Bloom, J. S., Richards, J. W., Nugent, P. E., et al. 2012, PASP, 124, 1175
Boisse, I., Bonfils, X., & Santos, N. C. 2012, A&A, 545, A109
Bonfils, X., Mayor, M., Delfosse, X., et al. 2007, A&A, 474, 293
Brandt, P. N., & Solanki, S. K. 1990, A&A, 231, 221
Butler, R. P., Bedding, T. R., Kjeldsen, H., et al. 2003, ApJL, 600, L75
Campbell, B., Walker, G. A. H., & Yang, S. 1988, ApJ, 331, 902
Cavallini, F., Ceppatelli, G., & Righini, A. 1985, A&A, 143, 116
Cegla, H. 2019, Geosc, 9, 114
Chaplin, W. J., Cegla, H. M., Watson, C. A., Davies, G. R., & Ball, W. H. 2019, AJ, 157, 163
Chaushev, A., Raynard, L., Goad, M. R., et al. 2019, MNRAS, 488, 5232
Claret, A., & Bloemen, S. 2011, A&A, 529, A75
Collier Cameron, A., Ford, E. B., Shahaf, S., et al. 2021, MNRAS, 505, 1699
Collier Cameron, A., Mortier, A., Phillips, D., et al. 2019, MNRAS, 487, 1082
Collobert, R., & Weston, J. 2008, in Proc. of the 25th Int. Conference on Machine Learning (New York: ACM), 160
Cosentino, R., Lovis, C., Pepe, F., et al. 2012, Proc. SPIE, 8446, 84461V
Cretignier, M., Dumusque, X., Allart, R., Pepe, F., & Lovis, C. 2020, A&A, 633, A76
Cretignier, M., Dumusque, X., Hara, N. C., & Pepe, F. 2021, A&A, 653, A43
Damasso, M., Lanza, A. F., Benatti, S., et al. 2020, A&A, 642, A133
Dattilo, A., Vanderburg, A., Shallue, C. J., et al. 2019, AJ, 157, 169
David, T. J., Cody, A. M., Hedges, C. L., et al. 2019, AJ, 158, 79
Davis, A. B., Cisewski, J., Dumusque, X., Fischer, D. A., & Ford, E. B. 2017, ApJ, 846, 59
Delisle, J. B., Ségransan, D., Dumusque, X., et al. 2018, A&A, 614, A133

Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., Tuccillo, D., & Fischer, J. L. 2018, MNRAS, 476, 3661
Donati, J. F., Hébrard, E., Hussain, G., et al. 2014, MNRAS, 444, 3220
Dravins, D. 1982, ARA&A, 20, 61
Dravins, D., Lindegren, L., & Nordlund, A. 1981, A&A, 96, 345
Dumusque, X. 2018, A&A, 620, A47
Dumusque, X., Boisse, I., & Santos, N. C. 2014, ApJ, 796, 132
Dumusque, X., Cretignier, M., Sosnowska, D., et al. 2021, A&A, 648, A103
Dumusque, X., Glenday, A., Phillips, D. F., et al. 2015, ApJL, 814, L21
Dumusque, X., Lovis, C., Ségransan, D., et al. 2011a, A&A, 535, A55
Dumusque, X., Udry, S., Lovis, C., Santos, N. C., & Monteiro, M. J. P. F. G. 2011b, A&A, 525, A140
Emilio, M, Kuhn, J.R., Bush, R.I., & Scholl, I.F. 2012, ApJ, 750, 135
Giorgini, J. D., Yeomans, D. K., Chamberlin, A. B., et al. 1996, AAS/DPS Meeting Abstracts, 28, 25.04
Glorot, X., Bordes, A., & Bengio, Y. 2011, in Proc. of the Fourteenth Int. Conf. on Artificial Intelligence and Statistics (Fort Lauderdale, FL: PMLR), 315, https://proceedings.mlr.press/v15/glorot11a.html
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (Cambridge, MA: MIT Press)
Grec, G., & Fossat, E. 1979, A&A, 77, 351
Hall, R. D., Thompson, S. J., Handley, W., & Queloz, D. 2018, MNRAS, 479, 2968
Hathaway, D. H. 2015, LRSP, 12, 4
Haywood, R. D., Collier Cameron, A., Queloz, D., et al. 2014, MNRAS, 443, 2517
Haywood, R. D., Collier Cameron, A., Unruh, Y. C., et al. 2016, MNRAS, 457, 3637
Haywood, R. D., Milbourne, T. W., Saar, S. H., et al. 2020, arXiv:2005.13386
Horne, K. 1986, PASP, 98, 609
Huélamo, N., Figueira, P., Bonfils, X., et al. 2008, A&A, 489, L9
Hunter, J. D. 2007, CSE, 9, 90
James, G., Witten, D., Hastie, T., & Tibshirani, R. 2013, An Introduction to Statistical Learning, Vol. 112 (Berlin: Springer)
Jarrett, K., Kavukcuoglu, K., Ranzato, M., & LeCun, Y. 2009, in 2009 IEEE 12th Int. Conf. on Computer Vision (New York: IEEE), 2146
Jones, D. E., Stenning, D. C., Ford, E. B., et al. 2017, arXiv:1711.01318
Jurgenson, C., Fischer, D., McCracken, T., et al. 2016, Proc. SPIE, 9908, 99086T
Kjeldsen, H., & Bedding, T. R. 1995, A&A, 293, 87
Kosiarek, M. R., & Crossfield, I. J. M. 2020, AJ, 159, 271
Langellier, N., Milbourne, T. W., Phillips, D. F., et al. 2021, AJ, 161, 287
Latham, D. W., Mazeh, T., Stefanik, R. P., Mayor, M., & Burki, G. 1989, Natur, 339, 38
Lefebvre, S., García, R. A., Jiménez-Reyes, S. J., Turck-Chièze, S., & Mathur, S. 2008, A&A, 490, 1143
Leighton, R. B., Noyes, R. W., & Simon, G. W. 1962, ApJ, 135, 474
Lindegren, L., & Dravins, D. 2003, A&A, 401, 1185
Livingston, W. C. 1982, Natur, 297, 208
Lomb, N. R. 1976, Ap&SS, 39, 447
Loshchilov, I., & Hutter, F. 2017, arXiv:1711.05101
Mann, A. W., Vanderburg, A., Rizzuto, A. C., et al. 2018, AJ, 155, 4
Marsh, T. R. 1989, PASP, 101, 1032
Mayor, M., Pepe, F., Queloz, D., et al. 2003, Msngr, 114, 20
Mayor, M., & Queloz, D. 1995, Natur, 378, 355
Medina, A. A., Johnson, J. A., Eastman, J. D., & Cargile, P. A. 2018, ApJ, 867, 32
Meunier, N., Desort, M., & Lagrange, A. M. 2010, A&A, 512, A39
Miklos, M., Milbourne, T. W., Haywood, R. D., et al. 2020, ApJ, 888, 117
Milbourne, T. W., Haywood, R. D., Phillips, D. F., et al. 2019, ApJ, 874, 107
Milbourne, T. W., Phillips, D. F., Langellier, N., et al. 2021, ApJ, 920, 21
Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Machine Learning (New York: ACM), 807
National Academies of Sciences, Engineering, and Medicine and others 2018, Exoplanet Science Strategy (Washington, DC: National Academies Press)
Newton, E. R., Mann, A. W., Tofflemire, B. M., et al. 2019, ApJL, 880, L17
Noyes, R. W., Hartmann, L. W., Baliunas, S. L., Duncan, D. K., & Vaughan, A. H. 1984, ApJ, 279, 763
Oliphant, T. E. 2006, A Guide to NumPy (USA: Trelgol Publishing)
Osborn, H. P., Ansdell, M., Ioannou, Y., et al. 2020, A&A, 633, A53
Oshagh, M., Boisse, I., Boué, G., et al. 2013, A&A, 549, A35
Pearson, K. A., Palafox, L., & Griffith, C. A. 2018, MNRAS, 474, 478
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, JMLR, 12, 2825, https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
Pepe, F., Cristiani, S., Rebolo, R., et al. 2021, A&A, 645, A96
Pepe, F., Mayor, M., Galland, F., et al. 2002, A&A, 388, 632
Phillips, D. F., Glenday, A. G., Dumusque, X., et al. 2016, Proc. SPIE, 9912, 99126Z
Polyak, B. T. 1964, Comput. Math. Math. Phys., 4, 1
Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., et al. 2018, AJ, 156, 123
Queloz, D., Henry, G. W., Sivan, J. P., et al. 2001, A&A, 379, 279
Rajpaul, V., Aigrain, S., Osborne, M. A., Reece, S., & Roberts, S. 2015, MNRAS, 452, 2269
Ramesh, A., Kambhampati, C., Monson, J. R., & Drew, P. 2004, Ann. R. Coll. Surg. Engl., 86, 334
Robertson, P., Mahadevan, S., Endl, M., & Roy, A. 2014, Sci, 345, 440
Rušin, V. 1972, BAICz, 23, 75
Saar, S. H., & Donahue, R. A. 1997, ApJ, 485, 319
Scargle, J. D. 1982, ApJ, 263, 835
Schanche, N., Collier Cameron, A., Hébrard, G., et al. 2019, MNRAS, 483, 5534
Setiawan, J., Henning, T., Launhardt, R., et al. 2008, Natur, 451, 38
Severnyi, A. B., Kotov, V. A., & Tsap, T. T. 1980, A&A, 88, 317
Shallue, C. J., Lee, J., Antognini, J., et al. 2019, JMLR, 20, 1, http://jmlr.org/papers/v20/18-789.html
Shallue, C. J., & Vanderburg, A. 2018, AJ, 155, 94
Strassmeier, K. G., Ilyin, I., & Steffen, M. 2018, A&A, 612, A44
Suárez Mascareño, A., Faria, J. P., Figueira, P., et al. 2020, A&A, 639, A77
Szentgyorgyi, A., Barnes, S., Bean, J., et al. 2014, Proc. SPIE, 9147, 914726
Thompson, S. J., Queloz, D., Baraffe, I., et al. 2016, Proc. SPIE, 9908, 99086F
Tuomi, M., Anglada-Escud é, G., Gerlach, E., et al. 2013, A&A, 549, A48
Ulrich, R. K. 1970, ApJ, 162, 993
Vanderburg, A., Mann, A. W., Rizzuto, A., et al. 2018, AJ, 156, 46
VanderPlas, J., Connolly, A. J., Ivezic, Z., & Gray, A. 2012, in Proc. of Conf. on Intelligent Data Understanding (New York: IEEE), 47
VanderPlas, J. T., & Ivezić, Ž. 2015, ApJ, 812, 18
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Wilken, T., Curto, G. L., Probst, R. A., et al. 2012, Natur, 485, 611
Wise, A., Plavchan, P., Dumusque, X., Cegla, H., & Wright, D. 2022, ApJ, 930, 121
Yu, L., Vanderburg, A., Huang, C., et al. 2019, AJ, 158, 25
Zechmeister, M., & Kürster, M. 2009, A&A, 496, 577
Zhao, J., & Ford, E. B. 2022, arXiv:2201.03780
Zhao, L. L., Fischer, D. A., Ford, E. B., et al. 2022, AJ, 163, 171
Zucker, S., & Giryes, R. 2018, AJ, 155, 147