



Detection of Strongly Lensed Arcs in Galaxy Clusters with Transformers

Peng Jia^{1,2,3} , Ruiqi Sun¹, Nan Li⁴, Yu Song¹, Runyu Ning¹, Hongyan Wei¹, and Rui Luo⁵¹ College of Physics and Optoelectronic, Taiyuan University of Technology, Taiyuan, 030024, People's Republic of China; robinmartin20@gmail.com, nan.li@nao.cas.cn² Peng Cheng Lab, Shenzhen, 518066, People's Republic of China³ Department of Physics, Durham University, DH1 3LE, UK⁴ National Astronomical Observatories, Beijing, 100101, People's Republic of China; why_1221@126.com⁵ Central South University, Changsha, 410083, People's Republic of China

Received 2022 April 28; revised 2022 November 7; accepted 2022 November 8; published 2022 December 22

Abstract

Strong lensing in galaxy clusters probes properties of dense cores of dark matter halos in mass, studies the distant universe at flux levels and spatial resolutions otherwise unavailable, and constrains cosmological models independently. The next-generation large-scale sky imaging surveys are expected to discover thousands of cluster-scale strong lenses, which would lead to unprecedented opportunities for applying cluster-scale strong lenses to solve astrophysical and cosmological problems. However, the large data set challenges astronomers to identify and extract strong-lensing signals, particularly strongly lensed arcs, because of their complexity and variety. Hence, we propose a framework to detect cluster-scale strongly lensed arcs, which contains a transformer-based detection algorithm and an image simulation algorithm. We embed prior information of strongly lensed arcs at cluster scale into the training data through simulation and then train the detection algorithm with simulated images. We use the trained transformer to detect strongly lensed arcs from simulated and real data. Results show that our approach could achieve 99.63% accuracy rate, 90.32% recall rate, 85.37% precision rate, and 0.23% false-positive rate in detection of strongly lensed arcs from simulated images and could detect almost all strongly lensed arcs in real observation images. Besides, with an interpretation method, we have shown that our method could identify important information embedded in simulated data. Next, to test the reliability and usability of our approach, we will apply it to available observations (e.g., DESI Legacy Imaging Surveys⁶) and simulated data of upcoming large-scale sky surveys, such as Euclid⁷ and the China Space Station Telescope.⁸

Unified Astronomy Thesaurus concepts: [Astronomy data analysis \(1858\)](#); [Strong gravitational lensing \(1643\)](#); [Convolutional neural networks \(1938\)](#)

1. Introduction

Gravitational lensing has become a powerful probe in many areas of astrophysics and cosmology (Mao & Witt 2012; Medezinski et al. 2013; Fu & Fan 2014; Rahvar 2015; Bartelmann & Maturi 2017; Mandelbaum 2018). The phenomenon has been detected since Walsh et al. (1979) and over a wide range of scales, from megaparsec in the weak-lensing regime (Bacon et al. 2000; Hamana et al. 2003; Castro et al. 2005; Schmidt 2008; Jee et al. 2016; Kilbinger et al. 2017; Troxel et al. 2018) to kiloparsec in strong lensing (Lynds & Petrosian 1986; Soucail et al. 1987, Fort et al. 1988; Hewitt 1988; Hudson et al. 1998; Barvainis & Ivison 2002; Oldham et al. 2017; Talbot et al. 2018; Stacey & McKean 2018) and down to parsec and subparsec scales probed by microlensing (Shvartzvald et al. 2017; Bruce et al. 2017; Han et al. 2018). Therefore, gravitational lensing can be used to measure the mass distribution in the universe (Newman et al. 2013; Han et al. 2015; Diego et al. 2018; Jauzac et al. 2018), improve the study of lensed high-redshift galaxies (Coe et al. 2013;

Jones et al. 2013; Dye et al. 2015; Stark et al. 2015), and constrain cosmological parameters (Suyu et al. 2013; Liao et al. 2014; Suyu et al. 2014; Magaña et al. 2015), among other applications. Cosmological strong lensing is an extreme representation of gravitational lensing, which leads to multiple highly magnified and distorted images of background sources. In particular, highly distorted galaxies by strong lenses are called gravitational lensed arcs, which have been used in various cosmological applications (Kneib & Natarajan 2011; Meneghetti et al. 2013). For instance, the frequency of strongly lensed arcs on the sky reflects the abundance (Dalal et al. 2004; Li et al. 2006; Fedeli et al. 2007; Hilbert et al. 2007; Fedeli et al. 2010), the concentration (Oguri et al. 2012; Sereno & Covone 2013; Meneghetti et al. 2014), and astrophysical properties (Rozo et al. 2008) of massive lenses and the redshift distribution and properties of sources (Wambsganss et al. 2004; Bayliss et al. 2011; Bayliss 2012). Expectedly, strongly lensed arcs can bring more reliable restraints on astrophysical and cosmological problems with the enormous data from next-generation surveys.

So far, the strongly lensed arcs have been detected almost exclusively by visual inspection of cluster images, although automated search algorithms have recently been proposed (Lenzen et al. 2004; Horesh et al. 2005; Alard 2006; Seidel & Bartelmann 2007; Horesh et al. 2010). However, scanning wide-field data by eye covering hundreds (or even thousands) of square degrees for arcs appears to be a hopeless endeavor. Regardless of the size of the data sets, detecting arcs with human inspection may involve potential biases due to seekers'

⁶ <https://www.legacysurvey.org/>⁷ <https://www.euclid-ec.org/>⁸ <https://nao.cas.cn/csst/>

capability. Some arcs may be obstructed by bright foreground galaxies or stars, which could not be detected by human experts. In contrast, automated tools for finding arcs bring objective and reproducible definitions of arc samples, especially in the case of blind searches on the large-scale sky across multiple survey projects (Meneghetti et al. 2013; Xu et al. 2016; Li et al. 2019), as well as for the comparison between observational and simulated data (Horesh et al. 2005, 2011; Li et al. 2019).

More recently, machine-learning algorithms have become the mainstream for identifying gravitational lenses automatically (Metcalf et al. 2018), given their strong performance in the field of general image recognition. In particular, detecting the galaxy-scale strong-lensing systems has been remarkably successful (Bom et al. 2017; Ostrovski et al. 2017; Jacobs et al. 2017; Hartley et al. 2017; Petrillo et al. 2017; Lanusse et al. 2018; Avestruz et al. 2019), and the number of high-quality candidates of strong-lensing systems is over a couple of thousand by combining the data products of DES, DESI, and KIDS (Li et al. 2019; Simon 2019; Huang et al. 2020; Li et al. 2020; Huang et al. 2021). Notably, most of the above studies are based on stamps centered at galaxies, which is reasonable for galaxy-scale lenses but unsuitable for strong lensing in galaxy clusters (cluster galaxies strong lenses; CGSLs), for the following three reasons:

1. Since the number of CGSLs is too small, it is not possible to solely use these known CGSLs as the training set.
2. Since not all CGSLs are around the centers of galaxy clusters (Meneghetti et al. 2020), we would lose lots of CGSLs that do not have brightest cluster galaxies (BCGs), if we use BCGs as the prior condition for detection of CGSLs.
3. Since CGSLs are rare and may extend to very large scale, if we split full-frame images to patches of images with smaller size for classification, we would obtain a lot of false-positive detection results, even when the false-positive rate is low.

Therefore, we propose a framework to detect CGSLs, which contains two parts: a detection algorithm and a simulation algorithm. The simulation algorithm would embed prior information of CGSLs known by scientists into images in the training data set. After training with simulated images, the detection algorithm could detect CGSLs from full-frame observation images across a large field of view without the centering and cutting-out process. With these two algorithms, we could detect CGSLs according to prior information provided by scientists.

For the image simulation part, we assume that arcs are features of CGSLs and we could use these features to detect CGSLs. We use the PICS (Li et al. 2016) to generate ideal images that contain CGSLs according to extragalactic catalog CosmosDC2 (Korytov et al. 2019) and their corresponding labels (mask matrix with the same size of simulated images) in the training set. Besides, we have also generated images that do not contain CGSLs and zero labels (zero matrix with the same size of simulated images) in the training set. The proportion of CGSLs in the real observation images is very small, which will affect the training efficiency. Therefore, we increased the CGSLs ratio during training and set it to 50%. For simulated images in the validation set, we also set the ratio of images that contain CGSLs to 50% to evaluate the performance of our algorithm. Besides, we would also generate simulated images with 1% of them containing CGSLs as a test set to test the performance of our algorithm in real applications.

For the detection algorithm, the convolutional neural network (CNN) is widely used as the basic structure. However, the performance of the CNN-based detection algorithm is limited by the receptive field of the convolutional kernel. For targets with variable scales and complex structures, the performance of CNN-based detection algorithms would be limited. In a previous paper, we found that for classification of simple point-like or streak-like astronomical targets observed by wide-field small-aperture telescopes, a trained recurrent neural network (RNN) could have better performance than the CNN (Jia et al. 2019). The RNN has a sequence structure. After training, the RNN may better capture features with larger size. However, the RNN treats the whole image as a long sequence, which would require a lot of GPU memory and last a long time during the training stage. Thanks to Vaswani et al. (2017), an attention-based neural network transformer has been proposed. The transformer does not need sequence-aligned recurrent architecture, which makes it easier to train, even with a large number of parameters. The transformer is first used for natural language processing and then for image processing. For target detection tasks, the DETection TRansformer (DETR) is widely studied since it was proposed by Carion et al. (2020). The DETR uses the transformer to reason about the relations between objects to be detected and the global image context to directly predict positions and types of targets. The mechanism of the DETR is similar to that of human attention, which would extract semantic information of images for detection and could achieve better performance in detection of complex and extended targets.

CGSLs are a type of celestial objects with complex and extended shapes, which contain some front galaxies in the center and some arc structures around the center. These features could be used as semantic information for attention-based source detection algorithms. In the algorithm developed by Thuruthipilly et al. (2021), features from attention-based encoder layers are extracted for classification of SGLs at galaxy scale from candidate images and have achieved better performance than those of CNN-based methods. Since CGSLs have much larger size and more irregular shape than SGLs at galaxy scale, we propose to integrate the DETR and the Deformable DETR (Zhu et al. 2020) with an ensemble learning strategy to build a detection algorithm with better performance. Because detection results would be checked by human scientists for further study, the true positive rate should be high and the recall rate should be moderate. Therefore, we further propose a two-step strategy for detection of CGSLs to increase its performance in real applications.

For the validation set with half of the total images containing CGSLs, our method could achieve more than 88% recall rate and more than 70% precision rate, when we directly use our method with an intersection over union (IOU) of 0.1 and a score of 0.7. Considering that many CGSLs contain BCGs and they could not be detected even by human inspections, the recall rate and the precision rate are acceptable. For the test set with 1% of total images containing CGSLs, our two-step detection strategy could achieve 99.63% accuracy and 0.23% false-positive rate, when the recall is 90.32% and the precision is 85.37%. We further use real observation images from the Hubble Space Telescope Frontier Fields project, the Hubble Space Telescope RELICS project, and the early release image from the James Webb Space Telescope to detect CGSLs. For real observation images, we find that almost all CGSLs could

be detected by our algorithm, except several false detection results brought by diffraction rings in these images, which are not included in the training data. Finally, we use the interpretation method to show that our detection algorithm could focus on important features (arcs) of CGSLs.

In Section 2, we first describe the simulation and data processing method to obtain training data for our algorithm. Then, we will analyze the detection requirements of CGSLs and adopt our evaluation criterion for the detection algorithm. The basic structure and the performance of the DETR are shown in Section 3. Section 4 describes a comparative investigation between DETR and Deformable DETR, including training and detection performance. In Section 5, we explore the correlation between the features of images and the detection performance with a machine-learning-interpretation module. In Section 6, we will show the performance of our algorithm in deploying it with simulated and real observation data. Finally, discussions and conclusions are delivered in Section 7.

2. Data Preparation Procedure and Evaluation Criterion for Detection of CGSLs

As we have discussed in Section 1, only tens of CGSLs have been discovered so far. The number is too small to be used as the training set. Besides, many CGSLs are discovered by the visual system of human beings, which would introduce statistical bias into training data. A neural network trained with these data may only be possible to find “similar” CGSLs. Besides, considering detection of CGSLs from images of multiple bands is beyond the capacity of human vision systems; a lot of CGSLs would be lost by a detection neural network that is trained by data obtained by human vision systems. Therefore, we use simulation data to train the neural network, which could embed prior information about scientists’ understanding of CGSLs into images in the training set. Meanwhile, we could also enlarge the volume and the diversity of the training data with the simulation algorithm. After training, the detection algorithm would be able to detect CGSLs that satisfy scientists’ prior assumption about properties of CGSLs and are missed by human vision inspections.

We will briefly introduce the simulation method in Section 2.1 and introduce the data processing strategy to generate training data in Section 2.2. Besides, although the mean average precision (mAP) is widely used as an evaluation criterion for general target detection algorithms, for CGSL detection tasks we will show that it would be better to use precision and recall under a predefined IOU as the evaluation criterion. In Section 2.3, we will discuss the evaluation criterion for the CGSL detection algorithm.

2.1. Simulation of CGSLs

To train and evaluate neural network models for detection of CGSLs, we have created an ideal synthetic data set without point-spread functions (PSFs) and noise using a simulation pipeline named PICS (Li et al. 2016). Similar to Madireddy et al. (2019), the simulation of CGSLs in this paper comprises six steps: (1) create populations of lenses and sources according to the given statistical properties of CGSLs, (2) build mass and light models of foreground lenses, (3) calculate deflection fields of the lenses, (4) construct light profiles of background source galaxies, (5) run ray-tracing simulations to create strongly lensed images based on the deflection fields and light profile of

sources, and (6) stack the lensed images of lensed arcs and images of galaxies on the line of sight, as well as the foreground images of lenses.

The populations of lenses and sources are built based on a state-of-the-art extragalactic catalog called CosmoDC2 (Korytov et al. 2019). CosmoDC2 provides a catalog of galaxy clusters, including the virial mass of dark matter halos and the apparent magnitudes, axis ratios, position angles, and redshifts of member galaxies. The mass model of a lens galaxy cluster is modeled as a dark matter halo plus a set of member galaxies. The mass model of a dark matter halo is elliptical NFW, and it requests virial mass, concentration parameter, and ellipticity. CosmoDC2 gives the virial mass, concentrations are calculated according to the c - M relation given by Child et al. (2018), and ellipticity is obtained by measuring the ellipticity of the spatial distribution of member galaxies in the cluster. Hence, the deflection angle map due to the dark matter halo can be described by M_{vir} , c_{vir} , q_{nfw} , z_l , and z_s , where z_l and z_s are the redshifts of the lens plane and source plane, respectively.

The mass model of member galaxies is a singular isothermal ellipsoid (SIE) as adopted in Collett (2015), since SIE is analytically tractable and consistent with models of individual lenses and lens statistics on length scales relevant for strong lensing (Koopmans et al. 2006; Gavazzi et al. 2007; Dye et al. 2008). Accordingly, the deflection maps due to member galaxies can be defined by positions, velocity dispersions, axis ratios, position angles, and redshifts of member galaxies, as well as redshifts of source galaxies, namely, x_1 , x_2 , σ_v , q_b , ϕ_b , z_b , and z_s . The parameters x_1 , x_2 , q_b , ϕ_b , z_b , and z_s are taken directly from the CosmoDC2 catalog. σ_v is derived from the L - σ scaling relation from the bright sample of Parker et al. (2007) given by

$$\sigma_v = 142(L/L_{\text{star}})^{1/3l} \text{ km s}^{-1}, \quad (1)$$

where $\log 10(L/L_{\text{star}}) = -0.4(\text{mag}_r - \text{mag}_*)$, with mag_r the apparent r -band magnitude of the galaxy given by the CosmoDC2 catalog. We adopt the assumption in More et al. (2016) that mag_* evolves with redshift as $\text{mag}_* = +1.5(z - 0.1) - 20.44$ (Faber et al. 2007). To guarantee significant lensing features, we set $z_s > z_l + 0.5$ and then randomly choose galaxies satisfying this criterion. The projected positions of sources in the lensing system are randomly chosen in the area where lensing magnifications are larger than 20 on the source plane. The light profiles of galaxies in the light cone are all modeled as a composite Sérsic profile containing bulges and disks, and all the parameters are from the CosmoDC2. For nonlensed galaxies, including member galaxies and line-of-sight galaxies, the images are rendered with composite Sérsic profiles on regular grids directly; for lensed arcs, the images are rendered with composite Sérsic profiles on ray-traced grids.

The final simulated data are generated by stacking images of background galaxies and images of generated CGSLs. We also provide corresponding masks of lensed arcs in image stamps to generate labels for the training data. Each simulated image includes four channels. The first three channels are images of g , r , and i bands, and the last channel is the mask of lensed arcs. The size of simulated images is 1280×1280 pixels, and CGSLs are mostly in the center of these simulated images. Besides, we have also generated images that contain galaxies without CGSLs. For these images, the first three channels are images of g , r , and i bands, and the last channel is a zero matrix

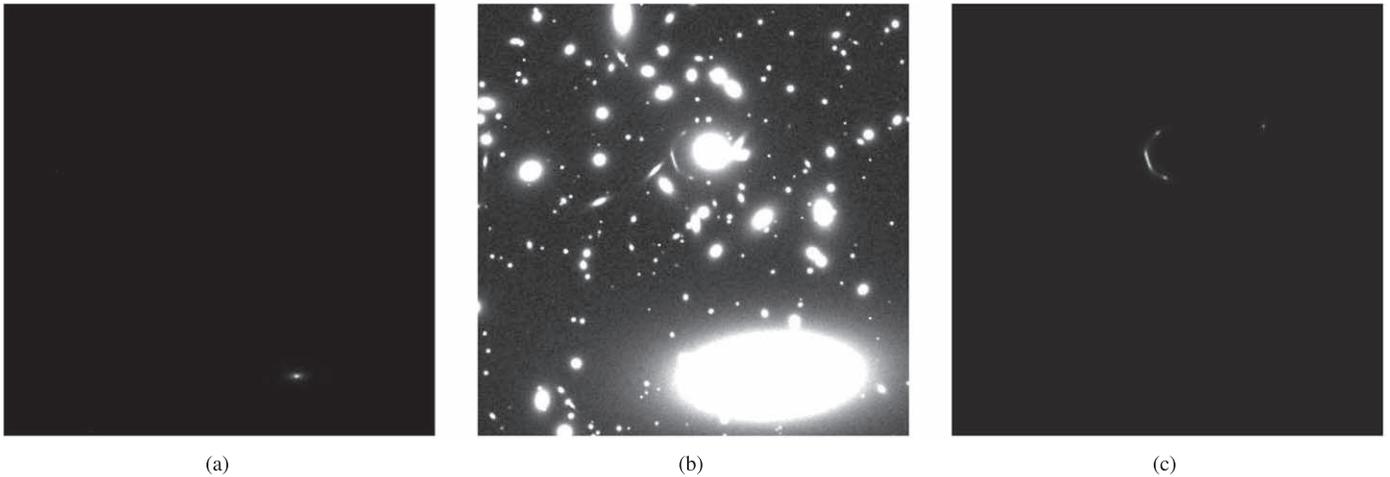


Figure 1. This figure shows (a) the original simulated image, (b) the image after grayscale transformation, and (c) the mask of lensed arcs. As we can see, the structure of this image is almost invisible in the original image. After grayscale transformation, we can easily see various galaxy structures and even directly observe the CGSL, which would be beneficial to development of the algorithm. We can see that the mask of lensed arcs ignores the foreground and shows the boundary of the CGSL, which is an arc in this image.

with the same size of images in the first three channels. Some additional data processing is required to generate training data from these simulated images, which we will discuss in Section 2.2.

2.2. Generation of Training Data from Simulated CGSL Images

The detection method proposed in this paper is a supervised learning algorithm. For a supervised learning algorithm, training data should include both its inputs (images of CGSLs) and its outputs/labels (positions and size of CGSLs). We first need to generate labels and images from simulated data. Contemporary detection algorithms use bounding boxes to indicate positions and sizes of targets, and bounding boxes are circumscribed rectangles of targets. However, masks of CGSLs are two-dimensional mask images with the same size as that of simulated images. Therefore, we would transform grayscale values of masks of lensed arcs with log transformation. Then, pixels with grayscale values larger than 10^{-3} will be set as part of targets, and other pixels in the mask images will directly be set as backgrounds, as shown in Figure 1. Then, we would generate circumscribed rectangles for pixels belonging to CGSLs as bounding boxes.

For inputs of CGSL detection algorithms, we have noticed that the difference between grayscale values in simulated images is large and it is hard to see features of CGSLs directly from simulated images. Generally grayscale transformation such as `zscale` in DS9 (Smithsonian Astrophysical Observatory 2000) would make structures of CGSLs easier to detect. Therefore, we would transform gray scales of simulated images with `zscale` transformation. Although neural networks could learn the grayscale transformation algorithm, it would be easier and faster to train neural networks, if we have made grayscale transformation before training.

After grayscale transformation, images of the same target in different bands will be stored as an image with several channels. Besides, since almost all strong-lensing targets are in the center of simulated images, it would introduce strong bias into the detection algorithm during the training stage. The detection algorithm would be more likely to predict positions of CGSLs in the center of these images. Therefore, we generate a

window of 400×400 pixels and randomly shift the window in the original images to cut stamp images for detection. With this method, there are some images without any CGSLs or only with part of CGSLs. Besides, many CGSLs would be distributed in different parts of these stamp images. Figure 2 shows several images with and without CGSLs. It should be noted that we have generated images with a size of 400×400 pixels as the training set to reduce the requirement of GPU memory during the training stage. Since the DETR and the Deformable DETR could accept images of any size as inputs, we would directly detect CGSLs from full-frame images, as long as we have enough GPU memory during the deployment stage. Finally, all images will be saved as PNG files, with three channels as inputs of the neural network, as shown in Figure 2. It should be noted that the difference of grayscale values in some pixels of original images could be quite large, which would introduce difficulties in development of detection algorithms. Transforming original simulated images to PNG files, whose grayscale values are integers within the scale of 0–255, can constrain data distribution and make our algorithm easier to train at the cost of low detection ability of dim targets and higher position regression error.

2.3. Performance Evaluation Criterion for Detection of CGSLs

The performance evaluation criterion of the detection algorithm is important for the algorithm development. Although the mAP is widely used to evaluate the performance of a general-purpose detection algorithm, it is better to use an appropriate evaluation criterion, according to real detection requirements. Hence, we select the precision rate and the recall rate under a predefined IOU ratio as the performance evaluation criterion. The IOU is defined as the ratio between the overlap area and the union area of the bounding box from the detection results and the labels:

$$\text{IOU} = \frac{\text{Intersection Area}}{\text{Union Area}}. \quad (2)$$

If the IOU is larger than a predefined criterion, we set the detection as a true positive detection (TP). Otherwise, we set the detection result as a false negative (FN) or a false-positive (FP) detection. Then, we would further define precision and

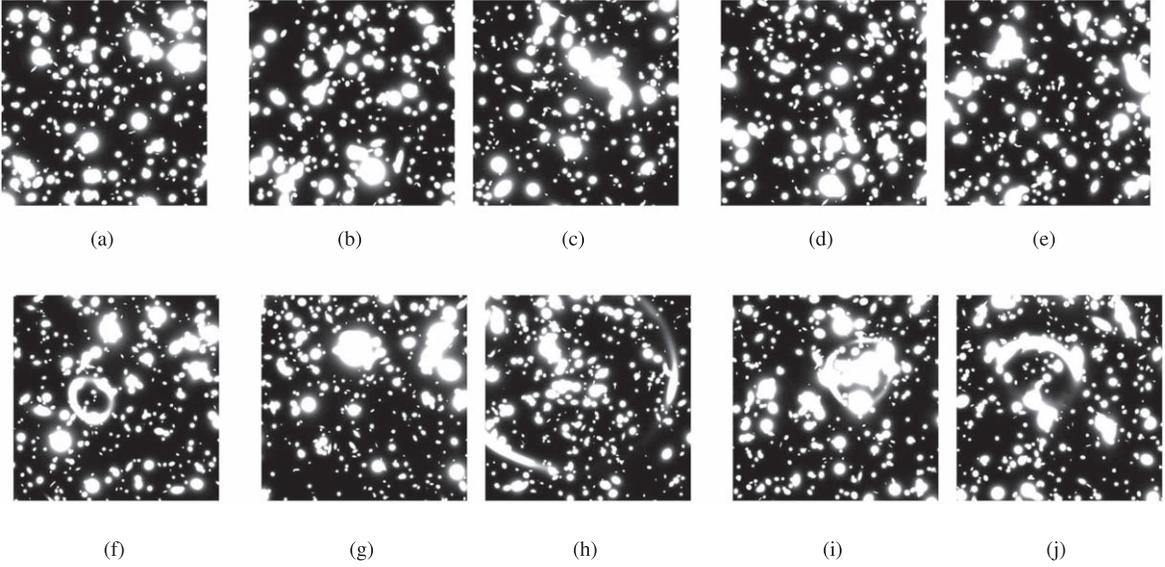


Figure 2. This figure shows several images without and with CGSLs. Panel (a)–(e) show images without CGSLs, and panels (f)–(j) show images with CGSLs. We find that in the images with CGSLs there are arcs, but in the images without CGSLs there are no similar structures.

recall rate to evaluate the performance of a detection algorithm with TP, FP, and FN. The precision rate is the percentage of true positive detection results to all detection results, and the recall rate is the percentage of true positive detection results to all targets. The precision and recall rates are defined as

$$\begin{aligned} \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \\ \text{Recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}. \end{aligned} \quad (3)$$

CGSLs are very rare, normally less than one in a full-frame observation image. Therefore, it would be more important to discover one CGSL instead of getting the accurate position of the CGSL in observation images. Therefore, we could first set a small value of IOU to define the precision and recall rates, such as 0.1 in this paper. If we set the IOU to be 0.1, the overlap between the detection result (bounding box) and the true result (true position and size of the CGSL) would be larger than 10% of the true result. Therefore, the detection result would roughly indicate the position of the CGSL. Then, we need to further use the segmentation algorithm to obtain CGSLs from detection results. For the image segmentation algorithm, we need to expand the size of the bounding box by 10 times to make sure the whole image of the CGSL is in the bounding box.

3. Detection of CGSLs with the DETR

As mentioned in Section 1, CGSLs have extended and complex structures and would often be obstructed by BCGs, as shown in Figures 2(g), 7(a), 14(g), and 14(h). In this section, we describe our strategy to use the DETR to detect CGSLs. We give a brief introduction to the DETR in Section 3.1. We further describe the training strategy and show the performance of the DETR in detecting CGSLs in Section 3.2.

3.1. Introduction of the DETR

The data flowchart of the DETR is shown in Figure 3. When we input an image into the DETR, its features of different pixels will first be extracted by the backbone neural network to form different feature vectors. It should be noted that since

these feature vectors are related to different pixels in the image, feature vectors of adjacent pixels would have similarities. However, feature vectors are abstract representations of contents by the backbone neural network, so it does not include the corresponding position information. Therefore, feature vectors and their corresponding position vectors will be connected together as a one-dimensional vector and sent to the transformer. In the transformer, these feature vectors with position information would be processed again by means of the attention mechanism. Finally, results will be sent to a feed-forward propagation (FFP) neural network for target position and type prediction.

The structure of the DETR used in this paper is shown in Figure 4. We propose to use the Resnet-50 as the backbone neural network for feature extraction because it has a residual structure, which makes it easier to train (He et al. 2016). Besides, the Resnet-50 could gain high accuracy even with very deep structure. The feature vectors extracted from the backbone neural network would be sent to the transformer along with the position information of these feature vectors. The position vectors that represent position information of these feature vectors are encoded by

$$\begin{aligned} PE(x, y)_{2i} &= \sin((x, y) / 1000^{2i/d_{\text{model}}}) \\ PE(x, y)_{2i+1} &= \cos((x, y) / 1000^{2i/d_{\text{model}}}), \end{aligned} \quad (4)$$

where $(x, y) \in [0, 1]^2$ is the normalized position of the current feature vector in the image and d_{model} is the dimension of position vectors, which has the same size as that of the feature vector or the dimension of the hidden state of the DETR. Parameter i represents the location index of each small piece at different positions in the current feature vector. We encode these pieces in even position and odd position with sin and cos functions, respectively, and finally obtain the position vector $PE(x, y)$ of the feature vector at (x, y) with d_{model} dimensions. Finally, we would directly add the position information into the corresponding feature vector.

The transformer has an encoder–decoder structure as shown in Figure 5. The encoder part is designed based on the attention

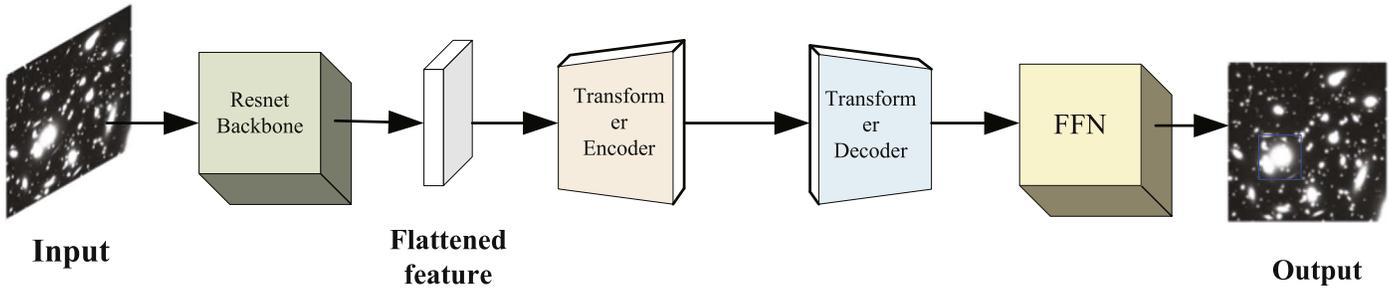


Figure 3. The data flowchart of the DETR. It includes an encoder and decoder part that could learn features of extended targets for detection.

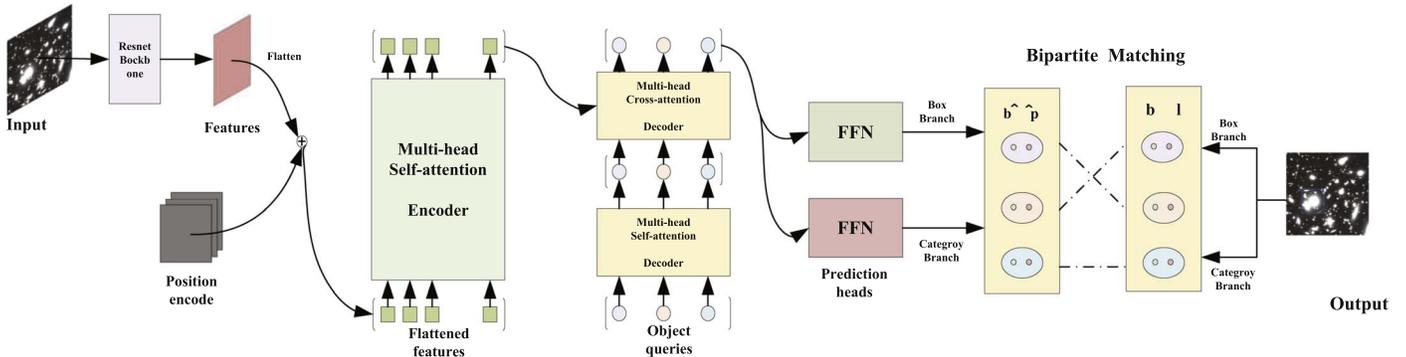


Figure 4. The structure of the DETR. The input image is first extracted by the backbone neural network to obtain features. After adding the encoded position information of feature vectors, feature vectors are sent to an encoder–decoder structure. The output decoded vectors will get the category and the position information of the target in the input image through two FFN branches. The last bipartite-matching structure of the network is to match the prediction output with the label of the input to help the network to achieve the correct prediction result. The variables \hat{p} and \hat{b} represent the category and position box predicted by the algorithm, respectively, l and b represent the real category and position box of the target, respectively, and lines represent the process of cross-matching between the predicted results and real labels of this bipartite-matching part. Each oval is an object, which could be either the background or a target. The circles inside the ovals represent the value of the position or category.

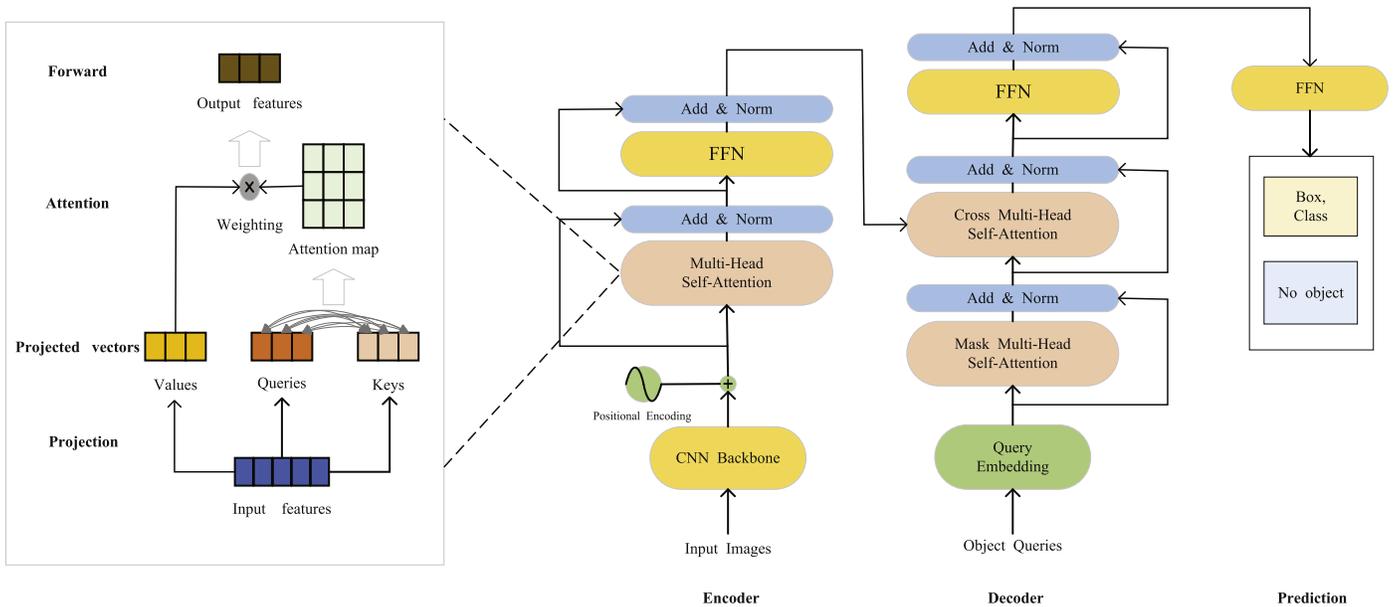


Figure 5. The encoder–decoder structure of the transformer. Features of images are first extracted by the CNN backbone, and then the multihead self-attention layer would generate attention of these features in the encoder. In the decoder, several object queries will sample features automatically, and the mask multihead self-attention layer will process these features with the self-attention mechanism. Finally, in the multihead cross-attention layer, features will further be processed to generate cross-attention to detect targets.

mechanism, and it is called the self-attention layer. If a feature vector is put into the self-attention layer, it would be transformed to three vectors: Value, Query, and Key. Similar to hidden states in RNN, Value, Query, and Key are used to project feature vectors to different directions and positions and obtain relations

between features. For each feature vector, its Query will multiply Keys of all feature vectors to calculate the correlation between the current vector and all other feature vectors. Therefore, for all feature vectors, we could get a weighted map, which is called the attention map, and values in each pixel of the weighted map

stand for correlations between features. Higher correlation values between features mean that they have attracted more attention and vice versa. The attention map will propagate to a softmax layer, and outputs of the softmax layer will be multiplied by values of all feature vectors to get final outputs. The self-attention layer would naturally obtain connections between features, regardless of their distance, which could exceed the receptive field of convolutional kernels in CNN. Several attention layers will be connected together to generate a head. We would use several heads to design our detection neural network, based on the multihead mechanism. With the multihead design, we could extract more information to focus on different positions in an image.

The decoder is also designed with the concept of attention. Inputs of the decoder are random initialized vectors, which are called object queries. Object queries are similar to “anchors” in the Faster-RCNN (Ren et al. 2015). Each object query represents a target predicted by the transformer-based detection algorithm. In other words, the number of object queries is the number of objects predicted by the model, including the prediction results of the background (no object). The number of queries is fixed and would be much larger than the actual number of targets in the input image. Prediction results would be sent to the feed-forward network (FFN). In the FFN, the predictions of the model will be bipartite-matched with input labels, where a majority of queries will be classified to the background or “no object” and only a few closest to the actual target will be regarded as predictions of real targets. During the training stage, object queries will automatically sample features of objects from the training set and propagate these features to the mask multihead self-attention layer. In the mask multihead self-attention layer, features will be processed again by the self-attention mechanism. Then outputs of the mask multihead self-attention layer and outputs from the encoder will be sent to the multihead cross-attention layer. The multihead cross-attention layer is designed according to the cross-attention mechanism. There are also three vectors in the cross-attention mechanism, Value, Query, and Key, and we would carry out the same operation process as the self-attention mechanism. The difference is that the Query comes from the decoder, while the Key and the Value come from the encoder. The output from the encoder and that from the decoder are crossed, so this is called the cross-attention mechanism. In the multihead cross-attention layer, features will further be processed to generate cross-attention to better focus on interested targets.

Finally, output feature vectors will be sent to two FFNs for object classification and position regression. Different from other CNN-based detection algorithms, we would use outputs from FFN and labels to calculate the loss of the DETR. Because the total number of predictions is fixed in the DETR and more predictions would require larger memory, the DETR is better at detection of sparsely distributed large astronomical targets, such as CGSLs. With N predictions, the Hungarian algorithm is used in the last bipartite-matching structure to match N outputs from the FFN layer and M labels, resulting in the Hungarian loss between the predictions \hat{y} and the label y . The Hungarian loss is a combinatorial algorithm that is used to calculate optimal matching between prediction results and ground truth (Kuhn 1955), which is defined by

$$L_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^N [L_{\text{class}}(i, \sigma(\hat{i})) + L_{\text{box}}(i, \sigma(\hat{i}))]. \quad (5)$$

There are two types of losses in the Hungarian loss: the matching classification loss L_{class} , and the bounding box loss L_{box} . For CGSL detection tasks, outputs could be classified to be either CGSL or background. Therefore, the classification loss is defined as cross-entropy between these two classes. The bounding box loss is used to evaluate the position accuracy of the DETR outputs. L_1 loss and L_{iou} loss are both used as defined by

$$L_{\text{class}}(i, \sigma(\hat{i})) = \sum_{i=1}^N - (c_i * \log(p_{\sigma(\hat{i})}) + (1 - c_i) * \log(1 - p_{\sigma(\hat{i})})),$$

$$L_{\text{box}}(i, \sigma(\hat{i})) = \sum_{i=1}^N (\lambda_{\text{iou}} L_{\text{iou}}(b_i, b_{\sigma(\hat{i})}) + \lambda_{L_1} \|b_i - b_{\sigma(\hat{i})}\|_1), \quad (6)$$

$$L_{\text{iou}}(b_i, b_{\sigma(\hat{i})}) = 1 - \text{IOU}, \quad (7)$$

where i is the index of the target, $\sigma(\hat{i})$ is the corresponding prediction index, c_i is the value of the target’s category, and $p_{\sigma(\hat{i})}$ is the value of the prediction’s category. λ_{iou} and λ_{L_1} are hyperparameters corresponding to weights of L_{iou} and L_1 norm, respectively. Parameter b_i is the position of the target, and $b_{\sigma(\hat{i})}$ is the predicted position. L_1 loss is the absolute error between the predicted position box and the position of the real target box. In real applications, we find that targets of different size have different sensitivity to the L_1 loss. A larger target has a larger boundary box, so a small deviation will cause a large L_1 loss, while a smaller coordinate box may not cause a large L_1 loss, even if the deviation is large. To balance this, we introduce the box L_{iou} loss, which is independent of the size of the bounding box. The L_{iou} loss is defined in Equation (7), and the IOU is defined in Equation (2).

3.2. Training and Performance Evaluation of the DETR

Because the number of CGSLs is small and the DETR has relatively complex structure, it would be hard to train the DETR. Therefore, we propose the following strategies to train the model:

1. We propose to add the dropout strategy with a dropout rate of 0.1 to prevent overfitting.
2. We use instance normalization to process each channel of an image (subtract the average value of each channel and then divide the standard deviation of these images).
3. We propose to use pretrained weights to initialize the DETR to reduce training time. The pretrained weights used in this paper are obtained from Carion et al. (2020).

The DETR is implemented with the Pytorch (Paszke et al. 2019) in a computer with one Nvidia RTX 3090 GPU card. With the strategies proposed above, we train the DETR with the Adam optimizer as the optimizer (Kingma & Ba 2014). There are 2000 images in the training set and 600 images in the validation set. We set half of the images to contain CGSLs and half of them to not contain CGSLs to help us to better evaluate the performance of our algorithm. The DETR is trained with 250 epochs and takes around 25 hr. After training, we use the evaluation criterion defined in Section 2.3 to evaluate the performance of the DETR in detection of CGSLs. It would take about 0.06 s to process an image with a size of 400×400 pixels. The performance of our algorithm in detection of

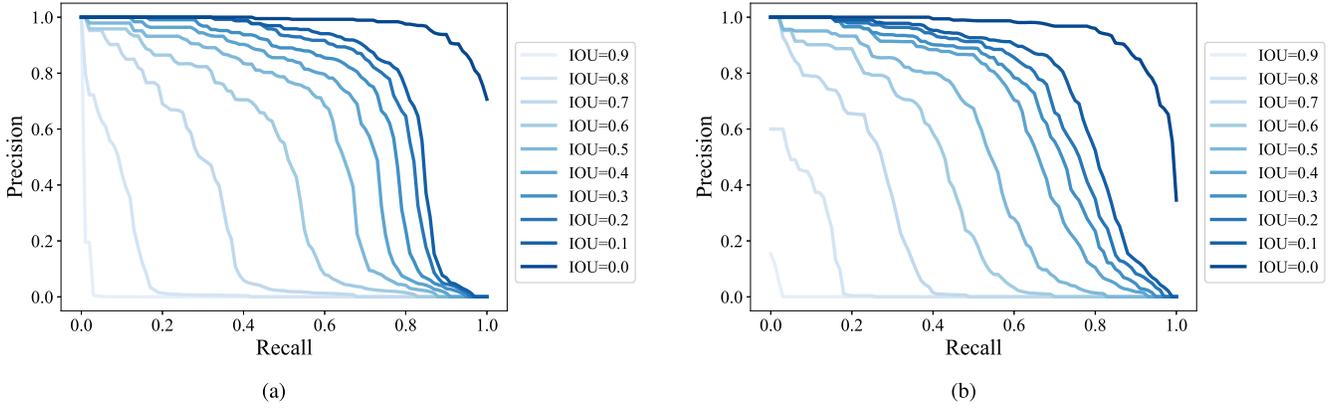


Figure 6. The performance of the DETR and the Deformable DETR in detection of CGSLs. The vertical axis represents the precision rate, and the horizontal axis represents the recall rate. The farther to the right of the curves, the higher the recall, and the higher the curves, the better the precision. We can use the area of the P - R curve to evaluate the performance. The larger the area surrounded by the curves, the better the overall performance of the algorithm. IOU represents the degree of overlap between the predicted targets and the real targets. The larger IOU is, the greater the overlap is, indicating that the location of the targets predicted by the algorithm is more accurate. We take different IOUs as the threshold, and only when the overlap degree between the targets predicted by the algorithm and the real targets is larger than the threshold do we consider the prediction to be accurate.

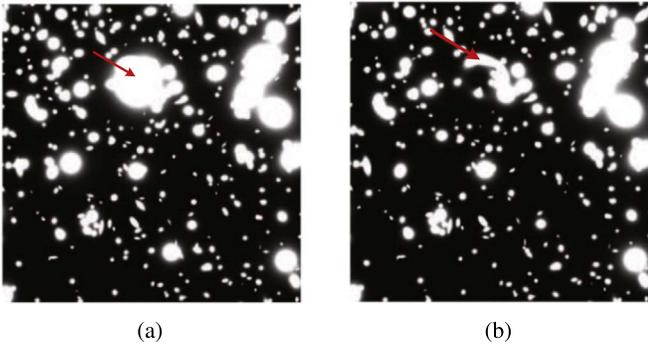


Figure 7. This figure shows a simulated image of the same strong gravitational lensing system with and without foreground central galaxies. Panels (a) and (b) show the same CSGL, but there are obvious foreground central galaxies in panel (a) and no foreground central galaxies in panel (b). We can easily see the morphological characteristics of the CSGL. The arrow in panel (a) points to the foreground central galaxies. We can see the arc of the strong gravitational lensing system after we remove foreground central galaxies, which is pointed to by the arrow in panel (b). Therefore, CSGLs with foreground central galaxies interference like in panel a pose high requirements and challenges to our detection algorithm.

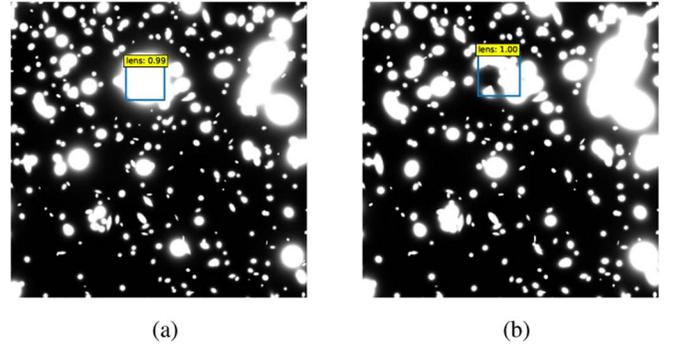


Figure 8. The results of our algorithm in detection of CSGLs with and without foreground central galaxies. We can see that our algorithm can detect CSGLs regardless of foreground galaxies.

CGSLs is shown in the left panel of Figure 6. It can be seen that the area surrounded by the P - R curve increases rapidly as the IOU threshold decreases, which indicates that the performance of our detection algorithm improves rapidly if we reduce position accuracy requirements, which is consistent with our discussions in Section 2.3.

If we set a higher IOU, the position accuracy of the detection algorithm would be higher at the cost of low recall rate. Considering that the number of CGSLs is small in real applications, we set the IOU threshold to be 0.1, which will enlarge the size of detection results by 10 times for further analyses. The area of the P - R curve can reach 0.85, when the IOU threshold is 0.1. Besides, it is worth noting that when our IOU threshold is set as 0 the detection algorithm becomes a classification algorithm, which means that our algorithm can directly identify whether there is a strong gravitational lensing system in the image regardless of its position. For the classification task, we can see that the area of the P - R curve could reach around 0.95.

Due to the existence of foreground central galaxies, many features of CGSLs, such as arcs and rings, are obstructed. Meanwhile, foreground central galaxies will also bring interference to the position prediction accuracy of our algorithm. Therefore, we further use simulated images of the same strong gravitational lensing arcs with and without foreground central galaxies to evaluate the performance of our algorithm. A pair of simulated images with and without foreground galaxies are shown in Figure 7. The performances of our algorithm in detection of CGSLs with and without foreground central galaxies are shown in Figures 8 and 9. In Figure 8, we find that our algorithm could detect the CGSL, regardless of the foreground galaxies. In Figure 9, we show the statistical results of our algorithm in detection of CGSLs with or without foreground galaxies. We find that the difference between detection results is small. Besides, we find that the performance of our algorithm does not drop significantly when there are BCGs in detection of CGSLs.

4. Detection of CGSLs with the Deformable DETR

For different sky survey projects, the DETR should be trained with simulated data with different observation conditions, such as observation bands, pixel scales, PSFs, and noise levels. However, it would take a long time to train the DETR. Therefore, we propose to use the Deformable DETR for CGSL

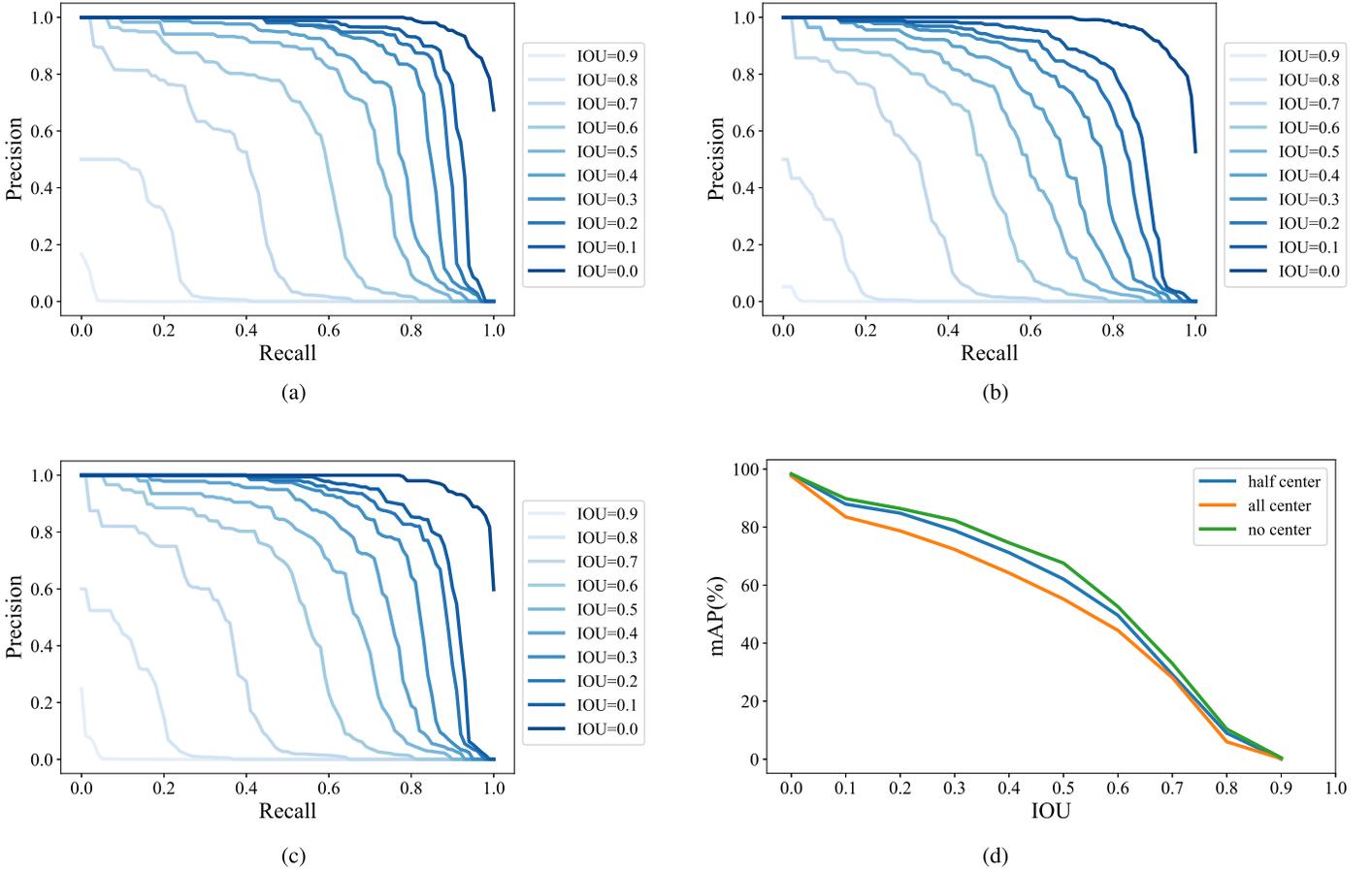


Figure 9. The performances of our algorithm in detection of CSGs with and without foreground central galaxies. Panels (a), (b), and (c) show the detection ability of our method on data sets with no foreground central galaxies, all of them with foreground central galaxies, and half of them without foreground central galaxies at different IOU thresholds, respectively. It can be seen that the model performs best when there are no foreground central galaxies and worst when there are foreground central galaxies, indicating that foreground central galaxies have a certain influence on the performance of our algorithm. In panel (d) the vertical axis represents the area of the P - R curve (mAP) for different data sets of the model, and the horizontal axis represents different IOU thresholds. It can be seen that when the IOU threshold is small or large, the difference of the mAP is small, indicating that the foreground central galaxies have little interference to the model at this time. When the IOU threshold is around 0.3, the performance difference of the mAP is the largest. When the algorithm is only used for the classification task of the strong gravitational lens, the mAP is close to 100%, which indicates its strong classification ability.

detection as a light-weighted algorithm. Besides, we further propose to use multiscale features in the Deformable DETR to increase its accuracy.

4.1. Introduction of the Deformable DETR

The DETR algorithm uses feature vectors at all positions to interact with each other for detection, which would require a long time and a lot of GPU memory. Because CGSLs are sparsely distributed (normally less than one CGSL in an observation image), it is possible to use sparse interactive operations with feature vectors of a few positions for detection. Based on this concept, we propose to use the Deformable DETR for detection. The Deformable DETR (Zhu et al. 2020) takes advantage of the sparse concept from the deformable convolution operation (Dai et al. 2017) and the spatial relation modeling concept from the transformer. The structure of the Deformable DETR is shown in Figure 10.

The structure of the Deformable DETR is similar to that of the DETR. But there are two main modifications in the Deformable DETR. First, to keep the performance of the Deformable DETR stable, we extract several instead of all feature vectors from last few layers in Resnet-50 as the multiscale feature, which is similar to the feature pyramid

network (FPN) used in celestial object detection for wide-field small-aperture telescopes (Jia et al. 2020). In this way, the model can obtain more features of input images at different scales and levels, which could help the model to obtain richer feature information. Second, we introduce the attention mechanism operation with adjacent feature vectors in the Deformable DETR. A feature vector will interact only with adjacent feature vectors, which could reduce complexity. This feature vector is called the reference vector, while the adjacent feature vectors sampled by the Deformable DETR are called sampling vectors. Meanwhile, since the CGSLs is large and sparsely distributed, sparsely distributed sampling vectors would not seriously affect the performance. Sampling features would be sent to attention layers of the decoder. After data propagation through these layers to attention layers of the decoder, multiscale attention operation would be carried out to give final results.

4.2. Performance Comparison between the DETR and the Deformable DETR

We train the Deformable DETR with the same data set as we used in Section 3.2. The optimizer and the loss function are the same as in the DETR. The model training process is shown in

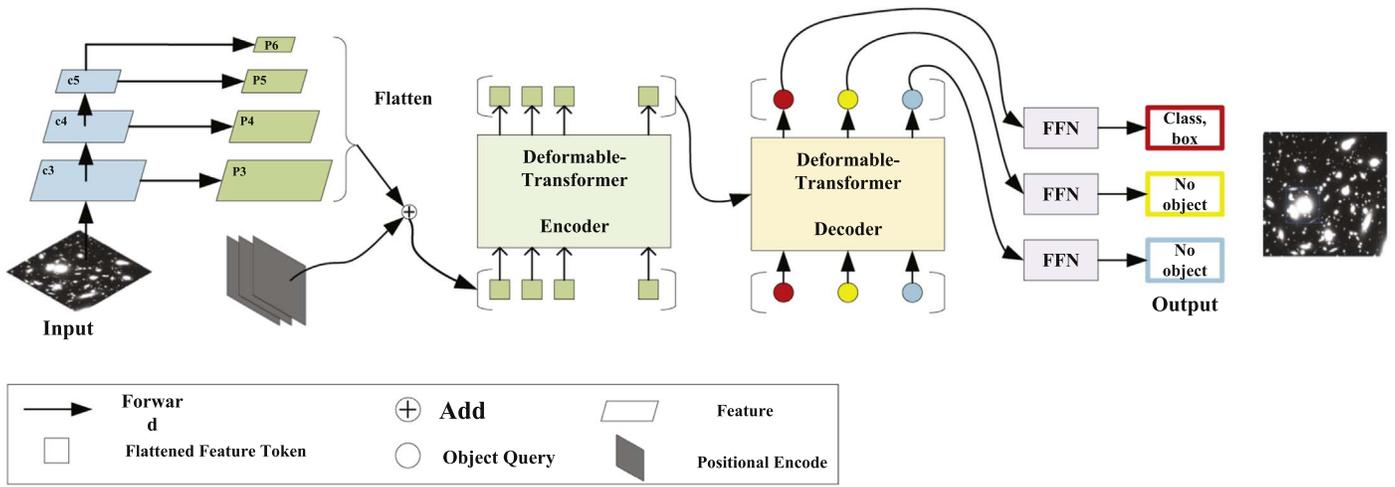


Figure 10. The schematic diagram of the Deformable DETR. This structural paradigm is the same as that of the DETR with the following differences. First of all, the backbone neural network in the Deformable DETR outputs feature vector maps c_3 , c_4 , and c_5 from its last three layers. We transform the depth (number of channels) of these three feature vector maps to the same size, thus forming three feature vector maps of p_3 , p_4 , and p_5 accordingly. Among them, we would convolve feature vector maps of c_5 with convolutional layers to further extract feature information and obtain p_6 for further processing. We would stack these feature vector maps together to form a multiscale feature, similar to the feature pyramid network (FPN). Second, in the encoder–decoder part, the attention mechanism is improved by introducing reference vectors and sampling vectors. Only the attention map between reference vectors and sampling vectors is calculated, thus greatly reducing the computation complexity and improving the training speed of the model. In this figure, “Flattened Feature Token” represents feature vectors that we input into the transformer. Object queries in different colors represent targets predicted by our model, which is far greater than the actual number of targets in the image. Therefore, in the bipartite matching, most original target instances will be led to “no object,” and only a few will be led to corresponding true targets.

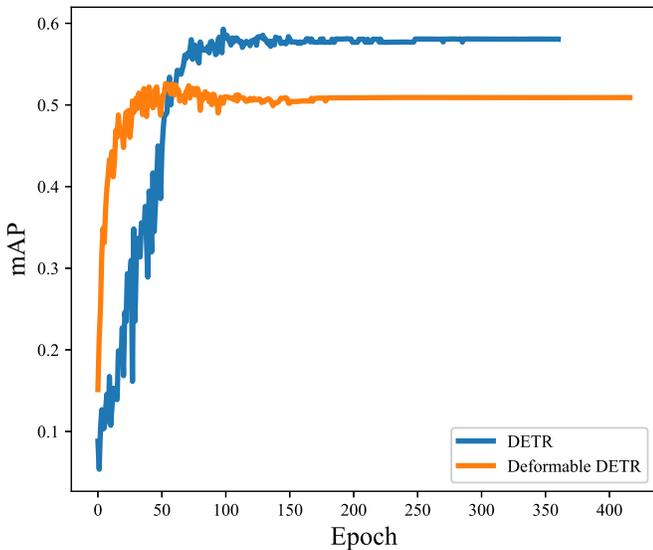


Figure 11. The learning curve of the DETR and that of the Deformable DETR. As we see in this figure, the Deformable DETR will converge after around 100 epochs and the DETR will require 200 epochs to converge. However, the DETR has a higher mAP after training.

Figure 11. With about 100 epochs, the Deformable DETR would converge. However, the DETR needs about 200 epochs to converge. Therefore, the convergence speed of the Deformable DETR is greatly improved, but there is a slight decrease in detection performance. Figures 6 and 12 show the detection performance of these two algorithms under different IOU thresholds. It can be seen that the overall performance of the Deformable DETR is lower than that of the DETR, because the introduction of the deformable attention mechanism effectively reduces the amount of computation but also loses the information of nonreference vectors and nonsampling vectors. Still, we can see that the performance degradation is small, reaching a maximum of no more than 10% at the IOU

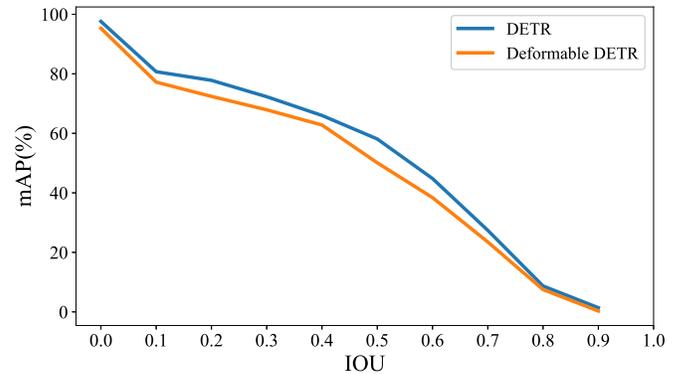


Figure 12. The mAP of the DETR and that of the Deformable DETR with different IOU criteria. As shown in this figure, we find that performance of the DETR is better than that of the Deformable DETR.

threshold of 0.5. We select several CGSLs that are detected by the DETR and missed by the Deformable DETR and also some CGSLs that are detected by the Deformable DETR and missed by the DETR in Figure 13. From these figures, we find that the DETR could detect arc structures and might be affected by foreground galaxies. The Deformable DETR would occasionally detect CGSLs with foreground galaxies when reference points are selected in some priority positions, which would better extract features of CGSLs for detection.

4.3. Increasing the Performance in Detection of CGSLs with Ensemble Learning for Real Applications

Since we find that the DETR and the Deformable DETR have different designs, their performance should be different for the same data set. Two machine-learning algorithms with different performance could be merged together with ensemble learning to further improve their performance. With this concept, we use Equation (8) to calculate final results with detection results of the DETR and thos of the Deformable

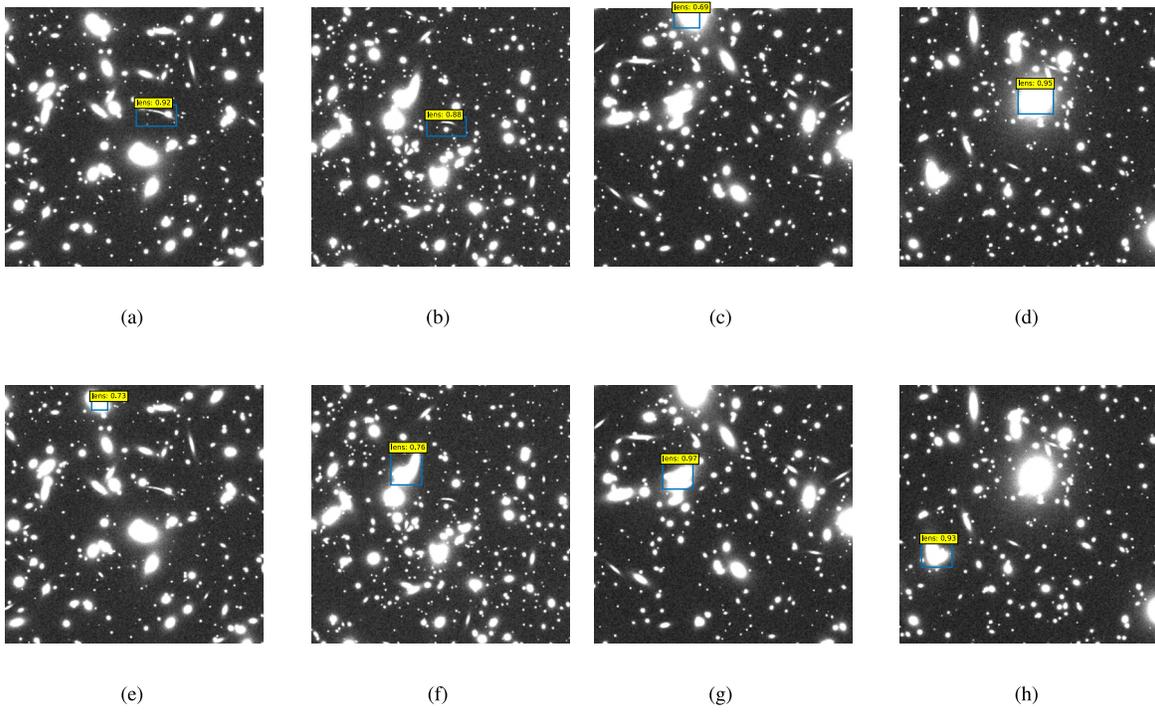


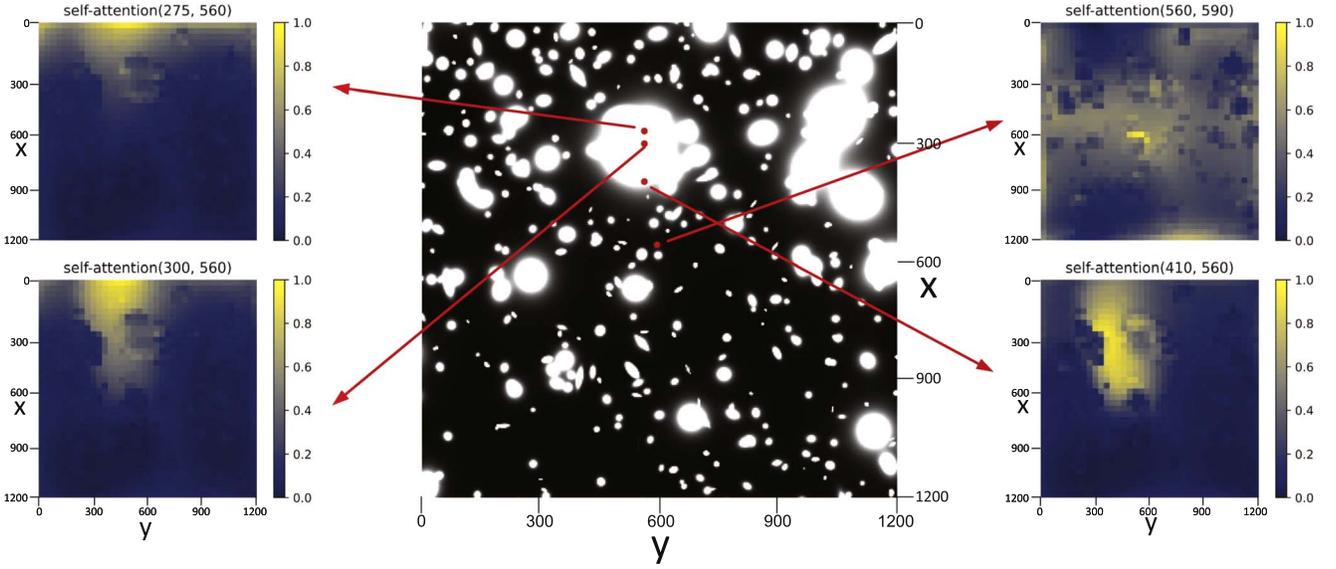
Figure 13. This figure shows detection results of the two different methods in four panels each. Panels (a)–(d) show the detection results of the DETR, and panels (e)–(h) show the detection results of the Deformable DETR. We find that the DETR gives correct results for panels (a) and (b), while the Deformable DETR could give correct results for panels (g) and (h). Meanwhile, for panels (g) and (h), we find that the arc is obscured by the foreground galaxy, so it is difficult for eyes to see it, but our algorithm can effectively identify and locate it.

Table 1
The Recall Rate and the Precision Rate (R/P) of the DETR with Different IOUs and Scores

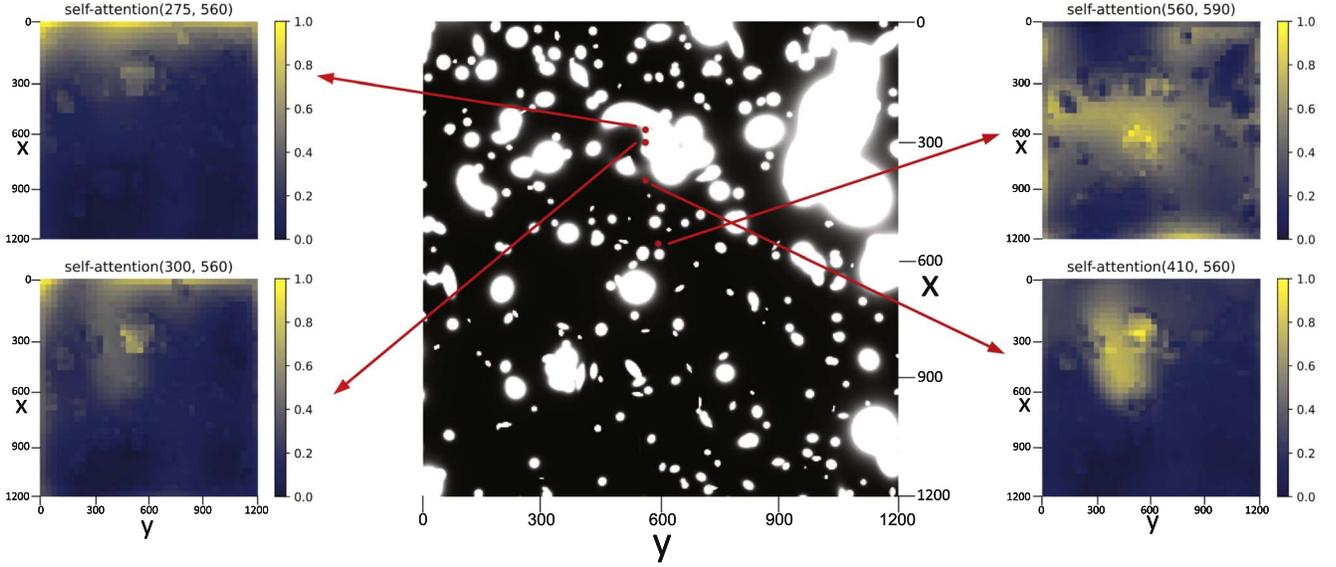
IOU	Score=0.9	Score=0.85	Score=0.8	Score=0.75	Score=0.7	Score=0.65	Score=0.6	Score=0.55
0.9	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0
0.8	4.9/5.9	5.8/5.0	7.5/4.7	8.2/4.5	9.4/4.2	10.5/4.0	12.1/3.8	13.6/3.7
0.7	13.7/18.3	16.3/15.8	20.9/15.4	30.0/15.0	26.1/14.4	28.7/13.7	32.0/13.1	35.0/12.6
0.6	23.5/35.5	27.4/30.7	33.9/30.0	37.1/29.6	41.7/29.1	45.5/28.4	49.4/27.1	52.8/26.2
0.5	32.6/55.9	39.8/53.8	46.8/51.4	49.8/49.8	54.9/49.5	58.4/47.8	62.5/46.2	65.8/44.9
0.4	37.5/69.4	45.1/67.0	52.9/65.6	56.1/64.0	60.8/63.2	64.1/60.9	68.2/60.0	71.3/58.1
0.3	40.1/77.4	47.8/74.7	55.8/73.9	59.0/72.3	63.8/71.2	67.1/69.6	71.1/68.2	74.1/66.8
0.2	41.2/82.3	49.3/79.2	57.2/78.3	60.4/76.4	65.1/75.8	68.5/74.2	72.6/73.2	75.5/72.0
0.1	42.2/84.4	49.9/81.0	58.0/80.6	61.3/79.4	66.0/78.9	69.6/77.9	73.6/77.1	76.4/75.7
0.0	42.8/86.6	50.7/83.7	58.8/83.4	62.1/82.4	66.8/81.8	70.5/81.6	74.4/80.6	77.2/79.4

Table 2
The Recall Rate and Precision Rate (R/P) of the Ensemble Model with Different IOUs and Scores

IOU	Score=0.9	Score=0.85	Score=0.8	Score=0.75	Score=0.7	Score=0.65	Score=0.6	Score=0.55
0.9	0.0/0.0	0.7/0.3	1.0/0.3	1.5/0.3	2.8/0.3	5.0/0.3	9.0/0.3	14.3/0.3
0.8	6.0/5.6	8.8/4.9	12.5/4.6	17.5/4.2	28.6/3.8	42.4/3.7	58.3/3.6	70/3.5
0.7	16.8/17.8	23.7/15.8	31.0/14.5	40.5/13.4	56.8/12.6	70.8/12.0	82.1/11.7	88.5/11.6
0.6	28.0/34.3	36.6/29.3	45.6/27.1	56.9/26.0	72.0/24.6	82.9/24.1	90.2/23.5	93.9/23.3
0.5	36.3/50.2	48.1/47.0	57.2/43.2	67.5/40.9	80.2/38.8	88.3/37.7	93.5/36.8	96.0/36.5
0.4	42.3/64.8	54.5/60.9	63.6/56.4	73.3/54.0	84.2/51.1	90.9/49.5	95.0/48.6	97.0/48.4
0.3	44.9/71.8	57.1/67.7	66.6/64.4	76.0/62.4	86.1/59.0	92.1/57.9	95.7/57.0	97.4/56.7
0.2	47.2/78.9	59.2/73.7	68.7/71.0	77.8/69.0	87.2/65.3	92.8/64.4	96.1/63.4	97.7/63.3
0.1	48.6/83.6	60.5/77.8	69.9/75.2	78.8/73.1	88.0/70.2	93.4/69.9	96.4/69.1	97.8/68.6
0.0	49.2/85.4	61.3/80.5	70.8/78.5	79.4/76.1	88.6/74.1	93.7/73.8	96.6/72.9	97.9/72.4



(a)



(b)

Figure 14. The self-attention map of four pixels in the last layer of the encoder. Panels (a) and (b) are CGSLs with and without foreground central galaxies, respectively. We find that the self-attention maps of them are similar, which indicates that our model can still identify CGSL features accurately and has good robustness even in the presence of foreground central galaxies. It also shows that the encoder is paying attention to the main structure of the CGSL, regardless of BCGs.

DETR:

$$\text{out} = \text{out}_{\max}(\text{Score}_{\text{DETR}}, \text{Score}_{\text{DeformableDETR}}), \quad (8)$$

where $\text{Score}_{\text{DETR}}$ and $\text{Score}_{\text{DeformableDETR}}$ are scores of the DETR and the Deformable DETR. With Equation (8), we would output candidates with high confidence of either of these two algorithms. For comparison, the DETR is used as a baseline for comparison as shown in Table 1. The results of the ensemble learning algorithm are shown in Table 2. We find that with an IOU of 0.1 and a Score of 0.7, the DETR has a recall rate of 66.0% and a precision rate of 78.9%, while the ensemble

learning algorithm has a recall rate of 88.0% and a precision rate of 70.2%. More CGSLs could be detected with ensemble learning with only a small drop in precision rate. For rare target detection, such as CGSLs, the ensemble learning would be a better choice.

5. Interpretation of the Mechanism of the Attention-based CGSL Detection Algorithm

Both the DETR and the Deformable DETR use the attention mechanism to detect targets. The attention mechanism is quite similar to the attention of human beings in observing a target.

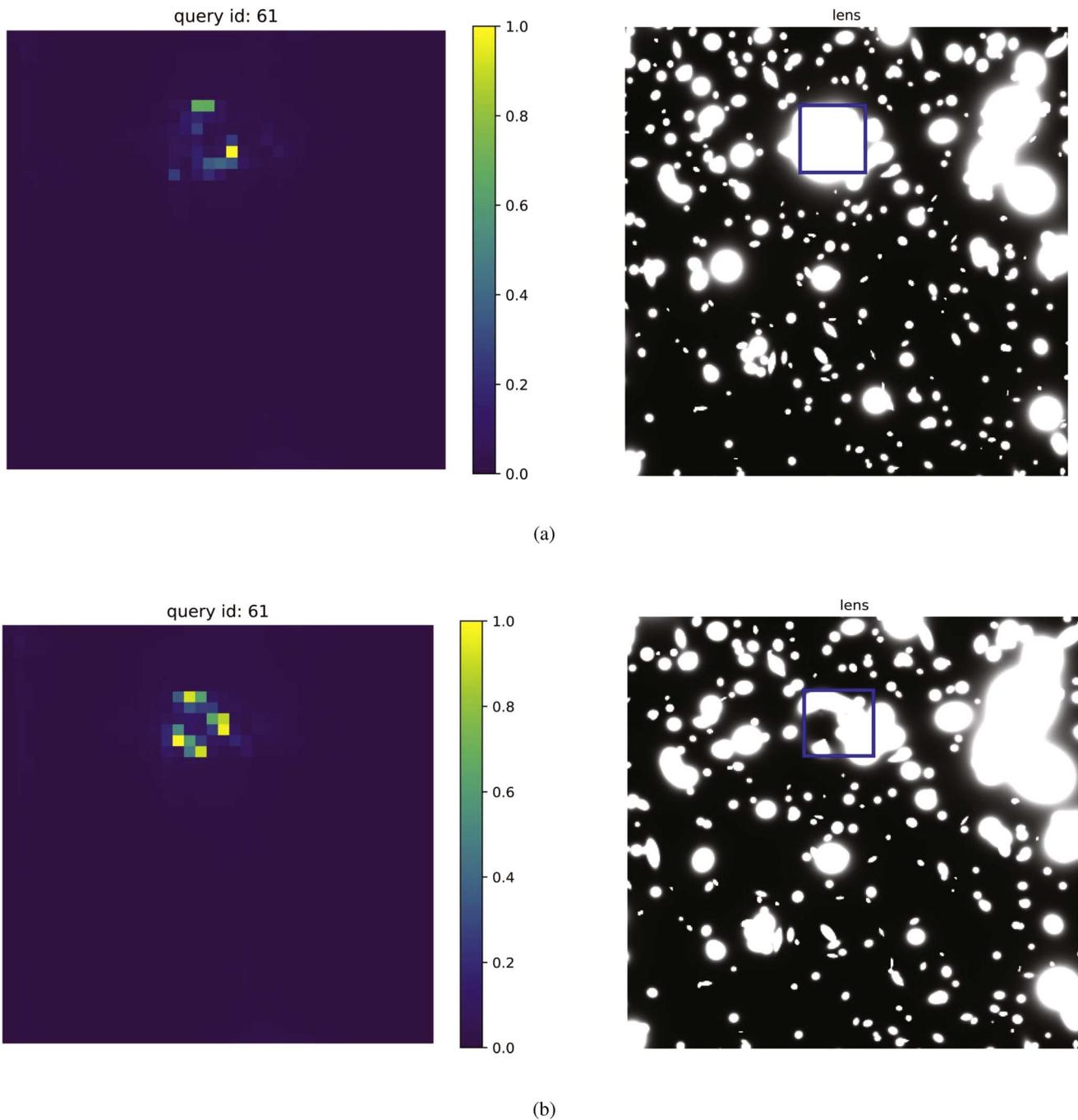
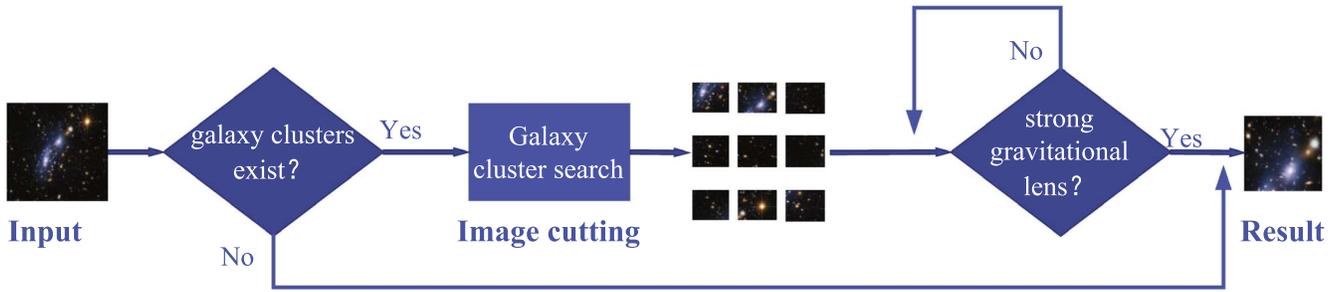


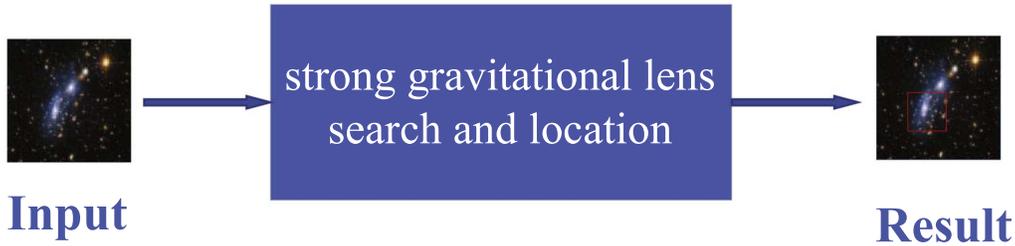
Figure 15. The cross-attention map of the last layer in the decoder. Input images are the same as those in Figure 14. In this figure, right panels are prediction results, and left panels are cross-attention maps of object queries matching predicted results in the decoder, whose legend represents the serial number of the object query. We can see that the decoder pays more attention to boundaries of CGSLs and gives similar results regardless of the interference of foreground central galaxies.

In this part, we would draw attention maps in different attention layers to show how the DETR and the Deformable DETR work and help us to better evaluate their performance. There are two types of attention maps from two different attention layers that we are going to visualize: the self-attention map in the last self-attention layer of the encoder part, and the cross-attention map in the last cross-attention layer of the decoder part. We will show both of these attention maps for two scenarios: a CGSL that is obstructed by the foreground galaxies, and a CGSL that is not obstructed by the foreground galaxies.

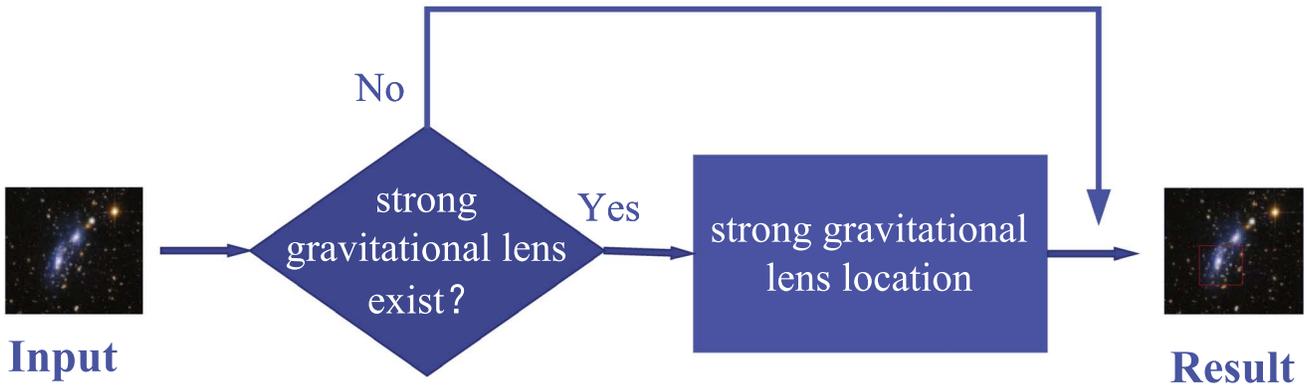
We will first show self-attention maps in the last self-attention layer of the encoder part. When we input an image into the model, the weight of the relation between feature vectors of different pixels in the image will be calculated in the last self-attention layer of the encoder as the attention map. If we select feature vectors from one pixel and calculate relations between other pixels, we would get an attention map with reference to that pixel. Therefore, we could select several pixels in the input image as sampling pixels and get their attention maps in the encoder's last self-attention layer. Then, we can



(a)



(b)



(c)

Figure 16. This figure shows the schematic diagram of three different approaches. Panel (a) shows the classic method, which first detects images of galaxy clusters. Then, these stamp images would be cut from original observation images and classified as either CGSLs or other targets, as discussed in Jacobs et al. (2017), Petrillo et al. (2017), Ostrovski et al. (2017), Bom et al. (2017), Hartley et al. (2017), Avestruz et al. (2019), and Lanusse et al. (2018). Panel (b) shows the diagram to use our algorithm in a one-step way, which directly detects CGSLs from observational images. Panel (c) shows the two-step strategy. We first obtain candidates with low IOU and then detect CGSLs from these images.

visualize the relationship between sampling pixels and other pixels in the input image to show us the attention of the detection algorithm. As shown in Figure 14, we select four pixels in the image to show their attention maps. We use the right-handed Cartesian coordinate system to define coordinates of these four pixels and use red circles to indicate their positions in the figure. Figures 14(a) and (b) are CGSLs with and without foreground central galaxies, respectively.

In Figure 14(a), we can see that when the sampling pixel (560, 590) is not in the CGSL, the attention distribution is uniform and random, indicating that there is no clear correlation or significant difference between the current

sampling pixel and other pixels. Therefore, the sampling pixel does not participate in the detection of CGSLs, and this pixel does not belong to the image of the CGSL. Sampling pixels (275, 560), (300, 560), and (410, 560) are inside the CGSL. We can see that as sampling pixels approach the CGSL, the distribution of the attention map gradually concentrates to a circle around the CGSL. This tells us that there exists a higher attention weight between the sampling pixel and other pixels within the CGSL. This strong relationship shows that these pixels are more closely related to each other, which could be used to characterize the CGSL system. The self-attention map within the CGSL is not exactly the same between different

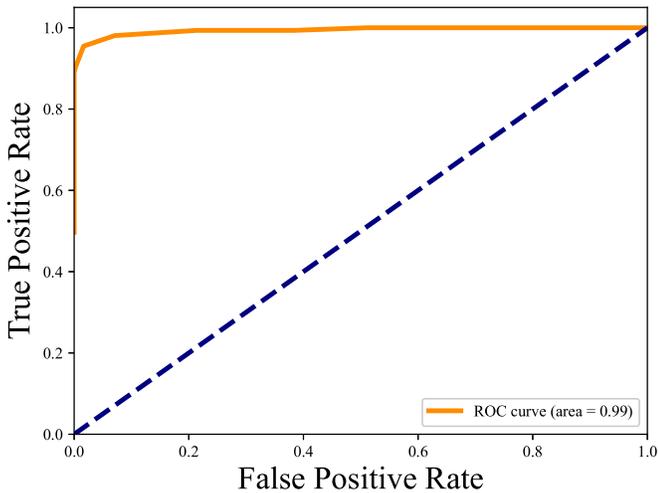


Figure 17. The ROC curve of our algorithm, which has an area of 0.99.

sampling pixels. It can be seen that the concentration of the attention distribution of sampling pixels at the edge of the CGSL system is relatively poor, and there is a certain correlation with other irrelevant pixels around, which reflects that pixels at the edge of the CGSL image have weaker effects in obtaining final detection results. The concentration of the attention distribution of sampling pixels in the center of the CGSL is very high in the self-attention map, indicating that these pixels are important in detection of CGSLs. It should be noted that for labels in the training data set only the boundary information of the target is given, instead of detailed pixel-level information. However, in self-attention maps, we find that our algorithm could obtain pixel-level information of CGSLs, although in a relatively rough way. It still shows that, thanks to the attention mechanism, our algorithm can learn and capture some important nonhuman and nonprior features of the target by itself and achieve simple image segmentation results. In Figure 14(b), we can draw a similar conclusion. Whether or not the CGSL is obstructed by foreground BCGs, we can output an effective self-attention map, which indicates the effectiveness of our method. We also find that the self-attention maps of Figures 14(a) and (b) are similar, which indicates that our model can still identify CGSL features accurately and robustly even in the presence of foreground central galaxies. Overall, Figures 14(a) and (b) maintain high similarity.

Then, we visualize the cross-attention map in the last cross-attention layer of the decoder part. We directly use the cross-attention map of predicted results to visualize attentions. As we have mentioned earlier, each object query will predict a target, and only a few of them are real targets. Here we will show queries that match real targets. Then, we could obtain outputs of these queries in the last decoder layer and visualize it to obtain the cross-attention map of predicted targets. As shown in Figure 15, compared with the self-attention encoder, which focuses on the main part of the object, the decoder based on the cross-attention mechanism pays more attention to boundaries of CGSLs. Figures 15(a) and (b) are CGSLs with foreground central galaxies and without foreground central galaxies, respectively. We find that the cross-attention map pays more attention to CGSL boundaries than other parts. Meanwhile, for the same CGSL system, regardless of whether there is a foreground central galaxy or not, the cross-attention map is very similar, which indicates that our method is robust in detection of CGSLs.

With visualization methods mentioned above, we find that attention-based detection algorithms first pay attention to the main body of CGSLs with the self-attention layer in the encoder. Then, the detection algorithms pay attention to the edge of CGSLs with the cross-attention layer in the decoder part. This is similar to the attention mechanism by which people observe things. However, attention-based detection algorithms could directly use CGSL images of multiple bands for detection. Although the image of the CGSL is strongly obstructed by the foreground galaxies, these detection algorithms could still detect the CGSLs. Thanks to this property, attention-based detection algorithms could achieve better performance in detection of CGSLs than could be obtained by humans and would lead to new discoveries of CGSLs.

6. Performance Test with Simulated and Real Observation Data

6.1. Performance Test with Simulated Data

In this subsection, we will consider using our method in real applications. Since CGSLs are rare in real conditions and detection results would be analyzed by scientists for further study, it would be more appropriate to design a framework with high precision, moderate completeness, and low false-positive rate. Considering that the recall rate is more important than position accuracy in practical applications, we do not need to use very high position accuracy conditions all the time. First, we use a low IOU to classify full-frame images and obtain candidate images that may contain CGSLs. In this way, the position accuracy is low, but the recall rate and precision rate are high. Then, we use a high IOU to detect the candidate images again, so as to accurately find the CGSLs in these candidate images, which ensures the position accuracy. The flowchart of the classic method, our previous method, and the two-step strategy is shown in Figure 16. We would first use the ensemble learning framework discussed above to detect CGSLs from observational images, and we would further use the following strategy to obtain final detection results. Detailed steps are shown below:

1. We set the IOU as 0.0 and obtain images that may contain CGSLs as candidates.
2. We set the IOU as 0.7 and detect CGSLs from these candidate images with either the deformable DETR or the DETR, whose detection result is adapted in the previous step.

To test the performance of the two-step approach in real applications, we have generated simulated data with 16,000 images, and 1% of them contain CGSLs. It would take about 180 minutes to process all these images. The test results show that our method can achieve 99.63% accuracy, when the recall rate is 90.32% and the precision rate is 87.53%. Besides, the false-positive rate is only 0.23%. We have further plotted the receiver operating characteristic (ROC) curve as shown in Figure 17. The vertical coordinate of the curve represents true positive rate and the horizontal coordinate represents false-positive rate. The low false-positive rate ensures that future applications of large-scale sky surveys will not produce a lot of false positives, which means that there will not be too much reliance on human inspection. Although the number of candidates to be checked by scientists has been greatly reduced, there are still lots of candidates that need to be checked by human investigation. However, as we have

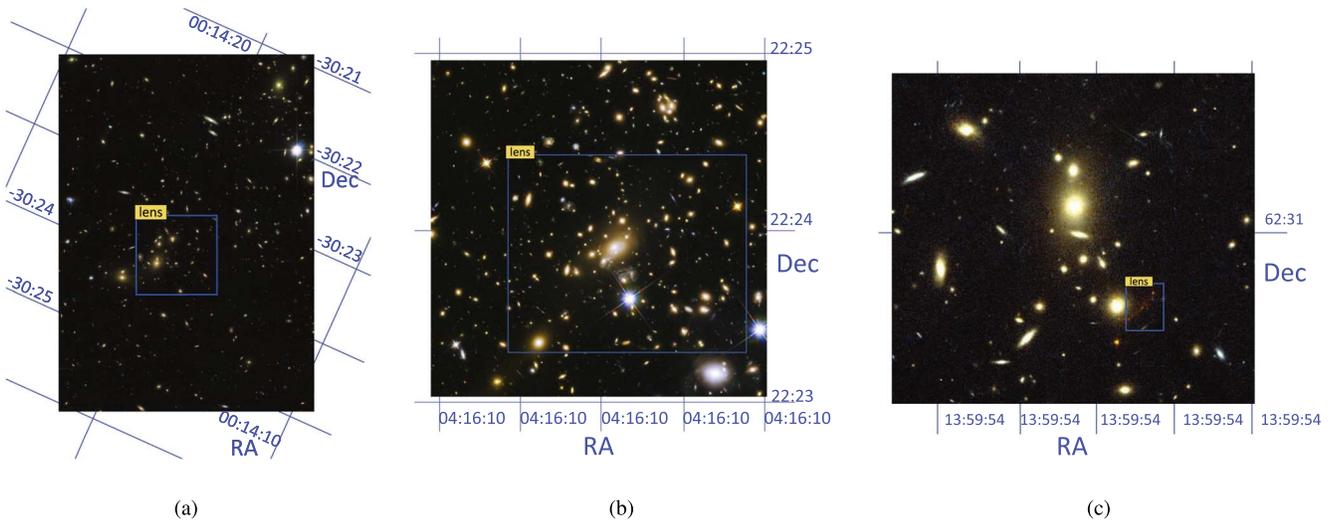


Figure 18. Detection results of our algorithm for images containing A2744 ($3'05 \times 4'38$ with 3662×5253 pixels), MACS J1149.5+2223 ($1'90 \times 1'90$ with 3800×3800 pixels), and CIG 1358+62 ($1'06 \times 0'96$ with 640×583 pixels). These results show that our algorithm could directly detect gravitational arcs from real observation data, when our algorithm is only trained with ideal simulated data without noise. Generally speaking, a model trained only with simulated data usually needs some methods to adapt to real observation data. However, our algorithm can detect strong gravitational lensing on real data without further training, which shows that it is effective to train an effective neural network with simulated data, if the simulation reflects real physical processes. Original image of A2744 by NASA, ESA, and D. Coe (STScI)/J. Merten (Heidelberg/Bologna). Original image of MACS J1149.5+2223 by NASA, ESA, S. Rodney (John Hopkins University, USA), and the FrontierSN team; T. Treu (University of California Los Angeles, USA), P. Kelly (University of California Berkeley, USA), and the GLASS team; J. Lotz (STScI) and the Frontier Fields team; M. Postman (STScI) and the CLASH team; and Z. Levay (STScI). Original image of CIG 1358+62 by Marijn Franx (University of Groningen, The Netherlands), Garth Illingworth (University of California, Santa Cruz), and NASA/ESA.

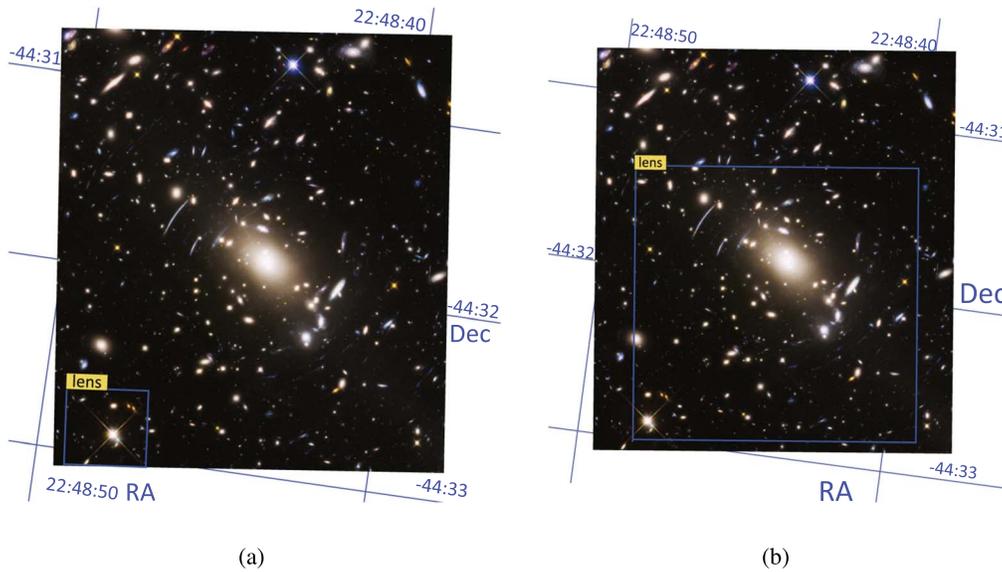


Figure 19. Detection results for images containing Abell S1063 ($2'07 \times 2'32$) in images of different size, 2243×2511 pixels in the left panel and 1280×1433 pixels in the right panel. A star with a bright arc nearby is wrongly classified as a CGSL because it is relatively large and contains a central bright source and arcs nearby. However, our algorithm could detect the CGSL from an image with a smaller size, when the star and the arc have smaller size. Original image of Abell S1063 by NASA, ESA, and J. Lotz (STScI).

discussed in Section 2, our algorithm could detect CGSLs that could not be directly seen by humans. Therefore, we need to integrate the attention map with observation images to generate augmented images and use citizen science platforms to investigate details of these images (Lintott et al. 2008; Smith et al. 2011; Brink et al. 2013; Marshall et al. 2015). We will further investigate this problem in future work.

6.2. Performance Test with Real Observation Data

We use real observation data obtained from the Frontier Fields survey project and the RELICS survey project obtained

by the Hubble Space Telescope and the early release deep field image from the James Webb Space Telescope to test the performance of our algorithm. First, we select several images obtained by the Frontier Fields survey project (Lotz et al. 2014; Koekemoer et al. 2014; Lotz et al. 2017) to test the performance of our algorithm. These images are downloaded from the website (<https://esahubble.org/>) as full-size original TIF files. We directly use the ensemble learning algorithm to connect the DETR and the Deformable DETR trained with simulated data to detect CGSLs in these images. Since these images do not have any bounding boxes, we check these

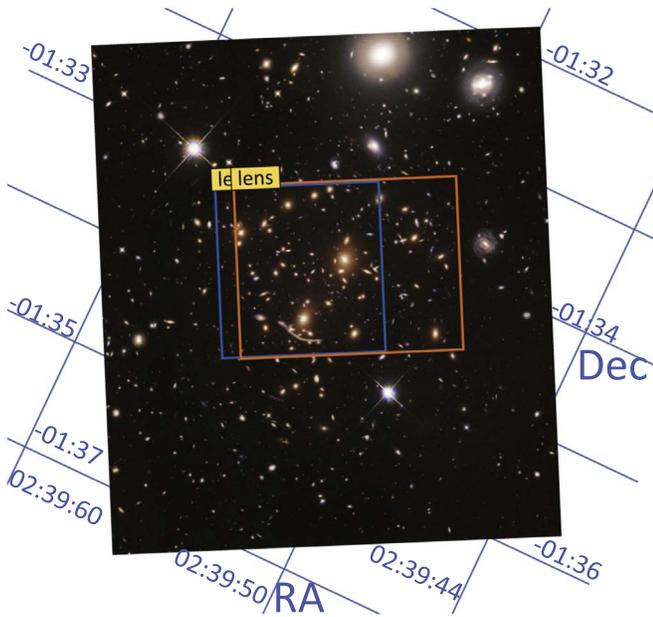


Figure 20. Detection results with top two scores for an image that contains A370 ($3'91 \times 4'45$ with 4164×4634 pixels). As shown in this figure, these two detection results locate the same CGSL with a little horizontal shift, which indicates that the detection results of our algorithm are reliable. Original image of A370 by NASA, ESA, A. Koekemoer, M. Jauzac, C. Steinhardt, and the BUFFALO team.

images by eye and use the dark matter distribution released on the same website as a reference to locate CGSLs. We have tested our algorithm with all these images, which include images of A2744, MACS J0416.1–2403, MACS J0717.5+3745, MACS J1149.5+2223, Abell S1063, and A370. Generally speaking, our algorithm could detect almost all gravitational arcs from these images, as shown in Figure 18.

The difference between training data and real data would introduce problems to machine-learning algorithms, when they are deployed. Although our algorithm has shown relatively good generalization performance and robustness in Figure 18, its performance is still affected. For the CGSL detection task, images that have similar arc structures might introduce errors to detection results. Since the training data do not contain any diffraction effects brought by the optics system, our detection algorithm has some risks to detect “arc” structures brought by the diffraction effect as shown in Figure 19(a). We find that our detection algorithm locates a star with an arc structure nearby. However, if we use our algorithm to detect CGSLs in the same image with lower resolution (smaller number of pixels here), our algorithm could detect CGSLs as shown in Figure 19(b). The problem of wrong detections brought by diffraction effects could be solved if we add more realistic effects to simulated images as training data, or if we use real observation data to train the neural network with a transfer learning strategy.

To better investigate the performance of our algorithm. We have further designed a test. Since there is normally only one CGSL in observation images, we select the detection result with the highest score as the output. However, we could also output several detection results with high scores to test whether the detection results are reliable. Based on this concept, we output two detection results with the top two scores of an image that contains A370 in the second test, as shown in Figure 20. In this figure, we find that our algorithm could detect the main part

of the CGSL. There is a shift of only tens of pixels in these detection results.

Finally, we test the performance of our algorithm in processing the same CGSL obtained by telescopes with different pixel scales. SMACS0723 has attracted a lot of attention recently, since the observation data by the James Webb Space Telescope were released on 2022 July 14.⁹ SMACS0723 has been observed by the Hubble Space Telescope in the RELICS survey project (Salmon et al. 2018, 2020). In this paper, we test the performance of our algorithm with images of the SMACS0723 observed by the Hubble Space Telescope and the James Webb Space Telescope. The image of SMACS0723 from the Hubble Space Telescope is obtained from the official site of the RELICS project as a color image.¹⁰ The image of SMACS0723 from the James Webb Telescope is obtained from the official site as a color image. Detection results and attention maps are shown in Figure 21. As shown in this figure, our algorithm could detect and locate CGSLs in both of these figures, and attentions of these detection results are similar. According to these results, we find that, even for real observation images, our algorithm could still obtain effective results, when trained only with simulated images.

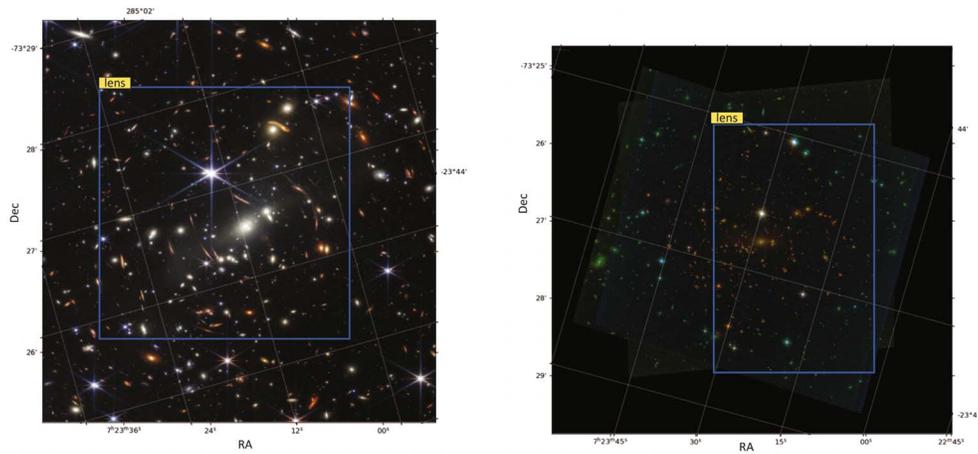
7. Conclusions and Future Work

CGSLs are valuable in scientific research of galaxies, galaxy clusters, and cosmology. Due to their significant sky coverage and depth, future sky surveys are expected to reveal many CGSLs. However, considering the complexity and variety of the strong-lensing signals in CGSLs and the contamination of the line-of-sight objects, it is challenging to detect strong-lensing signals efficiently and automatically. Hence, we propose a transformer-based algorithm for detecting CGSLs from enormous data to solve the issue. The algorithm is designed to learn features with a large extension because CGSLs are sparsely distributed and often have extended structures, which is adequate for the transformer-based detection algorithm. Our method uses ensemble learning to merge detection results from the DETR and the Deformable DETR to give final detection results. Besides, our method could detect CGSLs directly from images with any number of channels (bands), which makes it adequate in the application of detection of CGSLs from multicolor sky survey projects. We use simulated images to train our algorithm, and the results show that our algorithm takes advantage of the attention mechanism and could achieve an 88% recall rate and 70% precision rate in detecting CGSLs, although many of them are blended with foreground galaxies. We use self-attention and cross-attention to show features that attract our detection algorithm. The results show that our algorithm focuses on the arc-like structure of CGSLs.

We have also considered applications of our method in real applications. To further increase detection efficiency, we propose a two-step strategy, which first obtains candidate images that contain CGSLs with small IOU threshold and then detects CGSLs from these candidate images with high IOU threshold. With this strategy, our method achieves a 99.63% accuracy rate, 90.32% recall rate, 85.37% precision rate, and

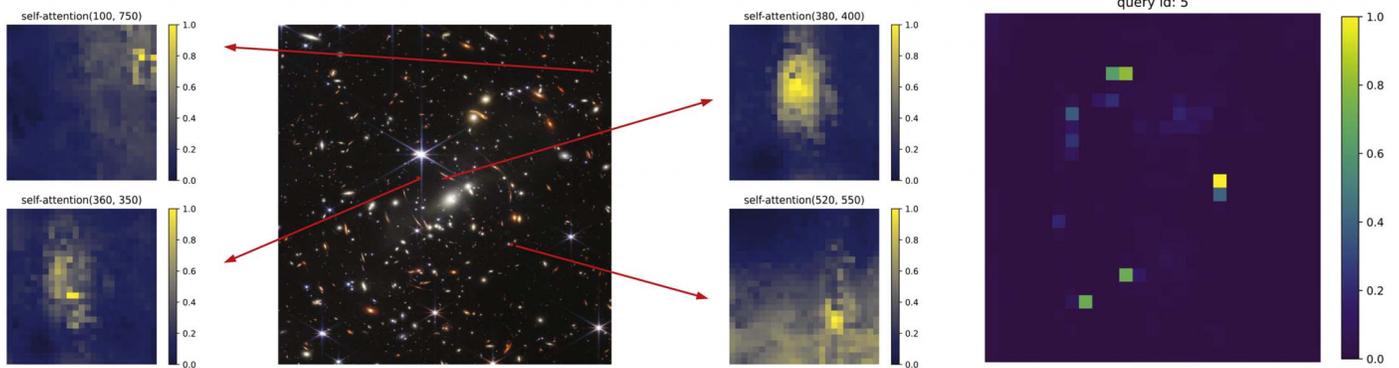
⁹ <https://www.nasa.gov/image-feature/goddard/2022/nasa-s-webb-delivers-deepest-infrared-image-of-universe-yet>

¹⁰ <https://relics.stsci.edu/data/smacs0723-73/>



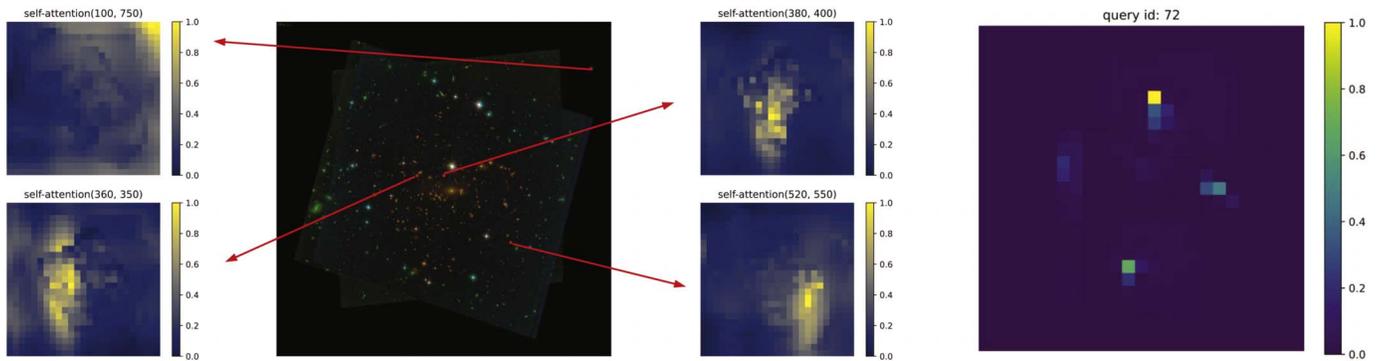
(a)

(b)



(c)

(d)



(e)

(f)

Figure 21. These figures show self-attention and cross-attention of our algorithm in detection of CGSLs from observation images. These two images contain the same target SMACS0723 observed by the Hubble Space Telescope and the James Webb Space Telescope. As shown in this figure, we find that our algorithm could detect locations of the CGSLs in both of these figures. Besides, attention maps are similar to each other, which indicates that our method could reveal similarities between these two images.

0.23% false-positive rate in detecting CGSLs from 16,000 images containing 1% CGSLs. We further use our method to detect CGSLs from real observation images from the Hubble Space Telescope and the James Webb Space Telescope. The results show that our method can identify most of the strongly

lensed arcs but misses a couple owing to diffraction rings, which can be improved with more training data containing realistic PSFs and noise. Moreover, when applying our method to the HST data, we find that elongated galaxies (panel (a) in Figure 19) can be the primary sources of false positives. This

failure points out two issues in the training set that need to be improved: (1) including bright stars beyond extragalactic objects, and (2) making a more aggressive definition for giant arcs to eliminate the contamination due to elongated galaxies. Alternatively, we expect to improve the performance of detecting CGSLs by involving humans in the loop, which is thoroughly studied in another project of ours.

Compared to other machine-learning-based algorithms, which try to find strong lenses by identifying stamp images centered at galaxies containing CGSLs, our algorithm focuses on recognizing arc-like features in an arbitrary field of view. Our strategy is specifically suitable for cluster-scale lenses because arcs in galaxy clusters are not guaranteed to be located around the centers of BCGs of galaxy clusters. On the other hand, compared to traditional arc-finder algorithms, our method can detect and locate the strong gravitational lensing system directly on the original image without the need to search and cut galaxy clusters and then classify them, so it is simpler and more efficient. At the same time, due to the application of the attention mechanism, our algorithm has better detection performance and robustness, particularly for the cases of faint arcs, complex arcs, and highly blended images, particularly when multiple-band information is taken into account in the future. Therefore, we optimistically foresee the application of the method to the data from upcoming large-scale surveys.

In the future, we will carry out simulations with multiple-band information, PSFs, and noise to generate more appropriate training sets for the data obtained by different instruments. Additionally, we plan to design multistep detection strategies and involve joint training by combining the data from various telescopes to improve the precision and recall of the detection of arcs in galaxy clusters. Eventually, the program will be applied to available and upcoming observations, such as the DESI Legacy surveys, the China Space Station Telescope, and Euclid, and the data products will be released for exploring the dark sectors of the universe.

This work is based on observations made with the NASA/ESA Hubble Space Telescope, obtained at the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5-26555. These observations are associated with programs GO-11103, GO-12166, GO-12884, and GO-14096. This work is based on observations made with the NASA/ESA/CSA James Webb Space Telescope. The data were obtained from the Mikulski Archive for Space Telescopes at the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5-03127 for JWST. These observations are associated with program No. 2736.

P.J. would like to thank Professor Ran Li from National Astronomical Observatories and Professor Huanyuan Shan, Professor Shiyin Shen, and Dr. Rafael S. de Souza from Shanghai Observatory, who provided very helpful suggestions for this paper. This work is supported by the National Natural Science Foundation of China (NSFC) with funding Nos. 12173027 and 12173062. We acknowledge the science research grants from the China Manned Space Project with Nos. CMS-CSST-2021-A01 and CMS-CSST-2021-B12. We acknowledge the science research grants from the Square Kilometre Array (SKA) Project with No. 2020SKA0110102.

This work is also supported by the special fund for Science and Technology Innovation Teams of Shanxi Province.

Data Availability

Data resources are supported by China National Astronomical Data Center (NADC) and Chinese Virtual Observatory (China-VO). This work is supported by Astronomical Big Data Joint Research Center, co-founded by National Astronomical Observatories, Chinese Academy of Sciences and Alibaba Cloud. The code and data used in this paper can be found at DOI:10.12149/101172.

ORCID iDs

Peng Jia  <https://orcid.org/0000-0001-6623-0931>

References

- Alard, C. 2006, arXiv:astro-ph/0606757
- Avestruz, C., Li, N., Zhu, H., et al. 2019, *ApJ*, 877, 58
- Bacon, D. J., Refregier, A. R., & Ellis, R. S. 2000, *MNRAS*, 318, 625
- Bartelmann, M., & Maturi, M. 2017, *SchpJ*, 12, 32440
- Barvainis, R., & Ivison, R. 2002, *ApJ*, 571, 712
- Bayliss, M. B. 2012, *ApJ*, 744, 156
- Bayliss, M. B., Gladders, M. D., Oguri, M., et al. 2011, *ApJL*, 727, L26
- Bom, C. R., Makler, M., Albuquerque, M. P., & Brandt, C. H. 2017, *A&A*, 597, A135
- Brink, H., Richards, J. W., Poznanski, D., et al. 2013, *MNRAS*, 435, 1047
- Bruce, A., Lawrence, A., MacLeod, C., et al. 2017, *MNRAS*, 467, 1259
- Carion, N., Massa, F., Synnaeve, G., et al. 2020, European Conference on Computer Vision (Berlin: Springer), 213
- Castro, P. G., Heavens, A. F., & Kitching, T. D. 2005, *PhRvD*, 72, 023516
- Child, H. L., Habib, S., Heitmann, K., et al. 2018, *ApJ*, 859, 55
- Coe, D., Zitrin, A., Carrasco, M., et al. 2013, *ApJ*, 762, 32
- Collett, T. E. 2015, *ApJ*, 811, 20
- Dai, J., Qi, H., Xiong, Y., et al. 2017, in Proceedings of the IEEE Int. Conf. on Computer Vision (New York: IEEE)
- Dalal, N., Holder, G., & Hennawi, J. F. 2004, *ApJ*, 609, 50
- Diego, J. M., Schmidt, K. B., Broadhurst, T., et al. 2018, *MNRAS*, 473, 4279
- Dye, S., Evans, N. W., Belokurov, V., Warren, S. J., & Hewett, P. 2008, *MNRAS*, 388, 384
- Dye, S., Furlanetto, C., Swinbank, A. M., et al. 2015, *MNRAS*, 452, 2258
- Faber, S. M., Willmer, C. N. A., Wolf, C., et al. 2007, *ApJ*, 665, 265
- Fedeli, C., Bartelmann, M., Meneghetti, M., & Moscardini, L. 2007, *A&A*, 473, 715
- Fedeli, C., Meneghetti, M., Gottlöber, S., & Yepes, G. 2010, *A&A*, 519, A91
- Fu, L. P., & Fan, Z. H. 2014, *RAA*, 14, 1061
- Gavazzi, R., Treu, T., Rhodes, J. D., et al. 2007, *ApJ*, 667, 176
- Hamana, T., Miyazaki, S., Shimasaku, K., et al. 2003, *ApJ*, 597, 98
- Han, C., Jung, Y. K., Udalski, A., et al. 2018, *ApJ*, 867, 136
- Han, J., Eke, V. R., Frenk, C. S., et al. 2015, *MNRAS*, 446, 1356
- Hartley, P., Flamar, R., Jackson, N., Tagore, A. S., & Metcalf, R. B. 2017, *MNRAS*, 471, 3378
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (Cham: Springer), 770
- Hewitt, J. N. 1988, in Yale Cosmic String Workshop (New Haven, CT: Yale), 156
- Hilbert, S., Metcalf, R. B., & White, S. D. M. 2007, *MNRAS*, 382, 1494
- Horesh, A., Maoz, D., Ebeling, H., Seidel, G., & Bartelmann, M. 2010, *MNRAS*, 406, 1318
- Horesh, A., Maoz, D., Hilbert, S., & Bartelmann, M. 2011, *MNRAS*, 418, 54
- Horesh, A., Ofek, E. O., Maoz, D., et al. 2005, *ApJ*, 633, 768
- Huang, X., Storfer, C., Gu, A., et al. 2021, *ApJ*, 909, 27
- Huang, X., Storfer, C., Ravi, V., et al. 2020, *ApJ*, 894, 78
- Hudson, M. J., Gwyn, S. D. J., Dahle, H., & Kaiser, N. 1998, *ApJ*, 503, 531
- Jacobs, C., Glazebrook, K., Collett, T., More, A., & McCarthy, C. 2017, *MNRAS*, 471, 167
- Jauzac, M., Harvey, D., & Massey, R. 2018, *MNRAS*, 477, 4046
- Jee, M. J., Dawson, W. A., Stroe, A., et al. 2016, *ApJ*, 817, 179
- Jia, P., Liu, Q., & Sun, Y. 2020, *AJ*, 159, 212
- Jia, P., Zhao, Y., Xue, G., & Cai, D. 2019, *AJ*, 157, 250
- Jones, T., Ellis, R. S., Richard, J., & Jullo, E. 2013, *ApJ*, 765, 48
- Kilbinger, M., Heymans, C., Asgari, M., et al. 2017, *MNRAS*, 472, 2126

- Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
- Kneib, J.-P., & Natarajan, P. 2011, *A&ARv*, **19**, 47
- Koekemoer, A. M., Avila, R. J., Hammer, D., et al. 2014, AAS Meeting, **223**, 254.02
- Koopmans, L. V. E., Treu, T., Bolton, A. S., Burles, S., & Moustakas, L. A. 2006, *ApJ*, **649**, 599
- Korytov, D., Hearin, A., Kovacs, E., et al. 2019, *ApJS*, **245**, 26
- Kuhn, H. W. 1955, *Nav. Res. Logist. Q.*, **2**, 83
- Lanusse, F., Ma, Q., Li, N., et al. 2018, *MNRAS*, **473**, 3895
- Lenzen, F., Schindler, S., & Scherzer, O. 2004, *A&A*, **416**, 391
- Li, G.-L., Mao, S., Jing, Y. P., Kang, X., & Bartelmann, M. 2006, *ApJ*, **652**, 43
- Li, N., Gladders, M. D., Heitmann, K., et al. 2019, *ApJ*, **878**, 122
- Li, N., Gladders, M. D., Rangel, E. M., et al. 2016, *ApJ*, **828**, 54
- Li, R., Napolitano, N. R., Tortora, C., et al. 2020, *ApJ*, **899**, 30
- Li, R., Shu, Y., Su, J., et al. 2019, *MNRAS*, **482**, 313
- Liao, N. H., Bai, J. M., Liu, H. T., et al. 2014, *ApJ*, **783**, 83
- Lintott, C. J., Schawinski, K., Slosar, A., et al. 2008, *MNRAS*, **389**, 1179
- Lotz, J., Mountain, M., Grogin, N. A., et al. 2014, AAS Meeting, **223**, 254.01
- Lotz, J. M., Koekemoer, A., Coe, D., et al. 2017, *ApJ*, **837**, 97
- Lynds, R., & Petrosian, V. 1986, *BAAS*, **18**, 1014
- Madireddy, S., Li, N., Ramachandra, N., et al. 2019, arXiv:1911.03867
- Magaña, J., Motta, V., Cárdenas, V. H., Verdugo, T., & Jullo, E. 2015, *ApJ*, **813**, 69
- Mandelbaum, R. 2018, *ARA&A*, **56**, 393
- Mao, S., & Witt, H. J. 2012, *MNRAS*, **420**, 792
- Marshall, P. J., Lintott, C. J., & Fletcher, L. N. 2015, *ARA&A*, **53**, 247
- Medezinski, E., Umetsu, K., Nonino, M., et al. 2013, *ApJ*, **777**, 43
- Meneghetti, M., Bartelmann, M., Dahle, H., & Limousin, M. 2013, *SSRv*, **177**, 31
- Meneghetti, M., Davoli, G., Bergamini, P., et al. 2020, *Sci*, **369**, 1347
- Meneghetti, M., Rasia, E., Vega, J., et al. 2014, *ApJ*, **797**, 34
- Metcalf, R. B., Croft, R. A. C., & Romeo, A. 2018, *MNRAS*, **477**, 2841
- More, A., Verma, A., Marshall, P. J., et al. 2016, *MNRAS*, **455**, 1191
- Newman, A. B., Treu, T., Ellis, R. S., et al. 2013, *ApJ*, **765**, 24
- Oguri, M., Bayliss, M. B., Dahle, H., et al. 2012, *MNRAS*, **420**, 3213
- Oldham, L., Auger, M. W., Fassnacht, C. D., et al. 2017, *MNRAS*, **465**, 3185
- Ostrovski, F., McMahon, R. G., Connolly, A. J., et al. 2017, *MNRAS*, **465**, 4325
- Parker, L. C., Hoekstra, H., Hudson, M. J., van Waerbeke, L., & Mellier, Y. 2007, *ApJ*, **669**, 21
- Paszke, A., Gross, S., Massa, F., et al. 2019, arXiv:1912.01703
- Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2017, *MNRAS*, **472**, 1129
- Rahvar, S. 2015, *IJMPD*, **24**, 1530020
- Ren, S., He, K., Girshick, R., & Sun, J. 2015, arXiv:1506.01497
- Rozo, E., Nagai, D., Keeton, C., & Kravtsov, A. 2008, *ApJ*, **687**, 22
- Salmon, B., Coe, D., Bradley, L., et al. 2018, *ApJL*, **864**, L22
- Salmon, B., Coe, D., Bradley, L., et al. 2020, *ApJ*, **889**, 189
- Schmidt, F. 2008, *PhRvD*, **78**, 043002
- Seidel, G., & Bartelmann, M. 2007, *A&A*, **472**, 341
- Sereno, M., & Covone, G. 2013, *MNRAS*, **434**, 878
- Shvartzvald, Y., Yee, J. C., Novati, S. C., et al. 2017, *ApJL*, **840**, L3
- Simon, J. D. 2019, *ApJ*, **863**, 89 VizieR Online Data Catalog
- Smith, N., Li, W., Filippenko, A. V., & Chornock, R. 2011, *MNRAS*, **412**, 1309
- Smithsonian Astrophysical Observatory 2000, SAOImage DS9: A utility for displaying astronomical images in the X11 window environment, Astrophysics Source Code Library, ascl:0003.002
- Soucail, G., Mellier, Y., Fort, B., Mathez, G., & Cailloux, M. 1987, *Msngr*, **50**, 5
- Stacey, H. R., & McKean, J. P. 2018, *MNRAS*, **481**, L40
- Stark, C. W., Font-Ribera, A., White, M., & Lee, K. G. 2015, *MNRAS*, **453**, 4311
- Suyu, S. H., Auger, M. W., Hilbert, S., et al. 2013, *ApJ*, **766**, 70
- Suyu, S. H., Treu, T., Hilbert, S., et al. 2014, *ApJL*, **788**, L35
- Talbot, M. S., Brownstein, J. R., Bolton, A. S., et al. 2018, *MNRAS*, **477**, 195
- Thuruthipilly, H., Zdrozny, A., & Pollo, A. 2021, arXiv:2110.09202
- Troxel, M. A., Krause, E., Chang, C., et al. 2018, *MNRAS*, **479**, 4998
- Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, arXiv:1706.03762
- Walsh, D., Carswell, R. F., & Weymann?, R. J. 1979, *Natur*, **279**, 381
- Wambsganss, J., Bode, P., & Ostriker, J. P. 2004, *ApJL*, **606**, L93
- Xu, B., Postman, M., Meneghetti, M., et al. 2016, *ApJ*, **817**, 85
- Zhu, X., Su, W., Lu, L., et al. 2020, arXiv:2010.04159