Extracting the Subhalo Mass Function from Strong Lens Images with Image Segmentation

Bryan Ostdiek,[*] Ana Diaz Rivero,[†] and Cora Dvorkin[‡]
Department of Physics, Harvard University, Cambridge, MA 02138, USA

(Dated: September 16, 2020)

Detecting substructure within strongly lensed images is a promising route to shed light on the nature of dark matter. However, it is a challenging task, which traditionally requires detailed lens modeling and source reconstruction, taking weeks to analyze each system. We use machine learning to circumvent the need for lens and source modeling and develop a neural network to both locate subhalos in an image as well as determine their mass using the technique of image segmentation. The network is trained on images with a single subhalo located near the Einstein ring. Training in this way allows the network to learn the gravitational lensing of light and, remarkably, it is then able to accurately detect entire populations of substructure, even far from the Einstein ring. In images with a single subhalo and without noise, the network detects subhalos of mass $10^6 M_\odot$ 62% of the time and 78% of these detected subhalos are predicted in the correct mass bin. The detection accuracy increases for heavier masses. When random noise at the level of 1% of the mean brightness of the image is included (which is a realistic approximation for the *Hubble Space Telescope*, for sources brighter than magnitude 20), the network loses sensitivity to the low-mass subhalos; with noise, the $10^{8.5} M_\odot$ subhalos are detected 86% of the time, but the $10^8 M_\odot$ subhalos are only detected 38% of the time. The false-positive rate is around 2 false subhalos per 100 images with and without noise, coming mostly from masses $m \leq 10^8 M_\odot$. With good accuracy and a low false-positive rate, counting the number of pixels assigned to each subhalo class over multiple images allows for a measurement of the subhalo mass function (SMF). When measured over five mass bins from $10^8 M_\odot$ to $10^{10} M_\odot$ the SMF slope is recovered with an error of 14.2 (16.3)% for 10 images, and this improves to 2.1 (2.6)% for 1000 images without (with 1%) noise.

[*] bostdiek@g.harvard.edu
[†] adiazrivero@g.harvard.edu
[‡] cdvorkin@g.harvard.edu

CONTENTS

I. INTRODUCTION

The Lambda Cold Dark Matter (ΛCDM) paradigm has been successful at explaining many cosmological observations. However, on smaller scales (galactic/sub-galactic), dark matter clustering depends on particulars of the dark matter model, so even if large-scale observations are consistent with a cold dark matter particle, probing small scales can reveal an exotic dark matter scenario. For instance, in warm dark matter [1–3], self-interacting dark matter [4], or ultra-light bosonic dark matter models [5, 6], overdensities below a certain threshold do not collapse to form bound structure, which creates a low-mass cutoff of the halo mass function. For instance, warm dark matter with a mass of few keV can cutoff the halo mass function around $10^7 - 10^8 M_\odot$ with larger masses having lower cutoffs [7–10]. Therefore, searching for the low-mass dark matter halos serves as a test for the ΛCDM paradigm and can help reveal the nature of dark matter.

Unfortunately, low-mass halos are particularly hard to find. At low masses $\left(m \lesssim 10^9 M_\odot\right)$, star formation is suppressed and halos are not very luminous [11–16]. Even in the Local Group, it is challenging to detect light from these halos. Gravitational interactions may then provide the best opportunity to detect subhalos that live within larger halos. Local searches for dark matter subhalos include analyzing the effect of dark substructure passing through extended cold stellar streams [9, 17–21], careful examinations of stellar motions [22], and combining collective motions of stars and stellar weak lensing [23–25].

Outside of the Local Group, the only method to detect dark matter subhalos – so far – is strong gravitational lensing of galaxies or quasars. In order to probe low-mass subhalos, the object acting as the lens needs be galaxy sized. In galaxy-galaxy lensing, light from a distance source is deflected by a foreground galaxy. Most of the mass, and therefore the shape of the lens, comes from the dark matter halo of the galaxy. Dark matter substructure then serves as a perturbation to the main lens. The amount of perturbation depends on the mass and location of the subhalo. As an example, when placed near the Einstein ring, a subhalo with a mass of $10^6 M_\odot$ $\left(10^8 M_\odot\right)$ causes changes in the observed light of order 10^{-4} $\left(10^{-2}\right)$. The size of the changes decreases dramatically with the distance from the ring. Observing such small changes is a challenging task.

The traditional technique to detect substructure in strong lensing images as a first step involves modeling a smooth lens and the background source of light. Residuals between an image generated by forward-modeling through the best-fit smooth model and the data can then be minimized by adding substructure to improve the fit [26–30]. To date, there have been two systems found with evidence for resolved substructure, one with a mass of $(3.51 \pm 0.15) \times 10^9 M_\odot$ [31] and one with a mass of $(1.9 \pm 0.1) \times 10^8 M_\odot$ [32].

The traditional methodology relies heavily on accurate modeling; inaccuracies can lead to extra residuals and false-positives [33]. The intensive modeling required for this technique makes the method slow and computationally expensive. Additionally, alternative ways of modeling can lead to different results [34]. For instance, both of the detected subhalos were originally modelled with pseduo-Jaffe profiles [35]. However, the mass recovered when modeling instead with an NFW profile is significantly higher (the $(3.51 \pm 0.15) \times 10^9 M_\odot$ subhalo could be shifted to around $10^{10} M_\odot$ [36]). One extra caveat about these methods is that they are sensitive to the *effective* mass of the subhalo. Ref. [37] showed that the true subhalo mass can be up to an order of magnitude larger than the effective subhalo which is reconstructed.

Other strategies to learn about the nature of dark matter from strong lenses bypass identifying and characterizing individual subhalos and instead look for the collective effect of a population of low-mass halos [38–45].[1] Due to the steep low-mass end of the halo and subhalo mass functions in CDM, we expect an abundance of low-mass halos. Although most of the individual halos are not detectable individually, their collective perturbations on an image can be detected. This is desirable since dark matter theories make population-level predictions. Another advantage of statistical methods is that they can be sensitive to lower masses (*e.g.* Ref. [44] showed that, on observable scales, the convergence power spectrum is most sensitive to the much more abundant population of $10^6 - 10^8 M_\odot$ halos than it is to the much smaller population of higher-mass halos). Despite their advantages, these statistical methods still require either removing the main lens or simultaneously inferring both the mains lens and substructure.

As finding substructure has proved to be so difficult, machine learning techniques are being applied to the problem. Many uses have been found for these methods, from including better/faster modeling of the lens to measuring substructure directly. The authors of Refs. [46, 47] showed that a convolutional neural network (CNN) can be used to infer model parameters for the main lens. With the inferred model parameters, one can then look for residuals [48, 49]. Modeling the whole system (light source and main lens) was done in Ref. [50] using deep probabilistic programming combined with autoencoders.

Alternative machine learning studies have aimed at looking for substructure directly. A CNN was used for binary classification to determine if a strongly lensed image contains substructure beyond the main lens or not [51]. It did not require individually modeling the main lens as a prior step to finding substructure, and allows for quickly identifying which images dedicated modeling should be focused on. The lower-mass reach for this method is around $10^8 M_\odot$. While this is closely related to the direct detection methods above, it is also possible to develop machine learning methods to replicate the statistical searches previously discussed, meaning that they aim to uncover population-level characteristics without requiring individually resolving subhalos. Ref. [52] built a network that assumes the presence of substructure and infers both the abundance of dark matter subhalos and the slope of the subhalo mass function. Similarly, Ref. [53] uses the effects of many subhalos to infer the low-mass cutoff of the subhalo mass function. This is done as a classification problem, where the output for a given image is the mass bin where function is cut off. CNNs were trained on images with different types of dark matter substructure to learn how to distinguish them in Ref. [54].

In this work and our companion paper [55], we present results for a new method to directly detect substructure in strong gravitational lens images. This is based on a machine learning technique of object detection, where specific objects are found in an image. This is done by training on many example images which have been labeled by hand. After the network has been trained, it is able to detect objects in images it has not seen before.

We focus on the particular technique of image segmentation, where rather than putting a box around the object, each pixel is classified as belonging to a specific class. We use a U-Net [56] architecture for this task, which consists of many convolutional layers, along with a down sampling and up sampling to help the network detect features at different scales. The U-Net has emerged as one of the best architectures for image segmentation and was designed to track cells in biological images. We use it to classify each pixel in an image as belonging to one of several predetermined classes the network is trained to identify. Each pixel in our simulated images can fall into one of eleven different classes: part of the main lens, a subhalo with a mass within one of nine mass bins, or neither (background). At the pixel level, this is a classification task. However, it allows us to both locate and get the mass of substructure in the gravitational lens. As opposed to the traditional direct detection methods, the mass predicted by the network corresponds to the true (simulated) subhalo mass, not the effective mass.

An example of our image segmentation is shown in Fig. 1. The network takes in as input a lensed image as given by the left column and labels each pixel, quickly identifying both the main lens and the substructure, as shown by

[1] Statistical evidence for substructure was found in Ref. [41], allowing for a bound on the warm dark matter mass to be $m > 2$ kev.

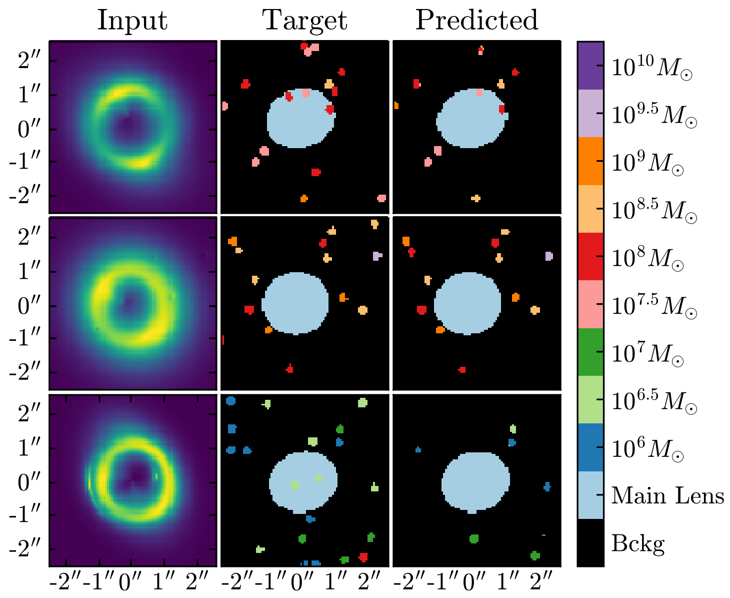

FIG. 1: Examples of segmenting gravitationally-lensed images. The left panels show simulated images which are fed into the neural network. Each pixel in an image is mapped to a label as either being the main lens within the Einstein radius, having a subhalo of a given mass, or as none of these (background). The middle panels show the true labels for the corresponding input image. The effects of only a few of the subhalos are visible by eye. The right panels show the corresponding output of our neural network. The network was trained only on images with a single subhalo, but it is still able to find substructure in images with a rich population of subhalos. Subhalos in dimmer pixels (either in the center of the ring or the edge of the image) are more likely to be missed. Subhalos that are close together sometimes get detected as a single subhalo with the combined mass. As an example, these images do not contain noise. The effect of noise is to reduce the sensitivity to low-mass subhalos.

the right panel. The mass of the subhalos is denoted by the color of the pixel. In the middle column, we have used truth knowledge of the lens to label the area within the Einstein radius of the main lens and the different subhalos. The network has good accuracy, but does struggle to detect substructure near the edges of the image. We also see that it does not appear to add in spurious subhalos. These two factors (good accuracy and low false-positive rate) allow the opportunity to learn about distribution of subhalo masses.

This paper is organized as follows. In Sec. II, we discuss our image generation pipeline. A detailed discussion of the network setup and the training regiment is contained in Sec. III. The effect of noise in the network training is explored in Sec. IV. We study the accuracy of the network and the false-positive rate in Sec. V. In Sec. VI, we apply the network to images with multiple subhalos and infer the subhalo mass function. We conclude in Sec. VII.

II. DATA GENERATION

The goal of our work is to detect dark substructure within strong lens images using the technique of image segmentation. This is a supervised learning problem, so in order to train the network, a set of training data, including the target labels, is needed. We generate strongly-lensed images using the software package Lenstronomy [57, 58]. We use images with 80×80 pixels with a field of view of $5'' \times 5''$. This corresponds to a resolution of $0.06''$ per pixel. Each image contains a background source light, a smooth lens, possibly substructure in the lens, possibly noise, and it is always convolved with a point spread function (PSF) of $0.07''$. In each image, the gravitational lens (main halo and subhalos) as well as the source light are unique. Each step in the simulation pipeline is detailed below.

Smooth Lens: The halo of the main lens is chosen as a singular isothermal ellipsoid (SIE) [59]. In Lenstronomy, the SIE is parametrized by the Einstein radius (θ_E) and the ellipticity moduli. We choose the size of the Einstein radius to be typical of observed strongly-lensed galaxy-galaxy systems, drawn from a uniform distribution

$$\theta_E \in U[0.95, 1.05]'' \; , \tag{1}$$

and an ellipticity drawn from

$$\epsilon_{1,2} \in U[0, 0.1] \ . \tag{2}$$

The center is chosen to be near the middle of the image so that the images/arcs lie within the field of view. The x and y positions are drawn randomly as

$$x, y \in U[-0.25, 0.25]'' \ . \tag{3}$$

In this work, we fix the distance to the lens at a redshift of $z_{\rm lens} = 0.2$. Our fiducial cosmology is given by *Planck* 2015 results in Ref. [60]. These lens parameters (and the location of the source light) were chosen such that the main lens has a mass of order $10^{13} M_\odot$ (depending on the specific Einstein radius in a given image).

Subhalos: When we add substructure to the lens, it is modeled as a truncated NFW profile [61] with a concentration parameter $c = 15$. The subhalos are truncated at five times the scale radius. The network is trained on images that have either zero or one subhalo. The subhalo masses are chosen to be log-uniform over the mass range $[10^{5.75} - 10^{10.25}] M_\odot$, such that we obtain equal numbers of images in each mass bin. The subhalos are placed near the Einstein radius, where their effects are largest, defined by pixels which are at least 20% as bright as the brightest pixel in the image.

In Sec. VI, we test a trained network on images with many subhalos. For these images, the masses are drawn according to a power-law given by

$$\frac{dN}{dM} = a_0 \left(\frac{M}{m_0} \right)^{\beta}, \tag{4}$$

which was found to be a good fit of the subhalo population in the Aquarius simulation [62] with power-law index $\beta = -1.9$, amplitude $a_0 = 3.26 \times 10^{-5} M_\odot^{-1}$, and pivot point of $m_0 = 2.52 \times 10^7 M_\odot$ although this specific normalization doesn't necessarily apply outside of the Aquarius simulation, the general form of Eq. (4) is universally found in N-body CDM simulations. In such simulations, the three-dimensional distribution of subhalo positions is nearly spherically symmetric, with a strong dependence on the radius. However, because their positions are projected onto a single plane, and the fact that strong lens images have a small field of view compared to the full extent of the halo perpendicular to the line of sight, a uniform distribution for subhalo positions is a good approximation (for example, see Ref. [44]). Thus, in our images with many subhalos, we will draw their locations uniformly across the whole image.

Target Labels: The target labels, which the network is trying to predict, are generated only from the smooth lens and substructure, and do not use information from the source light or the observed image. Pixels which are inside the (possibly elliptical) Einstein radius are identified as the main lens class. For each subhalo, we draw a circle with a radius of 2 pixels centered on its location and assign all the pixels within the circle as belonging to a given subhalo mass bin class. Any pixel which has not been labeled as the main lens or a subhalo is denoted as background. This method of identifying subhalos treats all subhalo masses as identical in that the more (less) massive subhalos do not get larger (smaller) circles, even though their effects are larger (smaller). We choose to do this for two reasons. The first is that it leads to more stable training of the network. When the pixel labels change size with different masses, there are many more training pixels for the heavier classes than the light classes. This creates an imbalance that would need to be corrected for, be it by weighting subhalo classes differently in the loss function or having different numbers of training images for each subhalo mass bin. The second reason is that it makes counting subhalos easier. The predicted subhalo count can be obtained by dividing the total number of pixels predicted to be part of a subhalo mass bin by the expected area per subhalo (4π pixels). Furthermore, as we will discuss extensively in Sec. VI, we are interested in extracting the subhalo mass function from an ensemble of images, for which we simply need the number of subhalos in each mass bin, and therefore do not need to faithfully reconstruct the surface mass density on the lens plane.

Source Light: This work is focused on images of lensed extended objects because they offer better chances for detecting substrucure than point source-like objects [30]. We do not use galaxy images (simulated or real), but allow for some structure in the source, placing between one and four clumps of light, as done in Ref. [51]. Each clump is modeled as a Srsic ellipse with a radius randomly drawn from a uniform distribution

$$R_{{\rm ser},i} \in U[0.1\ {\rm kpc}, 1.0\ {\rm kpc}]$$

and ellipticity

$$\epsilon_{x,i}, \epsilon_{y,i} \in U[-0.5, 0.5],$$

subject to the constraint $\sqrt{\epsilon_{x,i}^2 + \epsilon_{y,i}^2} \leq 0.4$. The location of each clump is drawn from a multivariate Gaussian with a covariance matrix with diagonal elements $\sigma_{xx}^2 = 0.01^2$, $\sigma_{yy}^2 = 0.01^2$, and off-diagonal element σ_{xy}^2 randomly chosen for each image, with a range of $[-0.25, 0.25]$. The source is placed at a distance of redshift $z_{\text{source}} = 0.6$. These parameters give images that resemble true strongly-lensed galaxies rather than 4 very far-away clumps of light that are lensed into four different images. The amplitude of the intensity of each clump is set to unity (in arbitrary units of surface brightness integrated over units of angle squared).

Noise and detector effects: We consider images both with and without noise to understand the strengths and limitations of our image segmentation-based model. When noise is added, it is drawn, per-pixel, from a Gaussian with mean $\mu = 0$. The standard deviation of the Gaussian is set to 1% of the mean brightness near the Einstein ring, unless otherwise stated. This sounds like an optimistic level of noise considering the current quality of galaxy-galaxy lens images (Ref. [34] looks for substructure in images that have a signal-to-noise ratio of at least 3). However, we show in Appendix C that our 1% Gaussian noise approximation results in similar signal-to-noise ratio profiles for sources brighter than magnitude 20 compared to a more detailed *Hubble Space Telescope* (HST)-like noise simulation. This choice also allows us to accurately quantify how the network is affected by noise. We do not include detector effects other than convolving the image with a Gaussian PSF kernel with a full-width half-maximum of $0.07''$, which is comparable to that of the HST. The PSF is applied to all images, regardless of the noise.

With the input images and the target labels, we are able to train the network. The specific model setup and the training details are described in more detail in the next section.

III. MODEL ARCHITECTURE AND TRAINING

Much of the work of the machine learning community is about object detection. For instance, a self-driving car needs to be able to identify cars, pedestrians, traffic lights, and so forth. One common method is placing boxes inside the image and then classifying what is inside the boxes, see Ref. [63] for a review. An alternative method is to classify all of the individual pixels in an image. This technique is referred to as image segmentation and a recent review of various methods can be found in Ref. [64]. While there are many models for segmenting images, the best-performing ones have some similar features. First, they are fully convolutional (in that there are no fully-connected layers). The best models also tend to have an encoder-decoder structure. This allows the models to extract features across different scales and return a high-resolution segmentation map. In particular, our network is based on the U-Net architecture [56], which has excellent sensitivity to small objects in images.

In this work, we use a U-Net to classify each pixel in a strongly lensed image into one of 11 classes. The classes are broken down as: belonging to the main lens, a subhalo with mass $\{10^6, 10^{6.5}, 10^7, 10^{7.5}, 10^8, 10^{8.5}, 10^9, 10^{9.5}, 10^{10}\} M_\odot$, or none of the above (which we will refer to as background throughout the paper). While the goal of the network is classification, it allows us to both locate subhalos and obtain their mass.

Before an image is put through the network, the image is pre-processed by dividing by the maximum pixel value. This normalization helps by forcing the brightness in all the images to have similar ranges, since despite the fact that all our source-light clumps have the same intensity, different lens parameters lead to differing amounts of magnification. Consequently, the network cannot base its classification on the absolute brightness of an image.

Once the image has been normalized, it is ready to be segmented. Our U-Net model architecture is implemented in PyTorch [65], and Fig. 2 depicts our specific set up. Each orange arrow represents three operations. The first one is a 2D convolution in which a number of 3×3 pixels filters with learnable weights are convolved with the image. The number of filters for each convolutional layer is denoted above each layer in the figure. The second operation is batch normalization [66], which normalizes the data after the convolution, leads to faster training, and helps regularize the network. The final operation represented by the orange arrows is applying the rectified linear unit (ReLU) activation function to the normalized data. This is given by

$$\text{ReLU}(x) = \begin{cases} 0, & x < 0 \\ x, & x >= 0 \end{cases} . \tag{5}$$

The convolutions in a given block are padded to preserve the number of pixels.

The green and red arrows depict the down- and up-sampling procedures, which cut the number of pixels in half and double the pixel count, respectively. The down-sampling is done with a 2×2 maximum pooling operation. The up-sampling is done with a transposed convolution operation. Note that the height and width of the data at each stage is marked at the beginning of each row in the figure. Repeating the convolutional blocks (orange arrows) on the down sampled data with the same filter size allows the network to detect features at larger scales. The up-sampling transmits the information from these other scales back to the previous scale. After the up-sampling, the layer is

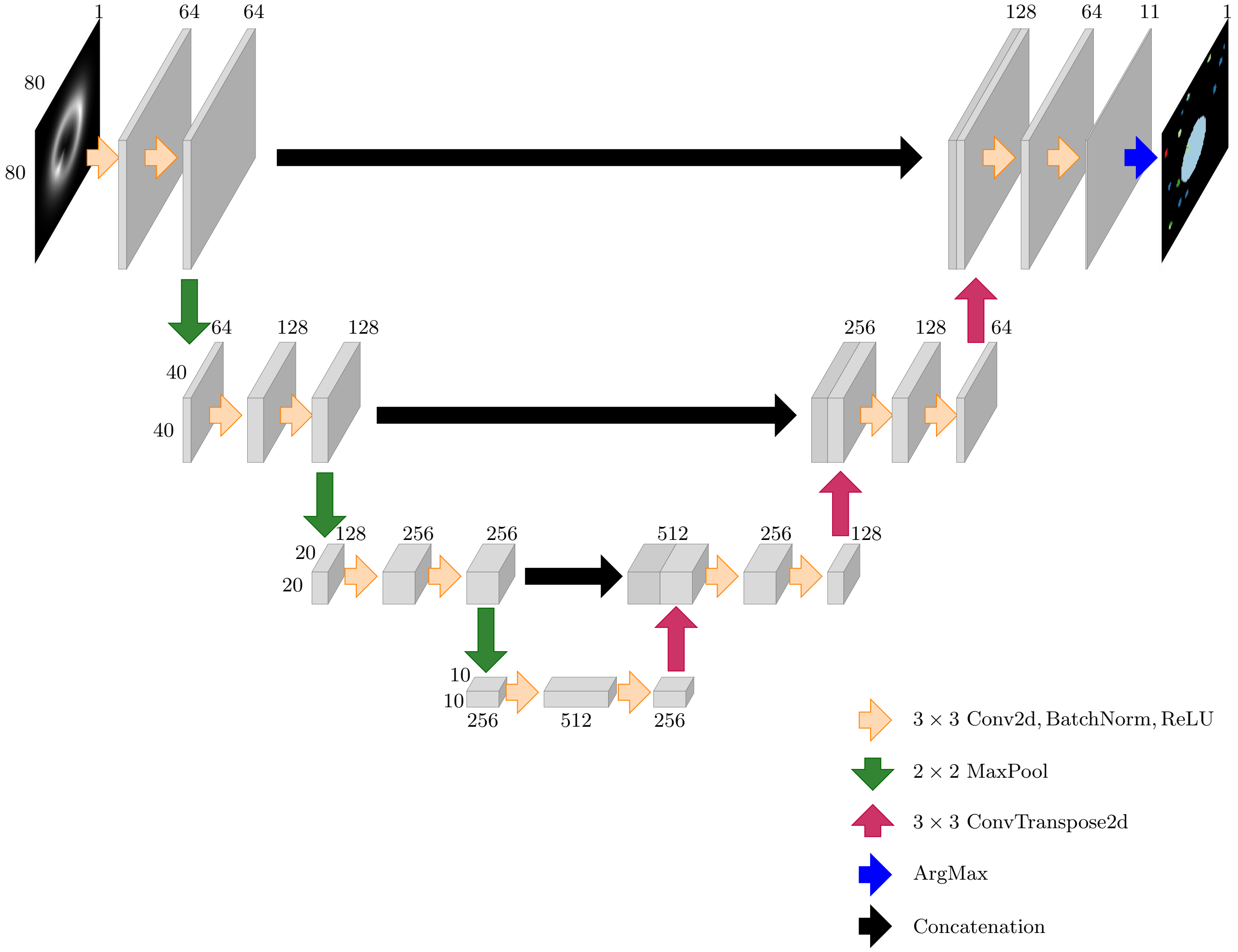

FIG. 2: Network architecture. It takes in an 80 × 80 pixels image with a single layer and returns an image of the same size. The pixel values in the output correspond to the predicted class.

concatenated with the last layer of the same height and width before down sampling (shown by the black arrows). This allows the network to localize the new features and to avoid losing pattern information.

After the last convolution, our images have a depth of 11 channels corresponding to the 11 classes. We apply the Softmax function along the channel such that the sum of a given pixel across all 11 channels is unity, and therefore its value for each channel can be thought of as a probability of belonging to the corresponding class. Explicitly, this is given by

$$\text{Softmax}\left(z_i\right) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \equiv \widehat{p}_i, \tag{6}$$

where z is the output for a pixel, the subscript denotes the pixel channel, and K is the total number of channels, 11 for the problem at hand. In this way, we interpret the channel to represent the predicted probability of belonging to a given class, denoted by $\widehat{p}_i$.

We train the network using a set of 9×10^4 images. Of these images, 9×10^3 have only the source light and a smooth lens. The remaining training images additionally contain exactly one subhalo. There are 9×10^3 images for each of the nine mass bins. We use an independent set of 10^4 images, with 10^3 from each of the sets mentioned above, to validate the model.

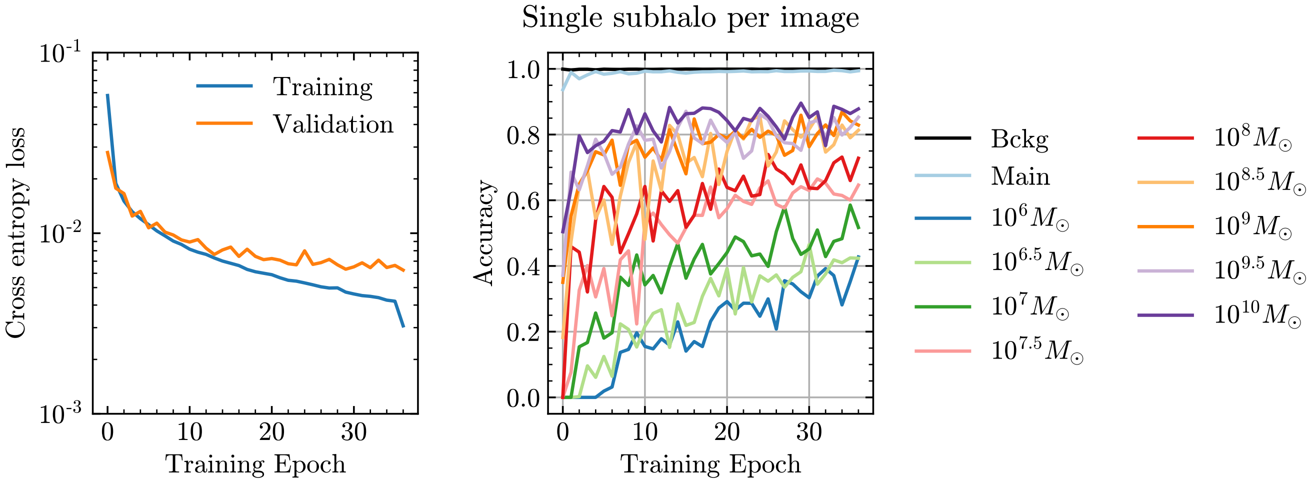

FIG. 3: Example of training on images with no noise. The left panel displays the categorical cross entropy loss as a function of the training epoch. The lowest validation loss occurs at after epoch 36, but the network is allowed to continue training until the validation loss has not improved for 15 epochs. The learned parameters from the epoch with the lowest validation loss are used when applying the network to new data. The right panel shows the per-pixel accuracy of the validation data for each of the 11 classes. The background and smooth lens pixels are predicted correctly nearly 100% of the time. The subhalos pixels are located and assigned the correct mass with accuracy between $40\% - 85\%$.

As a classification problem, the cross-entropy loss per pixel is used. This is given by

$$L = \frac{-1}{n \times p} \sum_{i=1}^{n} \sum_{j=1}^{p} \sum_{k=1}^{K} y_k^{(i,j)} \ \log\left(\widehat{p}_k^{(i,j)}\right), \tag{7}$$

where the sum over i goes over the n images, the sum over j runs over all of the p pixels in an image, the sum over k is the different possible classes, $y_k^{(i,j)}$ represents the true probability of pixel j in image i to belong to class k. As the true pixel is either in a given class or not, $y_k^{(i,j)}$ is either 0 or 1. Finally, $\widehat{p}$ is the probability predicted by the model.

We minimize the loss using the Adam optimizer [67] with a learning rate of 10^{-3} and the default β values. The batch size is set to 100 images. When the loss evaluated on the validation set has not improved for 5 epochs, the learning rate is dropped by a factor of 10, with a minimum rate of 10^{-6}. The training procedure is stopped when the validation loss has not improved for 15 epochs.

An example of the training is shown in Fig. 3 for images with no noise. The left panel shows the cross entropy loss as a function of the training epoch, where the blue and orange lines denote the training and validation sets, respectively. In addition to tracking the loss during training, we also compute the per-pixel accuracy of the validation data. We define the per-pixel accuracy as

$$\text{Pixel accuracy for class } k = \frac{\text{Number of pixels correctly predicted as class } k}{\text{Total number of truth-level class } k \text{ pixels}} \,. \tag{8}$$

For this, we define the a *correct* pixel assignment when the class with the largest probability $\left(\max_k \, \widehat{p}_k\right)$ for a given pixel matches the true assignment. There are 11 possible class assignments, so if the network is unsure of a given pixel's identity, all of the predicted probabilities could be around $1/11 \sim 9\%$. For now we still define a correct pixel assignment if the class with the largest probability matches the true class, even if the probability is low. We will examine setting a threshold on this assignment in a later section. We emphasize that the pixel accuracy for the subhalo classes requires getting both the location (the pixel) and the mass of the subhalo correct.

The pixel accuracy for each class is shown as a function of the training epoch for the validation images with no noise in the right panel of Fig. 3. In the first epochs, the model quickly learns to distinguish the main smooth lens from the background. After this, the effects of the subhalos are recognized, starting with the heaviest, which have the largest effects on the image. The accuracy for the $10^6 M_{\odot}$ subhalo pixels reaches 40% by the end of training. We note that at this stage we are only tagging pixels as belong to a subhalo (or not), but have not discussed the detection of

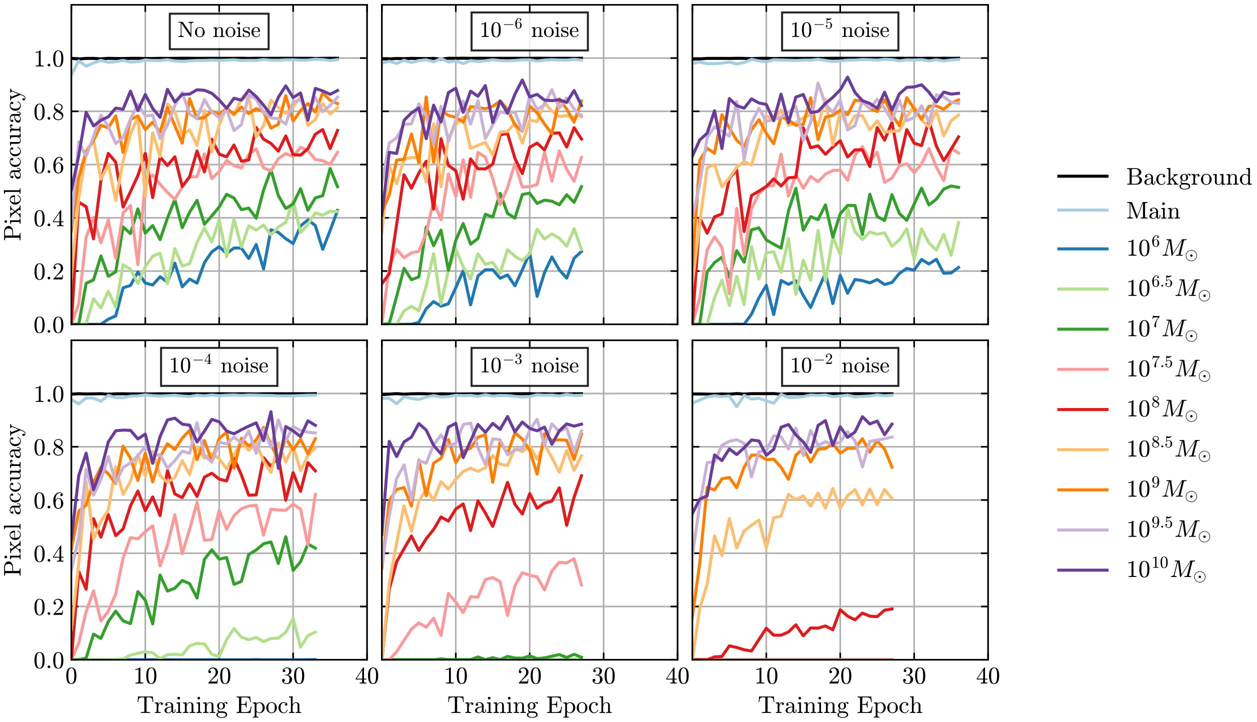

FIG. 4: Accuracy for the different pixel classes as a function of the training epoch. The label in each panel denotes the amount of noise included in the images. The network performance on the heavy subhalos is unaffected by noise while it loses sensitivity to the lower mass subhalos with increasing noise. Networks trained on less noise and applied to more noise perform very bad, even for the main lens and heavy subhalos. Conversely, training on more noise and applying to cleaner images performs according to the noise level the network was trained on.

a subhalo as a whole. The eventual goal, however, is to build a catalog of subhalos with their positions and masses. This is done later in Sec. V.

With the training procedure defined, we now move on to how noise in the images affects the training.

IV. THE EFFECTS OF NOISE

The last section showcased our training procedure and showed examples from the noiseless images. In general, we find that applying a network that was trained with less noise to images with more noise results in very poor performance. This is not surprising, but it implies that we need to train on images with noise if we want the network to detect subhalos in noisy images. To do this, we generate 9×10^4 training images and 10^4 validation images for different fixed levels of noise, as discussed in Sec. II. The noise is Gaussian with zero mean and standard deviation of a factor of $\left(10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\right)$ of the mean of the pixel brightness for the pixels that are at least as bright as 10% of the maximum pixel. The smallest levels of noise are unrealistic for currently observed strong lensing systems, although 1% results in similar signal-to-noise ratios as HST (see App. C). The noise levels are chosen to illustrate the correlation between the network's ability to identify pixels from a subhalo and the mean fractional change in intensity due to subhalos of a given mass. This helps us to understand what the network is doing and what causes it to fail.

Fig. 4 shows the per-pixel accuracy as a function of the training epoch for each of the classes. The amount of noise in each panel is indicated by the label. For reference, the upper-left panel shows the results with no noise, which were also shown in Fig. 3. The upper-middle and upper-right panels have very small amounts of noise, 10^{-6} and 10^{-5}, respectively. Despite these small levels of noise, the pixel accuracy for the $10^6 M_\odot$ and the $10^{6.5} M_\odot$ classes starts to decrease. The lower-left panel, with a noise at the level of 10^{-4}, has enough noise that the accuracy for the $10^6 M_\odot$ subhalos is almost zero. Similarly, the accuracy for the pixels of the $10^{6.5} M_\odot$ class is very low. The pixel accuracy for the heavier subhalos is not significantly changed.

As the noise is further increased to 10^{-3} (in the bottom-middle panel), the network loses sensitivity to subhalos with mass below $10^7 M_\odot$. However, we again see that the accuracy for the subhalos with $m > 10^8 M_\odot$ is not affected.

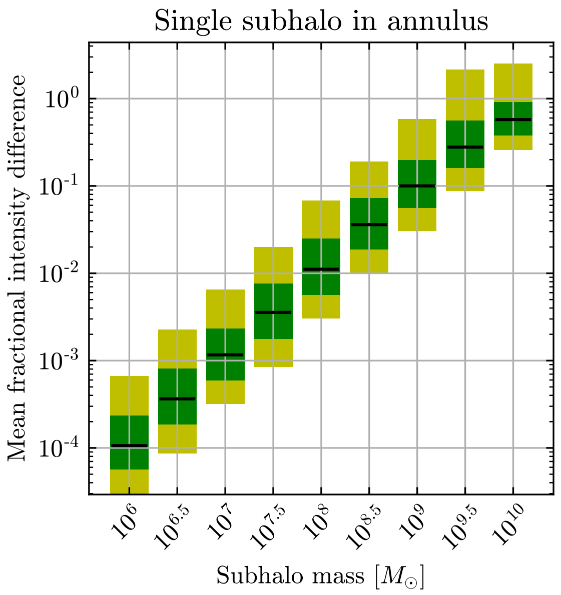

FIG. 5: Detecting subhalos require methods that are sensitive to small changes in pixel intensities. Images are generate without substructure. A single subhalo is then added, and the mean fractional change in the circle with a radius of 2 pixels around the subhalo are recorded. The bars show the mean from repeating this process 10^3 times. The green and yellow bands show the range of the 1 and 2 standard deviations.

In the final panel (lower-right), the noise level is 1%. With this much noise, the accuracy for the pixels coming from $10^8 M_\odot$ subhalos is significantly decreased (by 70%) and the accuracy of the $10^{8.5} M_\odot$ class pixels is decreased by around 25%. Again, the pixel accuracy for the heaviest subhalos do not change.

The noise level at which the network loses sensitivity to subhalos of a certain mass can be explained by the size of the perturbations caused by the subhalo. To demonstrate this, we examine 10^3 images before and after adding in a subhalo. As we have done so far, the subhalo is placed near the Einstein radius (defined by the pixels which are at least 20% as bright as the brightest pixel). We then compute the relative change in the observed image brightness for the pixels within a circle with a radius of 2 pixels centered on the location of the subhalo.[2] We plot the distribution for the mean relative change in each circle for each subhalo mass in Fig. 5. The bands mark 1 and 2 standard deviations.

Detecting a subhalo in the $10^6 M_\odot$ bin requires noticing changes in intensity in clusters of pixels of order 0.01%. This highlights the enormous potential of the U-Net architecture. Without noise, it is able to correctly identify around 40% of the pixels for the $10^6 M_\odot$ bin which cause tiny changes in the image. When the images contain noise at this same 10^{-4} level, it washes away the effects of these subhalos, as shown in the lower-left panel of Fig. 4. Similarly, Fig. 5 shows that subhalos in the $10^7 M_\odot$ and $10^8 M_\odot$ bins cause changes of order 0.1% and 1%, respectively. Fig. 4 likewise shows that the network loses sensitivity to these subhalos with these corresponding levels of noise. Throughout the rest of this work, we will compare the model with no noise or 1% noise. This amount of noise is realistic for strong lens images with apparent magnitudes brighter than 20.

Before moving on, we raise two points to the attention of the reader. The first is that the network needs to be trained on similar levels of noise to the data it will be used on. For instance, applying the network trained without noise to the noisy images leads to very poor performance. However, one cannot just be conservative and train on images with too much noise. We tested this explicitly by taking the network trained on images with 10^{-2} noise and applying it to images with no noise. The performance of the subhalos with masses greater than $10^8 M_\odot$ was very similar to that shown in the last panel of Fig. 4 for the images with 10^{-2} noise. Even though the images now have no noise, the network is not able to detect the low-mass subhalos. Training on images with varying levels of noise could alleviate this concern.

The second point we want to raise is that it is most likely possible to achieve sensitivity to higher levels of noise that we show here. Unlike random noise, the effects from perturbations to the lens are correlated across several pixels. Changing the size of the convolutional filters or the number of filters could help detect these small correlations on

[2] This corresponds to the pixels which are labeled as the subhalo in the target data of the training set.

top of the noise. Additionally, we did not implement any class weights into the loss function. It is possible to make the network place greater emphasis on identifying certain classes more than others. Our current setup has orders of magnitude more background pixels than subhalo pixels in the target data. Forcing the network to place greater emphasis on learning the more subtle subhalos, especially in the noisy images, could help. As a proof-of-principle, designing an optimal network for higher levels of noise is beyond the scope of this work. To this point, we have only shown results from the training and validation images. We now move onto to completely independent data to characterize the network's performance on images that it has not seen during either training or validation.

V. CHARACTERIZING THE NETWORK PERFORMANCE

After the network has been trained, we apply it to a series of images that the network has not seen during training or validation to evaluate its out-of-sample performance. We do this both for a network trained on images with no noise and a network trained on images with 1% noise. First, we show an example output of the network, which helps to visualize the different channel probabilities and see common ways for the network to mislabel pixels. We then compare the true target pixels to the predicted pixels to quantify the amount of correct and mislabeled pixels. Finally, we run the network on images without substructure to determine the rate at which the network will claim to detect subhalos when they are not there (false positives).

A. Example output

In Fig. 6, we show a detailed example of the network without noise applied to a noiseless image with a very light subhalo, in the $10^6 M_\odot$ bin. The upper row displays the observed image, which serves as input to the network (left), the truth-level target labels (middle), and prediction (right). We have assigned each pixel in the prediction image according to whichever class had the largest probability for that pixel. The lower three rows display the individual class probabilities, with red representing low probabilities (the network is certain that the pixel does not belong to that class) and blue is high probabilities (the network thinks this pixel belongs to the class). We chose a color map such that probabilities near 50% are white, showing that the network is unsure of those pixels.

In examining the target and prediction images of the top row, we see that the network successfully identified the subhalo in the image, despite its extremely low mass and its overlap with the main lens. The mass of the subhalo is indicated by the pixel colors (dark blue) representing a subhalo in the $10^6 M_\odot$ bin. In addition, we see that the predicted size is also very close to the target, although the shape is not exactly the same. For instance, the pixel on the top of the subhalo in the predicted image extends further into the main lens than it does in the true label. When examining the per-pixel accuracy in the next subsection, this will show up as a pixel which should be predicted to be part of the main lens, but is predicted to be in a subhalo class. Similarly, there are a couple of pixels along the bottom of the predicted subhalo that are not part of the target. In the discussion of the next section, these will show up as pixels which should be predicted as background, but are instead predicted as belonging to a subhalo class. In both of these cases, although the pixels were misidentified, the subhalo was still found and they do not represent the network introducing spurious substructure. If we examine the probability map for the $10^6 M_\odot$ class (2nd row, 3rd column), we see that the pixels around the edge of the subhalo are white, indicating that the network was not confident in these assignments. In general, we find that sometimes the subhalos are predicted to be a few pixels too large and sometimes a few pixels too small, but just as in the preceding discussion, if a single pixel that should be assigned to a subhalo class is misclassified into the main lens or background, this does not indicate that the subhalo was not found.

Similar features can be seen along the edge of the main lens. The shape and size of the lens is very similar between the target image and the predicted image. However, there are a few instances along the edge where the pixel assignments are incorrect. In the discussion below, these will be pixels that should be classified as background (main lens) but are misclassified as main lens (background). Looking at the probability maps in the 2nd row for the background and main lens, we can see that the regions where the network had classification errors have class probabilities near 50%.

This example contains another interesting feature; there is a single pixel that was predicted to belong to a $10^8 M_\odot$ subhalo. It is challenging to find the pixel on the predicted image (it is on the lower-right side of the main lens) but the probability maps shed light on it. The background probability map has a swath of pixels beneath the main lens that the network is unsure about. We see that these pixels are assigned some probability to belong to all of the subhalos classes with $m \leq 10^{8.5} M_\odot$, although for nearly all of them the background has the highest probability. However, one of the pixels has a probability in the $10^8 M_\odot$ bin that is slightly larger, so it gets assigned to that class despite the probability being low. Often these spurious pixels can be removed by imposing a probability threshold. For instance, the probability for these pixels is spread among many classes, and in no class is the probability at least

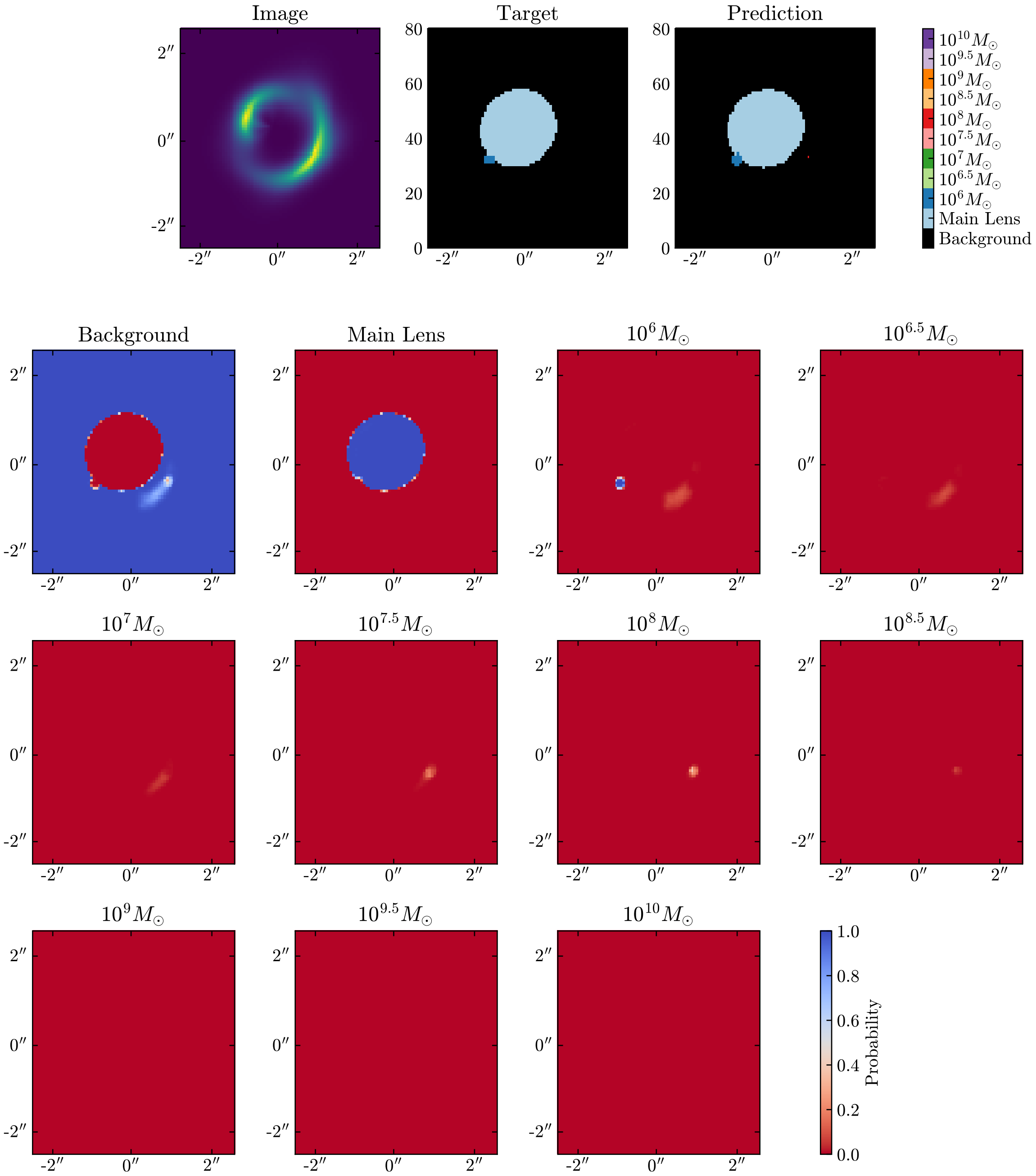

FIG. 6: The top row shows (from left to right) an image with no noise which is input to the network, the target labels for each pixel, and the prediction from the network. The lower three rows show the probability assigned to each class for each pixel. The network prediction for each pixel is determined by the class with the maximum probability for that pixel. The true subhalo has a mass in the $10^6 M_\odot$ bin and is detected in the prediction. A few pixels around the edge of the subhalo or main lens get misclassified. A single pixel is incorrectly predicted to belong to the $10^8 M_\odot$ class, but this prediction can be corrected by imposing a probability threshold.

50%. If we default all pixels to the background class unless a subhalo probability is greater than 50%, the wrong prediction will be removed. We note that most images do not contain random pixels like this one.

While instituting a probability threshold can remove spurious pixels, it also leads to a reduction in the overall accuracy for the true subhalo pixels, which is why we did not implement a probability threshold for our fiducial results. Recall that in Eq. (8) the accuracy was defined as the correctly assigned pixels for a class divided by the total true pixels for that class. Some pixels along the edge of subhalos will now get misclassified as background, reducing the accuracy.

B. Testing on a single subhalo

Now that the output of the network is better understood, we move on to quantify the the network's predictions. The purpose of this is to determine what the network is predicting for the pixels of any class: are most of them correct? And if they are predicted wrongly, what class are they assigned to? To assess this, we generate a new set of 10^4 images with the same amount of images with an without substructure as in the validation set: 10^3 have no substructure and the remaining images have a single subhalo in each and are evenly split between the nine mass bins, meaning there are 10^3 images in each.

In Fig. 7, we examine the per-pixel predictions on the images in the test set. The title in each panel states the true label of the pixels, while the x-axis denotes the predicted class. The solid (dashed) lines represent a prediction with no probability (50% probability) threshold. The blue line corresponds to images without noise while the orange line corresponds to images with 1% Gaussian noise. Each panel is normalized such that the sum of all the classes is unity.

The first two panels (at the top, from left to right) show the pixels that at truth-level correspond to the background or main lens classes, respectively. The network often misclassifies a few pixels around the edge of the main lens (e.g. Fig. 6 in the previous subsection). This can either be the main lens being reconstructed as slightly too large or too small, or getting the shape slightly off, although it is typically only a handful of pixels. Because there are so many background pixels, this corresponds to 0.1% of the background pixels getting misclassified as the main lens over the entire test set; this is the most common type of misclassification for the background pixels. Similarly, these errors around the edges of the main lens lead to $\lesssim 1\%$ of the pixels which should be predicted as the main lens getting misclassified as the background class.

The rest of the class assignments in these two panels are roughly uniform for the images without noise. In the images with noise, the same effect happens for subhalos with masses $> 10^8 M_{\odot}$. The explanation is similar to that of the main lens itself: when the network locates a subhalo, some of the pixels around the edge can get mislabeled. It is rare for the network to get the exact shape of the subhalo correct. This was also shown in Fig. 6. We emphasize that most of the pixels that are supposed to be background or the main lens, but are predicted to be a subhalo, do not represent additional false subhalos but are rather edge effects of this type. More evidence of this is given in the next subsection. Recall that, with 1% noise, the network does not detect subhalos with $m < 10^{7.5} M_{\odot}$. This means that there are no edge effects associated to these classes (and consequently no pixels getting misclassified as a subhalo with $m < 10^{7.5} M_{\odot}$), as can be seen by the cutoff in the orange histograms in both of these panels.

In these first two panels, we also see the effect of including a probability threshold when making a prediction. Specifically for the images without noise, the background and main lens pixels that get predicted into the $10^{6.5} M_{\odot}$-$10^{7.5} M_{\odot}$ classes are reduced by a factor of 1.5-2. This shows that network is uncertain of some of these pixels that are misclassified. The probability threshold does not have as large of an effect on the heavier subhalos because their effects are larger and allow the network to be more certain in its predictions pertaining to these mass classes.

The edge effects can also be seen in the other panels. For example, in the bottom-right panel, most of the pixels are correctly identified as belonging to a very massive subhalo. However, some are incorrectly marked as the main lens or background. These are from pixels around the edge of the subhalo. In addition to edge effects, the network sometimes gets the mass wrong by one mass bin. There is little difference in the results with and without noise, or with and without the probability threshold, for the heavy subhalos.

For subhalos with mass larger than $\gtrsim 10^{7.5} M_{\odot}$ for no noise, and $10^{8.5} M_{\odot}$ with noise, the most typical way for the prediction to be wrong is to predict the wrong mass bin. Progressing towards the panels with the lighter subhalos, we notice that the fraction of pixels incorrectly labelled as background and/or main lens increases. This makes sense as the magnitude of the deflection angles decreases with decreasing halo mass, so their effects are easier for the network to miss.

Fig. A1 in App. A shows the distribution of true labels for pixels predicted to belong to a given class, from which we see that in fact the predicted class is very likely to be correct. Thus, if the network predicts a group of pixels to have the same class, it is very likely that a subhalo is present there. The most common type of error for a pixel predicted to belong to a subhalo is that it should belong to a different (adjacent) mass bin.

Up to this point, we have only been discussing the per-pixel predictions. This makes sense from a machine learning perspective, but it does not necessarily address the physics goal of detecting subhalos themselves. This brings to light one potential challenge of using image segmentation to detect substructure in images of strong lensing: how does one go from pixels to subhalos? We have found that it is possible to either add extra pixels or miss pixels from a subhalo, especially around the edge. However the per-pixel accuracies and the example shown in Fig. 6 suggest that, on average, the size should be correct. This means that we can get a subhalo count by summing the number of pixels

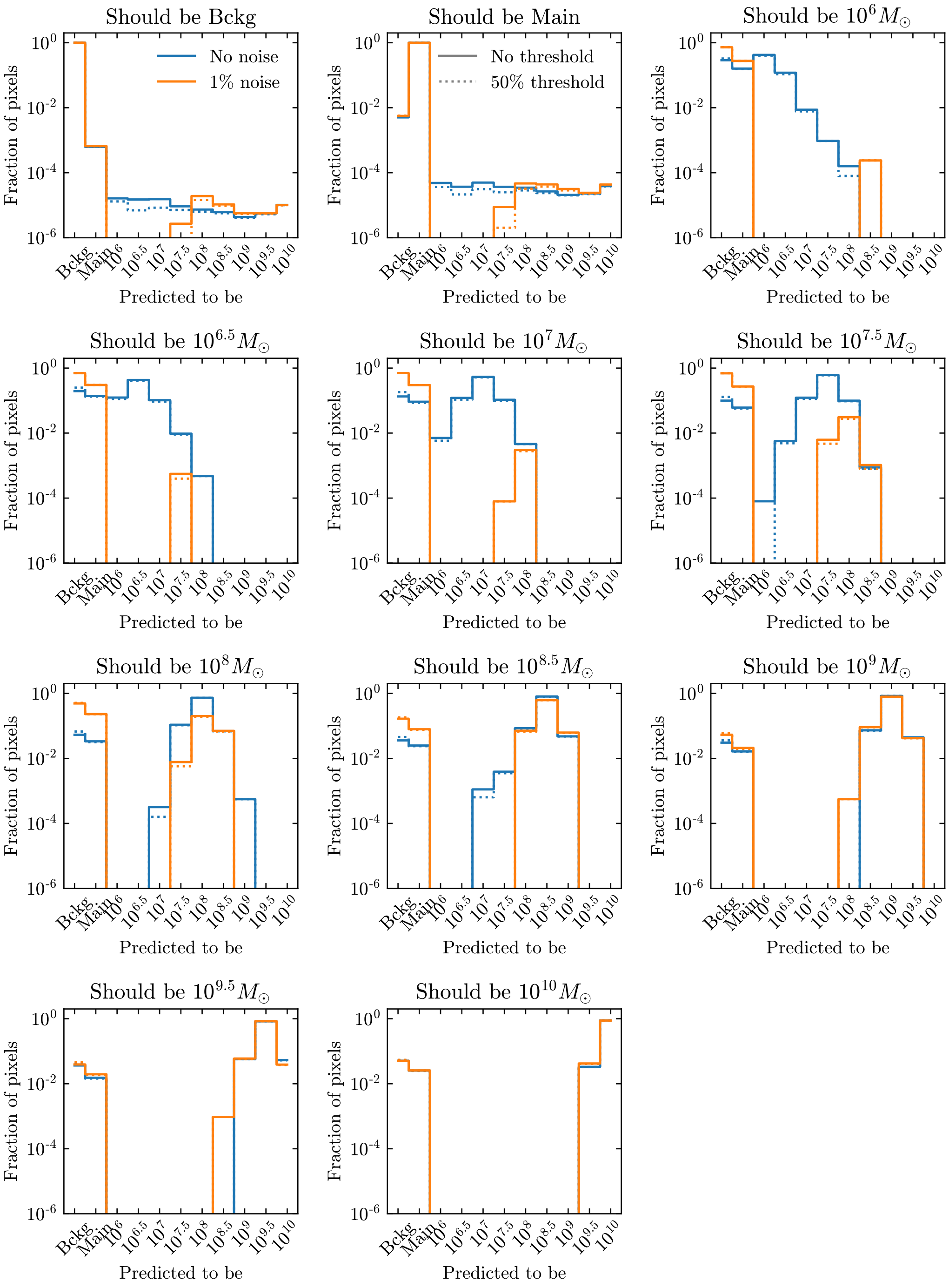

FIG. 7: Each panel corresponds to pixels which truth-level are of the indicated class. The x-axis denotes the class that the pixels are predicted to be. The solid and dashed lines represent the predictions with no probability threshold and a 50% probability threshold, respectively. The blue lines denote the network predictions on images without noise and the orange is for images with 1% Gaussian noise. Each line in each panel is normalized to unity.

predicted in each subhalo class and dividing by 4π pixels/subhalo (because we defined the target pixels such that the subhalos are comprised of a circle with a radius of 2 pixels).

We can then examine the subhalo detection accuracy over the test set. To define this accuracy, we take the pixels that at truth-level belong to a given subhalo class and count the number of these pixels assigned to each of the eleven possible classes. We then label the subhalo as belonging to the class with the largest count. Using this notion of a subhalo detection, Table I shows how the 10^3 subhalos of each mass bin were reconstructed. These numbers are for the

True Class	Not Detected	$10^6 M_\odot$	$10^{6.5} M_\odot$	$10^7 M_\odot$	$10^{7.5} M_\odot$	$10^8 M_\odot$	$10^{8.5} M_\odot$	$10^9 M_\odot$	$10^{9.5} M_\odot$	$10^{10} M_\odot$
$10^6 M_\odot$	384 [1000]	480 [0]	127 [0]	8 [0]	1 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]
$10^{6.5} M_\odot$	268 [1000]	141 [0]	475 [0]	105 [0]	11 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]
$10^7 M_\odot$	157 [996]	7 [0]	138 [0]	585 [0]	108 [1]	5 [3]	0 [0]	0 [0]	0 [0]	0 [0]
$10^{7.5} M_\odot$	90 [958]	0 [0]	7 [0]	138 [0]	659 [7]	105 [33]	1 [2]	0 [0]	0 [0]	0 [0]
$10^8 M_\odot$	29 [608]	0 [0]	0 [0]	0 [0]	117 [5]	780 [307]	73 [80]	1 [0]	0 [0]	0 [0]
$10^{8.5} M_\odot$	16 [134]	0 [0]	0 [0]	1 [0]	4 [0]	91 [75]	841 [717]	47 [74]	0 [0]	0 [0]
$10^9 M_\odot$	1 [12]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	78 [81]	874 [842]	47 [65]	0 [0]
$10^{9.5} M_\odot$	1 [4]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	62 [52]	879 [893]	58 [51]
$10^{10} M_\odot$	1 [1]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	37 [41]	962 [958]

TABLE I: The number of subhalos predicted to each class. For each row, there are 1000 images with a single subhalo in the indicated mass bin. The results for the network on images without noise and with noise are indicated by the number without and with brackets.

predictions without a probability threshold (see App. B for an analogous table including the probability threshold). The results for the network on images without noise and with 1% noise are indicated by the number without and with brackets, respectively. For the noiseless images, the network finds 616 (480 + 127 + 8 + 1) of the 1000 subhalos with a mass of $10^6 M_\odot$ (62%), of which 78% are in the correct mass bin. The subhalo detection accuracy, as well as the probability of getting the mass correct, increase for heavier subhalo masses. Subhalos with $m \geq 10^8 M_\odot$ are detected more than 97% of the time in images with no noise.

For images with 1% noise, nearly all of the subhalos lighter than $m \leq 10^{7.5} M_\odot$ are missed. Even though the 1% noise is on the same level as the effects on the brightness due to a $10^8 M_\odot$ subhalo, 39% of them are detected. Once the subhalos are heavy enough to produce effects larger than the noise level, the accuracy is similar to the noiseless images. We find that subhalos with $m \geq 10^{8.5} M_\odot (m \geq 10^9 M_\odot)$ are detected more than 86% (98%) of the time with 1% noise.

C. Null tests

The claim of the last section is that the pixels that should have been predicted as background, but were instead classified as a subhalo pixel, were dominantly from edge effects. Fig. 6 provided an example of one such case. In this section, we provide evidence of this claim systematically by determining the rate at which the network finds spurious substructure in images where there is only a smooth lens.

We generates a set of 10^3 images that contain only the source light and the smooth main lens, and another set that also contained 1% Gaussian noise. The trained networks were then applied to each of the images. We did not use a probability threshold for the predictions here to allow low-probability substructure to appear in the count which provides a more conservative estimate. The resulting counts are shown in Fig. 8 for the noiseless images in the left panel and the images with 1% noise in the right panel. The error bars were estimated using the square root of the estimated subhalo count, $\sqrt{N_{\rm subhalo}}$.

While the images do not contain subhalos, the network incorrectly predicts a few subhalos with low masses. The rate for this to happen is less than 10 subhalo per 10^3 images for each class separately, for the images without noise. The total rate of false subhalos is around 20 subhalos per 10^3 images, coming entirely from subhalos with masses $m \leq 10^8 M_\odot$. This gives us strong confidence that there is substructure in the lens when the network predicts a subhalo.

When noise is included, the total number of spurious subhalos increases slightly, to 25 per 10^3 images. Most of these are in the $10^8 M_\odot$ bin. This makes sense because subhalos with this mass cause changes in the pixel brightness on the order of 1%.

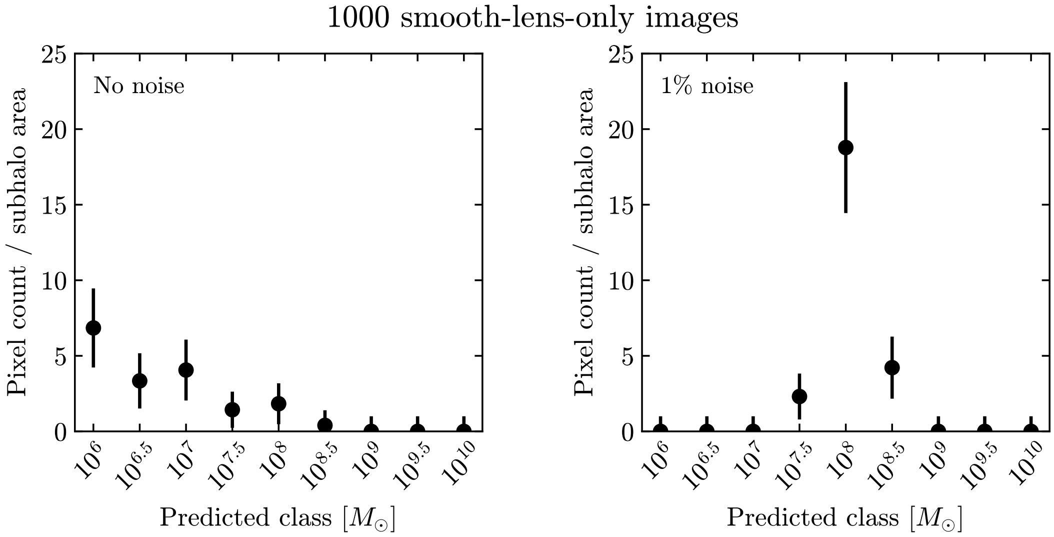

FIG. 8: Approximate predicted subhalo counts from 10^3 images that do not contain substructure in the lens. Uncertainties are estimated using $\sqrt{N_{\text{subhalo}}}$. The left and right panels correspond to noiseless images and images with 1% noise, respectively. For noiseless images, in each individual class the false-positive rate is less than 10 subhalos per 10^3 images, and the combined rate is around 20 subhalos per 10^3 images. The total false-positive rate in images with noise is around 25 subhalos per 10^3 images, most of which are in the $10^8 M_\odot$ mass bin.

D. Testing on images with many subhalos

Up to this point, we have presented results from networks trained on images with at most a single subhalo near the brightest pixels. However, galaxies are expected to have a large population of subhalos distributed throughout the main dark matter halo. We therefore investigate whether, without knowing this true distribution of subhalos, our network can be used to identify a population of substructure (i.e. more than a single subhalo per image).

To test this, we generate a set of images with up to 25 subhalos (with the actual number in each image drawn from a uniform distribution) with random masses and locations. Four of these images are shown in the left panels of Fig. 9. The middle panels show the true pixel labels and the right panels show the network output. We denote the number of subhalos in each of the target and predicted images. All of these images have no noise, but the results generalize to images with noise (losing efficiency for the lower masses, as explained above).

The network predictions for these images illustrate that light substructure far from the images is hard for the network to capture. Each row also illustrates different interesting properties. In the top row, on the lower-right side of the main lens, where there are two overlapping subhalos and the network only picks up on the heavier one of the two, which makes sense. The second row contains an example of a subhalo in which the network is unsure of the mass bin. The subhalo at the top of the main lens gets half of its pixels reconstructed in the $10^6 M_\odot$ class while the other half are put in to the $10^{6.5} M_\odot$ class, although the truth-level subhalo belongs to the $10^6 M_\odot$ class. In the third row, five subhalos are inside of the Einstein radius. The heaviest of these is reconstructed, but the others are missed. The interior of the ring is also far from the light, making detection harder. Additionally, it seems like the network could have compensated for these subhalos by slightly increasing the main lens, but instead it does find the $10^9 M_\odot$ subhalo, which is impressive. Finally, in the last row, we see examples of heavier subhalos, which are able to be detected closer to the edge of the image than the lighter subhalos.

From these examples, we see that the network trained on images with a single subhalo is able to detect a population of substructure. As the subhalos get further away from the bright pixels, the network loses sensitivity to them. The presence of subhalos close together is also a challenge for the network. When there is a hierarchy between their masses, the heavier subhalo will wash out the effects of the smaller one. This results in only a detection of the heavy subhalo. If nearby subhalos are similar in mass, they may get reconstructed as a single subhalo in the bin corresponding to the sum of their masses.

The variety of subhalo populations that were detected in Fig. 9 (both in terms of raw number of subhalos as well as their mass distribution) also highlights one of the strengths of our single-subhalo training methodology. We found empirically that when we instead trained a network on images with many subhalos drawn from a power-law mass function and then tested it on images with populations of subhalos drawn from a mass function with a different

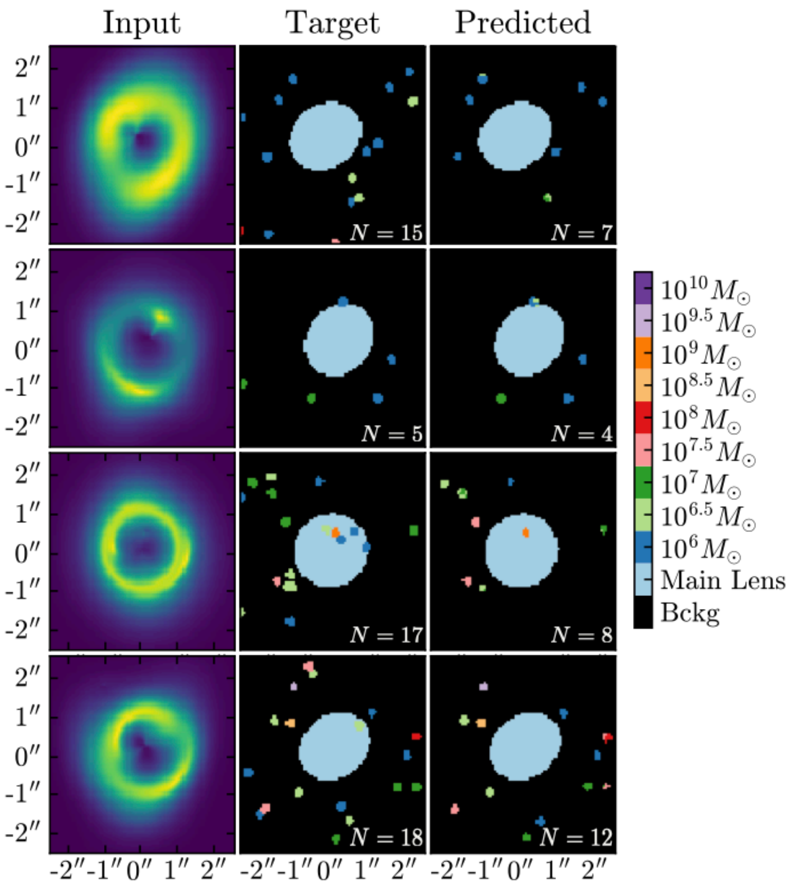

FIG. 9: The network trained on noiseless images with a single subhalo is now applied to noiseless images with rich distributions of substructure. The network has low sensitivity to subhalos that are far from the bright pixels around he Einstein ring. Subhalos that are near each other can get reconstructed as a single subhalo with their combined mass. The number of true/predicted subhalos are denoted in the figure.

power-law index, the detection accuracy became biased. For instance, if the network is trained on images with many subhalos drawn from a steep mass function, the few images with heavy subhalos will almost certainly have lots of light substructure as well. When we applied the network to images generated with a shallower mass function, the false-positive rate was higher for images with heavy subhalos: when it saw a heavy subhalo, it expected more light subhalos than there were, thus introducing false substructure. By only having a single subhalo in the training images, our fiducial network never learns to make decisions based on population characteristics, and thus generalizes extremely well to images with very different subhalo populations.

We have now shown the our network detects subhalos near the Einstein ring with high accuracy, and is capable of generalizing to images with different subhalo populations. Additionally, it has a very low false positive subhalo rate of 2.5 subhalos per 100 images. In the next section we demonstrate how such a network could be used to extract the subhalo mass function from an ensemble of strong lens images.

VI. DETERMINATION OF THE SUBHALO MASS FUNCTION

The population of subhalos under a CDM scenario is found to be well described by a power-law of the form

$$\frac{dN}{dm} \propto m^{\beta}, \tag{9}$$

with $\beta = -1.9$ [62]. However, models beyond CDM can affect this subhalo mass function. In previous sections, we have described how our U-Net model is able to accurately detect subhalos in simulated images. In this section, we show that we can determine the subhalo mass function using the network outputs from many images.

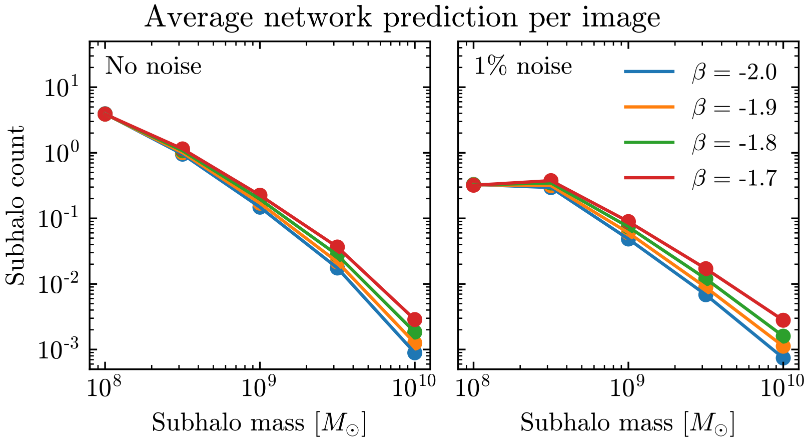

FIG. 10: The left and right panels display the average number of detected subhalos per class for 10^5 images with an without noise, respectively. The images can have between 0 and 25 truth-level subhalo uniformly distributed throughout the image. The masses of the subhalos are drawn from a power-law with an index denoted by β. The network has less sensitivity to lower mass subhalos, resulting in the curved shape. To infer the power-law index of an independent dataset, we find which value of β yields a curve closest to that observed in the data.

We first generate a mock catalog of images that can have more than a single subhalo. In each image, we draw the number of subhalos from a uniform distribution between 0 and 25, and place them uniformly throughout the image [43, 44]. The masses of the subhalos are drawn from a power-law with fixed $\beta = -1.9$. As the network with noise is not able to detect substructure with $m < 10^8 M_\odot$, and we want to compare the performance of the networks trained on noiseless and 1% noise images, we do not generate subhalos with masses that would end up in the $m \leq 10^{7.5} M_\odot$ bins. Our goal will be to infer the value of β from this catalog.

If we were able to perfectly reconstruct every subhalo, we could just fit the counts-per-bin to the functional form in Eq. (9). However, the accuracy is worse for lower mass subhalos, effectively changing the shape of the extracted mass function. An example of this is shown in Fig. 10. Here we simulated 10^5 images, using the same prescription as in our mock catalog, for β =-1.7, -1.8, -1.9, and -2.0. We then applied the network to each image and counted how many pixels were assigned to each class. This pixel count was converted to an approximate subhalo count by dividing by 4π. The left panel shows the average number of detected subhalos for the images with no noise, and the right panel shows the average counts for the images with 1% Gaussian noise. While the curves in the noiseless images are nearly straight (in log-log space), there is still some curvature. This is exaggerated even more in the images with noise, where in fact the subhalo counts plateau below $\sim 10^8 M_\odot$.

Rather than trying to fit the data with Eq. (9), we instead infer the most probable power-law index to have generated the observed subhalo counts extracted from a set of images with our U-Net model. To do so, we build a likelihood function which is the product of Gaussian likelihoods for each mass bin. Namely, the likelihood is given by

$$L(\beta) \equiv \prod_{i \in \text{mass bins}} G\Big(o_i \Big| \mu_i(\beta), \sigma_i(\beta)\Big), \tag{10}$$

where o_i is the observed number of detected subhalos in bin i, $\mu_i(\beta)$ is the expected value, and $\sigma_i(\beta)$ is the estimated standard deviation given β. To use the likelihood to infer β, we need to derive $\mu_i(\beta)$ and $\sigma_i(\beta)$. This is done in the next subsection. We emphasize that these derivations will include details that are specific to our mock catalog of images, and may not generalize to the real world, but we will discuss how to generalize to other catalogs.

A. Expectation and variance

Here, we derive the expectation and variance used in the likelihood function given in Eq. (10). First, we define the efficiency to tag a subhalo as

$$\varepsilon_i = \frac{\text{Number of subhalos predicted in class } i}{\text{Number of true subhalos in class } i}. \tag{11}$$

Because we train our network on images with a single subhalo, the network does not know about population level statistics, which is why the efficiency to zeroth order is not a function of β. If an image has N real subhalos, we define φ as the true fraction of subhalos in each mass bin, N_i. Note that φ is necessarily a function of β and is given by

$$N_i(\beta) = \varphi_i(\beta)\ N, \tag{12}$$

where N_i is the true number of subhalos in the ith bin. Thus, the number of subhalos expected to be predicted in a given class for an image with N total true subhalos can be written as

$$\langle n_i(\beta)|N\rangle = \varepsilon_i\ \varphi_i(\beta)\ N. \tag{13}$$

where n_i is the number of predicted subhalos in bin i.

From running a large number of images, we obtain a good estimate for

$$\begin{aligned}\epsilon_i(\beta) &= \varepsilon_i\ \varphi_i(\beta) \\ &= \langle n_i(\beta)|N\rangle/N\ ,\end{aligned} \tag{14}$$

where $\epsilon_i(\beta)$ is essentially the rate at which subhalos are detected in mass bin i given the value of β. For example, in Fig. 11 we applied the network to count the number of predicted subhalos in 10^5 images for various fixed values of β with an average of 12.5 subhalos per image. Rather than showing the results as a function of the subhalo mass bin (as done in Fig. 10), we show the average count per image as a function of β, with each mass bin shown in a different panel. We find that the data can be fit well with an exponential function of the form

$$\epsilon_i(\beta) = a_i + b_i\, e^{c_i\beta}\ . \tag{15}$$

Given the rate (ϵ_i) and the true number of subhalos (N) in a single image, the number of predicted pixels in a given class should be be Poisson-distributed about the expectation. Thus, the probability (p) to predict n_i subhalos is given by

$$p\big(n_i(\beta)\big) = P\big(n_i|\epsilon_i(\beta)\ N\big) = \frac{\big(\epsilon_i(\beta)\ N\big)^{n_i} e^{-\epsilon_i(\beta)\ N}}{n_i!}\ . \tag{16}$$

The expected number of detected subhalos in a single image with N true subhalos from Eq. (13) can then be rewritten in terms of the the weighted sum over the individual probabilities

$$\langle n_i(\beta)|N\rangle = \sum_{n_i} n_i\ p\big(n_i(\beta)\big)\ . \tag{17}$$

We now build in the assumptions about our mock data. In each image, we have placed between 0 and $N_{\rm max} = 25$ true subhalos, with the number drawn from a uniform distribution. The expected number of detected subhalos in class i for any image is then the average over the $N_{\rm max} + 1$ possible values (including 0 subhalos),

$$\langle n_i(\beta)\rangle = \sum_{N=0}^{N_{\rm max}} \frac{1}{N_{\rm max}+1} \sum_{n_i} n_i\ p\big(n_i(\beta)\big)\ . \tag{18}$$

In an analysis of real data, the actual distribution of subhalos would need to be obtained through detailed N-body simulations, which is beyond the scope of this work.

In any given image, the number of subhalos in a bin can vary widely. To compute the variance, we calculate $\langle n_i^2\rangle$ as a function of β, given by

$$\langle n_i^2(\beta)\rangle = \sum_{N=0}^{N_{\rm max}} \frac{1}{N_{\rm max}+1} \sum_{n_i=0}^{N} n_i^2\ p\big(n_i(\beta)\big)\ . \tag{19}$$

Then the variance is

$$\sigma_{n_i}^2(\beta) = \langle n_i^2(\beta)\rangle - \langle n_i(\beta)\rangle^2\ . \tag{20}$$

The range of possible true subhalos makes this variance large.

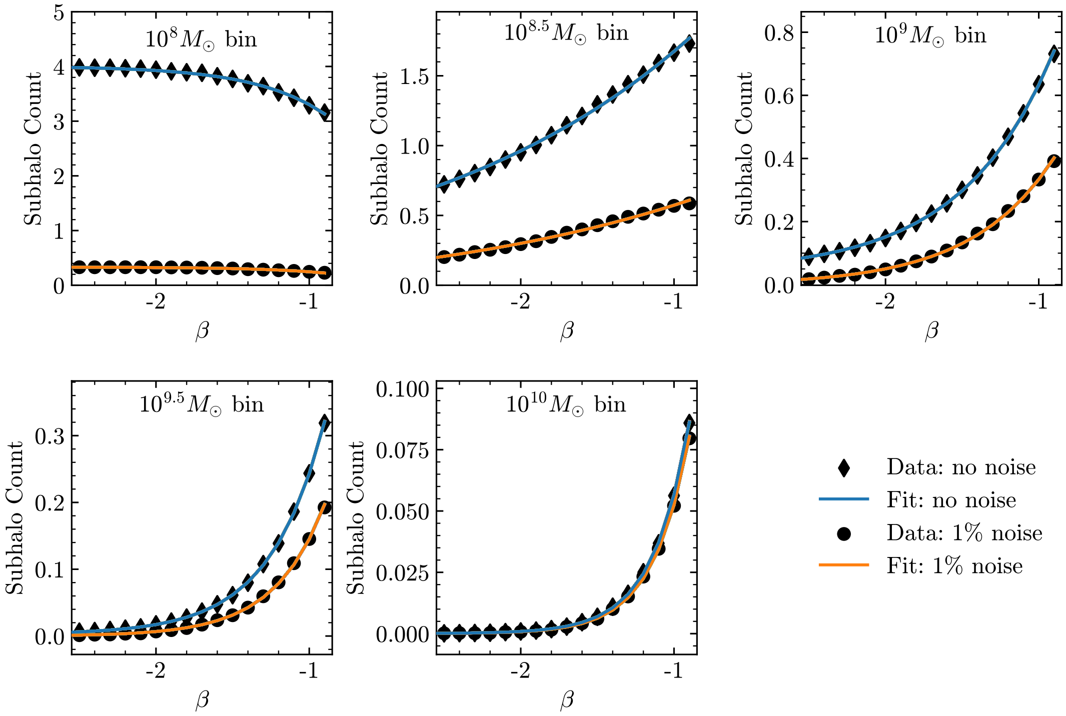

FIG. 11: Data points show the average number of subhalos detected by our network per image over 10^5 sample images with and without noise. The lines show exponential fits to the data as described in the text. These are used to define the expected subhalo counts and the standard deviations used in the likelihood function. The downward trend for increasing β in the $10^8 M_\odot$ panel is because of the fixed range for the number of subhalos.

Up to this point, we have examined the expectation and variance for the number of detected subhalos per bin in a single image. However, to determine the subhalo mass function we will analyze many images together. The expected number of pixels and the variance for N_{images} independent images is given by

$$\langle n_i(\beta)|N_{\text{images}}\rangle = N_{\text{images}}\langle n_i(\beta)\rangle; \tag{21}$$

$$\sigma^2_{n_i(\beta)|N_{\text{images}}} = N_{\text{images}}\sigma^2_{n_i}(\beta). \tag{22}$$

The value of the mean and standard deviations for each bin included in the likelihood function are then given by

$$\mu_i(\beta) = N_{\text{images}}\langle n_i(\beta)\rangle \tag{23}$$

and

$$\sigma_i(\beta) = \sqrt{N_{\text{images}}\sigma^2_{n_i}(\beta)}\ . \tag{24}$$

B. Inferring the power-law

At this stage, given a subhalo mass function power-law index and some number of images, we can compute the expected number of detected subhalos and the standard deviation for each bin. When we apply our network to a set of images, we can then determine which value of β yields expectations closest to the observed detections using the likelihood function defined in Eq. (10). Recalling that the Gaussian expectation and variance are a function of β and that the observed data does not change, the likelihood is then a function only of β,

$$L = L(\beta). \tag{25}$$

To test a hypothesized value of β, the likelihood ratio is used,

$$\ell(\beta) = \frac{L(\beta)}{L(\widehat{\beta})}, \tag{26}$$

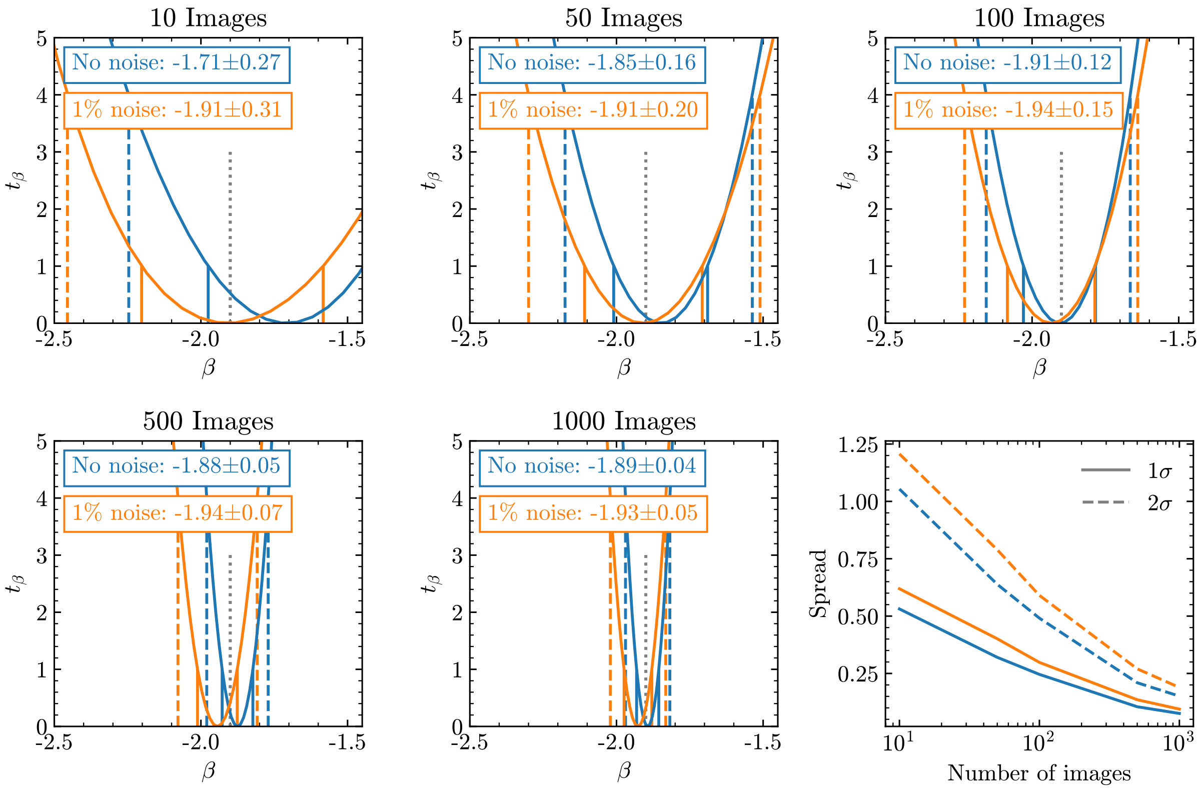

FIG. 12: Result of scanning the test statistic t_β when scanning over the power-law index β. The test statistic is defined in Eq. (27). The value of β that minimizes t_β produces expected count most similar to that in the mock observed data. The true value of β in the mock data is -1.9 and is marked by the grey dotted line. The 1 and 2 σ confidence intervals are the values of β between $t_\beta = 1$ and 4, respectively. The last panel shows that the uncertainty decreases as the number of images increases. The blue and orange lines shown the results for image with no noise or 1% Gaussian noise, respectively. The last panel shows the spread (maximum - minimum) of the 1- and 2-σ confidence bands. The uncertainty on the best fit for the images with noise is about 20% larger than the noiseless images.

where $\widehat{\beta}$ is the value of the slope which maximizes the likelihood. To find $\widehat{\beta}$, we use the Nelder-Mead [68] algorithm implemented in SciPy [69]. We then compute the test statistic t_β, defined as

$$t_\beta = -2 \log \ell\left(\beta\right), \tag{27}$$

with which we can determine confidence intervals.

Fig. 12 does this for images containing subhalo populations drawn from a subhalo mass function with $\beta_{\rm true} = -1.9$, for different numbers of images in each panel. The best fit is the value of β that minimizes t_β, while the $1\sigma(2\sigma)$ uncertainty is given by the range between $t_\beta = 1(4)$. The orange and blue lines denote the results from images with and without noise, respectively. The last panel shows the spread of the confidence intervals as a function of the number of images used for the fit.

The first panel uses only 10 images, and the resulting inferred values have large uncertainties. This is primarily due to the large variance in the number of subhalos per image, which is incorporated into the uncertainty. While the central value for the noiseless images is not close to the true value of $\beta_{\rm true} = -1.9$, it is within the 1σ confidence interval. Increasing the number to 50 images, the uncertainty drops by a factor of 1/3. The best fit values also start to get closer to the true value. The trend continues as more images are added, with the best fit converging to the true value and the uncertainty decreasing.

We again note that noise affects our ability to tag low-mass substructure. As shown in Fig. 10, the number of detected subhalos does not resemble a power-law for the noisy images. Despite this, we find that the uncertainty on the inferred value of β is only around 20% worse. With 500 images, the uncertainty on the power-law index is less than 0.1. Overall, we find that, when measured over five mass bins from $10^8 M_\odot$ to $10^{10} M_\odot$, the SMF slope is recovered with an error of 14.2 (16.3)% for 10 images, and this improves to 2.1 (2.6)% for 1000 images without (with 1%) noise

The results presented here strongly depend on our mock catalog. Specifically, for the images in our catalog, the number of subhalos in an image is drawn from a uniform distribution between 0-25 for the detectable mass bins ($m > 10^8 M_\odot$). To generalize this procedure to real lensing data, we would need to estimate the expected number of detectable subhalos per mass bin and the variance from N-body simulations.

VII. DISCUSSION AND OUTLOOK

We developed a method to detect subhalos in strong gravitational lens images. The method is based on image segmentation: we classify each pixel in an image as belonging to either the main lens, a subhalo within a given mass range, or neither. We trained a convolutional neural network with a U-Net architecture on images with either no substructure or a single subhalo near the lensed images. When the network is applied to an independent test set, it performs remarkably well.

We find that there are three common ways for the network to misclassify a pixel.

1. The subhalo is not detected and all of the pixels are assigned to the background or main lens classes. This happens more for light subhalos than large subhalos.

2. The pixel is on the edge of a subhalo and it is labeled as background instead of belonging to a subhalo. In these cases, the network finds the subhalo, but it is predicted to be a few pixels too large or too small.

3. Misidentifying the mass bin, generally by assigning the pixel to an adjacent mass bin. In these cases, the subhalo is still located correctly, but the mass is shifted up or down by a bin.

While the network is trained specifically looking pixel-by-pixel, we need to cluster the detected pixels into subhalos to extract some physical meaning to the pixel-based accuracies. On average, the subhalos have an area of 4π pixels, allowing for an easy conversion between the number of tagged pixels and the number of subhalos. Pixels around the edge of a subhalo can be missed while allowing for a detection of the subhalo. Because of this, the subhalo detection accuracy is better than the per-pixel accuracy.

Without noise, the class with the worst accuracy is that of subhalos in the $10^6 M_\odot$ mass bin, but 62% of these subhalos are located, of which 78% get assigned the correct mass. The accuracy improves for heavier subhalos. At a mass of $10^8 M_\odot$, the subhalos are detected 97% of the time, with 80% in the correct mass bin. The heaviest subhalos we considered have masses in the $10^{10} M_\odot$ bin, and 99.9% of these subhalos are detected, with 96% getting assigned to the correct mass bin. When noise is included, the network loses sensitivity to the low-mass subhalos. For instance, with 1% noise, only 39% of the $10^8 M_\odot$ subhalos are detected. However, the accuracy is similar for $m \geq 10^9 M_\odot$ with and without noise.

To put this in perspective, Refs. [28, 29] showed that traditional direct detection techniques can find subhalos with masses of a few times $10^8 M_\odot$ for a signal to noise ratio of as low as 3 when the subhalo has an NFW profile and on the Einstein ring. Additionally, these methods have detected subhalos with masses of $(3.51 \pm 0.15) \times 10^9 M_\odot$ [31] and $(1.9 \pm 0.1) \times 10^8 M_\odot$ [32]. We reach good accuracy for the same range of masses detected by traditional methods using similar amounts of noise; see App. C for a comparison of noise profiles. An advantage of our method is that we do not need to initially model the smooth lens to detect substructure in the system, which can take $\mathcal{O}$(weeks) to analyze in real systems. Instead, it takes us less than a second to run an image through the network. It is also important to keep in mind that our accuracies are presented for the *true* subhalo mass which was simulated. Direct detection methods are sensitive to the *effective* subhalo mass. Ref. [37] showed that the true mass can be biased by up to an order of magnitude compared to what is measured.

The network was also tested on images that do not contain substructure. False substructure was predicted at a rate of around 2.5 subhalos per 100 images. Most of these fall into the lightest mass bins that a network is sensitive to. The good detection accuracy and low false-positive rate implies that if the network predicts substructure is present an image, it is mostly likely truly due to a subhalo being present.

We also applied the network trained on images with single subhalos to images with many subhalos. We showed that the network has out-of-sample adaptability and can generalize to identify an abundance of substructure in a single image, although it does not see any such data during training. With this, we examined a method to determine the subhalo mass function power-law index from the network output of multiple images. The subhalo mass function is a key target for dark matter science because it can diagnose deviations from the standard cold dark matter scenario.

The technique uses a likelihood ratio for the count of detected subhalos in all of the bins, taking into account the expected counts given the power-law index and the network's detection accuracy for each class. We estimate that a 10% determination of the slope of the subhalo mass function will require around 50 (100) images without (with) noise. However, this is strongly dependent on astrophysical assumptions concerning the total mass of a halo contained in substructure. For instance, our catalog contained images with [0,25] subhalos with masses $m \geq 10^8 M_\odot$. If the true distribution of subhalos produces significantly more or less subhalos near the Einstein ring, the number of images necessary for an accurate determination of the power-law index of the subhalo mass function could change.

We also note that Ref. [52] uses likelihood-free inference to extract population-level substructure properties about the dark matter subahalo population. They use a neural network to approximate the likelihood ratio function to obtain both the slope and the amplitude of the subhalo mass function with $\mathcal{O}(100)$ images. In contrast, we infer about the subhalo mass function by explicitly resolving individual subhalo, which can then be further studied.

In the near future, the Vera Rubin Observatory is expected to find 10^4 galaxy-galaxy strong lenses [70] and Euclid expects 10^5 galaxies lensed by galaxies in the field-of-view [71]. The ability to quickly detect substructure in these lenses could dramatically improve our understanding of dark matter. Image segmentation is a promising method to study dark matter subhalos. We stress that work will be needed to increase the robustness to noisier images (dimmer sources). Additionally, we have ignored the possibility of extra perturbations to the lens along the line-of-sight [72–76], but not part of the main lens halo. These are both interesting problems that we save for future work.

ACKNOWLEDGEMENTS

We thank Simon Birrer, A. Cagan Sengul, and Arthur Tsang for help with Lenstronomy. We are thankful to Simon Birrer, Johann Brehmer, and Siddharth Mishra-Sharma for insightful comments on a previous version of this paper. BO was supported in part by the U.S. Department of Energy under contract DE-SC0013607. CD was partially supported by the Department of Energy (DOE) Grant No. DE-SC0020223.

Appendix A: Single subhalo pixel predictions

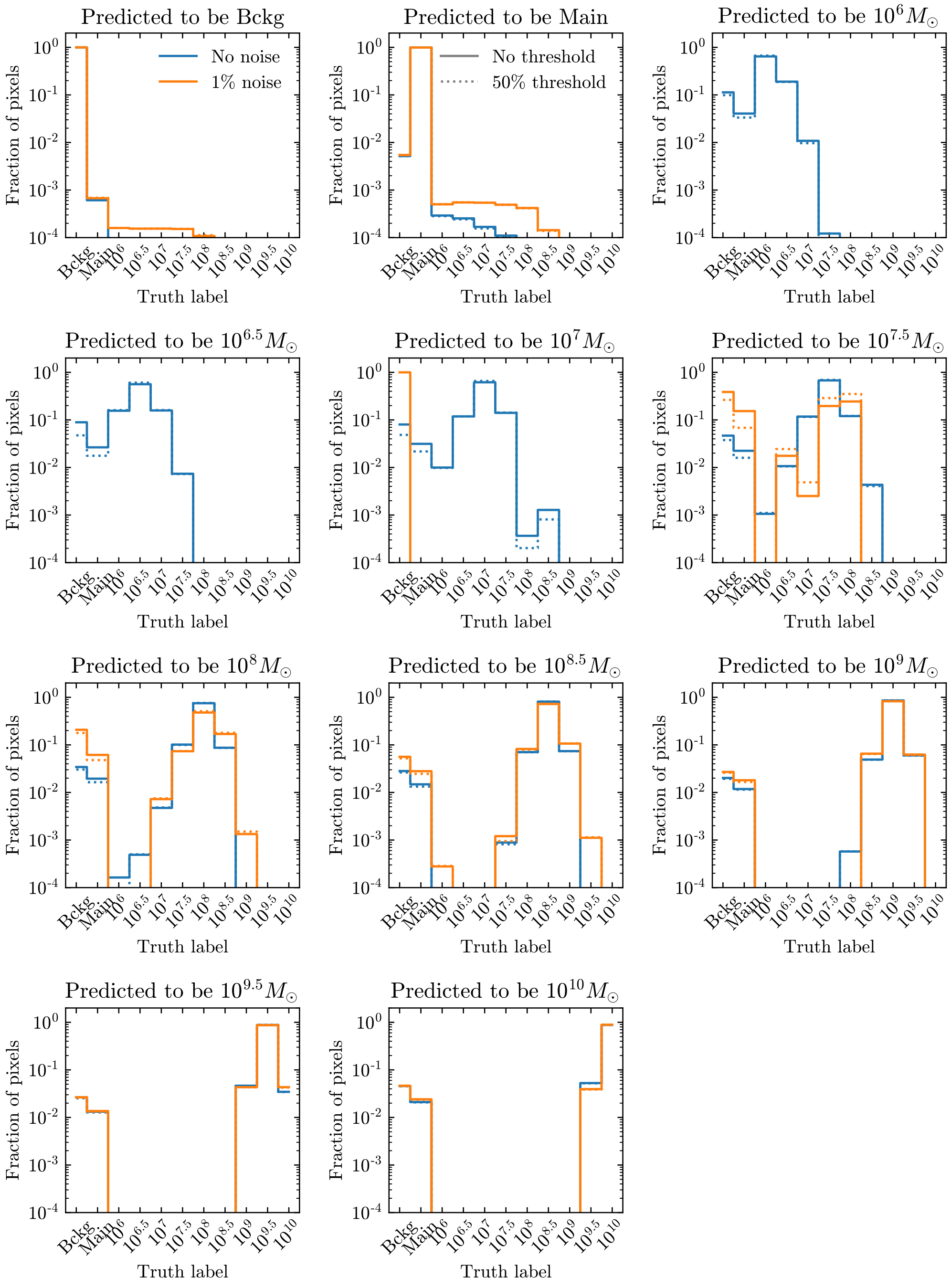

FIG. A1: Each panel corresponds to pixels which are predicted of the indicated class. The x-axis denotes the class that the pixels belong to at truth-level. Each panel is normalized to unity. The results for images without (with 1%) noise are shown in the blue (orange) lines. The solid lines denote when the predicted class has no probability threshold. The dotted lines require that the predicted probability is at least 50% before assigning a pixel to a subhalo class.

In the main text, we examine the per-pixel predictions on the images of the test set in Fig. 7. There, each panel corresponded to the true label of the pixel and the x-axis represented the predicted class for the pixel. Here we flip the information around. Each panel in Fig. A1 corresponds to pixels that are predicted to be part of the class indicated by the title. The x-axis then shows the true label and the panels are again normalized to unity. The blue and orange lines are for images with no noise and 1% noise, respectively. The solid lines do not use a probability threshold for the pixel prediction, while the dotted lines enforce that a pixel will not be predicted to a subhalo class unless the probability is at least 50%.

It is now clear that the class that is predicted is very likely to be correct. For the images with no noise, every panel has the class with highest probability correct for the prediction. The probability threshold reduces the number of pixels predicted to belong to the light subhalos. This slightly increases the fraction of pixels coming for the correct class while reducing the predictions from pixels which were supposed to belong to the background or main lens.

With noise, the network essentially does not predict pixels to belong to subhalos lighter than $m \leq 10^7 M_\odot$. While it predicts some pixels in the $10^{7.5} M_\odot$ bin, these are not very accurate, although the probability threshold does help. Once the subhalos are as heavy as $10^8 M_\odot$, the predicted class usually comes from pixels which at truth level were that same class.

Appendix B: Subhalo predictions with probability threshold

True Class	Not Detected	$10^6 M_\odot$	$10^{6.5} M_\odot$	$10^7 M_\odot$	$10^{7.5} M_\odot$	$10^8 M_\odot$	$10^{8.5} M_\odot$	$10^9 M_\odot$	$10^{9.5} M_\odot$	$10^{10} M_\odot$
$10^6 M_\odot$	410 [1000]	463 [0]	117 [0]	9 [0]	1 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]
$10^{6.5} M_\odot$	300 [1000]	135 [0]	455 [0]	100 [0]	10 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]
$10^7 M_\odot$	178 [996]	5 [0]	128 [0]	577 [0]	107 [1]	5 [3]	0 [0]	0 [0]	0 [0]	0 [0]
$10^{7.5} M_\odot$	103 [963]	0 [0]	6 [0]	132 [0]	653 [4]	105 [31]	1 [2]	0 [0]	0 [0]	0 [0]
$10^8 M_\odot$	33 [620]	0 [0]	0 [0]	0 [0]	115 [2]	778 [299]	73 [79]	1 [0]	0 [0]	0 [0]
$10^{8.5} M_\odot$	19 [140]	0 [0]	0 [0]	0 [0]	4 [0]	90 [70]	840 [716]	47 [74]	0 [0]	0 [0]
$10^9 M_\odot$	2 [13]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	78 [81]	872 [841]	48 [65]	0 [0]
$10^{9.5} M_\odot$	1 [4]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	62 [52]	879 [892]	58 [52]
$10^{10} M_\odot$	2 [2]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	0 [0]	36 [42]	962 [956]

TABLE A1: The number of subhalos predicted to each class. In order for a pixel to get assigned to a subhalo class, the probability must pass a 50% threshold. The subhalo counts are then computed by examining the pixels which at truth level correspond to the substructure. The subhalo is assigned to the class with the most predictions from these pixels. For each row, there are 1000 images with a single subhalo in the indicated mass bin. The results for the network on images without noise and with noise are indicated by the number without and with brackets.

This appendix shows the subhalo detection accuracy when using a probability threshold. The results are quite similar to those without the probability threshold in Tab. I. The probability threshold defaults individual pixels to the background class unless the probability for a subhalo class is larger than 50%. In general, this causes a few of the subhalos which were detected without the threshold to now be not detected. However, it also causes a few to jump to an adjacent mass bin. These subhalos have are nearly equally predicted to belong to two different classes, which happens when the mass is near the edge of a mass bin. If a pixel around the edge is now assigned to the background, the secondary class may now have more predicted pixels, resulting in a changed mass.

Appendix C: How realistic is 1% white noise?

Throughout this work, we have shown how the U-Net is able to detect dark matter subhalos in strong lens images without noise and when 1% white noise is included. Framing the noise as a fraction of the mean image brightness allowed us the easily see when the network would lose sensitivity to subhalos of a certain mass (Sec. IV). However, it is reasonable to ask how realistic this is. For instance, the gravitational imaging method searches for subhalos in pixels with a signal-to-noise ratio (S/N) of at least 3. We point out that using 1% noise does not imply that all of the pixels have S/N of 100. Instead, there is wide range, with some pixels having very strong signal compared to the noise, and other pixels that are swamped by noise.

To get a better sense of this, we simulated images including a more realistic noise similar to that of the *Hubble Space Telescope*. In particular, this requires us to set the brightness of the sky, the brightness of the source, the threshold brightness, and the exposure time. We generated 100 images using an exposure of 5400 seconds, assuming

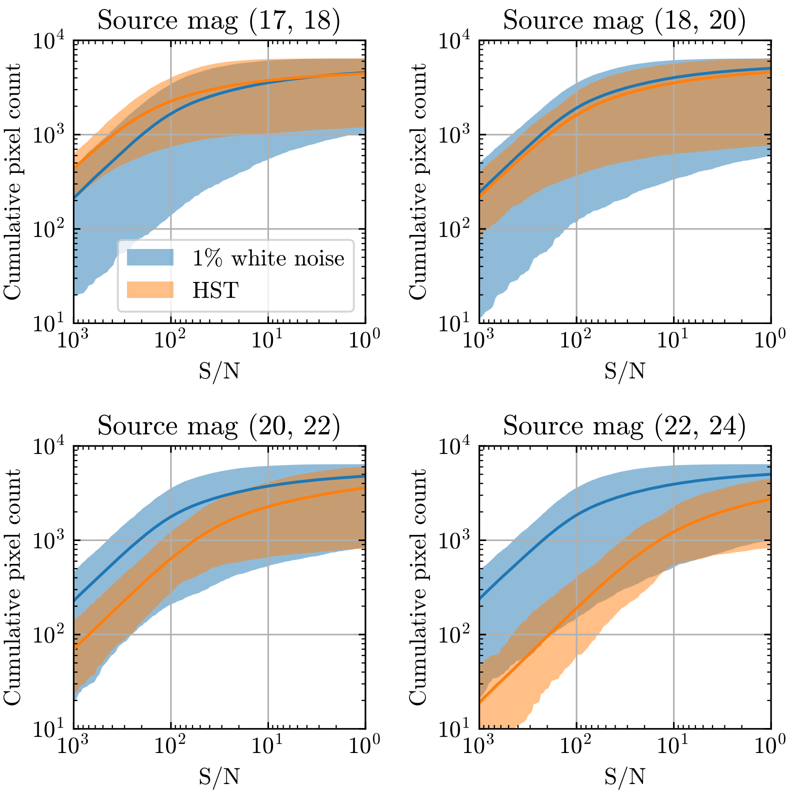

FIG. A2: The number of pixels in an 80×80 pixel strong lens image that have signal-to-noise (S/N) larger than the number on the x-axis. The blue denotes using Gaussian white noise with a standard deviation 1% of the mean image brightness. The orange is the result of using an HST-like simulation. The bands show the range over 100 images. For bright sources (lower magnitude), the HST-like noise results in larger S/N than 1% Gaussian white noise. In dimmer sources, the 1% Gaussian white noise is an optimistic assumption.

the background sky has a magnitude of 22, and that the threshold is 25.9 magnitude. For each image, we compute the signal-to-noise for each pixel and generate the cumulative distribution, starting from the highest ratio. In effect, this counts the number of pixels which have a signal-to-noise greater than a given number. These distributions are shown in Fig. A2, where the blue regions show our simple white noise assumption and the orange displays the HST-like noise. The bands span the range observed for the 100 images.

In the upper left panel, the source has a magnitude between 17 and 18, which is much brighter than the sky. This implies that the Poisson shot noise for the HST-like scenario dominates over the white noise. The orange band is near the top of or above the blue-band; there are more pixels with large S/N for HST-like noise than if we use 1% white noise, making substructure easier to detect. The upper right panel shows that our approximation is very similar to HST for sources with magnitudes between 18 and 20. The source light in the bottom left panel is now comparable with the background sky. The blue band is above the orange band, indicating that our noise approximation is optimistic for such sources. Finally, the lower right panel has very dim sources. In these images, the noise from the sky dominates and is much larger than our 1% estimate.

As a proof-of-concept, we used 1% Gaussian white noise, which yields similar noise profiles as HST for sources brighter than magnitude 20. This reduced a number of variables in our simulation, such as the source brightness. To apply image segmentation to real strong lens images, one should include all of these considerations.

[1] S. Dodelson and L. M. Widrow, Phys. Rev. Lett. **72**, 17 (1994), arXiv:hep-ph/9303287 [hep-ph].
[2] P. Bode, J. P. Ostriker, and N. Turok, ApJ **556**, 93 (2001), arXiv:astro-ph/0010389 [astro-ph].
[3] D. Hooper, M. Kaplinghat, L. E. Strigari, and K. M. Zurek, Phys. Rev. D **76**, 103515 (2007), arXiv:0704.2558 [astro-ph].
[4] S. Tulin and H.-B. Yu, Phys. Rep. **730**, 1 (2018), arXiv:1705.02358 [hep-ph].
[5] W. Hu, R. Barkana, and A. Gruzinov, Phys. Rev. Lett. **85**, 1158 (2000), arXiv:astro-ph/0003365 [astro-ph].
[6] L. Hui, J. P. Ostriker, S. Tremaine, and E. Witten, Phys. Rev. D **95**, 043541 (2017), arXiv:1610.08297 [astro-ph.CO].
[7] A. Schneider, R. E. Smith, A. V. Macciò, and B. Moore, Mon. Not. R. Astron. Soc. **424**, 684 (2012), arXiv:1112.0330 [astro-ph.CO].
[8] S. Bose, W. A. Hellwing, C. S. Frenk, A. Jenkins, M. R. Lovell, J. C. Helly, and B. Li, Mon. Not. R. Astron. Soc. **455**, 318 (2016), arXiv:1507.01998 [astro-ph.CO].
[9] N. Banik, J. Bovy, G. Bertone, D. Erkal, and T. J. L. de Boer, arXiv e-prints , arXiv:1911.02663 (2019), arXiv:1911.02663 [astro-ph.GA].
[10] D. Gilman, S. Birrer, A. Nierenberg, T. Treu, X. Du, and A. Benson, Mon. Not. R. Astron. Soc. **491**, 6077 (2020), arXiv:1908.06983 [astro-ph.CO].
[11] M. J. Rees and J. P. Ostriker, Mon. Not. R. Astron. Soc. **179**, 541 (1977).
[12] G. Efstathiou, Mon. Not. R. Astron. Soc. **256**, 43P (1992).
[13] A. Kravtsov, Advances in Astronomy **2010**, 281913 (2010), arXiv:0906.3295 [astro-ph.CO].
[14] V. Bromm, Reports on Progress in Physics **76**, 112901 (2013), arXiv:1305.5178 [astro-ph.CO].
[15] J. I. Read, G. Iorio, O. Agertz, and F. Fraternali, Mon. Not. R. Astron. Soc. **467**, 2019 (2017), arXiv:1607.03127 [astro-ph.GA].
[16] A. Fitts, M. Boylan-Kolchin, O. D. Elbert, J. S. Bullock, P. F. Hopkins, J. Oñorbe, A. Wetzel, C. Wheeler, C.-A. Faucher-Giguère, D. Kereš, E. D. Skillman, and D. R. Weisz, Mon. Not. R. Astron. Soc. **471**, 3547 (2017), arXiv:1611.02281 [astro-ph.GA].
[17] W. H. W. Ngan and R. G. Carlberg, ApJ **788**, 181 (2014), arXiv:1311.1710 [astro-ph.CO].
[18] R. G. Carlberg, ApJ **820**, 45 (2016), arXiv:1512.01620 [astro-ph.GA].
[19] J. Bovy, Phys. Rev. Lett. **116**, 121301 (2016), arXiv:1512.00452 [astro-ph.GA].
[20] D. Erkal, V. Belokurov, J. Bovy, and J. L. Sand ers, Mon. Not. R. Astron. Soc. **463**, 102 (2016), arXiv:1606.04946 [astro-ph.GA].
[21] A. Bonaca, D. W. Hogg, A. M. Price-Whelan, and C. Conroy, ApJ **880**, 38 (2019), arXiv:1811.03631 [astro-ph.GA].
[22] R. Feldmann and D. Spolyar, Mon. Not. R. Astron. Soc. **446**, 1000 (2015), arXiv:1310.2243 [astro-ph.GA].
[23] K. Van Tilburg, A.-M. Taki, and N. Weiner, JCAP **07**, 041 (2018), arXiv:1804.01991 [astro-ph.CO].
[24] C. Mondino, A.-M. Taki, K. Van Tilburg, and N. Weiner, Phys. Rev. Lett. **125**, 111101 (2020), arXiv:2002.01938 [astro-ph.CO].
[25] S. Mishra-Sharma, K. Van Tilburg, and N. Weiner, Phys. Rev. D **102**, 023026 (2020), arXiv:2003.02264 [astro-ph.CO].
[26] S. Mao and P. Schneider, Mon. Not. R. Astron. Soc. **295**, 587 (1998), arXiv:astro-ph/9707187 [astro-ph].
[27] L. A. Moustakas and R. B. Metcalf, Mon. Not. R. Astron. Soc. **339**, 607 (2003), arXiv:astro-ph/0206176 [astro-ph].
[28] L. V. E. Koopmans, Mon. Not. R. Astron. Soc. **363**, 1136 (2005), arXiv:astro-ph/0501324 [astro-ph].
[29] S. Vegetti and L. V. E. Koopmans, Mon. Not. R. Astron. Soc. **392**, 945 (2009), arXiv:0805.0201 [astro-ph].
[30] Y. Hezaveh, N. Dalal, G. Holder, M. Kuhlen, D. Marrone, N. Murray, and J. Vieira, ApJ **767**, 9 (2013), arXiv:1210.4562 [astro-ph.CO].
[31] S. Vegetti, L. V. E. Koopmans, A. Bolton, T. Treu, and R. Gavazzi, Mon. Not. R. Astron. Soc. **408**, 1969 (2010), arXiv:0910.0760 [astro-ph.CO].
[32] S. Vegetti, D. J. Lagattuta, J. P. McKean, M. W. Auger, C. D. Fassnacht, and L. V. E. Koopmans, Nature **481**, 341 (2012), arXiv:1201.3643 [astro-ph.CO].
[33] E. Ritondale, S. Vegetti, G. Despali, M. W. Auger, L. V. E. Koopmans, and J. P. McKean, Mon. Not. R. Astron. Soc. **485**, 2179 (2019), arXiv:1811.03627 [astro-ph.CO].
[34] S. Vegetti, L. V. E. Koopmans, M. W. Auger, T. Treu, and A. S. Bolton, Mon. Not. R. Astron. Soc. **442**, 2017 (2014), arXiv:1405.3666 [astro-ph.GA].
[35] N. Dalal and C. S. Kochanek, ApJ **572**, 25 (2002), arXiv:astro-ph/0111456 [astro-ph].
[36] S. Vegetti, G. Despali, M. R. Lovell, and W. Enzi, Mon. Not. R. Astron. Soc. **481**, 3661 (2018), arXiv:1801.01505 [astro-ph.CO].
[37] Q. E. Minor, M. Kaplinghat, and N. Li, ApJ **845**, 118 (2017), arXiv:1612.05250 [astro-ph.GA].
[38] Y. Hezaveh, N. Dalal, G. Holder, T. Kisner, M. Kuhlen, and L. Perreault Levasseur, JCAP **2016**, 048 (2016), arXiv:1403.2720 [astro-ph.CO].
[39] B. J. Brewer, D. Huijser, and G. F. Lewis, Mon. Not. R. Astron. Soc. **455**, 1819 (2016), arXiv:1508.00662 [astro-ph.IM].
[40] F.-Y. Cyr-Racine, L. A. Moustakas, C. R. Keeton, K. Sigurdson, and D. A. Gilman, Phys. Rev. D **94**, 043505 (2016), arXiv:1506.01724 [astro-ph.CO].
[41] S. Birrer, A. Amara, and A. Refregier, JCAP **2017**, 037 (2017), arXiv:1702.00009 [astro-ph.CO].
[42] T. Daylan, F.-Y. Cyr-Racine, A. Diaz Rivero, C. Dvorkin, and D. P. Finkbeiner, ApJ **854**, 141 (2018), arXiv:1706.06111 [astro-ph.CO].
[43] A. Diaz Rivero, F.-Y. Cyr-Racine, and C. Dvorkin, Phys. Rev. D **97**, 023001 (2018), arXiv:1707.04590 [astro-ph.CO].

[44] A. Díaz Rivero, C. Dvorkin, F.-Y. Cyr-Racine, J. Zavala, and M. Vogelsberger, Phys. Rev. D **98**, 103517 (2018), arXiv:1809.00004 [astro-ph.CO].
[45] S. Brennan, A. J. Benson, F.-Y. Cyr-Racine, C. R. Keeton, L. A. Moustakas, and A. R. Pullen, Mon. Not. R. Astron. Soc. **488**, 5085 (2019), arXiv:1808.03501 [astro-ph.GA].
[46] Y. D. Hezaveh, L. Perreault Levasseur, and P. J. Marshall, Nature **548**, 555 (2017), arXiv:1708.08842 [astro-ph.IM].
[47] L. Perreault Levasseur, Y. D. Hezaveh, and R. H. Wechsler, ApJ **850**, L7 (2017), arXiv:1708.08843 [astro-ph.CO].
[48] W. R. Morningstar, Y. D. Hezaveh, L. Perreault Levasseur, R. D. Blandford, P. J. Marshall, P. Putzky, and R. H. Wechsler, arXiv e-prints , arXiv:1808.00011 (2018), arXiv:1808.00011 [astro-ph.IM].
[49] W. R. Morningstar, L. Perreault Levasseur, Y. D. Hezaveh, R. Blandford, P. Marshall, P. Putzky, T. D. Rueter, R. Wechsler, and M. Welling, ApJ **883**, 14 (2019), arXiv:1901.01359 [astro-ph.IM].
[50] M. Chianese, A. Coogan, P. Hofma, S. Otten, and C. Weniger, arXiv e-prints , arXiv:1910.06157 (2019), arXiv:1910.06157 [astro-ph.CO].
[51] A. Diaz Rivero and C. Dvorkin, Phys. Rev. D **101**, 023515 (2020), arXiv:1910.00015 [astro-ph.CO].
[52] J. Brehmer, S. Mishra-Sharma, J. Hermans, G. Louppe, and K. Cranmer, ApJ **886**, 49 (2019), arXiv:1909.02005 [astro-ph.CO].
[53] S. Varma, M. Fairbairn, and J. Figueroa, arXiv e-prints , arXiv:2005.05353 (2020), arXiv:2005.05353 [astro-ph.CO].
[54] S. Alexander, S. Gleyzer, E. McDonough, M. W. Toomey, and E. Usai, ApJ **893**, 15 (2020), arXiv:1909.07346 [astro-ph.CO].
[55] B. Ostdiek, A. D. Rivero, and C. Dvorkin, to appear (2020).
[56] O. Ronneberger, P. Fischer, and T. Brox, arXiv e-prints , arXiv:1505.04597 (2015), arXiv:1505.04597 [cs.CV].
[57] S. Birrer and A. Amara, Physics of the Dark Universe **22**, 189 (2018), arXiv:1803.09746 [astro-ph.CO].
[58] S. Birrer, A. Amara, and A. Refregier, ApJ **813**, 102 (2015), arXiv:1504.07629 [astro-ph.CO].
[59] R. Kormann, P. Schneider, and M. Bartelmann, A&A **284**, 285 (1994).
[60] P. Ade *et al.* (Planck), Astron. Astrophys. **594**, A13 (2016), arXiv:1502.01589 [astro-ph.CO].
[61] J. F. Navarro, C. S. Frenk, and S. D. M. White, ApJ **462**, 563 (1996), arXiv:astro-ph/9508025 [astro-ph].
[62] V. Springel, J. Wang, M. Vogelsberger, A. Ludlow, A. Jenkins, A. Helmi, J. F. Navarro, C. S. Frenk, and S. D. M. White, Mon. Not. R. Astron. Soc. **391**, 1685 (2008), arXiv:0809.0898 [astro-ph].
[63] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, arXiv e-prints , arXiv:1807.05511 (2018), arXiv:1807.05511 [cs.CV].
[64] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, arXiv e-prints , arXiv:2001.05566 (2020), arXiv:2001.05566 [cs.CV].
[65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019) pp. 8024–8035.
[66] S. Ioffe and C. Szegedy, arXiv e-prints , arXiv:1502.03167 (2015), arXiv:1502.03167 [cs.LG].
[67] D. P. Kingma and J. Ba, arXiv e-prints , arXiv:1412.6980 (2014), arXiv:1412.6980 [cs.LG].
[68] J. Nelder and R. Mead, Comput. J. **7**, 308 (1965).
[69] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, Nature Methods **17**, 261 (2020).
[70] L. S. Collaboration, arXiv e-prints , arXiv:0912.0201 (2009), arXiv:0912.0201 [astro-ph.IM].
[71] A. Refregier, A. Amara, T. D. Kitching, A. Rassat, R. Scaramella, and J. Weller, arXiv e-prints , arXiv:1001.0061 (2010), arXiv:1001.0061 [astro-ph.IM].
[72] A. DAloisio and P. Natarajan, Monthly Notices of the Royal Astronomical Society **411**, 16281640 (2010).
[73] R. Li, C. S. Frenk, S. Cole, Q. Wang, and L. Gao, Mon. Not. Roy. Astron. Soc. **468**, 1426 (2017), arXiv:1612.06227 [astro-ph.CO].
[74] C. McCully, C. R. Keeton, K. C. Wong, and A. I. Zabludoff, Astrophys. J. **836**, 141 (2017), arXiv:1601.05417 [astro-ph.CO].
[75] G. Despali, S. Vegetti, S. D. M. White, C. Giocoli, and F. C. van den Bosch, Mon. Not. Roy. Astron. Soc. **475**, 5424 (2018), arXiv:1710.05029 [astro-ph.CO].
[76] A. c. Şengül, A. Tsang, A. Diaz Rivero, C. Dvorkin, H.-M. Zhu, and U. Seljak, Phys. Rev. D **102**, 063502 (2020), arXiv:2006.07383 [astro-ph.CO].