

Exploring the interpretability of deep neural networks used for gravitational lens finding with a sensitivity probe

C. Jacobs^{a,b,*}, K. Glazebrook^{a,b}, A. K. Qin^c, T. Collett^d

^aCentre for Astrophysics and Supercomputing, Swinburne University of Technology, P.O. Box 218, Hawthorn, VIC 3122, Australia

^bARC Centre of Excellence for All Sky Astrophysics in 3 Dimensions (ASTRO 3D), Swinburne University of Technology, Hawthorn, VIC 3122, Australia

^cFaculty of Science, Engineering and Technology, Swinburne University of Technology, P.O. Box 218, Hawthorn, VIC 3122, Australia

^dInstitute of Cosmology and Gravitation, University of Portsmouth, Portsmouth PO1 3FX

Abstract

Artificial neural networks are finding increasing use in astronomy, but understanding the limitations of these models can be difficult. We utilize a statistical method, a *sensitivity probe*, designed to complement established methods for interpreting neural network behavior by quantifying the sensitivity of a model’s performance to various properties of the inputs. We apply this method to neural networks trained to classify images of galaxy-galaxy strong lenses in the Dark Energy Survey. We find that the networks are highly sensitive to color, the simulated PSF used in training, and occlusion of light from a lensed source, but are insensitive to Einstein radius, and performance degrades smoothly with source and lens magnitudes. From this we identify weaknesses in the training sets used to constrain the networks, particularly the over-sensitivity to PSF, and constrain the selection function of the lens-finder as a function of galaxy photometric magnitudes, with accuracy decreasing significantly where the *g*-band magnitude of the lens source is greater than 21.5 and the *r*-band magnitude of the lens is less than 19.

Keywords:

methods: statistical, gravitational lensing, neural networks

1. Introduction

Machine learning, the name we give to algorithms designed to learn from data and make predictions without being explicitly programmed to do so, is playing an ever-increasing role in modern astronomy (for an overview, see Fluke & Jacobs, 2020). As the volume of “Big Data” available for analysis increases rapidly (Zhang & Zhao, 2015; Kremer et al., 2017), more efficient means of extracting scientifically relevant conclusions from this data have become increasingly urgent. One machine learning algorithm in particular—the artificial neural network (ANN; Rosenblatt, 1957; Fukushima, 1980)—has proven to be successful in many applications within and without our science. The power of ANNs lies in their ability to extract task-oriented feature sets at different granularities, mapping from input to output domains in a highly non-linear fashion and learning the optimal functional form automatically from supplied data. They can be scaled to almost arbitrary size and complexity. The now ubiquitous term *deep learning* (LeCun et al., 2015; Schmidhuber, 2015) refers to deep neural networks (DNNs), ANNs with many layers, and thus many trainable parameters. Due to the advent of GPU computing and the availability of large data sets, DNNs with up to billions of trainable parameters are now routinely applied in domains such as computer vision and natural language processing (Devlin et al., 2019).

One variant of DNN, the *convolutional neural network* (LeCun et al., 1989), which is optimized to exploit the relationships

between neighbouring pixels in image data, is extremely powerful in extracting meaning from images and has revolutionized the field of computer vision (LeCun et al., 2010; Krizhevsky et al., 2012; Voulodimos et al., 2018).

The application of deep learning to astronomy continues to accelerate. Just a few examples include such disparate applications as cosmological parameter estimation (Ntampaka et al., 2020; Wang et al., 2020); gravitational wave identification (George & Huerta, 2018); galaxy morphology classification (Dieleman et al., 2015; Zhu et al., 2019; Walmsley et al., 2020); stellar classification (Sharma et al., 2019) star-galaxy separation (Kim & Brunner, 2016); and photometric redshift estimation (Hoyle, 2016; Eriksen et al., 2020). ANNs are even being used, albeit tentatively, to directly infer physical laws, for instance by Iten et al. (2020) to ‘rediscover’ the heliocentric configuration of the solar system.

One area of astronomy where DNNs have had particular success is in the discovery of strong gravitational lenses. The study of strongly lensed galaxies is key in many areas of contemporary astrophysics, including but not limited to cosmography (Bonvin et al., 2016; Birrer et al., 2019; Collett et al., 2019), dark matter studies (Oguri et al., 2014; Li et al., 2016; Birrer et al., 2017), the mass-assembly of lensing galaxies (Sonnensfeld et al., 2013), and the astrophysics of the lensed sources themselves (Jones et al., 2018; Spilker, 2019). However, at galaxy scale strong lenses are relatively rare, < 1 in 1000 galaxies (Treu, 2010), and are difficult to distinguish from the galaxy population at large, displaying no large bias in color or lumi-

*colinjacobs@swin.edu.au; corresponding author

osity that enable them to be reliably singled out from catalogs. Lenses can only be definitively identified using a combination of color, morphology—such as Einstein rings/arcs, multiple images of the background source—and spectroscopy. Automating lens finding therefore requires either a significant investment of expert human time, such as recruiting citizen scientists to examine images (Marshall et al., 2016; Sonnenfeld et al., 2020), or a technique that can make full use of the morphological information present in multi-band survey imaging. Previous attempts at automating lens-finding in surveys involved the careful construction of algorithms to detect rings and arcs (Seidel & Bartelmann, 2007; Gavazzi et al., 2014; Brault & Gavazzi, 2015); model sources as prospective lenses (Marshall et al., 2009; Chan et al., 2015); or a combination of these (Sonnenfeld et al., 2018). These methods resulted in some dozens of new discoveries. However, the extraordinary success of CNNs in computer vision generally (LeCun et al., 2010) makes them a logical choice for the next generation of lens finders. CNNs have now been successfully employed to discover new lenses in surveys such as the Canada-France-Hawaii Legacy Survey (Jacobs et al., 2017), Dark Energy Survey (Jacobs et al., 2019a,b), Hyper Suprime-Cam Subaru Strategic Program (Sonnenfeld et al., 2018) and Kilo-Degree Survey (Petrillo et al., 2019). A few thousand new confirmed lenses or high-quality lens candidates have resulted from these searches.

The scientific imperative for lens discovery is accelerating; rare lenses, such as double or triple source plane configurations (Collett & Smith, 2020) can, even individually, provide strong constraints on cosmological parameters, but are extremely rare—of order one in 10^6 - 10^9 sources. Future pipelines, such as the upcoming Legacy Survey for Space and Time (LSST; Ivezić et al., 2019) will benefit from real-time assessments of the presence of strong lensing. When a transient candidate is identified, ideally a system will be in place to instantly and reliably assess whether the host galaxy may be multiply imaged by a foreground source and thus flagged for immediate follow-up. In order for deep learning-based lens finding to drive this next wave of scientific discovery, we will need to properly understand the inefficiencies, biases and errors of our lens finders. For instance, the selection function is not clear; is there a bias in the colors, magnitudes, and other features of discovered lenses? How is the search affected by the depth of the images or the seeing?

Answering questions such as these is not straight forward. Despite their successes in this and other fields, deep neural networks have a significant drawback, namely their lack of interpretability. The mapping of inputs to outputs that occurs in a deep neural network involves many non-linear transformations parameterized by of order millions of weights, making understanding the contribution of any feature or subset of features of the input to the final determination very challenging. In particular, the behaviour of a DNN when applied to an example that lies outside the distribution used for training is undefined and unpredictable. In computer vision, some attempts to interpret DNN functioning have relied on *salicency maps*, a family of techniques that determine the most important (i.e. salient) regions of the input in making the final class determination.

However, the utility of these methods is limited, especially in a scientific/astronomical context, since it is focused on producing a spatial saliency map without probing other physical parameters that may be of interest to physicists. We further discuss the difficulties with these techniques in section 2 below.

In this work we examine two DNNs used to find strong gravitational lenses in Dark Energy Survey (DES; Dark Energy Survey Collaboration et al., 2016) imaging and attempt to answer some of these questions. These networks enabled the discovery over over 500 high-quality strong lens candidates (Jacobs et al., 2019a), and 84 at redshifts ~ 0.8 and above (Jacobs et al., 2019b). The networks were able to produce samples for human inspection of considerable purity—up to one in five of the most highly-scored galaxies were rated as probable or definite lenses. However, the completeness of the search could only be estimated, as the selection function of the DNNs could not be known.

Here we develop and employ a technique designed to probe the sensitivity of a deep neural network to various properties of the inputs. The method collects summaries of network performance across test sets, and compares these performance summaries as a function of various input parameters. We apply the method systematically to neural networks trained to find galaxy-galaxy strong lenses in survey imaging, and test the sensitivity of the networks to color, PSF, noise, the occlusion of regions of the image, and the magnitudes of lens and lensed source.

The paper is structured as follows. In section 2, we provide some more detail on the challenges of interpreting neural network outputs. In section 3, we summarize the methodology used to probe the sensitivity of our trained networks to various physical parameters. In section 4 we detail the results and discuss the implications of the insights learned for future applications. Finally we offer concluding remarks in section 5.

2. Interpreting neural networks

Despite their success in many domains, not the least in astronomy, DNNs suffer from serious interpretability problems. Describing what features a neural network is learning is not straight forward, and it is in general not possible to anticipate its failure modes, nor biases in its learned features. In scientific applications quantifying these problems is of increasing urgency.

Understanding the decisions of deep networks is a field of active research. A distinction is often made between *interpretable* and *explainable* machine learning. An interpretable model is one which allows insight into how it performs under certain circumstances, including potential failure modes. An explainable model allows for a more detailed understanding of the model internals, specifically the detailed reasons behind a particular decision. Gilpin et al. (2018) elaborate on the distinction in more detail. Here we are focused on an aid to ANN interpretability.

Several approaches exist to interpret DNN behaviour. The most direct approach involves visual inspection of the feature maps - the features the network has been trained to recognise in a given input. At the initial layer these features tend to be

too simple to provide much explanatory power; in the case of image data, for instance, vertical edges or patches of color. At later layers the features are too abstract to be interpretable visually, despite being rich in semantic meaning. Figure 1 shows an example from the computer vision domain, for a neural network trained to recognise everyday objects. At this early layer, we see the network has detected features such as edges, but at a later layers spatial information has been lost and the feature map is not interpretable. Figure 2 shows the same for networks trained on simulated gravitational lenses, for both lens and non-lens images, with example feature maps at several layers throughout the network. These feature maps contain little or no quantitative information.

With networks trained for computer vision, some of the most widely adopted alternative approaches have focused on *saliency mapping*—determining and displaying those parts of the input most crucial (salient) in determining a given output. Early approaches tried calculating the significance of individual inputs (pixels) to a class score (“Image-Specific Class Saliency Visualisation”—Simonyan et al., 2013). As the contribution of any individual pixel to the final score is small, the outputs from this method were noisy and suffered from a lack of “dynamic range” in explanatory power; at best, they indicate a rather fuzzy, general area where there is a positive gradient with regard to the correct class score (see Simonyan et al., 2013).

Another variation on the theme is Gradient-weighted Class Activation Mapping (Grad-CAM; Selvaraju et al., 2017), which examines the feature maps at the last convolutional layer in a network, the point in the network immediately before the output is flattened and all spatial information is discarded. The algorithm weights these feature maps by their contribution to the final score, then maps them back spatially to the input image to produce a saliency map. Since CNNs usually discard spatial information in favour of more feature maps at later layers, the saliency map produced suffers from correspondingly lower resolution. Other techniques develop these general ideas (Smilkov et al., 2017; Zeiler & Fergus, 2014; Springenberg et al., 2015; Binder et al., 2016; Kindermans et al., 2017), but the central concept is the same; highlighting regions of interest in an input image.

Are saliency maps useful in a scientific context? They are confined to the exploration of spatial features, and typically lack granularity even then. Figure 3 depicts, firstly, a Grad-CAM saliency map generated on a neural network trained for visual classification of photographs, applied to an image of a tiger (as per Selvaraju et al., 2017). We can see that the tiger’s face is highly salient in determining the class depicted in the image, an intuitive result. On the right, we show some saliency maps applied to images of a strong gravitational lens, activated for the “is a lens” class; it is equally intuitive that the central galaxy and Einstein ring are salient in the determination, but does not provide quantitative insight into the biases and weaknesses of the methodology. The resolution of the saliency map is low, however it’s not clear that an improvement would allow significant new insights, as it is still limited to the spatial dimension of the input only.

For the rest of this paper, we contrast saliency mapping tech-

niques with a *sensitivity probe*, eschewing spatial information for patterns derived from other known properties of the input, and demonstrate how this approach may be more useful in quantifying the biases and strengths of a neural network applied to the astrophysical problem of strong gravitational lens detection. The sensitivity probe is designed to be complementary to saliency maps, particularly as an aid to the exploration of performance with regard to known physical properties.

3. Methodology

3.1. Data sets used

3.1.1. Dark Energy Survey

The networks analysed in this work were trained to detect strong gravitational lenses in Dark Energy Survey (DES; Dark Energy Survey Collaboration et al., 2016) imaging. DES is an optical and near-infrared survey of 5000 square degrees of sky in five bands (g , r , i , z and y). In previous works (Jacobs et al., 2019b, Jacobs et al. (2019a)) we searched the DES Year 3 coadd imaging (Sevilla-Noarbe et al., 2021), and it is these images that simulated lenses are designed to emulate. This imaging has a depth of 24.33 in g (for a signal-to-noise of 10) and a pixel scale of 0.263 arcsec per pixel, with median PSFs of 1.12 arcsec FWHM in g , 0.96 in r and 0.88 in i .

3.1.2. Lens models and test set

In the DES lens search we employed convolutional neural networks trained using different training sets. The first, which we call “Network A”, was trained on simulated strong lenses generated using the LENSPOP software (Collett, 2015), composed of real images of large elliptical galaxies chosen from a catalog, combined with a synthetic lensed source (henceforth, “simulated lenses”). The negative examples comprise the images of the elliptical galaxies only with no lensed source. The second, “Network B”, was trained on the same simulated lenses, with the negative examples comprising randomly chosen non-lens sources taken from the field, including elliptical and spiral galaxies, mergers, and stars. The networks/training sets are summarized in Table 1. For Network A, the presence of the lensed source is the key feature that indicates lensing, since the training set includes large elliptical galaxies in both positive and negative examples. For network B, the presence of a large elliptical (the most likely deflector in a strong lensing system) is discriminative, but the network learns about spiral arms and other potentially confusing features that are not features of lensing. Here we contrast the sensitivity of these two training approaches to several input properties.

We use a sensitivity probe on Networks “A” and “B” as described above in section 3.2. The models each have 10 convolutional layers and two fully connected layers of 256 neurons each. Each model has over 12 million trainable parameters. The training sets used consisted of $\sim 150,000$ simulated lenses and a similar number of non-lens galaxies. The output distinguishes between the lens and non-lens classes, i.e. produces a probability that a tested image contains a gravitational lens. A

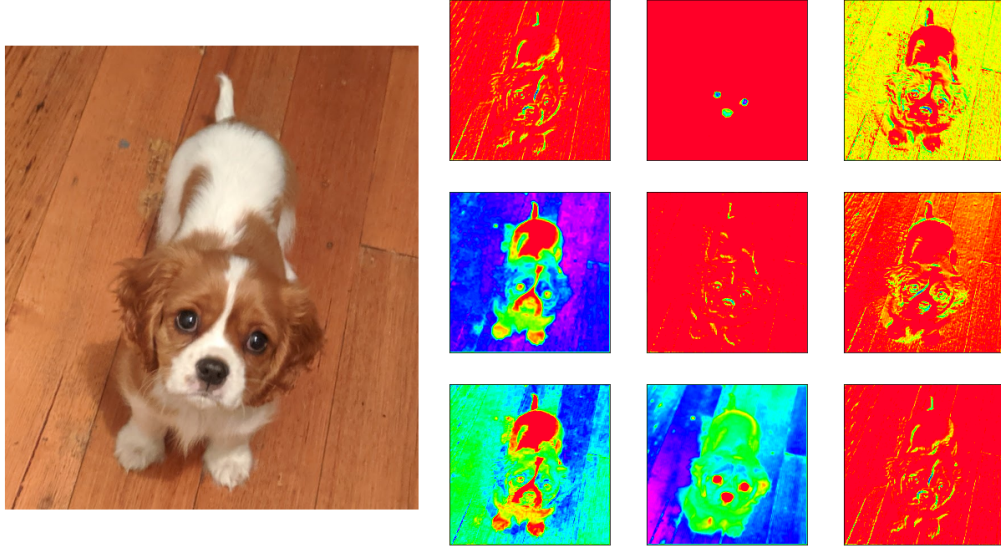


Figure 1: Examples of feature maps from a convolutional neural network. *Left*: An image supplied to a network (VGG—Simonyan & Zisserman, 2014) trained on images of everyday objects. *Right*: Nine feature maps, the result of convolving the input image with nine different convolutional kernels from the first layer of the network, showing that the network has learned to detect simple features at this level. Although these outputs are interpretable, in that we can see how features such as edges are detected by the network, it is difficult to draw a detailed understanding of the network from such examples.

Table 1: Summary of the training sets used to train the two networks (A and B), and the methodology used for positive and negative examples.

	Lenses	Non-lenses
Training set A	Elliptical galaxies with simulated lensed source added	Elliptical galaxies only
Training set B	Elliptical galaxies with simulated lensed source added	Random sources from field

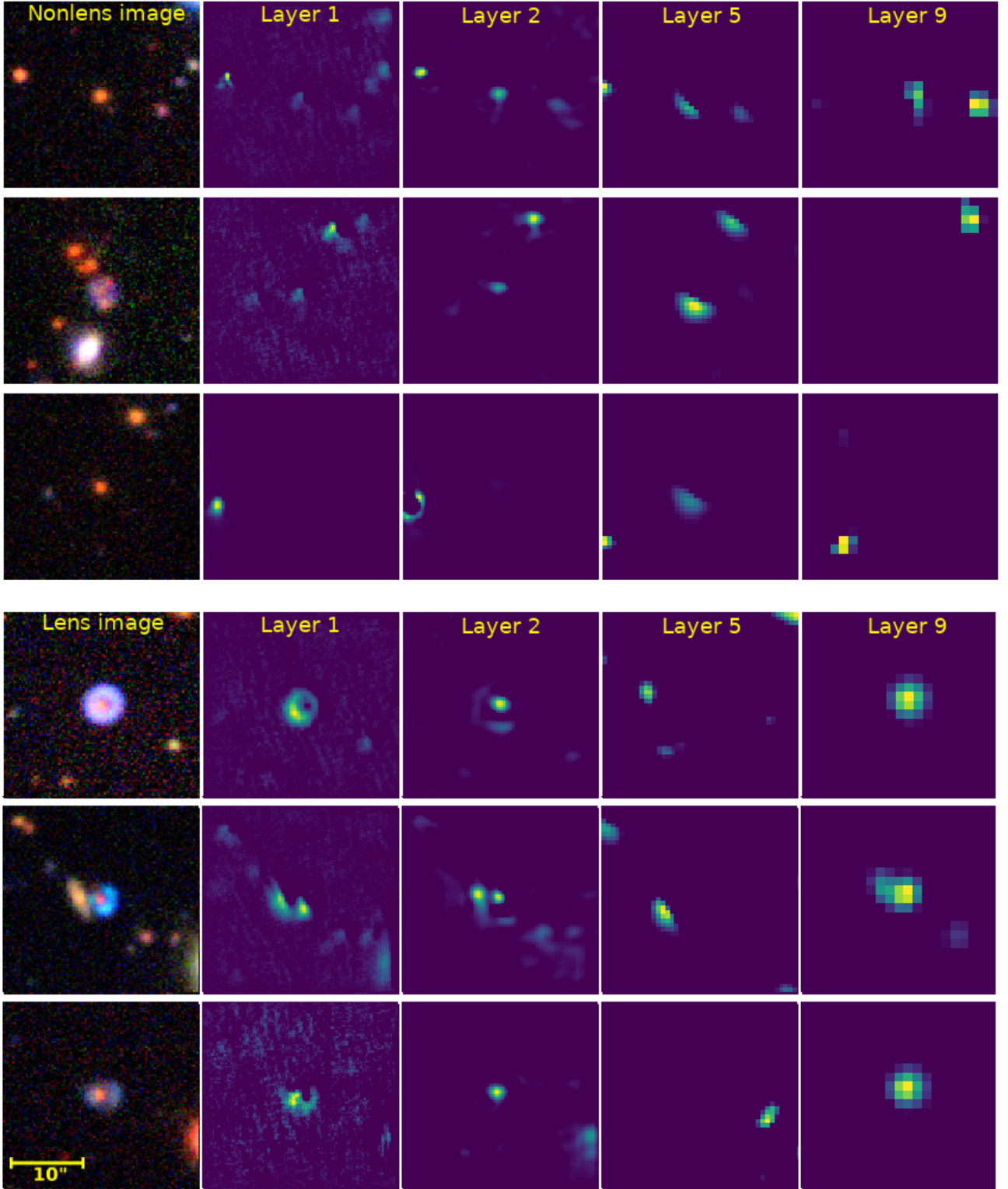


Figure 2: Examples of feature maps from a convolutional neural networks trained to detect strong gravitational lenses. *Top*: Three non-lens input images, convolved with convolutional kernels from different layers of the network as indicated. **Bottom**: Three simulated lenses and the resulting feature maps. Although the edges and colors detected by the network are visible, a quantitative understanding of network biases is difficult to extract from such maps.

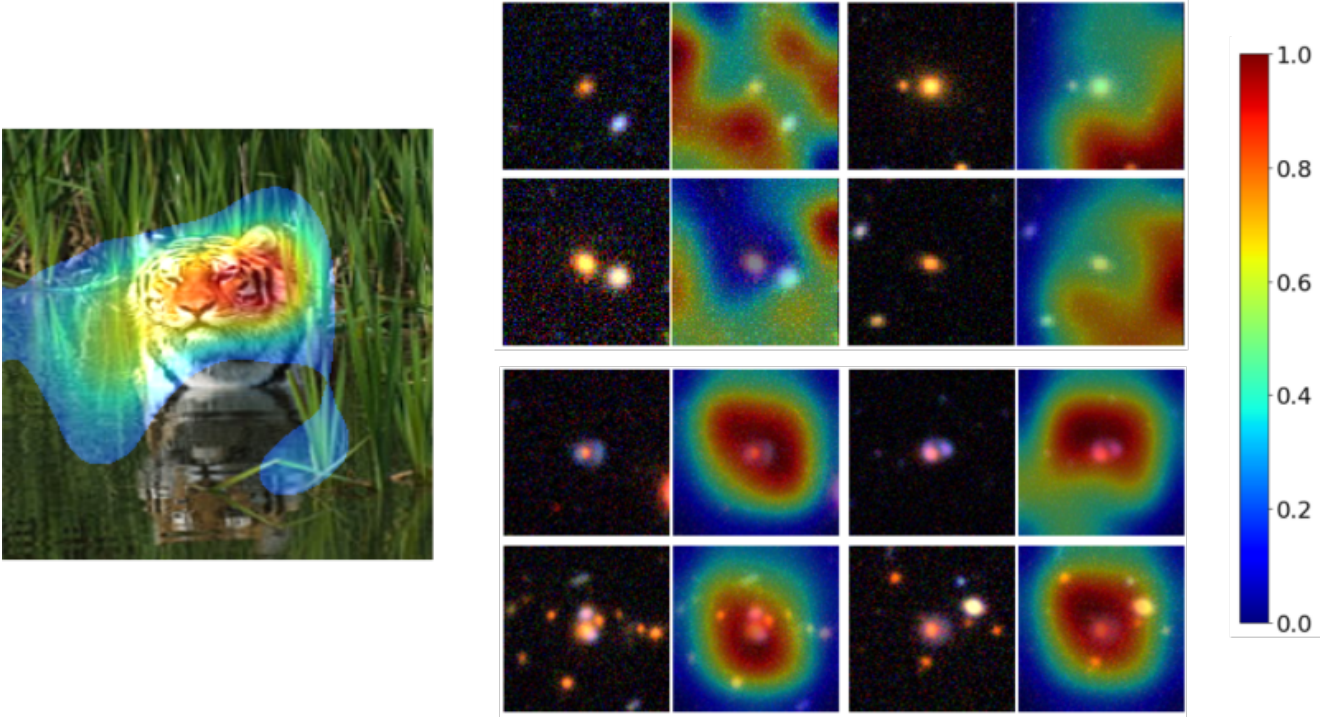


Figure 3: Examples of saliency maps generated using the Grad-CAM algorithm (Selvaraju et al., 2017). **Left:** An image of a tiger, passed to a network pre-trained on the ImageNet dataset (Russakovsky et al., 2015) with the salient region depicted—the tiger’s face. **Right:** Saliency maps for simulated lensing and non-lensing galaxies. *Top:* Four non-lensing galaxies. The saliency is distributed throughout the image. *Bottom:* Four lensing galaxies; the salient region is the central source. The colors depict the most salient regions, from blue (lowest positive saliency) to red (highest saliency).

further test set, which also includes simulated elliptical galaxies, is used in the test of PSF (section 4.2).

Except as where noted, the test set we use consists of 5000 simulated lenses, 5000 elliptical galaxies, 5000 real sources of all types, and 500 known lenses or high-quality lens candidates (human verified, and with a high spectroscopic confirmation rate) from Jacobs et al. (2019a). In all cases, we test images of dimensions 100x100 pixels, corresponding to 26.3 arcseconds on a side, in four bands (*griz*). These images were not used during the training process.

The purpose of the analysis is to better understand what the model learned and identify weaknesses in the training set that could assist in future searches. To that end, we apply the sensitivity probe, examining the models’ sensitivity to the following properties of the images (as described in detail in 4):

1. **PSF:** We generate simulations with a PSF distribution that varies from that used in the simulations used to train the network, and also degrade the image with a Gaussian blur.
2. **Einstein radius:** The Einstein radius of the simulated lens.
3. **Galaxy magnitudes:** The *g*-band source magnitude and *r*-band lens magnitude.

We also apply perturbations to the test set, to test the sensitivity to the following parameters:

1. **Noise,** by the addition of Gaussian noise;
2. **Color,** by artificially varying the colors of the images;

3. **Occlusion,** by zeroing out pixels in certain regions of the image

3.2. Sensitivity probe

The technique used to examine the neural networks in this paper we refer to as a *sensitivity probe*.¹ The key metric for the performance of a neural network classification model is whether it puts the input in the correct class. The output of a neural network trained for a classification task is a vector of dimension n , where n is the number of possible classes, and the value s^i , $i \in [1...n]$ is interpreted as a confidence that the input belongs to class i . In general the final layer includes a softmax activation such that $\sum_i s^i = 1$, so we can interpret s^i as a (pseudo-)probability. In the optimal case, the output of the network will be $s^k = 1$ where k is the index of the correct (ground truth) class, and $s^i = 0$ where $i \neq k$.

We use this correct-class score, s^k , as the test for the sensitivity probe. It is easily interpretable, as it represents the model’s confidence (in the range 0-1) that it has classified the input object in the correct class category. Other metrics would be equally valid, such as the categorical cross-entropy which is used as the loss metric to minimise during the training process. This value is directly correlated with the model accuracy, but is less easily interpreted as it is unbounded as the correct-class score approaches zero.

¹Not to be confused with parameter sensitivity analysis, which tests the sensitivity to the weights of the network itself.

We test how the correct-class score value varies across different test sets, or as the parameters of a given test set are varied in some way. If this value decreases for any given input or set of inputs, the performance of the network can be said to degrade, and vice versa. If we can track this degradation against some baseline, as the inputs are varied by some parameter, then we can obtain a quantitative understanding of how sensitive the performance of the network is to this parameter.

In summary, the algorithm employed is as follows. We divide our test set into subsets of approximately a few hundred objects, binned by the property to be investigated. Then, the score in the range (0, 1) for the correct class, lens or non-lens, is predicted by the model for each object. We calculate the mean score value μ per bin as well as the standard deviation σ in each bin. If the model performed perfectly μ and σ would be 1 and 0 respectively; in practice, μ is less than 1 and there is significant scatter in the predictions, representing both diversity in the test set and the inherent strength or weakness in the model when examining objects in a particular bin. If the performance of a model degrades by bin, this will be reflected in a lower μ (less confidence in the correct class) and higher σ (more variation in score across the test set). The sensitivity probe tests how these two quality metrics change across test sets in order to inspect how the score quality changes as a function of the binning parameter. This is a purely empirical result, summarising the performance of the network; further statistical analysis, such as performing Bayesian regression analysis of the relationship between score quality and input parameter against some prior, is possible. Here we focus on the conclusions that can be drawn from the score quality data only.

The sensitivity probe also allows investigation using a “perturber” function instead of binning by some known parameter of the object. The perturber function transforms each object in some way, for instance, by adding noise. This allows one to test the sensitivity of the model to some property that is not represented in catalog values to hand; in other words, we create a new distribution of input objects that differs from the existing distribution in some way and lets us see whether the model’s score quality is sensitive to this shift.

The detailed algorithm for these two use cases, binning by parameter or by perturbation, is described below.

For this analysis we use the SENSIE software package (Jacobs, 2020)². SENSIE, which is agnostic to the problem domain and architecture of the trained model, automates the process of calculating and plotting the accuracy of a neural network classifier controlled for an input variable or perturber function.

3.2.1. Sensitivity to class properties

The sensitivity probe determines the sensitivity of a (trained) model to some scalar parameter, p . This parameter may be some property of the inputs; for example, the g -band luminosity of a galaxy, or even the class index itself. In this case, we:

- Assemble a test set, T , of objects with a known ground truth (correct class label).

- Obtain the score given by the network for the correct (ground truth) class, $s^k \in (0, 1)$ for each example in T , by feeding it through the network.
- Bin the results by values of p ; for each bin p calculate the mean score across the N examples in the bin:

$$\langle s_p^k \rangle = \frac{1}{N} \sum_{i=0}^N s_i^k$$

as well as the corresponding standard deviation in s_p^k .

3.2.2. Sensitivity to perturbation of the input data

The sensitivity probe can test the sensitivity to a *perturbation* of the data, an arbitrary transformation applied to each example in T , parameterized by a scalar p , such that $T' = f(T, p)$ for a supplied perturbation function f . For example, we may wish to consider the sensitivity of the network to the rotation of input images; this could be parameterized by the rotation angle. In this case, the algorithm is as follows:

- Choose set of discrete values of P for testing over some range, \mathbf{P} .
- For each value in \mathbf{P} , p , obtain $T' = f(T, p)$.
- Calculate the score given by the network for the correct (ground truth) class k , $s_i^k \in (0, 1)$ for each example i in T' .
- Bin the results for each value of p and calculate the mean score $\langle s_p^k \rangle$ by bin, as well as the corresponding standard deviation of s_p^k .

In each case we have a set of discrete values or bins of some parameter p , and a measure of the performance of the network corresponding to data represented by this value, μ_p and σ_p . We may then investigate how the performance of the network varies as a function of p . This can be done through visual inspection of the results, however we can also perform linear regression on p versus μ_p to test whether the effect is significant—e.g. whether a zero gradient is consistent with the data. For an example, see section 4.3.

4. Results and discussion

4.1. Color

4.1.1. Test performed

We vary the colors of the images by applying a random “jitter” or scaling factor independently to the three bands (g , r , and i) consumed by the network. The data in each band is scaled by a value drawn from a normal distribution with a mean of 1 and a standard deviation σ between 0 and 1. For each value of σ , in increments of 0.05, we draw a scale factor for each band in image in the test set and apply the scaling (we multiply all pixels in the band by this factor), before testing the image to determine network accuracy. We thus test the sensitivity of the model to the amount of variation in the relative scaling of the three bands parameterized by σ , i.e. the fidelity of color of the image to the original simulation or observation.

The addition of this color jitter does not represent an astrophysically meaningful variation in the colours of the galaxies,

²<https://github.com/coljac/sensie>

which would be best accomplished by testing with a wide array of templates, stellar populations and/or star formation histories. The purpose of this test is simply to see whether the machine learning method is highly sensitive to the distribution of colors provided during training. A high sensitivity to the ‘correct’ colours would indicate that the model would benefit from seeing a wider, and physically motivated, distribution of galaxy colors in the training step.

4.1.2. Results

The network was sensitive to the fidelity to the original colors of the input images. Figure 4 (left) shows the sensitivity of the two networks to color, when tested on test sets consisting either of simulated lenses, non-lensing galaxies, or a mixture of both. The plots depict the mean score given by the network for the correct class (lens/non-lens), with error bars showing one standard deviation. On the left, we see the effect on the scores of simulated lenses for network A, trained on simulated lenses and elliptical galaxies, and network B, trained on simulated lenses and a diverse selection of real galaxies. The performance is similar, although network B appears slightly more robust at small values of the jitter factor. In both cases, the networks remain more than 90% confident in the lens images until the jitter factor exceeds 20%, at which point the accuracy decreases quickly. At a jitter factor higher than .2 in the case of network B and .4 in network A, the threshold of 0.5 is less than one standard deviation away from the mean score—the network is very unreliable at this point.

On non-lensing galaxies, the performance degrades similarly. The middle panel of Figure 4 shows the performance of the network A on non-lens elliptical galaxies and network B on other real field galaxies. A score of 1 in this case indicates certainty that the object is not a lens. The degradation in performance is similar to the lenses. Thus, when it comes to color information, deviating from the simulated colors used in the training set, or from the colors of real field galaxies, leads to confusion in the networks - both lenses and non-lenses become increasingly uncertain.

Figure 4 (right) shows the accuracy of the networks on a test set consisting of 50% simulated lenses and 50% non-lensing galaxies, i.e. combining the effects of the two separate test sets. The performance degrades linearly until the jitter reaches a factor of approximately 0.5, at which point the accuracy of the network has decreased from $\sim 100\%$ to $\sim 75\%$.

From the above we conclude that the network has learned that the colors of the objects in the image are determinative of lensing. Performance may be improved by introducing a wider range of colors into the simulations used for training, in case the network becomes overly sensitive to the choices made here. For instance, the colors of lensed sources in the simulations were drawn from star-forming galaxies in the COSMOS catalog (Ilbert et al., 2009), and when using simulated elliptical galaxies in some experiments, we relied on a single template for a 10-gigayear-old passive galaxy. Re-training with a wider family of galaxies (both lenses and source) may reduce the effect of this sensitivity to color and enable the discovery of more lenses with atypical colors, for instance red-red lenses.

4.2. PSF

4.2.1. Test performed

Our simulated strong lenses take an image of a real elliptical galaxy, estimate a realistic lensing mass, and simulate an image of a background source at higher redshift. This image is convolved with a PSF drawn from a distribution designed to match the survey properties and then added to the real image of the simulated galaxy. The simulations on which our model was trained use a PSF modeled as a Gaussian with a FWHM drawn from a distribution with means of 1.27, 1.08 and 0.98 arcseconds in *gri* respectively, consistent with the DES Science Verification imaging. To test the sensitivity to this parameter, that is, to determine if the trained model is more or less likely to recognize a gravitational lens when the PSF of the image matches the initial training set, we create further simulations, with synthetic lens and source galaxies, using PSFs drawn from a distribution with a mean of between ~ 0.7 and 3.3 arcsec, i.e. ~ 0.6 to 3 times that used in the training set.

For this test, we created two test sets composed of new simulations with this property, one containing simulations of the sort used to train the networks, with real galaxies as lenses and simulated sources (section 3.2); the other using both simulated sources and simulated elliptical galaxies in order to provide an additional test. This was done to create realistic images that have slightly different properties to the training set, as would occur when evaluating real galaxies with the model that have parameters not fully captured in training. The simulations were once again generated with LENSPOP.

Since the PSF only effects the simulated aspects of the images—the lensed source, and in the case of the test set described above, also a synthetic elliptical lensing galaxy—we also probe the effect of convolving the image with a 2D Gaussian kernel with a standard deviation, degrading the entire image including all foreground and background sources. We employ a Gaussian with standard deviation, σ , between 0 and 3 pixels (0-0.8 arcsec). We test how the performance of the network degrades as a function of this parameter σ . With this test, we hope to get an indication of the scale of the significant features used to determine whether a source is a strong lens or not.

Figure 12 depicts an example test set image that has been perturbed as described above with the blur, noise and occlusion perturbers.

4.2.2. Results

As expected, we find that the Gaussian blurring of the images degrades performance of the network. However, the response to the blurring is not entirely consistent across the networks. Figure 5 (left) shows the response of the networks when tested on simulated strong lenses with blur applied; this can be contrasted with the middle panel, which shows the response of the networks to simulated and real non-lenses. For both networks, increasing blur decreases the certainty of lensing, and this effect continues beyond the 50% threshold (the point at which the network would be guessing randomly between the two classes); i.e. the blurrier the image, the lower the probability of lensing assigned to the networks. The rate of this decline is different, however both networks show robust performance until the

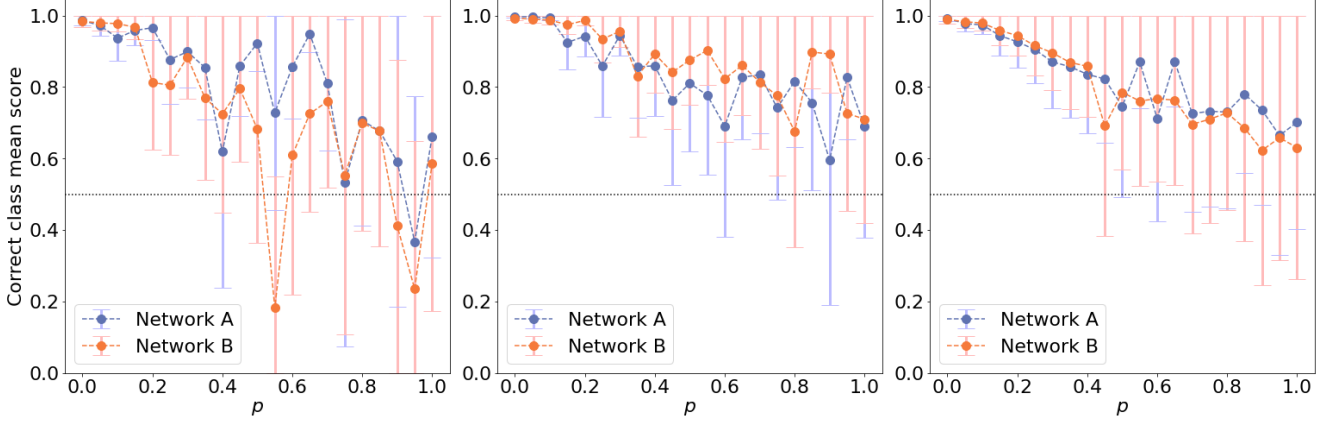


Figure 4: The effects of adding color ‘jitter’ to lens images, i.e. arbitrarily changing the scaling between bands in an image. Here the magnitude of the effect is parameterized by p , where p corresponds to the mean size of the effect, from 0 to 100% random variation. **Left:** Effect on the two networks when applied to simulated strong lenses. **Middle:** Effect on non-lens elliptical galaxies (Network A) and other non-lens real galaxies (Network B). **Right:** Combined effect on a test set containing 50% lenses and 50% non-lens images.

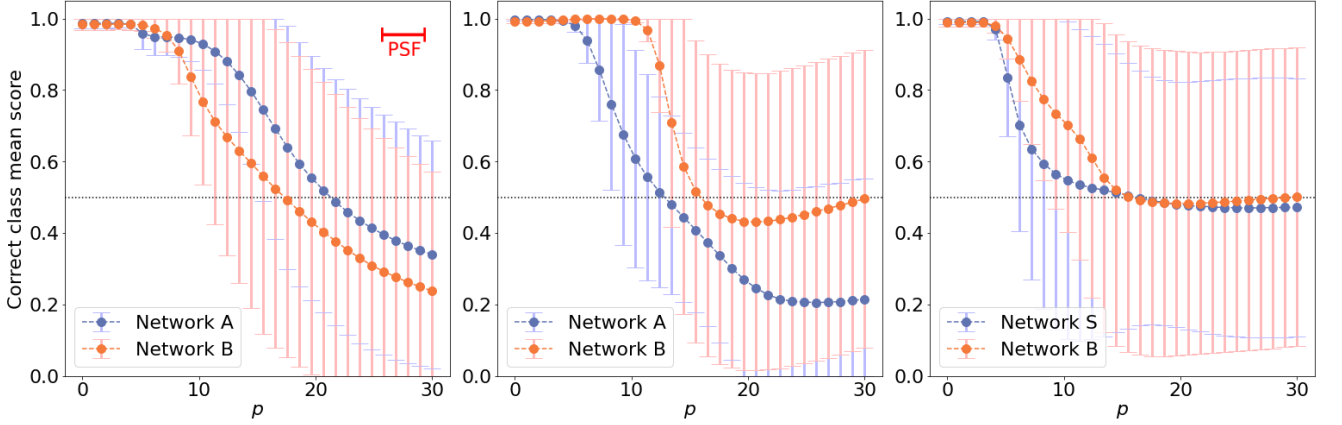


Figure 5: The effects of blurring test images by convolving with a Gaussian kernel. Here the magnitude of the effect is parameterized by p , where p corresponds to the RMS of the kernel in pixels. The mean PSF size (1.0 arcsec) is indicated on the plot for reference. **Left:** Effect on the two networks when applied to simulated strong lenses. **Middle:** Effect on non-lens images for Networks A and B. **Right:** Combined effect on a test set containing 50% lenses and 50% non-lens images.

width of the kernel reaches ~ 5 pixels, corresponding to an angular scale of 1.3 arcsec. This, presumably not coincidentally, is the approximate Einstein radius at which lensing becomes detectable in DES imaging.

In the case of the non-lenses in Figure 5 (middle panel), network B degrades to approximately 50% and remains there, however network A declines below the 50% threshold. In either case, the performance on a balanced test set containing a 50/50 split (rightmost panel of Figure 5) degrades smoothly to the expected 50% threshold.

Although the results are not surprising—applying a blur removes information from the image, and so the accuracy must necessarily decrease—one conclusion we can draw from this is that the networks have incorporated the features of the two training sets in different ways. Network A, trained only on simple simulated non-lenses as negative examples, thinks that non-lenses are more likely to be lenses the worse the resolution becomes. From this we can infer that it is quite sensitive to the properties of the simulated early-type galaxies that we used, such as the surface brightness profile, as well as the resolution of the simulated DES images. Once it is shown examples that deviate from this—even if no lensed source is introduced—it becomes increasingly confident the example is a lens. By contrast, its confidence in a blurred lens degrades more slowly. Network B, perhaps more intuitively, loses confidence in lenses as they become more distorted, but for non lenses it dips only slightly below the 0.5 “random guess” threshold.

With regards to the PSF, the results are more instructive. Figure 6 shows the response of the networks to simulated strong lenses with different PSFs and two different simulation methodologies. The first type of simulations are the same as those used to train the networks (simulated lensed sources and catalog elliptical galaxies, see Table 1); the second, labelled ‘sim2’ in the figure, use simulated elliptical galaxies. The x-axis represents a scale factor relative to the fiducial PSF used to create the simulations on which the networks were trained, a mean of approximately 1 arcsec. As the PSF gets wider, so the performance of the network degrades for both test sets, an expected result, although the effect is much larger for Network B. However, in the cases of both networks, a PSF better than the training set value also leads to decreased performance. One interpretation of this result is that the network is ‘smart’ enough to eschew lenses with unphysically sharp features. However, this also indicates a strong sensitivity to the simulated values. In the case of the lens search conducted in Jacobs et al. (2019a), the simulations used a simulated PSF designed to match that reported in the DES Science Verification data (~ 1 arcsec), but the search was conducted on Year 3 coadd images, which had a better PSF (~ 0.8 arcsec). From this experiment we conclude that several lenses may well have been missed due to this effect. This suggests a serious weakness in the training set that could be remedied firstly by better matching the PSF to the target imaging, but also by introducing a wider distribution of PSFs into the training set, forcing the network to adapt to a wider range of conditions.

4.3. Einstein Radius

4.3.1. Test performed

The Einstein radius, which is a function of the geometry of the lensing system and the lens mass, is calculated when the lenses are simulated. This data is available for simulated lenses only; for genuine lenses and lens candidates the Einstein radius is not precisely known (nor easily calculated) without high-resolution imaging. Although we can only test simulations this way, we can still obtain insights into how well the lensfinder works for less obvious lenses (smaller Einstein radius), and for those that are less represented in the training set.

We create a test set of 10000 simulated lenses with an Einstein radius $1.0 < E_r < 2.7$, a typical range for likely, detectable lenses in Dark Energy Survey DECam imaging. We bin the sims by Einstein radius, using 30 bins of 0.052 arcsec wide (approximately 300 per bin) and assign each bin’s midpoint to the simulations in that bin. These bin values are passed to the sensitivity probe.

4.3.2. Results

For this test we pass the networks 10,723 simulated lenses with known Einstein radii R_E . Figure 7 depicts the sensitivity of the networks to Einstein radius. On the left, network A and on the right, network B. In the first case, there appears to be a slight decrease in lens certainty as R_E increases; in the second, a slight increase up to 1.75 arcsec, then a decrease. The magnitude of the effect is low; in the case of network B, assuming the standard deviation as errors in the outputs, the data is consistent with a zero gradient inside a 95% credible interval. In the case of network S, the gradient, $\partial\langle s^k \rangle / \partial R_E$ is significant but small, with mean accuracy increasing at 1.2% per arcsecond across the tested interval.

Understanding the different response between the two networks may require further experimentation. Network A was only trained with images of elliptical galaxies, and so barring a few coincidental alignments in the training set—the simulated galaxies were placed in DES tiles so that interlopers, stars and artifacts were present—the presence of any arc is likely to be indicative of lensing. In the case of Network B, this is not the case, as the training set contained many thousands of spiral galaxies. Smaller Einstein radii were over-represented in the training set, whereas spiral arms could reach arbitrary size in the DES imaging. Thus, at larger radii the network becomes less certain, as the risk of confusion with a spiral arm or tidal tail is larger. In any case, this test does not reveal a deficiency or over-sensitivity in the training set for either network.

We note results from real (human inspected) lens candidates are not included here as the Einstein radius is not known and is difficult to constrain with ground-based imaging.

4.4. Lens and source magnitudes

4.4.1. Test performed

For simulated lenses, we know the (observed) magnitudes of the lens and the source. Here, we test the sensitivity of the lensfinder to the r -band magnitude of the lensing galaxy and the g -band magnitude of the lensed source. To test the sensitivity to

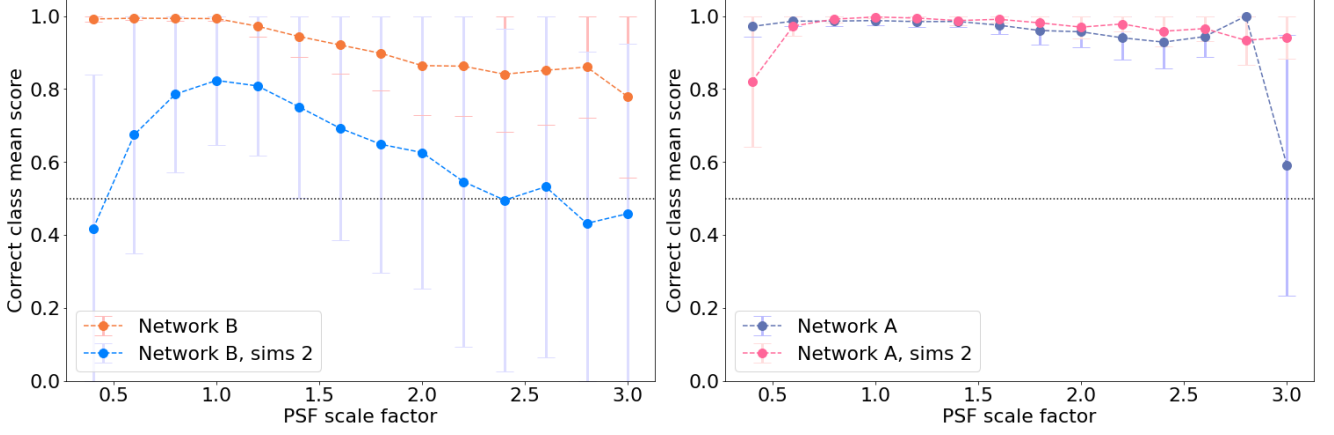


Figure 6: The effects of the PSF in simulated images on the network accuracy, Here the magnitude of the effect is parameterized by p , where p corresponds to the ratio of the test set PSF mean to the mean PSF used to train the networks originally. **Left:** Effect on the Network B, trained on simulated strong lenses and real field galaxies. **Right:** Effect on network A, trained on simulated lenses and elliptical galaxies. In both cases, the performance of the networks decreases when the PSF is *better* than the fiducial value used for training.

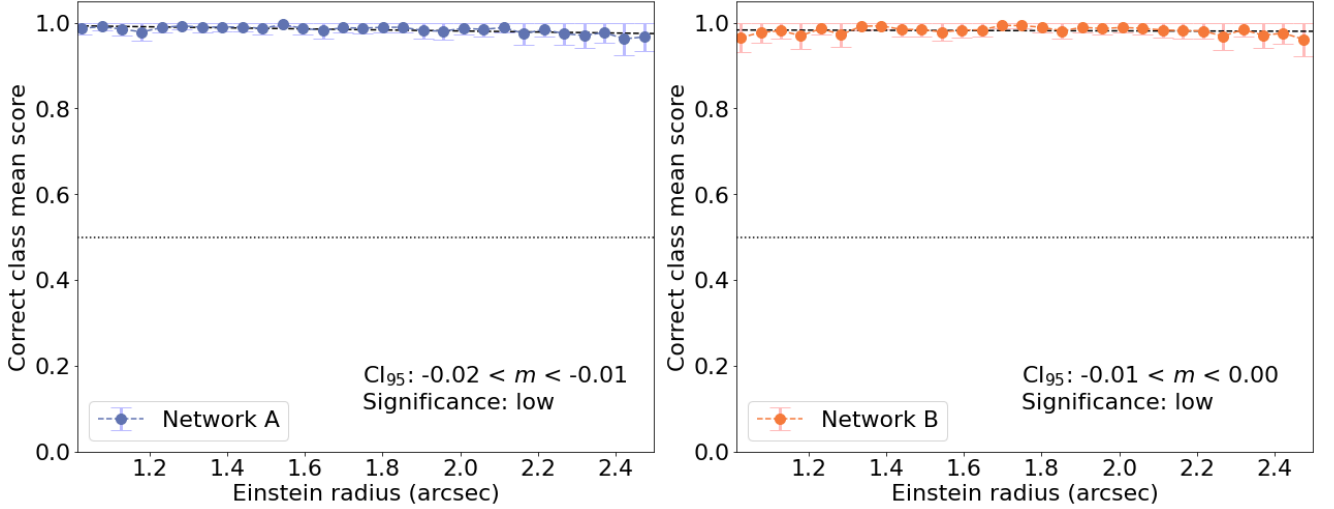


Figure 7: The effects of the simulated lens Einstein radius on network accuracy. Here we compare the mean score given by the networks to simulated lenses binned up by Einstein radius, with bins 0.052 arcsec wide/ **Left:** Effect on the Network A, trained on simulated lenses and elliptical galaxies. **Right:** Effect on network B, trained on simulated lenses and real field galaxies. Both networks display a strong insensitivity to this parameter. The 95% credible interval for a the gradient of a linear fit to the data is shown; it is consistent with zero or nearly zero in both cases.

these parameters, we collect 4000 simulated lenses where $19 < r < 21$ and 4000 simulated lenses where $20 < g_{\text{lensed}} < 22$. We create 20 bins 0.1 magnitudes wide across each magnitude domain, with each containing ~ 200 sources respectively. The midpoint of the magnitude bin is passed to the sensitivity probe as the parameter to test.

4.4.2. Results

By examining the sensitivity to the magnitudes of source and lens, we hope to gain a better understanding of the selection function of the lens-finder. Figure 8 depicts the response of the two networks to simulated lenses as a function of lens r -band and source g -band magnitude. The magnitude ranges explored reflect the parameters of detectable lenses used to train the networks.

The response of the two networks are somewhat dissimilar; network B is less sensitive across most of the range. Network A degrades in performance as the source gets fainter, with significant effects from a g -band magnitude of 21.5. Network B becomes less accurate at a similar range, although the trend is less clear. With the lens r -mag, Network A shows a significant degradation in performance where the lens is very bright, $r < 19$; this is not evident for network B. Network A also showed a degradation in performance for very faint lenses, where $r > 22.5$. Network B displays no sensitivity to bright lenses but fainter than 21 in r performance degrades quickly.

In the Jacobs et al. (2019a) search, we conducted the search by examining candidates above a certain score threshold (for instance, 0.99—very certain candidates) and then lowering the threshold in increments until the rate of discovery makes further examinations no longer worthwhile. From the sensitivity probe, we can map these thresholds to a selection function: At threshold .9, network A is unlikely to find many lenses with a g -band magnitude greater than 22. The scatter in the scores makes it difficult to establish a firm cutoff, but for a given score threshold we can establish a point where $\sim 50\%$ of lenses are unlikely to be found for any given magnitude value.

This experiment was repeated using simulations developed with a different methodology, as per section 4.2. Results of this test are presented in Figure 9. Although these simulations are less realistic than those described above, this may serve to highlight further weaknesses of the network by showing them examples dissimilar to those used in training, and thus less susceptible to over-fitting to features of the simulations. This analysis provides some evidence that of the limitations of a network trained only on one type of galaxy (bright ellipticals). The high accuracy for network A even for faint objects indicates that without having to account for the variety of morphologies and colors found in galaxies in general, the network can likely make some assumptions that will not survive contact with diverse (and lower signal-to-noise) sources from the field. In Jacobs et al. (2019a), we used network B specifically to account for weaknesses in the simulations: a network trained with the same elliptical galaxies in both the positive and negative training sets could not learn that some feature of these galaxies alone was discriminative. However, the use of network B, which was exposed to the full range of non-lens galaxy morphologies in

training, was essential to balance the unrealistic simplicity of the all-simulation training set.

4.5. Occlusion

4.5.1. Test performed

Occlusion—that is, masking out parts of an image to determine the effect on the score—has been used as a simple form of saliency mapping (see section 2). In theory, one could zero out each pixel of an image one by one and calculate the effect on the network’s score, and thereby create a map of $\partial\hat{y}/\partial p_{ij}$ over the image for each pixel p_{ij} . In practice, the effect of a single pixel on the classification is negligible, so the masking of larger regions is required to produce an interpretable map.

Here, we employ two strategies to test occlusion sensitivity. Firstly, we create annular masks with a width of 5 pixels and a radius varying between 0 and 50 pixels. An annulus is chosen since the sources, particularly strong lens systems, vary systematically with distance from the centre of the lens (lens light decreases, and lensed images appear around the Einstein radius of the system). Within the masks, the input pixels are set to zero. We therefore test the sensitivity to information in the image as a function of radius from the centre of the (real or potential) lensing galaxy.

We also test the sensitivity to occlusion by a disk of a radius between 0 and 35 pixels. As the disk grows in size and more information is removed from the images, we are able to test the sensitivity to the complete removal of information within a certain radius r . From this we can quantify the amount of lensing information present outside r ; that is, we can test how sensitive the network is to information inside r being present. We can compare this directly to the Grad-CAM saliency maps from Figure 3.

4.5.2. Results

The experiment with ring occlusion provides a quantitative measure of some of the morphological features learned by the network. In Figure 10 (right panel), we see that for both networks the effect of occlusion (at any radius) does not affect the ability to pick a non-lens galaxy. However, it has an impact on the scores of lenses: at a radius of 1.3 arcsec (so, information between 1.3 and 2.1 arcsec is missing) this fact is most pronounced, with accuracy dropping to $\sim 75\%$. This corresponds with the Einstein radii of the typical, and most common, lenses simulated and discovered in the survey. Accuracy is decreased by more than 10% across the range 0.5-1.7 arcsec. This confirms the networks have learned that the presence of lensed source flux at these radii is a key indicator of a strong lens, as expected. The lack of any uncertainty introduced into the non-lens scores when information is subtracted confirms this intuition. The fact that the effect is maximum in the expected range indicates that there is no concerning bias in the training set.

In Figure 11, left, we see how the certainty that Networks A and B have in identifying a lens drops as a function of disk occlusion radius. For network A, the mean score for lenses has dropped to .85 by a radius of .9 arcsec and .12 by 2 arcsec. If we take this information as an approximation of the lensing information content as a function of r , we see that $\sim 85\%$ of the

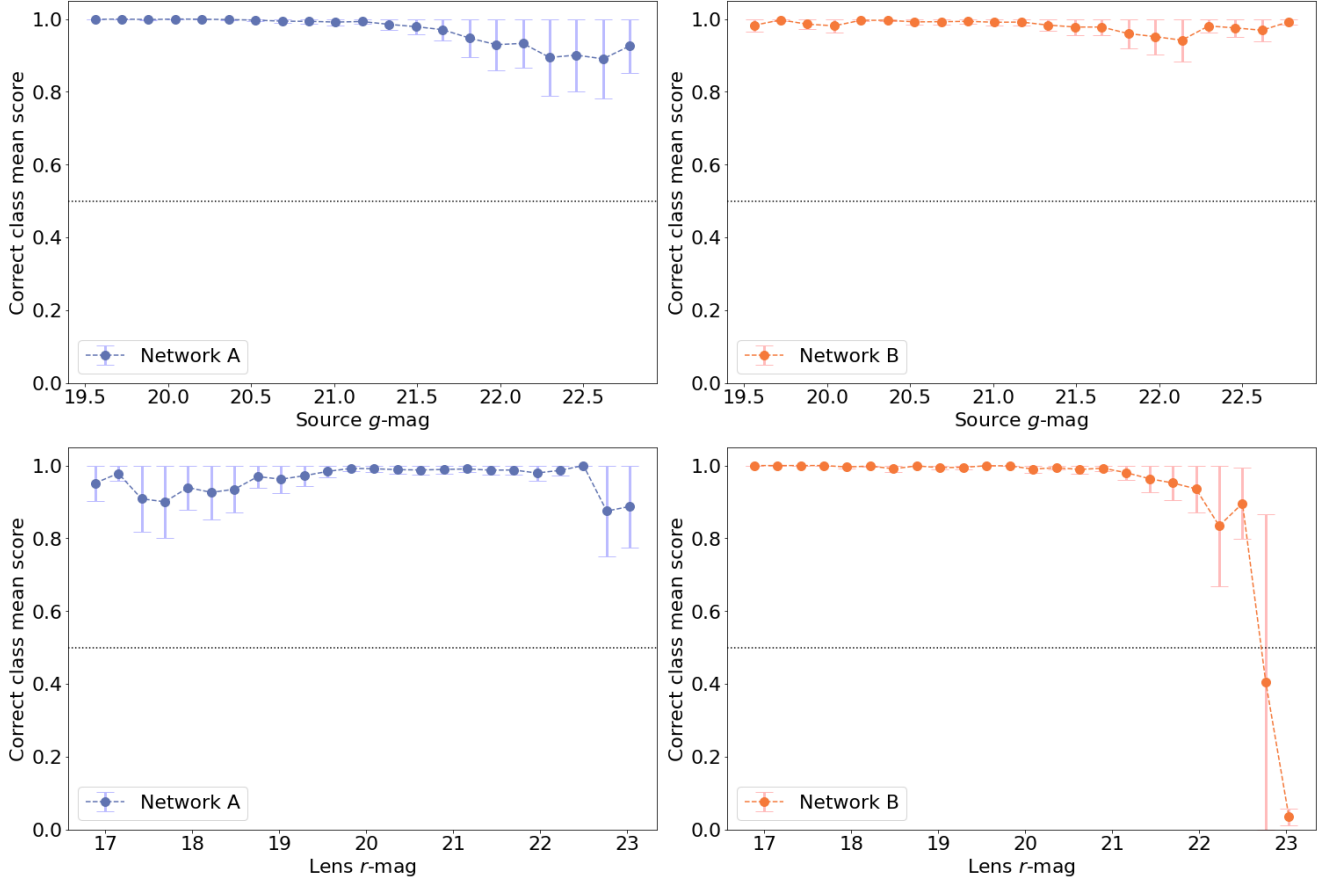


Figure 8: The accuracy of the networks in detecting simulated lenses, as a function of galaxy magnitude. **Top left:** Network A, trained on simulated lenses and elliptical galaxies, as a function of the integrated g -band magnitude of the lensed source. **Top right:** Network B, trained on simulated lenses and real field galaxies. **Bottom left:** Network A, accuracy as a function of the lens r -band magnitude. **Bottom right:** Network B, Accuracy as a function of the lens r -band magnitude.

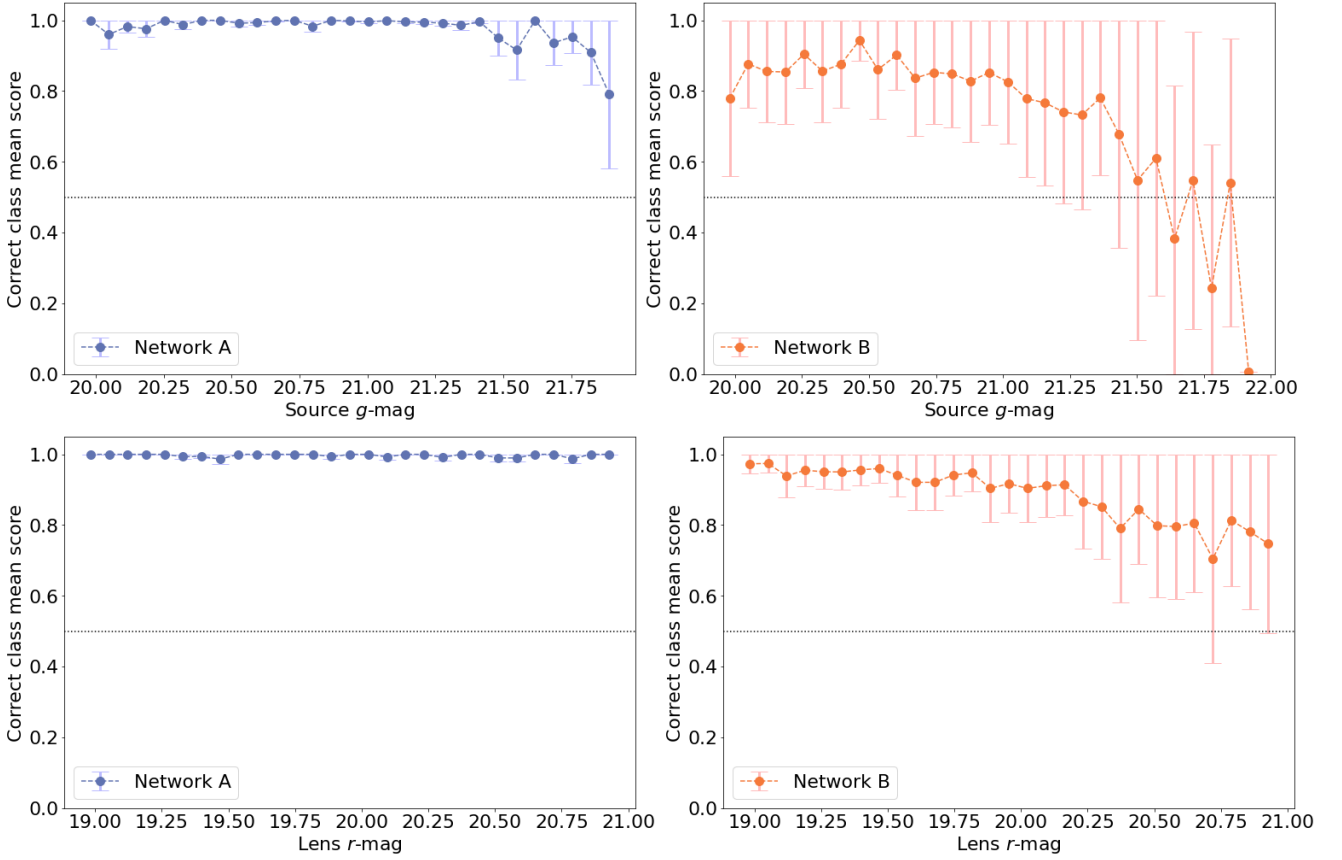


Figure 9: The accuracy of the networks in detecting simulated lenses, as a function of galaxy magnitude, for simulations containing synthetic elliptical galaxies. **Top left:** Network A, trained on simulated strong lenses, as a function of the integrated g -band magnitude of the lensed source. **Top right:** Network B, trained on simulated lenses and real field galaxies. **Bottom:** Accuracy as a function of the lens r -band magnitude. **Left:** Network A, **Right:** Network B.

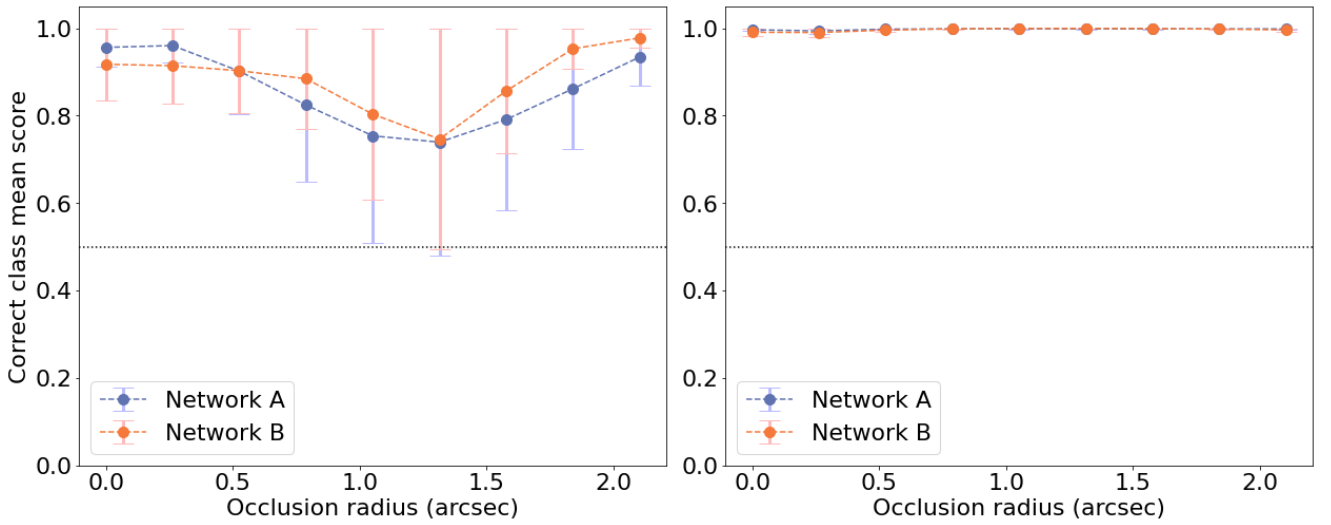


Figure 10: Effect on the network scores of the occlusion of an annulus of pixels with the given radius in arcsec (1 pixel = .263 arcsec). **Left:** Effect on the score of simulated lenses, showing the greatest effect at $\sim 1.3 - 2.1 \text{ arcsec}$, consistent with a typical detectable strong lens. **Right:** Effect on the scores of non-lenses. The networks are not sensitive to lack of radial information at any radius for a non-lens.

information used by Network A lies within a radius of 1 arcsec (over the supplied test set). We can compare this to the Grad-CAM test from Figure 3. On the right of the figure we contrast the Grad-CAM results for two lenses with this sensitivity map result. The Grad-CAM map corresponds to the individual input, while the sensitivity map result is statistical; we can nevertheless see that the information content is more tightly constrained to the central regions than the Grad-CAM maps may indicate.

For both networks, the accuracy on non-lenses is not affected by disk occlusion; removing central flux never results in a decreased certainty that an object is a non-lens.

4.6. Noise

4.6.1. Test performed

The addition of Gaussian noise enables us to probe the sensitivity of the model to signal-to-noise; what is the threshold beyond which the network can no longer reliably distinguish a lens, and how does this compare to a human expert? We interrogate this through the addition of increasing amounts of Gaussian noise to the images. The test set images are normalized such that the flux in each band to a mean of zero and a standard deviation of one, i.e. $X' = (X - \mu)/\sigma$, and then apply Gaussian noise, parameterized by a standard deviation between 0 and 30 to each pixel in an image. This corresponds to a typical change in mean signal-to-noise ratio per pixel from 20 down to ~ 0.5 across the test set. We test the sensitivity to the magnitude of this Gaussian noise.

4.6.2. Results

The addition of noise to the test set decreased the accuracy of the network as expected. However, the two networks behaved differently, shedding some light on the different features learned from the two training sets. For network A, the addition of noise leads to confusion - both simulated lenses and non lens galaxies are scored as uncertain (0.5) as the images get noisier (Figure 13, left). However, network B scores simulated lenses as more likely to be non-lenses as the noise increases and the real non-lens galaxies becomes *more* certain as noise increases. For network A, if it cannot discern lens morphology, it becomes uncertain; for network B, it becomes certain that the object is not a strong lens. For a balanced test set, the overall accuracy will be the same in both cases, but in practice the network B strategy is likely to lead to fewer false positives, albeit at the cost of completeness, in lower signal-to-noise regimes. The fact that strong lenses are very rare also points to the Network B strategy being more robust; in the absence of clear lensing features, the candidate would ideally be rejected, rather than given an uncertain score.

4.7. Effects on real images of gravitational lenses

Several of our perturber-based sensitivity tests were applied to a small test set containing images of real strong lenses. Figure 14 shows the results of applying the noise, blur and color jitter tests as described previously. The noise and blur tests are similar to the tests performed on simulations, with the interesting feature of an increase in blurred accuracy for Network A

up to a point, then the expected decrease. This does not imply that blurring the inputs would improve performance in reality, since we see that for non-lenses the blurring decreases the correct score—the false positive rate would increase dramatically in this case. Color jitter can also be seen to have a substantial impact on the networks’ accuracy; the scatter evident in the figure is a function of the small size of the test set.

4.8. Limitations of the method

The sensitivity probe is a relatively simple method and has several limitations. As presented here in the lens-finding case, the properties to be tested must be known *a priori*; the method does not, in an unsupervised way, segregate the test set in a way that allows discovery of hidden biases, for instance by automatically segregating the test set into groups of over- and under-performing objects. However, in astronomy we often do have a catalog of properties associated with the objects of interest: photometric magnitudes, flux errors, photometric and spectroscopic redshifts, colors, etc. In these cases the method is both useful and readily applied. Errors in these catalog properties will affect the sensitivity probe analysis, however due to its statistical nature it can still provide interpretable results even with noisily-labelled data. Even dividing the test set into as few as two or three bins would enable strong biases to be visualised.

Correlations between biases are also not analyzed. Since the model produces a single score for each object, the correlations in score quality for the lensing application simply reflect the correlations between galaxy and observational properties. However, segregating the test set into bins in two or more dimensions (i.e. by two or more properties) would be possible and for some applications such a ‘performance surface’ may offer useful insights.

4.9. Results summary

We tested the sensitivity of two trained neural networks to the *g*-band source magnitude, *r*-band lens magnitude, Einstein radius and PSF of simulated strong lenses, and also to the addition of noise, a Gaussian blur, and random rescaling of different bands in the images (color jitter). The networks were highly sensitive to the color jitter and PSF, showed some sensitivity to the galaxy magnitudes, and were insensitive to the simulated Einstein radius. The addition of noise and blur, which removed information from the images also caused a smooth decrease in network performance. From this we conclude in particular that the PSF is likely to be a significant weakness in the training set design, that the networks may benefit from a wider range of colors in the training sets, and that the networks are probably unreliable where the source is too faint ($g > 21.5$) or the lens too bright ($r < 19$). These results are summarized in Table 2.

The networks tested in this work were successful in facilitating the discovery of ~ 500 high-quality strong lens candidates in DES imaging. The completeness of that search is difficult to ascertain; it can be estimated using simulations (as per Collett et al., 2019) or perhaps more certainly when other techniques (such as mining spectroscopic data) are employed exhaustively to a significant subset of the survey area in future. We cannot

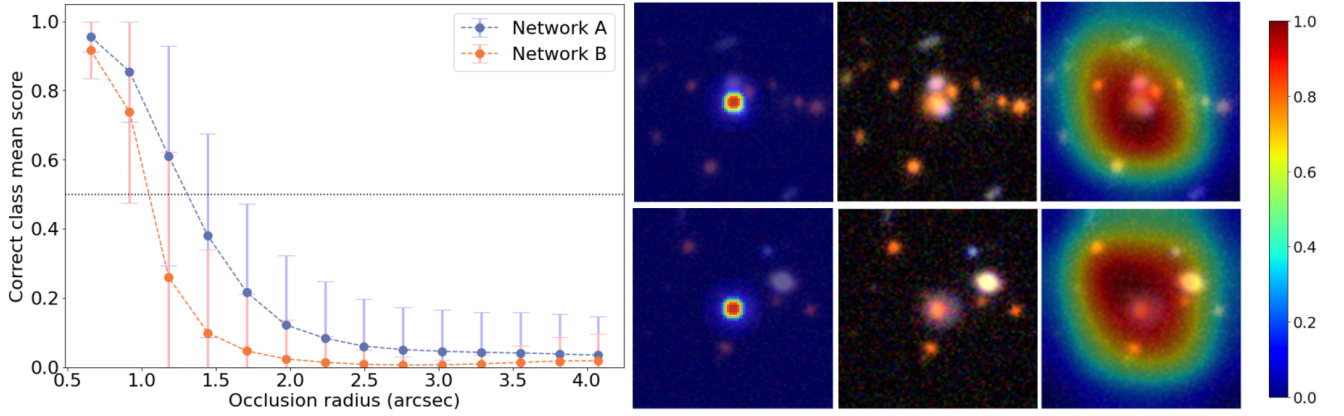


Figure 11: Effect on the network scores for simulated strong lenses of the occlusion of a disk of pixels with the given radius in arcsec. **Left:** For Network A, by radius 1.5 arcsec the majority of lenses are classified as non lenses. For Network B, the decline is faster, at just over 1 arcsec. **Right:** A comparison between the saliency maps produced using the Grad-CAM algorithm (right panel) and the importance of radial information from the disk occlusion sensitivity analysis (left panel). The color scale represents both the Grad-CAM salience and the information content by radius for Network A. Grad-CAM indicates information within several arcsec is highly salient for these two lenses; our analysis indicates that 90% of the relevant information lies within 2 arcsec for our test set.

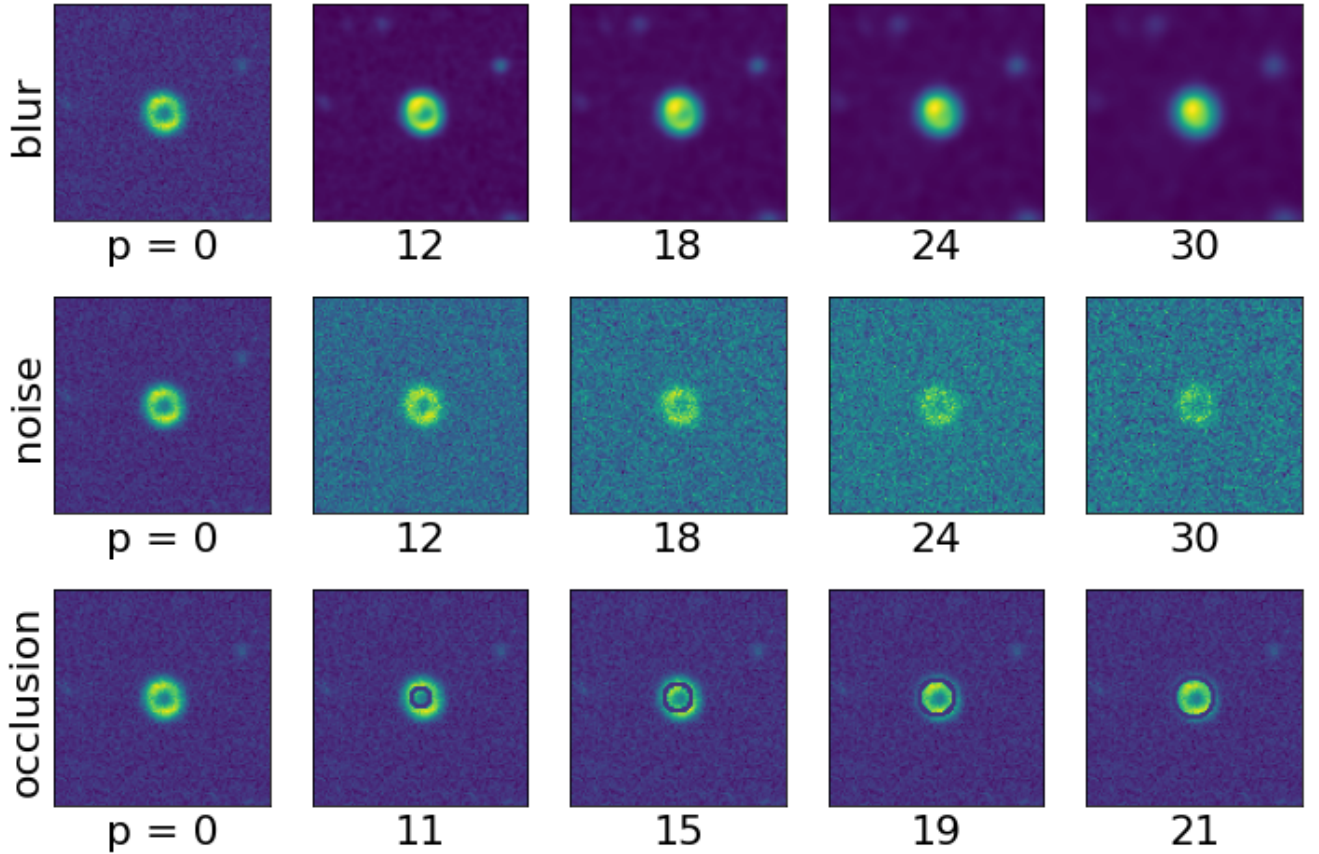


Figure 12: An example simulated lens from the test set before and after perturbation with one of the perturber functions. In order from top: Convolution with 2D-Gaussian (blur), with Gaussian width 0 to 3 pixels; addition of Gaussian noise, degrading S/N from 14 to ~ 2 ; occlusion of an annulus 5 pixels wide with radius shown (pixels). Images are 100 pixels across (~ 26 arcsec in DES).

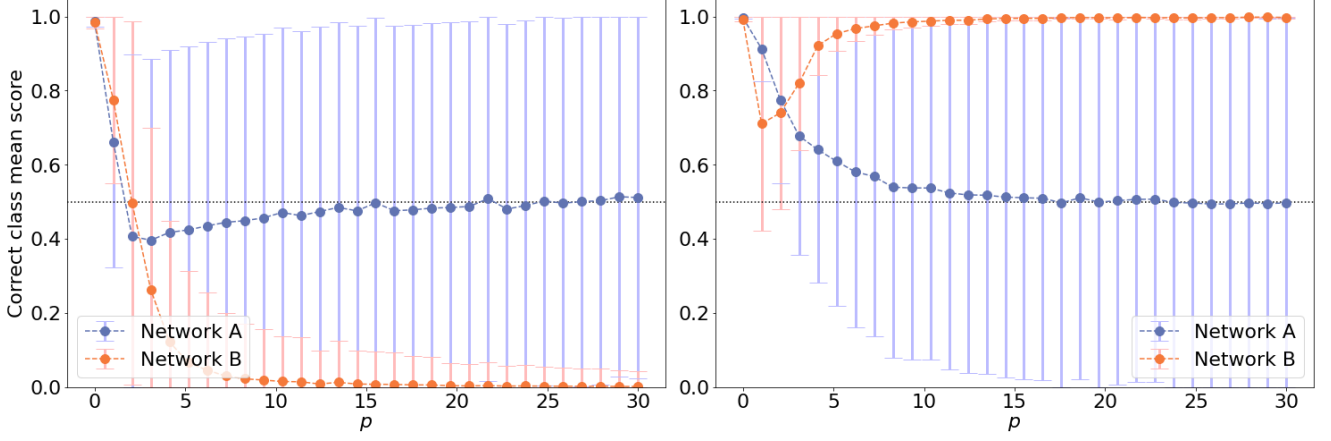


Figure 13: The effects of added noise. Here, the parameter p corresponds to the magnitude of Gaussian noise, corresponding to a typical decrease in signal-to-noise to ~ 0.5 when $p = 30$. **Left:** The effect of noise on the scores of simulated lenses. Two different strategies are evident. While Network A becomes more uncertain as noise is added, Network B becomes more certain the example is not a lens. **Right:** The same pattern is evident for the non-lenses. Network A becomes more uncertain (mean of 0.5, high scatter), whereas Network B becomes certain the examples are not lenses.

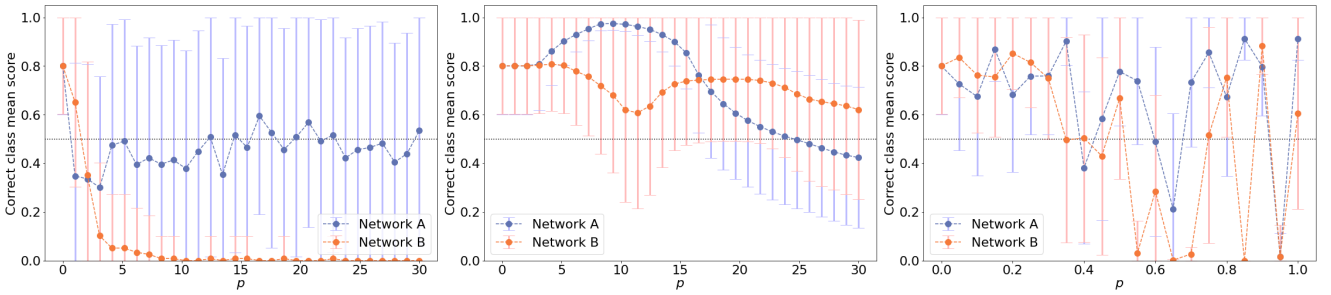


Figure 14: Effects of several sensitivity tests on the two networks using a test set consisting of 116 images of strong lens candidates from DES. **Left:** The effect of adding noise. **Middle:** The effect of convolution with a Gaussian blur. **Right:** The effect of color jitter. For the blur and noise addition, the results are consistent with what we see on the simulations, with Network A's apparent increase in performance for small blurs explained by a corresponding jump in false positives under this transformation. The effect of color jitter is strong, but due to the small size of the test set and the large amount of scatter, it is difficult to quantify beyond noting the drastic impact on accuracy.

easily quantify the properties of the sources missed or falsely rejected, nor can we easily understand in many cases why false positives are scored highly. The use of a sensitivity probe has allowed us to probe regions of parameter space where the performance of the networks degrades, putting some constraints on a selection function in terms of signal-to-noise, galaxy magnitudes, and PSF/seeing. This will enable the refinement of some estimates of the number of discoverable lenses in future surveys, and also provide a benchmark to which the next generation of lens-finders can be compared.

5. Conclusion

The use of deep neural networks in astronomy is growing exponentially, playing an increasing role in many subject areas. Understanding the biases and weaknesses of deep learning models is important to properly place the results in context and use them in downstream scientific analysis. In this article we use a sensitivity probe—which tests how the accuracy of a neural network varies as a function of a specified property of the input data—to probe two neural networks trained to recognise images of gravitational lenses in the Dark Energy Survey. We test the sensitivity of the networks to seeing (a Gaussian blur); a simulated point spread function; color; noise; occlusion of an annulus of pixels; the lens r -band magnitude; the lensed source’s g -band magnitude; and the Einstein radius of (simulated) strong lenses. We find that the networks are highly sensitive to color, indicating that they have learned that the colors of a typical strong lens (a red elliptical lens with a blue star-forming lensed source) are significant; this may indicate that an atypical lens, such as one with a red source, would be scored low. The network is also sensitive to the simulated PSF, with performance degrading smoothly as the PSF broadens from the fiducial value used to train the network, but also degrades rapidly as the PSF improves, indicating that the network is overly sensitive to the simulated PSF and should ideally be trained with a broader range of simulations to address this weakness. Response to noise, blur, and the lens and source magnitudes degrades smoothly as expected, however at a g -band magnitude of ~ 21.5 the performance is observed to degrade significantly, allowing us to constrain the selection function of the lens-finder. The network is not sensitive to the Einstein radius of an input lens image, reporting similar accuracy across a range from 1.0 to 2.7 arcsec. We tested the networks’ sensitivity to the occlusion (zeroing-out) of a ring three pixels wide; the network was most sensitive to this effect when the radius of the occlusion ring was at 5 pixels, equivalent to 1.3 arcsec. This is typical of a strong lens detected in the DES imaging and confirms that the network has learned this is a highly significant region of the image. These insights highlight potential utility of a sensitivity probe in finding weaknesses in deep learning-based algorithms and their training sets.

Future work could assist the next generation of lens-finders by quantifying the significance of assumptions underlying the simulations which for now compose the only feasible training set. The Sersic index of the simulated source, the presence of multiple lensed sources, clumpiness in the sources, and more

complicated mass distributions in the lens plane could all be tested.

Although the sensitivity probe method relies only on the collection of performance statistics calculated from the outputs of the networks, and is therefore simple to implement, it has not been systematically applied before in lens-finding or other DNN-driven astronomical applications. A sensitivity probe such as that presented here could prove useful more generally in future deep-learning based experiments by informing experimental design and identifying weaknesses in training sets that could be remedied before scientific application. The sensitivity probe also has the potential to help us understand the features a network has learned are the most salient in a particular context, from which we may potentially be able to ascribe astrophysical significance. Combined with new techniques for estimating the uncertainties in ANN outputs such as Bayesian neural networks, the chief weakness of deep learning approaches to science—the lack of interpretability—may be significantly ameliorated.

6. Acknowledgements

The author acknowledges support from Karl Glazebrook’s Australian Research Council Laureate Fellowship FL180100060. This research was supported by use of the Nectar Research Cloud and by Swinburne University of Technology. The Nectar Research Cloud is a collaborative Australian research platform supported by the NCRIS-funded Australian Research Data Commons (ARDC).

References

- Binder A., Montavon G., Lapuschkin S., Müller K.-R., Samek W., 2016, in *International Conference on Artificial Neural Networks*. Springer, pp 63–71
- Birrer S., Amara A., Refregier A., 2017, *J. Cosmol. Astropart. Phys.*, 2017, 037
- Birrer S., et al., 2019, *Monthly Notices of the Royal Astronomical Society*, 484, 4726
- Bonvin V., et al., 2016, *MNRAS*, p. stw3006
- Brault F., Gavazzi R., 2015, *Astronomy & Astrophysics*, 577, A85
- Chan J. H. H., Suyu S. H., Chiueh T., More A., Marshall P. J., Coupon J., Oguri M., Price P., 2015, *The Astrophysical Journal*, 807
- Collett T. E., 2015, *ApJ*, 811, 20
- Collett T. E., Smith R. J., 2020, arXiv:2004.00649 [astro-ph]
- Collett T., Montanari F., Räsänen S., 2019, *Physical Review Letters*, 123, 231101
- Dark Energy Survey Collaboration et al., 2016, *MNRAS*, 460, 1270
- Devlin J., Chang M.-W., Lee K., Toutanova K., 2019, in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pp 4171–4186, doi:10.18653/v1/N19-1423, <https://www.aclweb.org/anthology/N19-1423>
- Dieleman S., Willett K. W., Dambre J., 2015, *Monthly Notices of the Royal Astronomical Society*, 450, 1441
- Eriksen M., et al., 2020, arXiv:2004.07979 [astro-ph]
- Fluke C. J., Jacobs C., 2020, *WIREs Data Mining and Knowledge Discovery*, 10, e1349
- Fukushima K., 1980, *Biol. Cybernetics*, 36, 193
- Gavazzi R., Marshall P. J., Treu T., Sonnenfeld A., 2014, *ApJ*, 785, 144
- George D., Huerta E. A., 2018, *Physics Letters B*, 778, 64
- Gilpin L. H., Bau D., Yuan B. Z., Bajwa A., Specter M., Kagal L., 2018, in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. pp 80–89, doi:10.1109/DSAA.2018.00018

Table 2: A summary of the tests performed and conclusions drawn from the sensitivity probe.

Test	Parameter tested	Range	Significance	Comment
Color jitter	Band fluxes scaled to alter color	0-110%	high	Greater variation of colors may assist training robustness.
Einstein radius	Radius in arcsec	1.9-2.75	none	Network functions well across a large range.
PSF	Scaling factor from fiducial PSF	0.6-3	high	Network overly sensitive to simulated PSF, greater range indicated for training.
Occlusion	Radius of occluded annulus	0-2 arcsec	high	Network learns to rely on information at typical Einstein radius (1.3 arcsec).
Noise	Gaussian noise added	$20 \lesssim S/N \lesssim 0.5$	med	Performance degrades smoothly; different strategy between two networks.
Blur	Convolution with Gaussian kernel	$0 < \text{kernel} < 7.9''$	med	Performance degrades > 1.3 arcsec.
Lens mag	r -band magnitude of lens galaxy	21-23	low-med	Network A not sensitive across this range. Network B mean score dropped to 0.8 by $r = 20.5$.
Source mag	g -band magnitude of lens galaxy	22-24	med-high	Network A robust until $g > 21.5$; Network R until $g \sim 21$.

- Hoyle B., 2016, *Astronomy and Computing*, 16, 34
- Ilbert O., et al., 2009, *ApJ*, 690, 1236
- Iten R., Metger T., Wilming H., del Rio L., Renner R., 2020, *Phys. Rev. Lett.*, 124, 010508
- Ivezić Ž., et al., 2019, *ApJ*, 873, 111
- Jacobs C., 2020, *Journal of Open Source Software*, 5, 2180
- Jacobs C., Glazebrook K., Collett T., More A., McCarthy C., 2017, *Mon Not R Astron Soc*, 471, 167
- Jacobs C., et al., 2019a, *ApJS*, 243, 17
- Jacobs C., et al., 2019b, *Mon Not R Astron Soc*, 484, 5330
- Jones T., Stark D. P., Ellis R. S., 2018, *The Astrophysical Journal*, 863, 191
- Kim E. J., Brunner R. J., 2016, arXiv:1608.04369 [astro-ph]
- Kindermans P.-J., Schütt K. T., Alber M., Müller K.-R., Erhan D., Kim B., Dähne S., 2017, arXiv:1705.05598 [cs, stat]
- Kremer J., Stensbo-Smidt K., Gieseke F., Steenstrup Pedersen K., Igel C., 2017, preprint, 1704, arXiv:1704.04650
- Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Pereira F., Burges C. J. C., Bottou L., Weinberger K. Q., eds., *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., pp 1097–1105, <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks>
- LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, *Neural Computation*, 1, 541
- LeCun Y., Kavukcuoglu K., Farabet C., et al., 2010, in *ISCAS*. pp 253–256, http://research2.fit.edu/ice/sites/default/files/Convolutional%20networks%20and%20applications%20in%20vision_0.pdf
- LeCun Y., Bengio Y., Hinton G., 2015, *Nature*, 521, 436
- Li R., Frenk C. S., Cole S., Gao L., Bose S., Hellwing W. A., 2016, *Monthly Notices of the Royal Astronomical Society*, 460
- Marshall P. J., Hogg D. W., Moustakas L. A., Fassnacht C. D., Bradač M., Tim Schrabback Blandford R. D., 2009, *ApJ*, 694, 924
- Marshall P. J., et al., 2016, *Monthly Notices of the Royal Astronomical Society*, 455
- Ntampaka M., Eisenstein D. J., Yuan S., Garrison L. H., 2020, *ApJ*, 889, 151
- Oguri M., Rusu C. E., Falco E. E., 2014, *Monthly Notices of the Royal Astronomical Society*, 439
- Petrillo C. E., et al., 2019, *MNRAS*, 484, 3879
- Rosenblatt F., 1957, *Cornell Aeronautical Lab*
- Russakovsky O., et al., 2015, *Int J Comput Vis*, 115, 211
- Schmidhuber J., 2015, *Neural Networks*, 61, 85
- Seidel G., Bartelmann M., 2007, *Astronomy & Astrophysics*, 472, 12
- Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D., 2017, in *Proceedings of the IEEE International Conference on Computer Vision*. pp 618–626, http://openaccess.thecvf.com/content_iccv_2017/html/Selvaraju_Grad-CAM_Visual_Explanations_ICCV_2017_paper.html
- Sevilla-Noarbe I., et al., 2021, *ApJS*, 254, 24
- Sharma K., Kembhavi A., Kembhavi A., Sivarani T., Abraham S., Vaghmare K., 2019, arXiv e-prints, p. arXiv:1909.05459
- Simonyan K., Zisserman A., 2014, arXiv:1409.1556 [cs]
- Simonyan K., Vedaldi A., Zisserman A., 2013, arXiv preprint arXiv:1312.6034
- Smilkov D., Thorat N., Kim B., Viégas F., Wattenberg M., 2017, arXiv:1706.03825 [cs, stat]
- Sonnenfeld A., Treu T., Gavazzi R., Suyu S. H., Marshall P. J., Auger M. W., Nipoti C., 2013, *The Astrophysical Journal*, 777, 98
- Sonnenfeld A., et al., 2018, *Publ Astron Soc Jpn Nihon Tenmon Gakkai*, 70
- Sonnenfeld A., et al., 2020, arXiv e-prints, 2004, arXiv:2004.00634
- Spilker J., 2019, *Proceedings of the International Astronomical Union*, 15, 187
- Springenberg J., Dosovitskiy A., Brox T., Riedmiller M., 2015, in *ICLR (workshop track)*. <https://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a/>
- Treu T., 2010, *Annual Review of Astronomy and Astrophysics*, 48, 87
- Voulodimos A., Doulamis N., Doulamis A., Protopapadakis E., 2018, *Deep Learning for Computer Vision: A Brief Review*, doi:10.1155/2018/7068349, <https://www.hindawi.com/journals/cin/2018/7068349/>
- Walmsley M., et al., 2020, *Mon Not R Astron Soc*, 491, 1554
- Wang Y.-C., Xie Y.-B., Zhang T.-J., Huang H.-C., Zhang T., Liu K., 2020, arXiv e-prints, 2005, arXiv:2005.10628
- Zeiler M. D., Fergus R., 2014, in Fleet D., Pajdla T., Schiele B., Tuytelaars T., eds., *Vol. 8689, Computer Vision – ECCV 2014*. Springer International Publishing, Cham, pp 818–833, http://link.springer.com/10.1007/978-3-319-10590-1_53
- Zhang Y., Zhao Y., 2015, *Data Science Journal*, 14, 11
- Zhu X.-P., Dai J.-M., Bian C.-J., Chen Y., Chen S., Hu C., 2019, *Astrophys Space Sci*, 364, 55