The Astrophysical Journal, 992:40 (9pp), 2025 October 10 https://doi.org/10.3847/1538-4357/adf95c

DOLPHIN: A Fully Automated Forward-modeling Pipeline Powered by Artificial Intelligence for Galaxy-scale Strong Lenses

Anowar J. Shajib[1,2,3,4,7], Nafis Sadik Nihal[4], Chin Yi Tan[2,5], Vedant Sahu[3], Simon Birrer[6], Tommaso Treu[3], and Joshua Frieman[1,2]

[1] Department of Astronomy & Astrophysics, University of Chicago, Chicago, IL 60637, USA; ajshajib@uchicago.edu
[2] Kavli Institute for Cosmological Physics, University of Chicago, Chicago, IL 60637, USA
[3] Department of Physics & Astronomy, University of California, Los Angeles, CA 90095, USA
[4] Center for Astronomy, Space Science and Astrophysics, Independent University, Bangladesh, Dhaka 1229, Bangladesh
[5] Department of Physics, University of Chicago, Chicago, IL 60637, USA
[6] Department of Physics and Astronomy, Stony Brook University, Stony Brook, NY 11794, USA
Received 2025 April 3; revised 2025 July 18; accepted 2025 July 28; published 2025 October 3

Abstract

Strong gravitational lensing is a powerful tool for probing the internal structure and evolution of galaxies, the nature of dark matter, and the expansion history of the Universe, among many other scientific applications. For almost all of these science cases, modeling the lensing mass distribution is essential. For that, forward modeling of imaging data to the pixel level is the standard method used for galaxy-scale lenses. However, the traditional workflow of forward lens modeling necessitates a significant amount of human investigator time, requiring iterative tweaking and tuning of the model settings through trial and error. An automated lens-modeling pipeline can substantially reduce the need for human investigator time. In this paper, we present DOLPHIN, an automated lens-modeling pipeline that combines artificial intelligence with the traditional forward-modeling framework to enable full automation of the modeling workflow. DOLPHIN uses a neural network model to perform visual recognition of the strong lens components, then autonomously sets up a lens model with appropriate complexity and fits the model with the modeling engine, LENSTRONOMY. Thanks to the versatility of LENSTRONOMY, DOLPHIN can autonomously model both galaxy–galaxy and galaxy–quasar strong lenses.

Unified Astronomy Thesaurus concepts: Astronomy software (1855); Astronomy data modeling (1859); Strong gravitational lensing (1643); Neural networks (1933)

1. Introduction

Strong lensing occurs when a massive object, such as a galaxy or galaxy cluster, lies between a distant light source and an observer. The gravitational field of the massive object bends the light from the source, creating multiple images of the source for the observer. The positions and shapes of these images depend on the mass distribution of the lensing object and the geometric configuration of the lens, source, and observer. Consequently, strong lensing is a powerful tool for probing several key questions in cosmology and astrophysics. For example, it provides a unique opportunity to study the internal structure and evolution of galaxies at intermediate redshifts ($z \sim 0.5$; A. J. Shajib et al. 2024), the nature of dark matter (S. Vegetti et al. 2024), and the expansion history of the Universe (S. Birrer et al. 2024).

Imaging data, particularly in infrared to ultraviolet wavelengths, provides the most accessible type of strong-lensing observables in terms of telescope time usage. Traditionally, forward modeling of this imaging data to constrain the mass distribution of the lensing object is an essential step for almost all the science applications of strong lensing. At the galaxy scale, it is a standard procedure to model the imaging data up to the pixel level. In contrast, at the cluster scale, the lensing observables are summarized into astrometric positions of the lensed galaxies or knots for constraining the lensing mass distribution. Several forward-modeling software programs are used in the literature, for example, LENSTRONOMY (S. Birrer & A. Amara 2018; S. Birrer et al. 2021), PYAUTOLENS (J. Nightingale et al. 2018), GLEE (S. H. Suyu et al. 2010), GLAFIC (M. Oguri 2010), and LENSTOOL (J.-P. Kneib et al. 2011). On top of being computationally intensive, forward modeling of strong-lensing imaging data is also a time-consuming process requiring significant human intervention. The process involves several steps, such as visual identification of the lens and source components, selection of appropriate model components for them, and the evaluation of goodness of fit, which may lead to further fine-tuning of the model setup, with all these steps potentially repeated many times until a sufficiently good lens model is achieved (e.g., A. J. Shajib et al. 2019).

The significant requirement of investigator time for lens modeling has motivated the development of automated techniques that can perform the modeling process more efficiently. In particular, ongoing and upcoming large-area sky surveys such as Euclid, the Rubin Observatory Legacy Survey of Space and Time (LSST), and the Roman Space Telescope will increase the sample sizes of known lenses by 2 orders of magnitude or more (M. Oguri & P. J. Marshall 2010; T. E. Collett 2015; K. T. Abe et al. 2025), making automated

[7] NHFP Einstein Fellow.

lens-modeling algorithms essential. For example, a decision-tree-based algorithm (A. J. Shajib et al. 2019) can be used to automate the fine-tuning of the model setup (e.g., T. Schmidt et al. 2023 using LENSTRONOMY; S. Ertl et al. 2023 using GLEE; automated selection of a subset of the model settings and the subsequent optimization is also performed by PYAUTOLENS). However, these automated algorithms still require a set of initial inputs from a human investigator to initiate the lens model before successfully obtaining a good lens model. Recently, machine learning (ML) algorithms have also been used for the extraction of the lens model parameters or parameters that directly relate to galaxy properties or cosmology (e.g., Y. D. Hezaveh et al. 2017; J. Poh et al. 2022; S. Erickson et al. 2025; Euclid Collaboration et al. 2025). These ML algorithms have built-in automation, given that, once properly trained, the neural networks (NNs) can extract the parameters without further human intervention in the extraction process. However, for accurate and precise parameter extraction, a significant amount of effort and time investment from a human investigator may not be avoided when preparing the training data set.

In this work, we present DOLPHIN[8] (A. J. Shajib et al. 2025b), an automated lens-modeling pipeline that integrates artificial intelligence (AI) with forward modeling, offering a hybrid approach for fully automating the lens-modeling process. DOLPHIN uses an NN to perform visual recognition of the strong lens components to inform setting up an optimal lens model, which is then fitted with a conventional forward-modeling software program. DOLPHIN uses LENSTRONOMY as its modeling engine, which is a versatile and efficient software package for modeling strong lenses. Furthermore, LENSTRONOMY[9] has the unique combination of being open-source while being capable of modeling both galaxy–quasar and galaxy–galaxy lens systems. Thus, DOLPHIN, while also being open-source, leverages the unique features of LENSTRONOMY to provide an automated modeling pipeline for both galaxy–quasar and galaxy–galaxy lenses. DOLPHIN has been first developed and applied on samples of the galaxy–galaxy lenses in a semiautomated mode, where the visual recognition was performed by human modelers to set up the lens models (A. J. Shajib et al. 2021; C. Y. Tan et al. 2024). In this paper, we complete the AI module of DOLPHIN to transform it into a fully automated modeling pipeline.

Our presented methodology to fully automate lens modeling by combining AI with a forward-modeling framework is the first in the literature. This hybrid approach has several advantages over purely ML-based approaches. First, our approach still uses the conventional forward-modeling approach for the final optimization of the lens model, which is a well-established and reliable method for lens modeling. This approach is based on the full likelihood of the data, which leads to an optimal extraction of the information. Second, although the NN approach will be paramount for sample sizes of $\gtrsim\mathcal{O}(10^5)$, our automated hybrid approach will provide a way to validate these NN-based models for a smaller subset with the traditional forward-modeling approach.

[8] https://github.com/ajshajib/dolphin
[9] https://github.com/lenstronomy/lenstronomy

The paper is organized as follows. In Section 2, we provide an overview of DOLPHIN's workflow and associated modules and demonstrate its automated modeling capability with an example lens system. In Section 3, we describe DOLPHIN's AI that is trained to perform visual recognition of lens components in the imaging data. We conclude the paper with a discussion in Section 4.

2. Overview of DOLPHIN's Workflow

In this section, we briefly describe the workflow of DOLPHIN and the associated modules. The overview of DOLPHIN's workflow is illustrated in Figure 1. The pipeline is designed to be fully automated, where the user is only required to provide the imaging data and the initial or accurate point-spread function (PSF) applicable for the lens systems to be modeled. We give an overview of the main modules in Section 2.1, describe the fitting recipes in Section 2.2, and provide an example of lens modeling with DOLPHIN in Section 2.3.

2.1. Description of the Main Modules

In this section, we provide a general overview of the workflow of DOLPHIN's automated modeling procedure. DOLPHIN consists of three main modules as illustrated in Figure 1.

1. `ai`. This module contains an NN model for semantic segmentation, that is, identifying the lens components in the imaging data and classifying them. The segmentation is then processed by the `ai.Modeler` class to set up an optimal lens model. This setting up of an optimal model is largely based on the decision tree algorithm developed by A. J. Shajib et al. (2019). Effectively, `ai.Modeler` extracts the numbers and positions of the lens system's components (e.g., the central and satellite deflectors, the quasar images) from the segmentation, then sets up the model incorporating these components, and finally saves the model configuration in a `config.yaml` file.
2. `processor`. This module takes the provided model setup, and the class `processor.Config` sets up the lens model in the specifications required by LENSTRONOMY. Then, this module optimizes the model using optimizers such as the Particle Swarm Optimization (PSO; J. Kennedy & R. Eberhart 1995) and samples from the model posterior using the Markov Chain Monte Carlo (MCMC) method. The PSO optimization routine is crafted in the `processor.Recipe` class, which guides the initial optimization of the lens model for a more efficient convergence toward a stable solution (described in Section 2.2). The class `processor.Processor` then runs the fitting recipe for the given model configuration and saves the model output.
3. `analysis`. This module (i.e., the class `analysis.Output`) provides some quality-of-life tools to analyze the results of the lens modeling, such as the goodness of fit, the convergence of the optimizer, and the distribution of the lens model parameters.

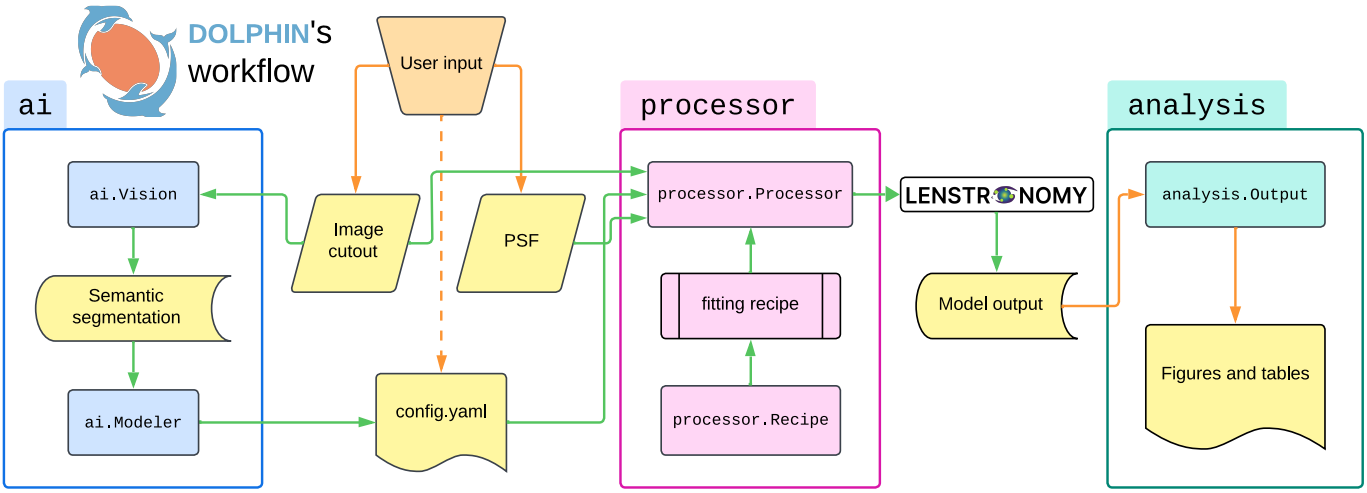

Figure 1. The overall workflow of DOLPHIN. The green arrows represent automated procedures performed by DOLPHIN, and the orange arrows represent manual procedures performed by a user. The yellow shapes represent products that are stored on the hard drive. The workflow consists of three main modules: `ai`, `processor`, and `analysis`. The `ai.Vision` module contains a trained NN model that identifies the lens components in the imaging data (i.e., semantic segmentation). The `ai.Modeler` then sets up the lens model and optimization configurations in the `config.yaml` file. This `config.yaml` file can be further tweaked manually by the user (depicted with the dashed orange arrow), if necessary. The `processor` module then optimizes the lens model with the LENSTRONOMY modeling engine employing the fitting recipe provided by the `processor.Recipe` class and then saves the model output. Finally, the `analysis` module provides tools to the user to analyze the model outputs and produce the desired figures and tables.

2.2. Fitting Recipes

Although a single PSO optimization can, in principle, provide a solution after a sufficient number of iterations, the convergence speed of the optimizer can be improved by using a fitting recipe that guides the optimization process. For example, instead of all the model parameters being free at the beginning, the optimization can be iteratively guided toward convergence by allowing subsets of the parameters to be free while keeping the others fixed at assumed or previously optimized values (from a previous iteration). The fitting recipe is a set of instructions that guides the optimization process to a stable solution. An iterative fitting recipe is also necessary when the PSF is required to be iteratively constructed as done for galaxy–quasar systems (e.g., A. J. Shajib et al. 2019; T. Schmidt et al. 2023). In addition to the "`galaxy-quasar`" fitting recipe, a purpose-built "`galaxy-galaxy`" fitting recipe is also provided for galaxy–galaxy lensing systems (A. J. Shajib et al. 2021; C. Y. Tan et al. 2024).

2.3. Running the Pipeline

The user first needs to prepare the image cutouts and the PSFs for each lens system in `hdf5` files that include the specific information and data sets required by LENSTRONOMY. These specifications of the input files, along with the desired directory structure (i.e., `io_directory`) to store the input and output files, are described in DOLPHIN's documentation. The lens model specifications of each lens are described in a `yaml` file (i.e., `config.yaml`) stored within the `io_directory`. In the fully automated pipeline, these `yaml` files are generated by the AI. However, a user can produce these `yaml` files to run DOLPHIN in a semiautomated mode or tweak the AI-generated `yaml` files as necessary before running the model optimization.

As an example, the following PYTHON code runs DOLPHIN for the quadruply lensed quasar system J2205−3727 (C. Lemon et al. 2023):

```
from dolphin.ai import Vision
from dolphin.ai import Modeler
from dolphin.processor import Processor

io_directory_path = "path/to/io_directory"

vision = Vision(io_directory_path,
↪  source_type="quasar")
vision.create_segmentation_for_single_lens(
↪  lens_name="J2205-3727", band_name="F814W")

modeler = Modeler(io_directory_path)
modeler.create_config_for_single_lens(
↪  lens_name="J2205-3727", band_name="F814W")

processor = Processor(io_directory_path)
processor.swim(lens_name="J2205-3727",
↪  model_id="example", recipe_name="galaxy-quasar")
```

The few lines of code above are all a user needs to run an automated lens model with DOLPHIN. This makes for a huge contrast with the amount of code needed to set up and run a lens model manually, for example, with LENSTRONOMY.[10] After running the pipeline, the user can plot an overview of the lens model by running the following code:

[10] For example, a JUPYTER notebook for manual lens modeling, which contains $\gtrsim$1600 lines of code.

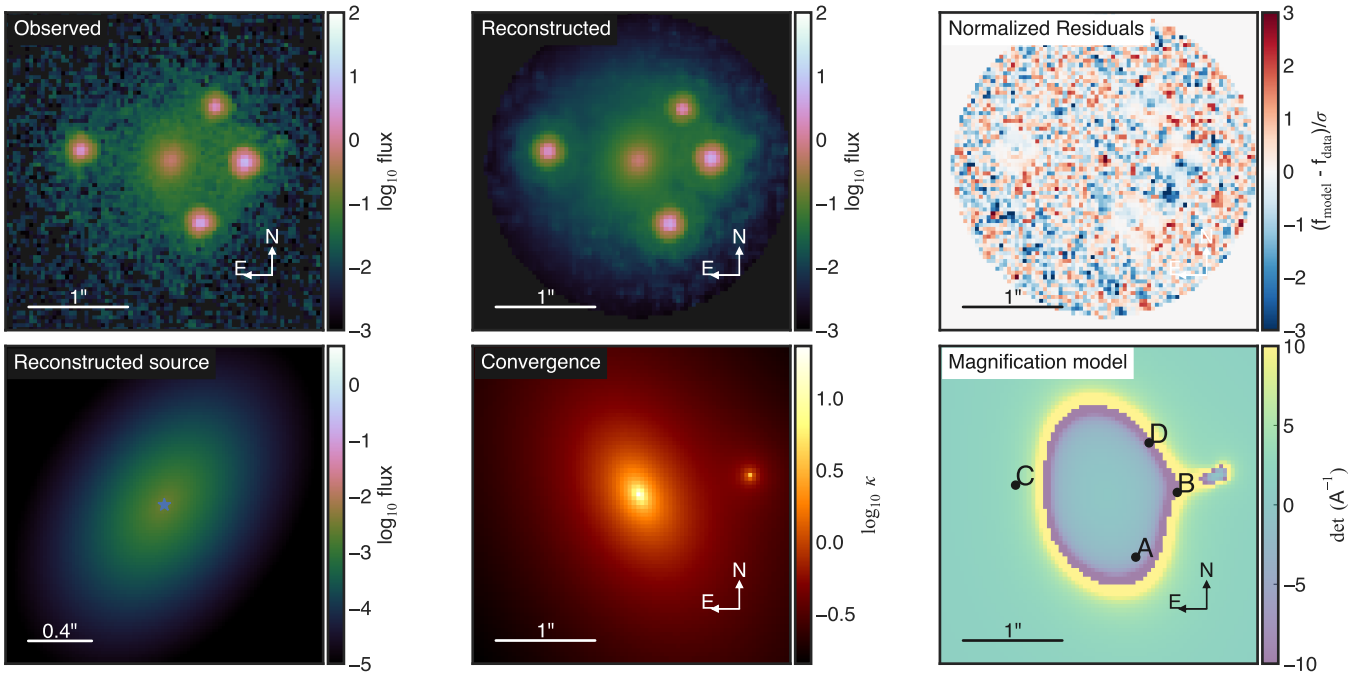

Figure 2. An overview of the lens model for the galaxy–quasar system J2205−3727. The top row shows the observed image cutout in the F814W filter, the optimized-model-based reconstruction, and the residual image. The bottom row shows the reconstructed flux distribution of the host galaxy, the convergence, and the magnification model. The blue star in the "Reconstructed source" panel points to the location of the quasar with respect to its host galaxy. The model is optimized with PSO, and the PSF is iteratively constructed with the "`galaxy-quasar`" fitting recipe from the `processor.Recipe` class.

```
from dolphin.analysis import Output

output = Output(io_directory_path)
fig =
↪  output.plot_model_overview(lens_name="J2205-3727",
↪  model_id="example")
```

For the modeling here, we use the image cutout and initial PSF estimate for the system J2205−3727 from A. J. Shajib et al. (2019) and T. Schmidt et al. (2023). Figure 2 shows the overview of the optimized lens model that is fitted to the noise level, as produced by the `output.plot_model_overview()` function. Notably, DOLPHIN's AI (described next in Section 3) has correctly identified a satellite deflector slightly northwest of the westernmost quasar image and has included it in the lens model, as can be noticed in the "Convergence" panel of Figure 2. Here, DOLPHIN models the mass distribution of the central deflector with an elliptical power law (EPL; N. Tessore & R. B. Metcalf 2015) and that of the satellite deflector with a singular isothermal ellipsoid (SIE; R. Kormann et al. 1994). Additionally, a residual shear field (alternatively called external shear; A. J. Shajib et al. 2024) is added to the mass model. The light profiles of the central and satellite deflectors and the extended source galaxy are modeled with Sérsic functions (J. L. Sérsic 1968). The model is optimized with PSO for 100 iterations with 100 particles. We use the "`galaxy-quasar`" fitting recipe from `processor.Recipe`, which includes the PSF reconstruction step in the optimization process (e.g., as performed in A. J. Shajib et al. 2019). We do not run the MCMC sampling here, as the PSO optimization is sufficient to illustrate that DOLPHIN can autonomously set up the lens model and optimize the model to produce a stable solution. In real use cases of DOLPHIN, a user may run the MCMC sampling to obtain the posterior distribution of the lens model parameters.

3. ML-based Visual Recognition of Lens Components

In this section, we describe the ML-based visual recognition model that identifies the lens components in the imaging data within the `ai.Vision` class. We describe the network architecture and activation functions in Section 3.1, the training data set in Section 3.2, the training procedure in Section 3.3, and the validations and tests in Section 3.4.

3.1. NN Architecture

We implement a U-Net architecture using TENSORFLOW and KERAS (M. Abadi et al. 2016). The U-Net architecture is a popular choice for semantic segmentation, which consists of an encoder path (downsampling) and a decoder path (upsampling; O. Ronneberger et al. 2015).

Our model (shown in Figure 3) takes single-band images of size 128×128 as inputs. Future upgrades of DOLPHIN will include new NN models to allow multiband imaging for improved performance in segmentation. The encoder path in the U-Net employs two 3×3 convolutional layers followed by a 2×2 max-pooling layer in each downsampling step. Each upsampling step in the decoder path consists of a 2×2 transposed convolution, an attention block (O. Oktay et al. 2018), and then a 3×3 convolutional layer. The attention block is applied to the feature maps from both the encoder and decoder paths, and its output is concatenated with the feature map from the decoder path before passing on to the next layer. Each convolutional layer has the rectified linear unit (ReLU) activation function to induce nonlinearity (V. Nair & G. E. Hinton 2010). To prevent overfitting, we add dropout layers between the two convolutional layers in each block of

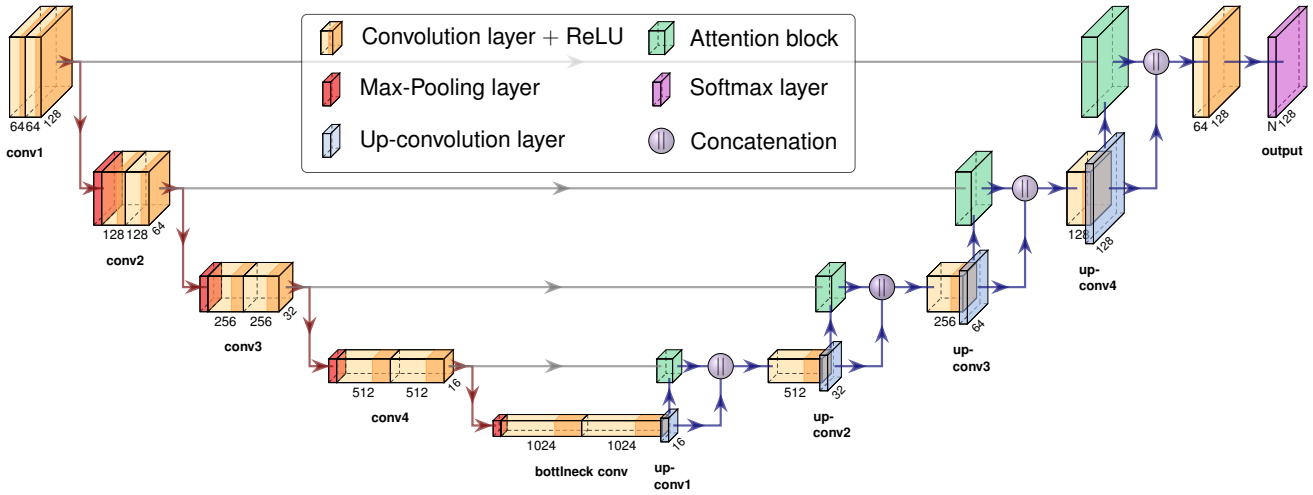

Figure 3. U-Net architecture of our NN to perform semantic segmentation of imaging data for galaxy-scale strong lenses. The U-Net model consists of an encoder path (downsampling; red arrows) and a decoder path (upsampling; blue arrows). The encoder path consists of convolutional layers followed by max-pooling layers. We use two consecutive convolutional layers in each block with a dropout layer in between. The decoder path consists of upsampling layers followed by convolutional layers. The gray arrows represent the skip connections from a downsampling block to an upsampling one at the same level of the U-Net. The output layer has $N = 4$ (for galaxy–galaxy lenses) or $N = 5$ (for galaxy–quasar lenses) channels, each providing the probability of one of these classes for every pixel: background, central deflector, quasar image, lensed arc, and satellite of the central deflector.

the encoder path, with a rate of 0.1 for the first two layers, 0.2 for the next two layers, and increasing to 0.3 for the bottleneck layer.

The attention block we use to filter relevant features is a soft attention mechanism. The encoder and decoder features are transformed using 1×1 convolutions. Then, we add the transformed features and apply the ReLU activation function. Next, an attention mask is generated using a sigmoid activation function. Finally, we multiply the attention mask by the decoder feature map to produce the output of the attention block.

For an image of the galaxy–quasar system, we identify five classes of components: background, central lens galaxy, quasar image, lensed arc from the quasar host galaxy, and satellite of the central lens galaxy. Therefore, the output layer for the network trained with galaxy–quasar systems has five channels, one containing the probability for each class. For an image of the galaxy–galaxy system, we simply exclude the quasar image class; thus, the output layer has four channels. We apply the softmax activation function (J. Bridle 1989) before the final output layer so that the values in the output channel represent the probability for the corresponding class at each pixel.

The loss function we used combines dice loss (C. H. Sudre et al. 2017) and focal loss (T.-Y. Lin et al. 2018) as it is suitable for multiclass classification and imbalanced data sets. For the focal loss, we set `alpha = 0.25` and `gamma = 2` for optimal results.

3.2. Training Data Set

We create a synthetic data set using LENSTRONOMY for training. Our simulated images mimic the image quality and resolution of the Hubble Space Telescope's (HST) Wide-Field Camera 3 (WFC3) in the F814W filter. Future upgrades to DOLPHIN will deliver new NN models trained with multiple HST filters and also with ground-based imaging qualities, such as that of the Rubin Observatory LSST. We also provide our code for data set creation and training to enable users to regenerate their own training set for the desired imaging specifications (e.g., choice of bands, ground-based or space-based resolution) and sample-level characteristics.

We use an EPL mass profile to simulate the central deflector galaxy and add an external shear field on top of it. We also add satellite deflectors with SIE mass profiles. The probability for the number of satellites in each lens system is $p(N_{\rm sat}) \propto 1/(1 + N_{\rm sat})$ with $N_{\rm sat}^{\rm max} = 3$. We simulate the central and satellite deflectors' light profiles with Sérsic functions. We simulate the source galaxy's light distribution using real spiral galaxy images from the HST legacy imaging from A. J. Shajib et al. (2022). The quasar images are simulated as point sources convolved with a realistic simulated PSF from TINY TIM (J. E. Krist et al. 2011). The Einstein radius for the main lens galaxy is randomly drawn from a uniform distribution between $0\farcs75$ and $1\farcs5$, typical for galaxy-scale lenses currently known. Similarly, the probability distributions for all the other parameters in the simulation are tuned to reproduce those observed in the galaxy-scale lens samples presented by A. J. Shajib et al. (2019) and C. Y. Tan et al. (2024). Finally, we add noise to the images that accounts for the background, read, and Poisson noise corresponding to 1429 s of exposure. We set the pixel scale to $0\farcs05$ with the image dimension being 128×128 pixels.

We create the segmentation mask, that is, the training labels, from the noiseless simulated images. We define five classes for the mask: central deflector, quasar host, quasar, satellite, and background. We create the labels through ad hoc conditions applied to each class's relative flux levels and signal-to-noise ratios. Figure 4 illustrates some examples of the created segmentation masks (the "Ground truth" columns). The specific algorithm for this can be found in the `ai_training/data set.py` module of the released code suite on GitHub.

3.3. Training Procedure

We simulated 110,000 images for each of the galaxy–galaxy and galaxy–quasar lens types. We used 90,000 of them for training, 10,000 for validation, and 10,000 for testing. We randomly remove the deflector's light from a fraction of the

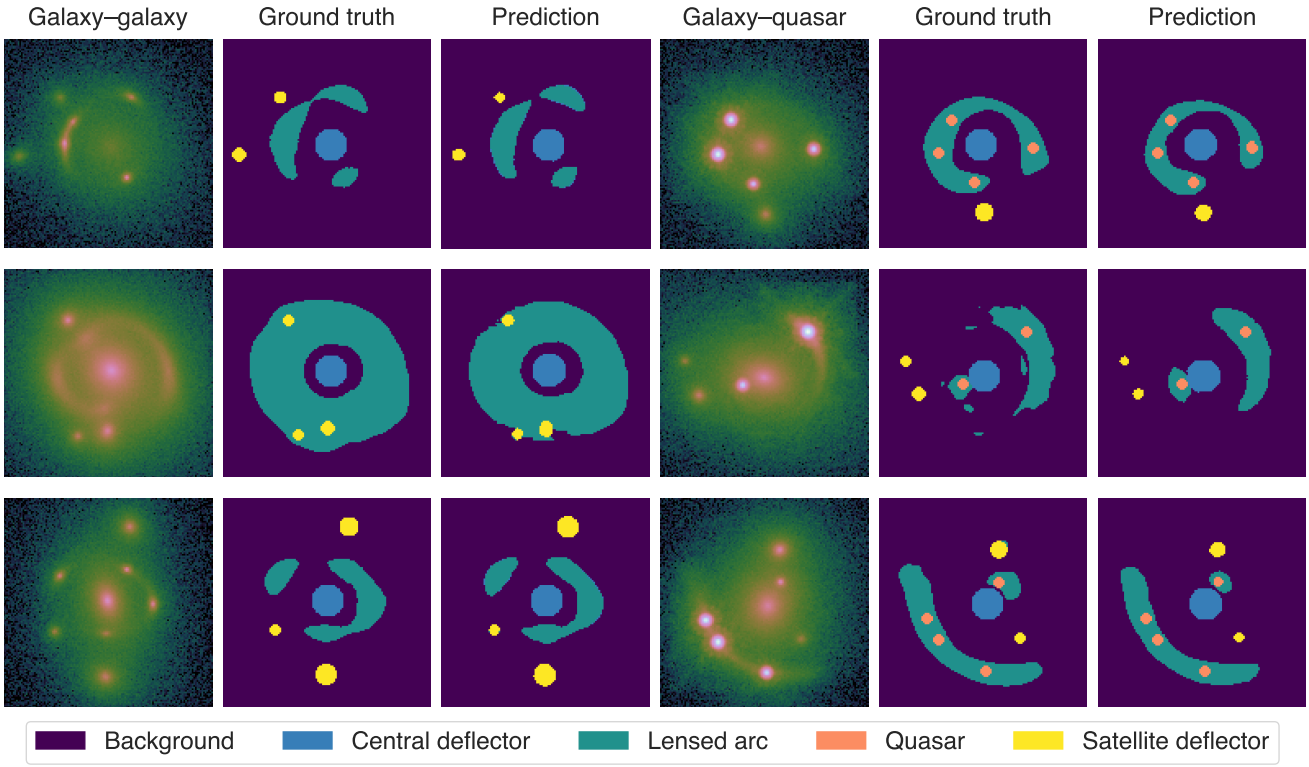

Figure 4. Validation results for both galaxy–galaxy (first three columns) and galaxy–quasar (last three columns) lens systems. The first and fourth columns show the inputs from the validation set of synthetic images, the second and fifth columns show the ground-truth segmentation masks (i.e., labels), and the third and sixth columns show the NN-predicted segmentation masks. The color for each class in the segmentation mask is labeled in the legend. The NN model shows strong performance in segmenting the lens components, with the F1 score for each class being above 86%.

images (10%) so that the network does not learn to label the central pixels within an image as the central deflector by default, given the predominance of this correlation in the training data set. We further augment our data by applying a zoom effect up to 50% consistently on both the input images and labels to generalize our model's performance for real input images that would be resized to 128×128 pixels from an arbitrary size.

We train the model on the training set with the adaptive momentum (Adam; D. P. Kingma & J. Ba 2014) optimizer over 50 epochs, with a batch size of 32. We used early stopping with a patience of 10 epochs to prevent overfitting and restore the best weights by monitoring validation loss. Additionally, we applied a learning rate reduction technique with a factor of 0.5 when the validation loss did not improve for five consecutive epochs, with a minimum learning rate of 1×10^{-6}. The model was trained using an Nvidia L4 GPU on the Google Colaboratory platform, which took ∼7 hr of wall time for each of the two source types.

3.4. Validations and Tests

We validate the model on the test set to evaluate its performance. We calculate the precision, recall, F1 score, and intersection over union (IoU) for each class. The precision is the ratio of true positives to the sum of true and false positives. The recall is the ratio of true positives to the sum of true positives and false negatives. The F1 score is the harmonic mean of precision and recall. The IoU is the overlap between the predicted bounding region and the ground-truth bounding region relative to their combined area. The IoU is a commonly used metric in computer vision to measure object detection accuracy. Table 1 shows these performance metrics for our trained NN model. The model performs strongly in segmenting the lens components, with the F1 score for each class being above 86% and the IoU score being above 76%. For reference, an F1 score above 80% and an IoU score above 70% are generally considered good performance for image segmentation tasks (e.g., A. Geiger et al. 2012; O. Ronneberger et al. 2015).

Table 1
Performance Metrics of Our Trained NN Model in Semantic Segmentation

Lens Type	Classes or Components	Precision (%)	Recall (%)	F1 Score (%)	IoU (%)
Galaxy–quasar	Background	98.42	99.67	99.04	98.10
	Central deflector	91.52	98.67	94.96	90.41
	Lensed arc	97.06	81.05	88.34	79.11
	Quasar	97.27	97.51	97.39	94.91
	Satellite deflector	91.68	81.90	86.51	76.23
Galaxy–galaxy	Background	99.25	99.60	99.43	98.86
	Central deflector	79.56	98.39	87.98	78.54
	Lensed arc	97.68	90.01	93.69	88.13
	Satellite deflector	91.44	88.66	90.02	81.85

In Figure 5, we show the segmentation results for real data for six galaxy–galaxy lenses from the Sloan Lens ACS (SLACS; A. S. Bolton et al. 2006) sample and six galaxy–quasar lenses from the STRong lensing Insights into the Dark Energy Survey (STRIDES; A. J. Shajib et al. 2019; T. Schmidt

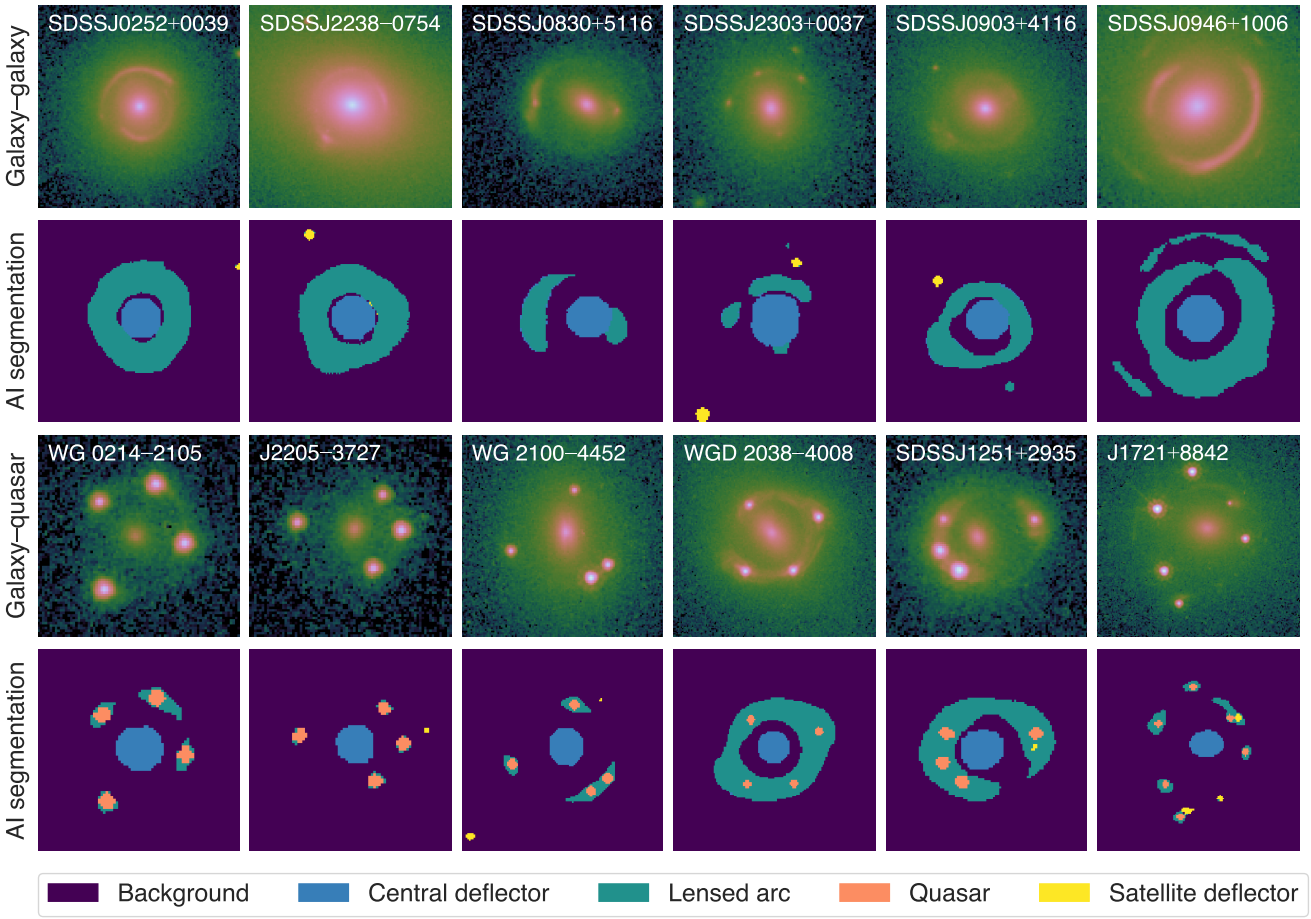

Figure 5. Segmentation results by our trained NN model for six galaxy–galaxy lenses (top two rows) from the SLACS sample and six galaxy–quasar lenses (bottom two rows) from the STRIDES sample. Here, the input images are HST imaging in the F814W filter, with the mean background subtracted. Our NN model showcases superb performance in identifying the lens components that are labeled in the legend. The last galaxy–galaxy lens illustrated here is a compound lens, the "Jackpot" lens, and the last galaxy–quasar lens is another compound lens, the first Einstein zigzag, with six images appearing of the same background quasar. Although these two highly complex systems fall outside the typical use cases for DOLPHIN, as extracting their scientific richness would require well-tailored, handcrafted models, they serve as illustrative examples of the exceptional performance of DOLPHIN's NN model in segmenting lens imaging data.

et al. 2023) sample. DOLPHIN reshapes the real HST images from their intrinsic pixel-grid dimension to 128×128 pixels before feeding them into the NN model. Figure 5 illustrates the superb performance of the NN model in the semantic segmentation of the real images, where all satellite galaxies and quasar images are correctly identified in these images. For an interesting test, we provided the "Jackpot" lens (R. Gavazzi et al. 2008), a compound lens (top row, last column in Figure 5), and the first Einstein zigzag (F. Dux et al. 2025), another compound lens with six images of the same background quasar. Our NN model detects both sets of arcs from the two background galaxies in the "Jackpot" lens. For the Einstein zigzag lens, our NN model correctly detects all six quasar images, but it confuses the small arcs from the intervening source galaxy as satellite deflectors. We note that the intervening source galaxy, in fact, acts as a deflector for the background quasar, which is located farther behind. However, such confusion by the NN model is not a concern here, as such complicated lenses are not expected to be modeled with an automated pipeline, given the extreme level of complexity in the lensing configuration requiring carefully handcrafted models enabling their science cases in precision cosmology (e.g., T. Schmidt et al. 2025). Although these very rare (1 in 500 lenses; F. Dux et al. 2025) and complicated systems will be out of the intended use case of DOLPHIN, they make for illustrative examples of the superb segmentation performance of our NN model. In the small fraction of severely failed cases of image segmentation from the NN models, common occurrences of misidentified components include very faint arcs not being detected, highly compact nearby satellites being misidentified as lensed quasars, or compact knots on the lensed arcs being misidentified as satellites. These failure cases may be mitigated in future upgrades by rebalancing the training data set to increase the relative presence of these failure modes, by utilizing multiband data to incorporate color information, by applying judicious weights through the loss function to correct data set imbalances, by increasing the training data set size, or through a combination of all the above.

4. Discussion and Conclusion

In this paper, we present DOLPHIN, the first fully automated forward-modeling pipeline for galaxy-scale strong lenses, capable of modeling both galaxy–galaxy and galaxy–quasar types. DOLPHIN combines an AI trained with deep learning with a traditional forward-modeling engine, namely, LENSTRONOMY, to provide a hybrid approach to achieve full automation in lens modeling. DOLPHIN performs semantic

segmentation for the strong lens components with an NN model, sets up an optimal lens model using the segmentation, and then fits the model with the forward-modeling engine LENSTRONOMY. We provided an overview of the workflow of DOLPHIN and the associated modules and a demonstration of the automated lens modeling in Section 2. We then described the NN model for visual recognition in Section 3. Our trained NN model shows strong performance in segmenting the lens components. We also showed that the model can identify the lens components in the real data images of both galaxy–galaxy and galaxy–quasar systems. This performance level in semantic segmentation of the imaging data is sufficient to set up an optimally tuned lens model for most galaxy-scale lens systems, including those that contain up to two or three satellite galaxies of the central deflector. Given the ease of use in setting up and running DOLPHIN, both in personal computers and on high-performance computing clusters, DOLPHIN will vastly reduce the amount of human investigator time in modeling large samples of lenses, as already demonstrated with its semiautomated (i.e., using human visual recognition) modeling mode (A. J. Shajib et al. 2021; C. Y. Tan et al. 2024; N. B. Hogg et al. 2025). Thus, the full automation achieved through the NN model presented in this paper will further reduce the human investigator time spent modeling large samples.

Like any software program, DOLPHIN will receive continuous and future upgrades based on necessity and user feedback. For example, although the `Processor` class can currently perform modeling with multiband imaging data as demonstrated in C. Y. Tan et al. (2024), the NN model is currently trained only with single-band specification in the optical band (i.e., the WFC3/F814W filter). The optical band tends to be the most useful for lensing information as it sits in the sweet spot for maximizing the signal-to-noise for both the lensing and source galaxy with less blending between the two, given the "red" and "blue" colors of the lens and source galaxies, respectively. As a result, the optical (F814W) filter also tends to be the primary choice for full ML-based extraction of the lensing parameters (e.g., S. Erickson et al. 2025). Future improvements in DOLPHIN's computer vision capability will allow multiband imaging input to harness the color information for better detection of the lens components. In addition, a separate purpose-built network for ground-based imaging specifications will extend the pipeline's scope of usage. Furthermore, more classes of objects can be included in the training to improve the robustness of component detection, for example, stars, cosmic rays, etc., that are sometimes present in the image cutout of real systems.

The DOLPHIN pipeline is especially useful for the large lens samples that will be discovered by the upcoming large-area sky surveys, such as the Euclid, Rubin, and Roman observatories. Several previous studies explored using ML algorithms for fully automated lens modeling (e.g., Y. D. Hezaveh et al. 2017; J. Poh et al. 2022; S. Erickson et al. 2025). An AI-powered forward modeler like DOLPHIN has important complementarity with these ML-based algorithms and also with traditional handcrafted lens models (e.g., S. Birrer et al. 2019; A. J. Shajib et al. 2020). This is because these three approaches will shine in different regimes of sample sizes that are largely determined by the somewhat correlated rarity and complexity of the lens systems, the availability of high-resolution imaging, and the specific science cases in which they will be employed. More complex lens systems, for example, those with multiple lens and source planes (e.g., A. J. Shajib et al. 2020; F. Dux et al. 2025), albeit requiring handcrafted models, are richer in information; thus, their scientific usefulness justifies the manual and time-consuming modeling approach by a human investigator. A major fraction of the lensing systems have sufficiently simple lens configurations that can be modeled with DOLPHIN. For example, C. Y. Tan et al. (2024) achieved an ∼80% success rate in modeling galaxy–galaxy lenses from a considered sample size of $\mathcal{O}(10^2)$. DOLPHIN is especially suitable for such sample sizes of $\mathcal{O}(10^2)$–$\mathcal{O}(10^3)$. As a forward-modeling pipeline, DOLPHIN will still require running sampling methods, such as the MCMC, to obtain the best-fit lens model parameters and their uncertainties. As a result, sample sizes much larger than $\mathcal{O}(10^3)$ can potentially be limited by computational resources for CPU-based modeling engines. However, GPU-accelerated, autodifferentiable modeling engines, such as GIGA-LENS (A. Gu et al. 2022), HERCULENS (A. Galan et al. 2022), and JAXTRONOMY,[11] will likely be up to the task. In particular, JAXTRONOMY is a direct port of LENSTRONOMY; thus, the modeling engine for DOLPHIN can be seamlessly swapped from LENSTRONOMY to JAXTRONOMY.

In addition, ML-based algorithms will be particularly useful when it comes to sample sizes of $\mathcal{O}(10^5)$ that are forecasted to be discovered from the Euclid, Rubin, and Roman observatories (T. E. Collett 2015; A. J. Shajib et al. 2025a). However, automated forward-modeling pipelines like DOLPHIN will be paramount to validate the ML-based models for a smaller subsample, as done by S. Erickson et al. (2025). In summary, we envision that all three approaches for lens modeling will be essential and complementary in the near future, with each shining at different regimes of sample sizes—$\mathcal{O}(10)$, handcrafted modeling for complex and scientifically rich systems; $\mathcal{O}(10^2)$–$\mathcal{O}(10^4)$, automated forward modeling with DOLPHIN; and $\mathcal{O}(10^4)$–$\mathcal{O}(10^5)$, ML-based lensing parameter extraction.

Acknowledgments

We thank the anonymous referee for helpful comments, which improved this manuscript. Support for this work was provided by NASA through the NASA Hubble Fellowship grant HST-HF2-51492 awarded to A.J.S. by the Space Telescope Science Institute (STScI), which is operated by the Association of Universities for Research in Astronomy, Inc., for NASA, under contract NAS5-26555. A.J.S. also received support from NASA through the STScI grants HST-GO-15320 and HST-GO-16773. This project was supported by the U.S. Department of Energy (DOE) Office of Science Distinguished Scientist Fellow Program. C.Y.T. was supported by NASA through the STScI grant HST-AR-16149. T.T. acknowledges support from the National Science Foundation through grant AST-2407277 and by the Gordon and Betty Moore Foundation through grant 8548.

Author Contributions

A.J.S.: conceptualization, data curation, funding acquisition, project administration, software, visualization, writing—original draft, writing—review and editing. N.S.N.: methodology, writing—original draft, writing—review and editing. C.Y.T.: software, writing—review and editing. V.S.: data curation.

[11] https://github.com/lenstronomy/jaxtronomy

S.B.: supervision, writing—review and editing. T.T.: supervision, funding acquisition, writing—review and editing. J.F.: supervision, funding acquisition, writing—review and editing.

Facilities: HST (WFC3).

Software: LENSTRONOMY (S. Birrer & A. Amara 2018; S. Birrer et al. 2021), TENSORFLOW (M. Abadi et al. 2016), KERAS (https://github.com/keras-team/keras), NUMPY (T. E. Oliphant 2015), MATPLOTLIB (J. D. Hunter 2007), H5PY (A. Collette 2013), SEABORN (M. Waskom et al. 2014), JUPYTER (T. Kluyver et al. 2016), PLOTNEURALNETWORK (H. Iqbal 2018), TINY TIM (J. E. Krist et al. 2011).

ORCID iDs

Anowar J. Shajib https://orcid.org/0000-0002-5558-888X
Chin Yi Tan https://orcid.org/0000-0003-0478-0473
Simon Birrer https://orcid.org/0000-0003-3195-5507
Tommaso Treu https://orcid.org/0000-0002-8460-0390

References

Abadi, M., Barham, P., Chen, J., et al. 2016, TensorFlow: A System for Large-scale Machine Learning, arXiv:1605.08695
Abe, K. T., Oguri, M., Birrer, S., et al. 2025, OJAp, 8, 8
Birrer, S., & Amara, A. 2018, PDU, 22, 189
Birrer, S., Millon, M., Sluse, D., et al. 2024, SSRv, 220, 48
Birrer, S., Shajib, A. J., Gilman, D., et al. 2021, JOSS, 6, 3283
Birrer, S., Treu, T., Rusu, C. E., et al. 2019, MNRAS, 484, 4726
Bolton, A. S., Burles, S., Koopmans, L. V. E., Treu, T., & Moustakas, L. A. 2006, ApJ, 638, 703
Bridle, J. 1989, Advances in Neural Information Processing Systems, Vol. 2 (San Mateo, CA: Morgan-Kaufmann) https://proceedings.neurips.cc/paper/1989/hash/0336dcbab05b9d5ad24f4333c7658a0e-Abstract.html
Collett, T. E. 2015, ApJ, 811, 20
Collette, A. 2013, Python and HDF, Vol. 5 (Sebastopol, CA: O'Reilly)
Dux, F., Millon, M., Lemon, C., et al. 2025, A&A, 694, A300
Erickson, S., Wagner-Carena, S., Marshall, P., et al. 2025, AJ, 170, 44
Ertl, S., Schuldt, S., Suyu, S. H., et al. 2023, A&A, 672, A2
Euclid Collaboration, Busillo, V., Tortora, C., et al. 2025, arXiv:2503.15329
Galan, A., Vernardos, G., Peel, A., Courbin, F., & Starck, J. L. 2022, A&A, 668, A155
Gavazzi, R., Treu, T., Koopmans, L. V. E., et al. 2008, ApJ, 677, 1046
Geiger, A., Lenz, P., & Urtasun, R. 2012, in IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 3354
Gu, A., Huang, X., Sheu, W., et al. 2022, ApJ, 935, 49
Hezaveh, Y. D., Levasseur, L. P., & Marshall, P. J. 2017, Natur, 548, 555
Hogg, N. B., Shajib, A. J., Johnson, D., & Larena, J. 2025, arXiv:2501.16292
Hunter, J. D. 2007, CSE, 9, 90
Iqbal, H. 2018, HarisIqbal88/PlotNeuralNet v1.0.0, Zenodo, doi:10.5281/zenodo.2526396
Kennedy, J., & Eberhart, R. 1995, in Proc. ICNN'95—Int. Conf. on Neural Networks (Piscataway, NJ: IEEE)
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, ed. F. Loizides & B. Schmidt (Amsterdam: IOS Press BV), 87
Kneib, J.-P., Bonnet, H., Golse, G., et al. 2011, LENSTOOL: A Gravitational Lensing Software for Modeling Mass Distribution of Galaxies and Clusters (Strong and Weak Regime), Astrophysics Source Code Library, ascl:1102.004
Kormann, R., Schneider, P., & Bartelmann, M. 1994, A&A, 284, 285
Krist, J. E., Hook, R. N., & Stoehr, F. 2011, Proc. SPIE, 8127, 81270J
Lemon, C., Anguita, T., Auger-Williams, M. W., et al. 2023, MNRAS, 520, 3305
Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. 2018, arXiv:1708.02002
Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Machine Learning, ICML'10 (Madison, WI: Omnipress), 807
Nightingale, J., Dye, S., & Massey, R. 2018, MNRAS, 478, 4738
Oguri, M. 2010, PASJ, 62, 1017
Oguri, M., & Marshall, P. J. 2010, MNRAS, 405, 2579
Oktay, O., Schlemper, J., Le Folgoc, L., et al. 2018, arXiv:1804.03999
Oliphant, T. E. 2015, Guide to NumPy (2nd ed.; Scotts Valley, CA: CreateSpace Independent Publishing Platform)
Poh, J., Samudre, A., Ciprijanovic, A., et al. 2022, arXiv:2211.05836
Ronneberger, O., Fischer, P., & Brox, T. 2015, arXiv:1505.04597
Schmidt, T., Treu, T., Birrer, S., et al. 2023, MNRAS, 518, 1260
Schmidt, T., Treu, T., Birrer, S., et al. 2025, A&A, 700, A92
Sérsic, J. L. 1968, Atlas de Galaxias Australes (Cordoba: Observatorio Astronomico)
Shajib, A. J., Birrer, S., Treu, T., et al. 2019, MNRAS, 483, 5649
Shajib, A. J., Birrer, S., Treu, T., et al. 2020, MNRAS, 494, 6072
Shajib, A. J., Glazebrook, K., Barone, T., et al. 2022, ApJ, 938, 141
Shajib, A. J., Nihal, N. S., & Tan, C. Y. 2025b, dolphin: An Automated Lens Modeling Pipeline Powered by AI for Galaxy-scale Strong Lenses v1, Zenodo, doi: 10.5281/zenodo.16587211
Shajib, A. J., Smith, G. P., Birrer, S., et al. 2025a, RSPTA, 383, 20240117
Shajib, A. J., Treu, T., Birrer, S., & Sonnenfeld, A. 2021, MNRAS, 503, 2380
Shajib, A. J., Vernardos, G., Collett, T. E., et al. 2024, SSRv, 220, 87
Sudre, C. H., Li, W., Vercauteren, T., Ourselin, S., & Cardoso, M. J. 2017, Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (Berlin: Springer), 240
Suyu, S. H., Marshall, P. J., Auger, M. W., et al. 2010, ApJ, 711, 201
Tan, C. Y., Shajib, A. J., Birrer, S., et al. 2024, MNRAS, 530, 1474
Tessore, N., & Metcalf, R. B. 2015, A&A, 580, A79
Vegetti, S., Birrer, S., Despali, G., et al. 2024, SSRv, 220, 58
Waskom, M., Botvinnik, O., Hobson, P., et al. 2014, seaborn: v0.5.0 (November 2014), Zenodo, doi:10.5281/zenodo.12710