



# Do Androids Dream of Magnetic Fields? Using Neural Networks to Interpret the Turbulent Interstellar Medium

J. E. G. Peek<sup>1,2</sup>  and Blakesley Burkhart<sup>3,4</sup> 

<sup>1</sup> Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA

<sup>2</sup> Department of Physics & Astronomy, Johns Hopkins University, 3400 North Charles Street, Baltimore, MD 21218, USA

<sup>3</sup> Center for Computational Astrophysics, Flatiron Institute, 162 Fifth Avenue, New York, NY 10010, USA

<sup>4</sup> Department of Physics and Astronomy, Rutgers University, 136 Frelinghuysen Road, Piscataway, NJ 08854, USA

Received 2019 May 2; revised 2019 August 9; accepted 2019 August 13; published 2019 September 3

## Abstract

The interstellar medium (ISM) of galaxies is composed of a turbulent magnetized plasma. In order to quantitatively measure relevant turbulent parameters of the ISM, a wide variety of statistical techniques and metrics have been developed that are often tested using numerical simulations and analytic formalism. These metrics are typically based on the Fourier power spectrum, which does not capture the Fourier phase information that carries the morphological characteristics of images. In this work we use density slices of magnetohydrodynamic turbulence simulations to demonstrate that a modern tool, convolutional neural networks, can capture significant information encoded in the Fourier phases. We train the neural network to distinguish between two simulations with different levels of magnetization. We find that, even given a tiny slice of simulation data, a relatively simple network can distinguish sub-Alfvénic (strong magnetic field) and super-Alfvénic (weak magnetic field) turbulence >98% of the time, even when all spectral amplitude information is stripped from the images. In order to better understand how the neural network is picking out differences between the two classes of simulations we apply a neural network analysis method called “saliency maps.” The saliency map analysis shows that sharp ridge-like features are a distinguishing morphological characteristic in such simulations. Our analysis provides a way forward for deeper understanding of the relationship between magnetohydrodynamic turbulence and gas morphology and motivates further applications of neural networks for studies of turbulence. We make publicly available all data and software needed to reproduce our results.

*Unified Astronomy Thesaurus concepts:* [Interstellar magnetic fields \(845\)](#); [Computational methods \(1965\)](#)

## 1. Introduction

Magnetohydrodynamic (MHD) turbulence is now part of the established paradigm of the interstellar medium (ISM) of galaxies and can influence the behavior of gas and dust over scales ranging from beyond a kiloparsec to below astronomical unit (Armstrong et al. 1995; Elmegreen & Scalo 2004; Lazarian 2007; Burkhart 2014). MHD turbulence is understood to be important for fundamental astrophysical processes including star formation, galactic pressure support, the transport of heat and metals, formation of molecular hydrogen, the acceleration and diffusion of cosmic rays, and the structure of magnetic field lines (Lazarian & Vishniac 1999; Mac Low & Klessen 2004; Lazarian 2006; Yan 2009; Burkhart et al. 2010; Burkhart & Lazarian 2012; Federrath & Klessen 2012; Bialy et al. 2017; Pingel et al. 2018).

Despite the importance of MHD turbulence for galactic processes, it is difficult to devise metrics that can accurately characterize plasma fluid properties from astronomical observations. MHD turbulence is notoriously difficult to study, even in a controlled laboratory setting (Nornberg et al. 2006; Bayliss et al. 2007) and even more challenging to quantify and understand when dealing with line-of-sight (LOS) effects, radiative transfer, and telescope beam smearing (Hill et al. 2008; Burkhart et al. 2013a; Offner et al. 2014; Koch et al. 2019). In light of these challenges, direct numerical simulations of turbulence have been critical for our understanding of the physical conditions and statistical properties of MHD turbulence in astrophysical environments (Cho et al. 2002; Mac Low & Klessen 2004; Ballesteros-Paredes & Hartmann 2007;

Kowal & Lazarian 2007; McKee & Ostriker 2007; Federrath et al. 2008; Burkhart et al. 2009; Collins et al. 2012). Numerical MHD turbulence simulations generally can resemble observations in terms of the scaling of the power spectrum and the overall density structure (e.g., filaments). However numerical simulations of turbulence lack the spatial dynamic range of the real ISM and therefore cannot reach the observed Reynolds numbers of nature. The Reynolds number ( $Re^{1/2}$ ) is the ratio of the large eddy turnover rate to the viscous dissipation rate. Therefore, large  $Re$  correspond to negligible viscous dissipation of large eddies over the eddy turnover time and simulations are therefore too viscous, usually due to numerical viscosity. However, the statistical properties of the ISM, including the density histogram (or PDF; Vazquez-Semadeni et al. 1997; Burkhart et al. 2009, 2015, 2017; Federrath et al. 2009; Kainulainen et al. 2009; Burkhart & Lazarian 2012; Chen et al. 2018), velocity/density power spectrum (Stanimirovic et al. 1999; Lazarian & Pogosyan 2000; Stanimirović et al. 2004; Lazarian & Pogosyan 2008; Burkhart et al. 2010, 2013b; Chepurnov et al. 2015; Pingel et al. 2018), three-point functions/phase analysis (Burkhart et al. 2009; Burkhart & Lazarian 2016; Portillo et al. 2018; Koch et al. 2019), and principle component analysis (Heyer & Brunt 2004, 2012; Yuen et al. 2018), are well represented by MHD turbulence simulations. Therefore, statistical studies may be the current best method for understanding the properties of turbulence and for connecting observations and numerical simulations.

The most common statistical tool of turbulence studies for nearly a century has been the spatial or temporal Fourier power

spectrum. This is because the power spectrum enables the examination of the turbulence energy cascade as a function scale or frequency and can reveal the sources (injection scale) and sinks (dissipation scale) of energy and the self-similar behavior of the inertial range scaling. Furthermore, the power spectrum of ISM density and LOS Doppler-shifted velocity can inform on the spatial and kinematic scaling of turbulence and sonic Mach number (Lazarian & Pogosyan 2004, 2006; Kowal & Lazarian 2007; Goodman et al. 2009; Heyer et al. 2009; Burkhart et al. 2010, 2014; Collins et al. 2012; Chepurnov et al. 2015; Koch et al. 2019).

The power spectrum is defined as

$$P(k) = \sum_k \tilde{F}(k) \cdot \tilde{F}^*(k) \quad (1)$$

where  $k$  is the wavenumber and  $\tilde{F}(k)$  is the Fourier transform of the field under study, e.g., density, velocity, magnetic energy, etc.

Equation (1) demonstrates a critical limitation of the Fourier power spectrum: it contains only the Fourier *amplitudes* and neglects the Fourier phases. This is problematic for studies of MHD turbulence because interactions among MHD waves can produce correlations in Fourier phases that are completely missed by the power spectrum (Burkhart & Lazarian 2016). The structure imprinted on the phase information will be entirely lost to a power spectral analysis. Furthermore, the degree of phase coherence or randomness in MHD turbulence is important for MHD wave-wave interactions and particle transport. In order to study both phase and amplitude information a number of statistics beyond the power spectrum have been proposed for ISM studies, including the three-point correlation function or bispectrum (Burkhart et al. 2009; Portillo et al. 2018) and the phase coherence (Burkhart & Lazarian 2016). Additional statistical tools and methodologies for studies of turbulence have been developed in the last several decades and include course graining (Aluie 2017; Eyink & Drivas 2018; Bian & Aluie 2019), higher-order moments (Kowal & Lazarian 2007; Burkhart et al. 2009, 2010; Gaensler et al. 2011), wavelets (Kowal & Lazarian 2010; Farge & Schneider 2015; Le et al. 2018), topological techniques (Kowal & Lazarian 2007; Chepurnov et al. 2008; Burkhart et al. 2012), clump and hierarchical structure finders (Rosolowsky et al. 2008; Goodman et al. 2009; Burkhart et al. 2013a), Tsallis distributions (Esquivel & Lazarian 2011; Tofflemire et al. 2011), and structure functions as tests of intermittency and anisotropy (She & Leveque 1994; Cho & Lazarian 2003; Esquivel & Lazarian 2005; Kowal & Lazarian 2010; Burkhart et al. 2014; Verdini et al. 2015; Hellinger et al. 2018; Bian & Aluie 2019). While these tools have been successful at obtaining some of the turbulence parameters such as the driving scale and sonic Mach number, they are less adept at obtaining magnetic field information that might be encoded in the phases and are somewhat difficult to interpret physically.

A generic problem in studying the phase content of images is that there is no single metric that encompasses all the information, and thus feature vectors must be constructed “by hand,” informed either by our best theoretical understanding of the underlying processes or by observations that clue us in to important morphological properties (e.g., Clark et al. 2014) or are otherwise difficult to glean physical meaning from, such as the bispectrum (Burkhart et al. 2010). Nevertheless, the phase information is crucial for reproducing images with content that

is understandable by humans. For example, Oppenheim & Lim (1981) showed that preserving the phase in an image, but scrambling or distorting the power, generated images that are comprehensible to the human eye, but preserving the power and distorting the phase led to incomprehensible images. This result gives us the intuition that a tool or statistic that can distinguish between visually distinct structures may be able to tap into the information locked up in the Fourier phases.

Tools that mimic the human capacity to distinguish between physically and semantically different scenes have long been the goal of Computer Vision and Machine Vision. These disciplines have tended to build up semantic information from low-level, custom-built feature vectors from segmented images into ever more complex scene analysis tools (e.g., Sonka et al. 2008). The rise of neural networks, deep learning, and, in particular, convolutional neural networks (CNNs), has completely revolutionized our ability to extract semantic information from images and upended the computer vision paradigm (see e.g., LeCun et al. 2015, for a broad overview). CNNs allow for hierarchical feature extraction without any custom feature vectors, and as such they can be used for astronomical images as easily as other images. Indeed, more than 40 papers have been published using CNNs in the astronomical literature in 2018, twice the number in all previous years combined. While many of these articles are exploratory in nature, a fair number report increased speed and/or increased accuracy over previous methods.

A common complaint among physical scientists is that while neural networks, and machine learning systems as a whole, can accurately propagate labels and perform regressions, they represent a “black box.” These systems often have millions of fit parameters, and thus understanding what the system has “learned” is far from trivial. In recent years, significant progress has been made in extracting the knowledge built up by a neural network by applying new analysis tools to understand the network’s intuition (see e.g., [distill.pub](#)). Understanding the way neural networks operate in different physical contexts may allow us to gain insight into how information is encoded in astronomical images, and thereby build more powerful physical theories.

In this work we demonstrate the power of CNNs for studies of ISM turbulence in the absence of power spectral information, i.e., just using the pure phase information. We take a number of steps to remove information that is known to be useful in the studies of magnetized turbulence in order to demonstrate how CNNs can harness currently unknown aspects of the information contained in images to measure the physics of turbulence. We will show that we can detect changes in the Alfvén Mach number without velocity information, histogram information, the power spectrum, or stretched features along the background magnetic field.

This Letter is organized as follows. In Section 2 we describe the simulation data we use in the analysis. In Section 3 we describe CNNs and the network architecture and methods we use. In Section 4 we report our quantitative and qualitative results. We discuss and conclude in Section 5.

## 2. Simulation Data

We use the database of 3D numerical simulations of isothermal compressible (MHD) turbulence with resolution  $512^3$  presented in a number of past works (Kowal et al. 2007; Burkhart et al. 2009, 2013a). This database of simulations is

part of the Catalog for Astrophysical Turbulence Simulations (CATS; B. Burkhart et al. 2019, in preparation).<sup>5</sup> We refer to these works for the details of the numerical set-up and here provide a short overview. The code is a third-order accurate ENO scheme (Cho & Lazarian 2003) that solves the ideal MHD equations in a periodic box with purely solenoidally driving and uses an isothermal equation of state. We vary the input values for the sonic Mach number ( $M_s = v/c_s$ , where  $v$  is the flow velocity and  $c_s$  is the sound speed) and Alfvénic Mach number ( $M_A = v/v_A$ , where  $v_A$  is the Alfvén speed). The magnetic field consists of the uniform background field and a turbulent field, i.e.,  $B = B_{\text{ext}} + b$ . Initially,  $b = 0$  and the initial density field is uniform.

We test the utility of neural networks for finding magnetic field structure in density using snapshots of each of two simulations at a resolution of  $512^3$ . The simulations have a fixed sonic Mach number of  $M_s = 7$ . There are two different magnetic field values used in this investigation:  $M_A \approx 0.7$  (sub-Alfvénic) and  $M_A \approx 2.0$  (super-Alfvénic), otherwise the simulations examined have identical parameters. We use four snapshots from each simulation at 5, 5.5, 6, and 7 eddy turnover times.

The sonic and Alfvénic Mach numbers are important control parameters for simulations of MHD turbulence. They are also related to the plasma  $\beta$ , i.e., the ratio of the thermal pressure ( $P_{\text{thermal}}$ ) to magnetic pressure ( $P_{\text{mag}}$ ), which is defined as  $\beta = \frac{P_{\text{thermal}}}{P_{\text{mag}}}$ . The plasma  $\beta$  can be defined as  $\beta = \frac{2M_s^2}{M_A^2}$ . The sonic Mach number can be measured directly from the LOS velocity dispersion of spectral lines and measurements of temperature in the ISM. However, because magnetic fields are difficult to directly detect (Clark et al. 2012; Crutcher 2012; Soler et al. 2013; Lazarian et al. 2018), the Alfvénic Mach number is more elusive. The simulation data used are available here: [10.7910/DVN/UKOPY](https://doi.org/10.7910/DVN/UKOPY).

### 3. Method: CNNs

Our goal is to build a system that can distinguish between simulations sub- and super-Alfvénic turbulence, and thereby learn what information exists in the image plane about the turbulence. To do this we use a CNN deployed with the Keras and Tensorflow frameworks (Abadi et al. 2015; Chollet et al. 2015).

CNNs (also called *convnets*) are a subclass of neural networks that have a specific limited connectivity, originally patterned after the structure of the neural structure of the visual processing structures in brains (Hubel & Wiesel 1962). They were first made popular in the astronomical literature by Dieleman et al. (2015), who used a CNN to reproduce visual classifications of galaxy morphology.

A standard neural network is composed of *neurons*: a neuron takes in some inputs and provides an output

$$y = f\left(\sum_{i=1}^d x_i \cdot w_i\right), \quad (2)$$

where  $x_i$  is the input vector,  $w_i$  are a set of trainable weights, and  $f$  is some nonlinear, typically monotonically increasing, activation function (e.g., LeCun et al. 2015). In this way a network can be built up connecting the outputs of many

neurons on some layer  $j$  to many neurons on the next layer  $j + 1$ . A “fully connected layer” is a set of weights between layers of neurons that connects all neurons on level  $j$  to level  $j + 1$ . A network will typically narrow, until the number of final neurons is equal to the number of classes the network is being designed to distinguish. The process of training a network consists of providing labeled data to the network and measuring the resultant class. If the class is incorrect the weights are adjusted through back-propagation of error. In most modern training scenarios a small subset of the training data is presented to the network at a time, and the weights are adjusted in a process called *stochastic gradient descent*.

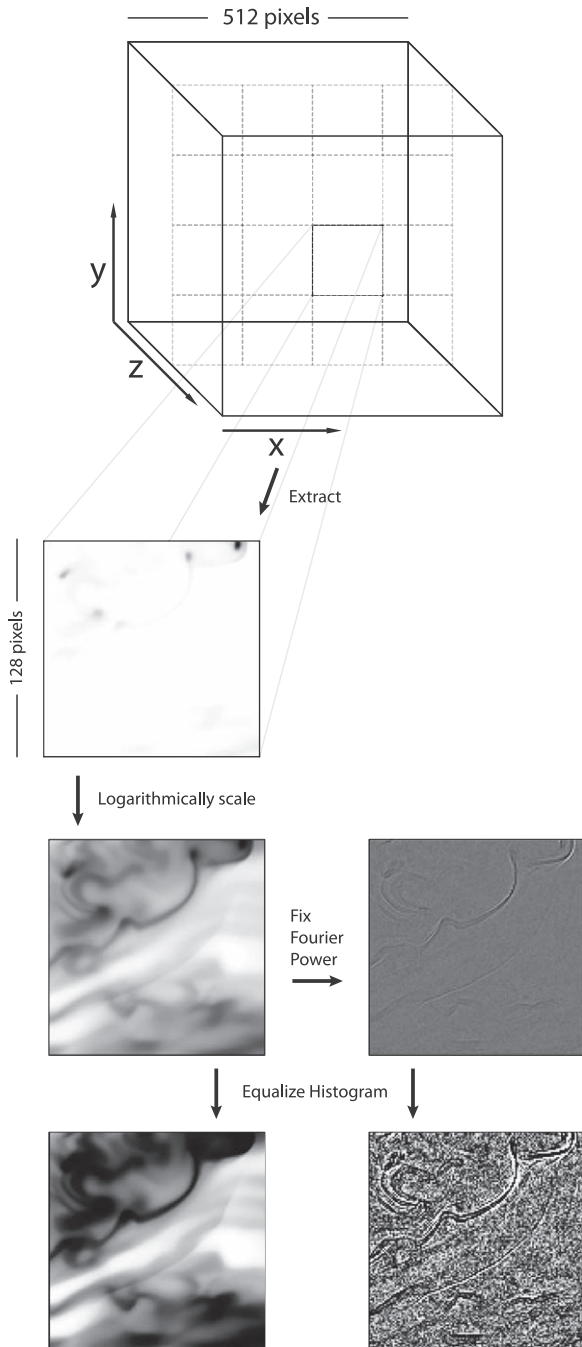
A CNN departs from this standard structure by the introduction of two extra layer types, *convolutional layers* and *pooling layers*.<sup>6</sup> If we construct our input as an  $M \times M$  image, a first convolutional layer would map each  $N \times N$  sub-patch of the image to a neuron in the next layer below with a grid of weights  $w_{ij}$ . This in effect generates a convolution of the image above. In practice many such convolutions would be applied to the layer, represented by  $w_{ij,k}$ , thus mapping to a  $M \times M \times K$  grid of neurons. A pooling layer decreases the resolution of the image by only passing the maximum activation of the neurons within a patch to the next layer. Thus, by alternating pooling and convolutional layers, the grid of neurons becomes smaller in the image plane and deeper in the convolutional dimension until fully connected layers are used at the end of the network. A more complete description of CNNs appropriate for an astronomical audience can be found in Section 5 of Dieleman et al. (2015).

CNNs take input images that typically go through a number of preprocessing steps. Our preprocessing steps are designed both to make the network effective, but also to strip out as much information as possible that is not relevant to our study of phase information (see Figure 1). To make images to start with we extract  $128 \times 128$  slices from the  $512 \times 512 \times 512$  grid point density cubes perpendicular to the magnetic field. Turbulent eddies are stretched along the magnetic axis (Goldreich & Sridhar 1995; Lazarian & Vishniac 1999), and therefore we slice across the magnetic field line axis in order to avoid anisotropic effects on the density field. Each  $512 \times 512$  plane is cut up into 16 images. We repeat this process across all four snapshots, generating  $16 \times 512 \times 4 = 32,768$  for each of the two simulations. Because the bounds of the simulation box are periodic, the data cube can be displaced in  $x$  and  $y$  with periodic bounds without injecting sharp edges into the data. In order to present the largest number of uncorrelated images to the network, we shift our box by 64 grid points in  $x$  and separately in  $y$  before again extracting the 16 images for each consecutive slice along  $z$ . This technique of “data enhancement” is common in neural network training, and yields 32,768 images per cube, or 131,072 per simulation. We perform this procedure on the density cubes for each of the four snapshots for both the super-Alfvénic and sub-Alfvénic simulations. The data from the first three snapshots comprise the training data, while the data from the last snapshot comprises the test data. We use the last snapshot as the test data because it is a full eddy turnover time from the training snapshots, and thus provides a largely independent test. The image pixel values are scaled by a base 10 logarithm.

<sup>5</sup> More information on the CATS project can be found at [www.mhdturbulence.com](http://www.mhdturbulence.com).

<sup>6</sup> See <https://cs231n.github.io/convolutional-networks/> for a discussion.





**Figure 1.** Preprocessing steps for the data. Each slice of each simulation data cube is split into 16  $128 \times 128$  grid point regions and extracted as an image. We then take the logarithm of the image pixels. These images are then split into two groups, one is left as is, the other is then run through the “Fixed Fourier Power” procedure. Then both sets of images are histogram equalized.

In addition to this we make a separate, parallel data set using the same methodology, but with an additional preprocessing step. For each  $x$ - $y$  slice of the cube we apply a fast Fourier transform, and set the Fourier power to unity. We then return to the image domain through an inverse fast Fourier transform. Thus, all the images for this fixed Fourier power (FFP) training and test set have no power spectral information.

Lastly, we perform histogram equalization using the exposure method from *sci-kit-image* (van der Walt et al. 2014), such that each image has a roughly equal distribution of values from 0 to 1. This is a common preprocessing technique

deployed here mainly for the purpose of optimizing the CNN computationally, but it also destroys all information that may be encoded in the pixel amplitude histogram (often called a probability distribution function (PDF) in the ISM literature). For a turbulent density field this PDF should be close to lognormal (Vazquez-Semadeni 1994; Padoan et al. 1997; Scalo et al. 1998; Federrath et al. 2008; Burkhart et al. 2009). The procedure is diagrammed in Figure 1; example images are shown in Figure 2.

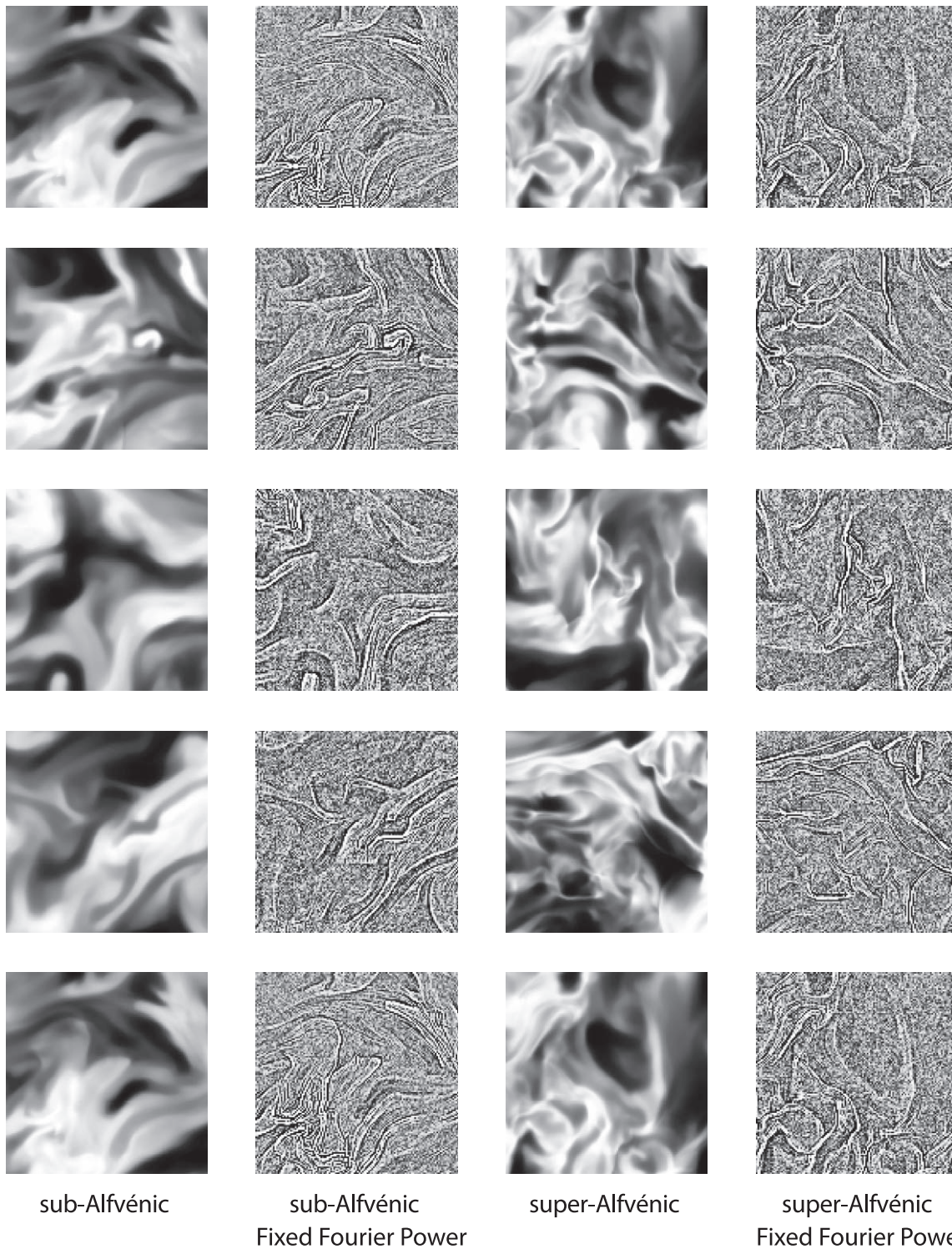
We use a relatively simple CNN, based on an example CNN architecture provided with the Keras package (Chollet et al. 2015), for distinguishing the MNIST set of handwritten digits (LeCun et al. 1998). In this architecture pairs of deep convolutional layers are alternated with pooling layers. Dropout, a commonly used method that leaves randomly selected neurons out of the training process, is applied to the pooling layers to reduce overfitting (Srivastava et al. 2014). This sequence is repeated three times until resultant 9216 neurons can be connected in a dense layer to 512 neurons (see Figure 3). Dropout is also applied to this layer, which leads to the final two classes of neurons: one neuron for sub-Alfvénic turbulence, one for super-Alfvénic turbulence. Overall there are 4858,978 trainable weights in the network. Neuron activation functions were rectified linear units,  $f(x) = \max(0, x)$ , except the final layer, which was softmax (i.e., a normalized exponential function):

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^2 e^{z_j}}. \quad (3)$$

Loss was computed by categorical cross-entropy and adadelta (Zeiler 2012) was used for optimization. Adadelta is an optimizer that adjusts learning rates adaptively based on the updates to network gradients. At present it is a typical choice for a network optimizer. The network was trained using batch sizes of 128 images with 12 total epochs (i.e., runs through the full training set).

Deep learning protocols typically reserve a portion of the data as “validation” data, and a portion of data as “test” data in addition to the training data. The validation data is used to determine how well the network can perform against data that it has not seen before. Because a network has many hyperparameters—many layers, many optimizer choices, many activation options, etc.—“test” data are also set aside to remediate the effects overfitting through so-called hyperparameter tuning. In this work we did not perform any hyperparameter tuning. The network was modified from the original example network only to make it capable of handling larger  $128 \times 128$  pixel images. This is because the purpose of this work is not to produce the best possible network for later classification purposes, but rather to determine what information exists, and where it exists, in the images of the simulation available to even a simple network. Our goal is to understand whether and how the CNN is picking up information between two classes of simulations, not to perfect the network. Thus, in this work we have only set aside one set of data, which we call the “test” data.

We further explored these data using a method called “saliency maps,” developed in Simonyan et al. (2013). Simply put, the saliency associated with each pixel in each image is the derivative of the (correct) classification with respect to the pixel amplitude. The purpose of the method is to determine which parts of each image contributes most to its classification, thus allowing us to build intuition about what features in an image



**Figure 2.** Images of example slices. The left two columns are drawn from simulations of sub-Alfvénic turbulence, and the right two columns from super-Alfvénic turbulence. All images are histogram equalized. The second and fourth column have had their power spectra flattened to fixed Fourier power (FFP). The five rows are randomly selected from the data set of 2048 test images.

contain the physical underlying information that the network is seizing upon, or whether the network is focused on spurious information. A short exploration confirmed previous work that “guided back-propagation” (Springenberg et al. 2014) gives the sharpest saliency maps, relatively free of network architecture-induced artifacts. All of the code necessary to extract the images from the simulations, normalize them, build the

networks, train the networks, test the networks, and generate figures with data can be found here: [10.5281/zenodo.2658177](https://zenodo.org/record/2658177)

#### 4. Results

The networks were trained on both the normal and FFP data sets in 15 minutes apiece on a 16 CPU compute node with 60 GB of RAM with an NVIDIA Tesla P100 GPU deployed on



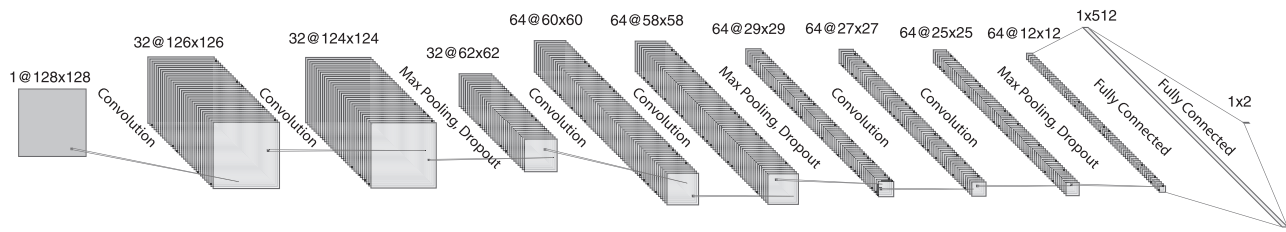


Figure 3. Architecture of the neural network used.

the *Google Compute Engine*. The final test accuracy of the network trained on normal images was 97.94%, and the final test accuracy of the network trained on FFP images was 99.50%. We also performed a secondary test, using training data from only one cube and test data from the other. The code continued to perform well, even with half the training data and achieved a final test accuracy of 95.25%.

While we did not engage in a hyperparameter space search, we did find interesting results regarding the networks and training of the CNN. We found that FFP data took much longer to train, and spent many training epochs with very little decrease in loss function or increase in accuracy on the training data, before rapidly finding a much deeper minimum on the parameter space. Depending on the random seed, we find the network can take many epochs to converge trained on FFP images. We expect that this relates to the fact that these images do not have the hierarchy of scales that CNNs are designed to capture. This is to say, the large-scale (small wavenumber) power has been removed from these images, which makes the CNN less efficient. We urge caution in working with FFP data and CNNs, as we do not yet fully understand what causes the network to converge or fail. We also note that in initial exploratory tests we had degraded the image resolution significantly to speed our analysis, which resulted in significantly worse performance in both normal and FFP analyses. We expect that this is due to the fact that small-scale features are important to distinguish the super-Alfvénic and sub-Alfvénic scenarios. Both trained networks, in the model format readable by keras, can be found here: [10.7910/DVN/UKOPY](https://github.com/10.7910/DVN/UKOPY).

An example of 18 randomly selected super-Alfvénic and 18 sub-Alfvénic sub-images and their associated saliency maps are shown in Figure 4. The saliency map is shown as a single red contour over the image in grayscale. We find that the saliency map very clearly highlights the sharp, ridge-like features in the images in the super-Alfvénic case. Conversely, the saliency maps for the sub-Alfvénic case highlight dark regions on the edge of bright regions, implying that softening these sharper edges would enhance the sub-Alfvénic label confidence. These features highlight how the CNN is distinguishing between the classes of models.

In the test case that we explore here (sub-Alfvénic versus super-Alfvénic turbulence), the magnetic field produces fluctuations in the density field that can be anisotropic in the case of strong sub-Alfvénic turbulence and more isotropic for super-Alfvénic turbulence. However, because we take slices that are perpendicular to the mean magnetic field, we are sensitive only to density fluctuations produced by the local magnetic field and not the large-scale mean field anisotropy. The CNN is able to distinguish the local magnetic field fluctuations even after removal of the large-scale power and histogram information. This is a unique success for the CNN, as other statistical tools used for magnetic field studies, such as

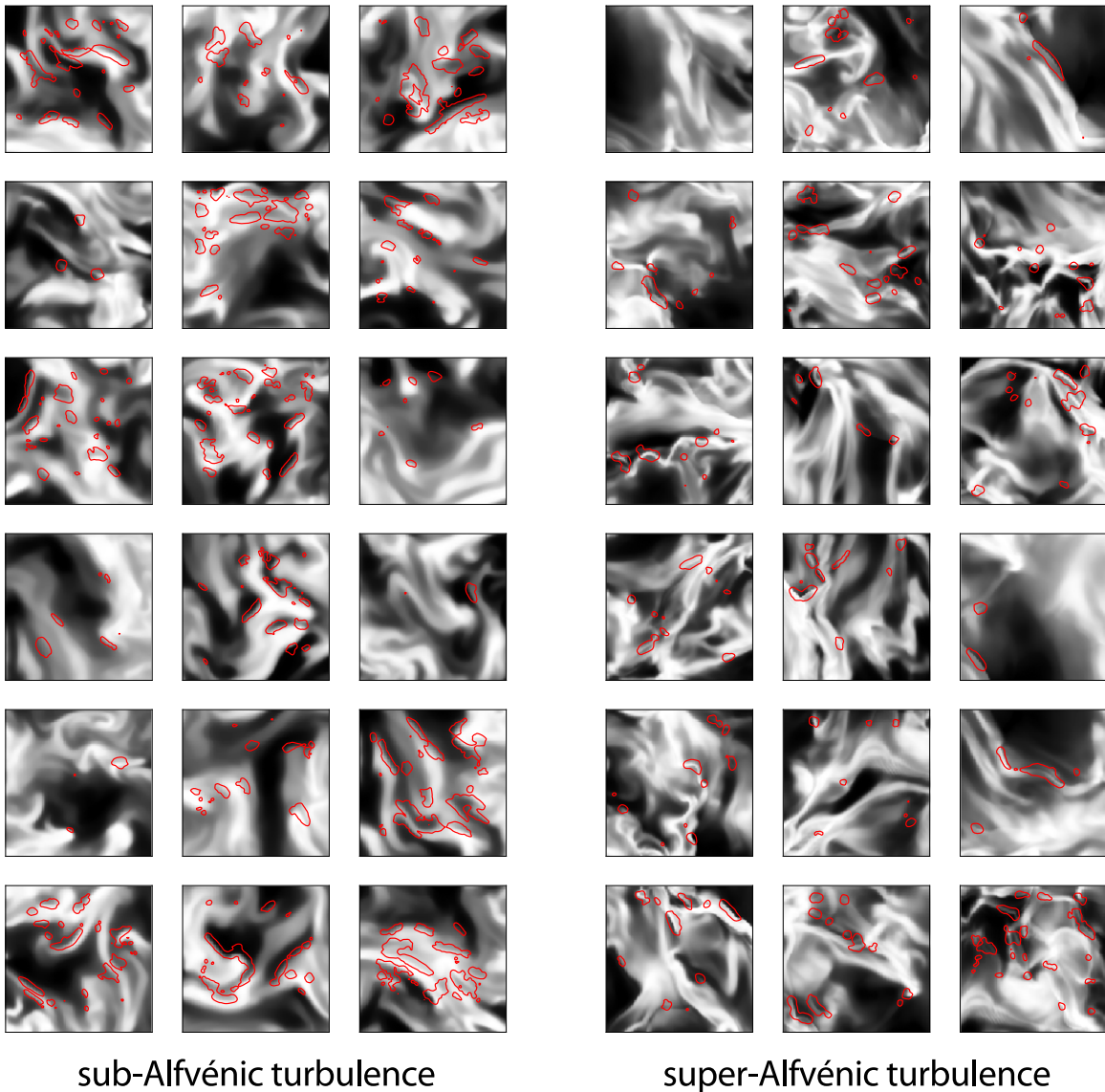
correlation functions, are only sensitive to global magnetic anisotropy (Burkhart et al. 2014).

## 5. Discussion and Conclusions

The trained networks were able to tell sub-Alfvénic versus super-Alfvénic simulations apart with  $\sim 98\%$  accuracy when presented only a single  $128 \times 128$  image,  $\sim 0.01\%$  of the entire simulation volume. These networks were presented with no velocity information, no PDF information, and, in the case of the FFP data sets, no power spectral information. The only information available was embedded in the Fourier phase domain. *This experiment has shown that there is a large amount of valuable information in Fourier phase space not available to techniques that rely on the power spectrum alone.*

These results allow two new avenues for using neural networks. Conservatively, neural networks provide a strong lower bound on the amount of information about the physical system contained in the image plane. By running a neural network we can determine that there is more information available than is being captured by a given metric if the neural network is easily able to distinguish between the data sets. This can provide encouragement to find more powerful analytic metrics. More speculatively, when trained on much more realistic simulations, one could use these CNNs to determine the underlying physical parameters of the ISM. We caution against taking this approach wholesale, without significant caveats. At present, numerical simulations are limited in their ability to reproduce observations of complex ISM structures and substructures, in terms of the physics captured, the resolution achievable, the accuracy of synthetic imaging, and our understanding of initial conditions relevant to a given ISM domain. We have shown that in some cases metrics derived from analytics have much less statistical power than CNNs, but they have the distinct advantage of being motivated by theoretical expectations from the physics of the system. Thus it is much easier to quantify in what ways these metrics may be *biased*, or what confounding factors may occur.

We recommend that CNNs and other deep and machine learning techniques trained directly on pixel-level data be used to measure the observed ISM *in parallel* with analytically based methods like power spectra. If the two methods agree on the underpinning physical parameters like driving scale, sonic Mach number, and Alfvénic Mach number, one can have much more confidence in the result. In principle, our method of scraping out information can be extended, such that the image information fed to the CNN is entirely separate from the information in the analytically based metrics. Such a scenario would give two fully independent measures of the parameters of interest. If these independent metrics gave significantly different answers, this would give a researcher pause in declaring that they had successfully measured a given



**Figure 4.** Images of 18 randomly selected slices from sub-Alfvénic turbulence simulations (left) and 18 from super-Alfvénic turbulence simulations (right). A contour of the saliency map for each image at a saliency of 200 is shown in red. On the left we see that the network would find even higher confidence in the sub-Alfvénic turbulence label if the dark regions on the boundaries of bright regions were enhanced, thus it is using the smoothness of these boundaries as a marker of sub-Alfvénic turbulence. Conversely, on the right, we see the network would be even more confident in the super-Alfvénic label if the brightness of sharp, thin density ridges were enhanced. These maps provide a way to understand how the network has learned to label these images, and where the distinguishing morphological information is in the images.

parameter without determining what systematic uncertainties or biases might be impacting one metric or the other.

Finally, our saliency analysis has shown that neural networks can be very powerful in providing *intuition* for where in the image plane phase information is most distinguishing between different classifications. In the work described here, we found that thin, ridge-like features are the main discriminator that the network uses to determine the magnetization of the simulation. In future, these kinds of experiments could allow us to develop a better understanding of how different turbulent systems manifest in the image plane. Extrapolating further, saliency can be used with any observational data as well, allowing astronomers to better understand what morphological characteristics (e.g., spiral arm shapes) correlate with what physical characteristics (e.g., star formation rate).

B.B. acknowledges the generous support of the Center for Computational Astrophysics in the Flatiron Institute. This work took part under the program The Interstellar Medium Beyond 3D and the program Milky Way-*Gaia* of the PSI2 project funded by the IDEX Paris-Saclay, ANR-11-IDEX-0003-02. We acknowledge the Deep Skies Lab as a community of multi-domain experts and collaborators who have facilitated an environment of open discussion, idea-generation, and collaboration. This community was important for the development of this project. J.P. thanks Jamie Kinney of Google for help with *Google Compute Engine*. This project made use of Numpy van der Walt et al. (2011) and Astropy (Astropy Collaboration et al. 2013; Collaboration et al. 2018).

*Software:* Numpy, astropy.

## ORCID iDs

J. E. G. Peek  <https://orcid.org/0000-0003-4797-7030>  
 Blakesley Burkhardt  <https://orcid.org/0000-0001-5817-5944>

## References

- Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from [tensorflow.org/](https://tensorflow.org/)
- Aluie, H. 2017, *NJPh*, **19**, 025008
- Armstrong, J. W., Rickett, B. J., & Spangler, S. R. 1995, *ApJ*, **443**, 209
- Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, *A&A*, **558**, A33
- Ballesteros-Paredes, J., & Hartmann, L. 2007, *RMxAA*, **43**, 123
- Bayliss, R. A., Forest, C. B., Nornberg, M. D., Spence, E. J., & Terry, P. W. 2007, *PhRvE*, **75**, 026303
- Bialy, S., Burkhardt, B., & Sternberg, A. 2017, *ApJ*, **843**, 92
- Bian, X., & Aluie, H. 2019, *PhRvL*, **122**, 135101
- Burkhardt, B., Falceta-Gonçalves, D., Kowal, G., & Lazarian, A. 2009, *ApJ*, **693**, 250
- Burkhardt, B., & Lazarian, A. 2012, *ApJL*, **755**, L19
- Burkhardt, B., & Lazarian, A. 2016, *ApJ*, **827**, 26
- Burkhardt, B., Lazarian, A., & Gaensler, B. M. 2012, *ApJ*, **749**, 145
- Burkhardt, B., Lazarian, A., Goodman, A., & Rosolowsky, E. 2013a, *ApJ*, **770**, 141
- Burkhardt, B., Lazarian, A., Leão, I. C., de Medeiros, J. R., & Esquivel, A. 2014, *ApJ*, **790**, 130
- Burkhardt, B., Lazarian, A., Ossenkopf, V., & Stutzki, J. 2013b, *ApJ*, **771**, 123
- Burkhardt, B., Lee, M. Y., Murray, C., & Stanimirovic, S. 2015, *ApJ*, **811**, 28
- Burkhardt, B., Stalpes, K., & Collins, D. C. 2017, *ApJL*, **834**, L1
- Burkhardt, B., Stanimirović, S., Lazarian, A., & Kowal, G. 2010, *ApJ*, **708**, 1204
- Burkhardt, B. K. 2014, PhD thesis, The Univ. Wisconsin—Madison
- Chen, H. H.-H., Burkhardt, B., Goodman, A., & Collins, D. C. 2018, *ApJ*, **859**, 162
- Chepurmov, A., Burkhardt, B., Lazarian, A., & Stanimirovic, S. 2015, *ApJ*, **810**, 33
- Chepurmov, A., Gordon, J., Lazarian, A., & Stanimirovic, S. 2008, *ApJ*, **688**, 1021
- Cho, J., & Lazarian, A. 2003, *MNRAS*, **345**, 325
- Cho, J., Lazarian, A., & Vishniac, E. T. 2002, *ApJ*, **564**, 291
- Chollet, F. 2015, Keras, <https://keras.io>
- Clark, P. C., Glover, S. C. O., Klessen, R. S., & Bonnell, I. A. 2012, *MNRAS*, **424**, 2599
- Clark, S. E., Peek, J. E. G., & Putman, M. E. 2014, *ApJ*, **789**, 82
- Collaboration, A., Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, *AJ*, **156**, 123
- Collins, D. C., Kritsuk, A. G., Padoan, P., et al. 2012, *ApJ*, **750**, 13
- Crutcher, R. M. 2012, *ARA&A*, **50**, 29
- Dieleman, S., Willett, K. W., & Dambre, J. 2015, *MNRAS*, **450**, 1441
- Elmegreen, B. G., & Scalo, J. 2004, *ARA&A*, **42**, 211
- Esquivel, A., & Lazarian, A. 2005, *ApJ*, **631**, 320
- Esquivel, A., & Lazarian, A. 2011, *ApJ*, **740**, 117
- Eyink, G. L., & Drivas, T. D. 2018, *PhRvX*, **8**, 011023
- Farge, M., & Schneider, K. 2015, *JPhPh*, **81**, 435810602
- Federrath, C., & Klessen, R. S. 2012, *ApJ*, **761**, 156
- Federrath, C., Klessen, R. S., & Schmidt, W. 2008, *ApJL*, **688**, L79
- Federrath, C., Klessen, R. S., & Schmidt, W. 2009, *ApJ*, **692**, 364
- Gaensler, B. M., Haverkorn, M., Burkhardt, B., et al. 2011, *Natur*, **478**, 214
- Goldreich, P., & Sridhar, S. 1995, *ApJ*, **438**, 763
- Goodman, A. A., Rosolowsky, E. W., Borkin, M. A., et al. 2009, *Natur*, **457**, 63
- Hellinger, P., Verdini, A., Landi, S., Franci, L., & Matteini, L. 2018, *ApJL*, **857**, L19
- Heyer, M., Krawczyk, C., Duval, J., & Jackson, J. M. 2009, *ApJ*, **699**, 1092
- Heyer, M. H., & Brunt, C. M. 2004, *ApJL*, **615**, L45
- Heyer, M. H., & Brunt, C. M. 2012, *MNRAS*, **420**, 1562
- Hill, A. S., Benjamin, R. A., Kowal, G., et al. 2008, *ApJ*, **686**, 363
- Hubel, D. H., & Wiesel, T. N. 1962, *The Journal of Physiology*, **160**, 106
- Kainulainen, J., Beuther, H., Henning, T., & Plume, R. 2009, *A&A*, **508**, L35
- Koch, E. W., Rosolowsky, E. W., Boyden, R. D., et al. 2019, *AJ*, **158**, 1
- Kowal, G., & Lazarian, A. 2007, *ApJL*, **666**, L69
- Kowal, G., & Lazarian, A. 2010, *ApJ*, **720**, 742
- Kowal, G., Lazarian, A., & Beresnyak, A. 2007, *ApJ*, **658**, 423
- Lazarian, A. 2006, *ApJL*, **645**, L25
- Lazarian, A. 2007, *JQSRT*, **106**, 225
- Lazarian, A., & Pogosyan, D. 2000, *ApJ*, **537**, 720
- Lazarian, A., & Pogosyan, D. 2004, *ApJ*, **616**, 943
- Lazarian, A., & Pogosyan, D. 2006, *ApJ*, **652**, 1348
- Lazarian, A., & Pogosyan, D. 2008, *ApJ*, **686**, 350
- Lazarian, A., & Vishniac, E. T. 1999, *ApJ*, **517**, 700
- Lazarian, A., Yuen, K. H., Ho, K. W., et al. 2018, *ApJ*, **865**, 46
- Le, A., Roytershteyn, V., Karimabadi, H., et al. 2018, *PhPI*, **25**, 122310
- LeCun, Y., Bengio, Y., & Hinton, G. 2015, *Natur*, **521**, 436
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, *IEEEP*, **86**, 2278
- Mac Low, M.-M., & Klessen, R. S. 2004, *RvMP*, **76**, 125
- McKee, C. F., & Ostriker, E. C. 2007, *ARA&A*, **45**, 565
- Nornberg, M. D., Spence, E. J., Kendrick, R. D., Jacobson, C. M., & Forest, C. B. 2006, *PhRvL*, **97**, 044503
- Offner, S. S. R., Clark, P. C., Hennebelle, P., et al. 2014, in *Protostars and Planets VI*, ed. H. Beuther et al. (Tucson, AZ: Univ. Arizona Press), **53**
- Oppenheim, A. V., & Lim, J. S. 1981, *IEEEP*, **69**, 529
- Padoan, P., Nordlund, A., & Jones, B. J. T. 1997, *MNRAS*, **288**, 145
- Pingel, N. M., Lee, M.-Y., Burkhardt, B., & Stanimirović, S. 2018, *ApJ*, **856**, 136
- Portillo, S. K. N., Slepian, Z., Burkhardt, B., Kahraman, S., & Finkbeiner, D. P. 2018, *ApJ*, **862**, 119
- Rosolowsky, E. W., Pineda, J. E., Foster, J. B., et al. 2008, *ApJS*, **175**, 509
- Scalo, J., Vazquez-Semadeni, E., Chappell, D., & Passot, T. 1998, *ApJ*, **504**, 835
- She, Z.-S., & Leveque, E. 1994, *PhRvL*, **72**, 336
- Simonyan, K., Vedaldi, A., & Zisserman, A. 2013, arXiv:1312.6034
- Soler, J. D., Hennebelle, P., Martin, P. G., et al. 2013, *ApJ*, **774**, 128
- Sonka, M., Hlavac, V., & Boyle, R. 2008, *Image Processing, Analysis and Machine Vision* (Berlin: Springer)
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. 2014, arXiv:1412.6806
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *Journal of Machine Learning Research*, **15**, 1929
- Stanimirovic, S., Staveley-Smith, L., Dickey, J. M., Sault, R. J., & Snowden, S. L. 1999, *MNRAS*, **302**, 417
- Stanimirović, S., Staveley-Smith, L., & Jones, P. A. 2004, *ApJ*, **604**, 176
- Tofflemire, B. M., Burkhardt, B., & Lazarian, A. 2011, *ApJ*, **736**, 60
- van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, *CSE*, **13**, 22
- van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., et al. 2014, *PeerJ*, **2**, e453
- Vazquez-Semadeni, E. 1994, *ApJ*, **423**, 681
- Vazquez-Semadeni, E., Ballesteros-Paredes, J., & Rodriguez, L. F. 1997, *ApJ*, **474**, 292
- Verdini, A., Grappin, R., Hellinger, P., Landi, S., & Müller, W. C. 2015, *ApJ*, **804**, 119
- Yan, H. 2009, *MNRAS*, **397**, 1093
- Yuen, K. H., Chen, J., Hu, Y., et al. 2018, *ApJ*, **865**, 54
- Zeiler, M. D. 2012, arXiv:1212.5701