deeplenstronomy: A dataset simulation package for strong gravitational lensing

Robert Morgan[*1, 2], **Brian Nord**[3, 4], **Simon Birrer**[5], **Joshua Yao-Yu Lin**[6], **and Jason Poh**[4]

1 University of Wisconsin-Madison **2** Legacy Survey of Space and Time Data Science Fellowship Program **3** Fermi National Accelerator Laboratory **4** University of Chicago **5** Stanford University **6** University of Illinois Urbana-Champaign

DOI: 10.21105/joss.02854

Software

- Review
- Repository
- Archive

Editor: Pierre de Buyl

Reviewers:

- @shreyasbapat
- @jiwoncpark

Submitted: 10 November 2020
Published: 03 February 2021

Background

Astronomical observations and statistical modeling permit the high-fidelity analysis of strong gravitational lensing (SL) systems, which display an astronomical phenomenon in which light from a distant object is deflected by the gravitational field of another object along its path to the observer. These systems are of great scientific interest because they provide information about multiple astrophysical and cosmological phenomena, including the nature of dark matter, the expansion rate of the Universe, and characteristics of galaxy populations. They also serve as standing tests of the theory of General Relativity and modified theories of gravity.

Traditional searches for SL systems have involved time- and effort-intensive visual or manual inspection of images by humans to identify characteristic features — like arcs, particular color combinations, and object orientations. However, a comprehensive search using the traditional approach is prohibitively expensive for large numbers of images, like those in cosmological surveys — e.g., the Sloan Digital Sky Survey (York et al., 2000), the Dark Energy Survey (Abbott et al., 2018), and the Legacy Survey of Space and Time (LSST) (Ivezić et al., 2019). To automate the SL detection process, techniques based on machine learning (ML) are beginning to overtake traditional approaches for scanning astronomical images. In particular, deep learning techniques have been the focus, but they require large sets of labeled images to train these models. Because of the relatively low number of observed SL systems, simulated datasets of images are often needed. Thus, the composition and production of these simulated datasets have become integral parts of the SL detection process.

One of the premier tools for simulating and analyzing SL systems, `lenstronomy` (Birrer & Amara, 2018), works by the user specifying the properties of the physical systems, as well as how they are observed (e.g., telescope and camera) through a `python`-based application programming interface (API) to generate a single image. Generating populations of SL systems that are fit for neural network training requires additional infrastructure.

Statement of need

Due to the inherent dependence of the performance of ML approaches on their training data, the deep learning approach to SL detection is in tension with scientific reproducibility without a clear prescription for the simulation of the training data. There is a critical need for a tool that simulates full datasets in an efficient and reproducible manner, while enabling the use of all the features of the `lenstronomy` simulation API. Additionally, this tool should simplify

*Corresponding author

user interaction with lenstronomy and organize the simulations and associated metadata into convenient data structures for deep learning problems. Multiple packages have been developed to generate realistic training data by wrapping around lenstronomy: baobab (Park, 2021) generates training sets for lens modeling and hierarchical inference and the LSST Dark Energy Science Collaboration's SLSprinkler (Kalmbach et al., 2020) adds strongly lensed variable objects into catalogs and images. Nonetheless, the need for a simple, general tool capable of efficiently simulating any astronomical system in a reproducible manner while giving the user complete freedom to set the properties of objects remains.

Summary

deeplenstronomy generates SL datasets by organizing and expediting user interaction with lenstronomy. The user creates a single yaml-style configuration file that describes the aspects of the dataset: number of images, properties of the telescope and camera, cosmological parameters, observing conditions, properties of the physical objects, and geometry of the SL systems. deeplenstronomy parses the configuration file and generates the dataset, producing both the images and the parameters that led to the production of each image as outputs. The configuration files can easily be shared, enabling users to easily reproduce each other's training datasets.

The premier objective of deeplenstronomy is to help astronomers make their training datasets as realistic as possible. To that end, deeplenstronomy contains built-in features for the following functionalities: use any stellar light profile or mass profile in lenstronomy; simulate a variety of astronomical systems such as single galaxies, foreground stars, galaxy clusters, supernovae, and kilonovae, as well as any combination of those systems; fully control the placements of objects in the simulations; use observing conditions of real astronomical surveys; draw any parameter from any probability distribution; introduce any correlation; and incorporate real images into the simulation. Furthermore, deeplenstronomy facilitates realistic time-domain studies by providing access to public spectral energy distributions of observed supernovae and kilonovae and incorporating the transient objects into time series of simulated images. Finally, deeplenstronomy provides data visualization functions to enable users to inspect their simulation outputs. These features and the path from configuration file to full data set are shown in Figure 1.

deeplenstronomy makes use of multiple open-source software packages: lenstronomy is used for all gravitational lensing calculations and image simulation; numpy (Harris et al., 2020) Arrays are used internally to store image data and perform vectorized calculations; pandas (McKinney & others, 2010) DataFrames are utilized for storing simulation metadata and file reading and writing; scipy (Virtanen et al., 2020) is used for integration and interpolation; matplotlib (Hunter, 2007) functions are used for image visualization; astropy (Astropy Collaboration et al., 2013) is used for cosmological calculations and color image production; h5py (Collette, 2014) is utilized for saving images; and PyYAML (Simonov & Net, 2006) is used to manage the configuration file. While not used directly, some python-benedict (Caccamo, 2018) functionalities helped to create deeplenstronomy's data structures and internal search algorithms.

deeplenstronomy is packaged and disseminated via PyPI. Documentation and example notebooks are available on the deeplenstronomy website. Any bugs or feature requests can be opened as issues in the GitHub repository (Morgan, 2020).

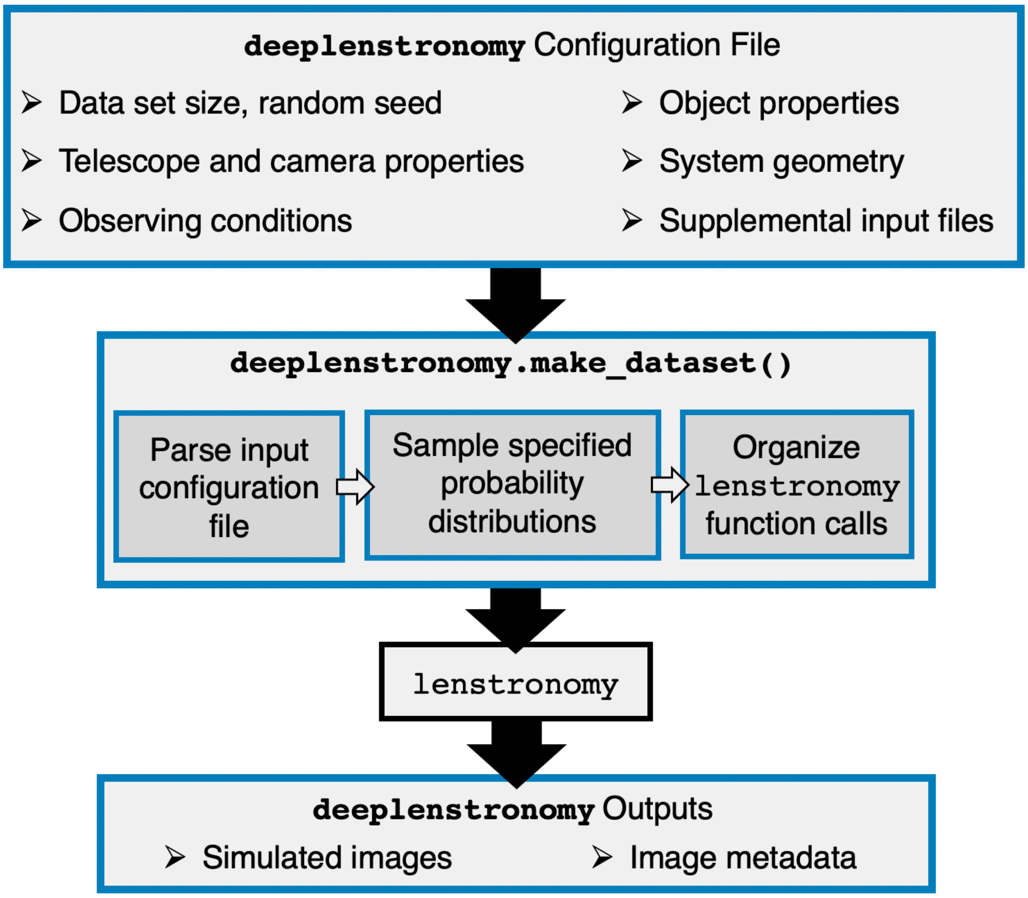

Figure 1: The `deeplenstronomy` process. Dataset properties, camera and telescope properties, observing conditions, object properties (e.g., `lenstronomy` light and mass profiles, point sources, and temporal behavior), the geometry of the SL systems, and optional supplemental input files (e.g., probability distributions, covariance matrices, and image backgrounds) are specified in the main configuration file. `deeplenstronomy` then intreprets the configuration file, calls `lenstronomy` simulation functionalities, and organizes the resulting images and metadata.

Acknowledgements

R. Morgan thanks the LSSTC Data Science Fellowship Program, which is funded by LSSTC, NSF Cybertraining Grant #1829740, the Brinson Foundation, and the Moore Foundation; his participation in the program has benefited this work. R. Morgan also thanks the Universities Research Association Fermilab Visiting Scholar Program for funding his work on this project.

We acknowledge the Deep Skies Lab as a community of multi-domain experts and collaborators who've facilitated an environment of open discussion, idea-generation, and collaboration. This community was important for the development of this project. We acknowledge contributions from Joao Caldeira during the early stages of this project.

Work supported by the Fermi National Accelerator Laboratory, managed and operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

References

Abbott, T. M. C., Abdalla, F. B., Alarcon, A., Aleksić, J., Allam, S., Allen, S., Amara, A., Annis, J., Asorey, J., Avila, S., Bacon, D., Balbinot, E., Banerji, M., Banik, N., Barkhouse, W., Baumer, M., Baxter, E., Bechtol, K., Becker, M. R., … Dark Energy Survey Collaboration. (2018). Dark Energy Survey year 1 results: Cosmological constraints from galaxy clustering and weak lensing. *Physical Review D*, *98*(4), 043526. https://doi.org/10.1103/PhysRevD.98.043526

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., … Streicher, O. (2013). Astropy: A community Python package for astronomy. *Astronomy & Astrophysics*, *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Birrer, S., & Amara, A. (2018). Lenstronomy: Multi-purpose gravitational lens modelling software package. *Physics of the Dark Universe*, *22*, 189–201. https://doi.org/10.1016/j.dark.2018.11.002

Caccamo, F. (2018). *Python-benedict*. https://github.com/fabiocaccamo/python-benedict

Collette, A. (2014). *Python and HDF5*. O'Reilly.

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Fernández del Río, J., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*, 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., Alonso, D., AlSayyad, Y., Anderson, S. F., Andrew, J., Angel, J. R. P., Angeli, G. Z., Ansari, R., Antilogus, P., Araujo, C., Armstrong, R., Arndt, K. T., Astier, P., Aubourg, É., … Zhan, H. (2019). LSST: From Science Drivers to Reference Design and Anticipated Data Products. *The Astrophysical Journal*, *873*(2), 111. https://doi.org/10.3847/1538-4357/ab042c

Kalmbach, J. B., Park, J. W., Wiesner, M., & Chiang, J. (2020). *LSSTDESC/SLSprinkler: LSST DESC Strong Lensing Sprinkler for Simulations* (Version v1.1.0) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4480392

McKinney, W., & others. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, *445*, 51–56. https://doi.org/10.25080/majora-92bf1922-00a

Morgan, R. (2020). *Deeplenstronomy*. https://github.com/deepskies/deeplenstronomy

Park, J. W. (2021). *Jiwoncpark/baobab: v0.1.2* (Version v0.1.2) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.4476822

Simonov, K., & Net, I. (2006). *PyYaml*. https://github.com/yaml/pyyaml

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

York, D. G., Adelman, J., Anderson, Jr., John E., Anderson, S. F., Annis, J., Bahcall, N. A., Bakken, J. A., Barkhouser, R., Bastian, S., Berman, E., Boroski, W. N., Bracker, S., Briegel, C., Briggs, J. W., Brinkmann, J., Brunner, R., Burles, S., Carey, L., Carr, M. A., … SDSS Collaboration. (2000). The Sloan Digital Sky Survey: Technical Summary. *The Astronomical Journal*, *120*(3), 1579–1587. https://doi.org/10.1086/301513