LSST DESC Notes

SLRealizer: LSST Catalog-level Realization of Gravitationally-lensed Quasars

Jenny Kim,[1] Phil Marshall,[1,2] Mike Baumer,[1,2] Steve Kahn,[1,2] and Rahul Biswas[3]
(LSST Dark Energy Science Collaboration)

[1] *Kavli Institute for Particle Astrophysics & Cosmology, P. O. Box 2450, Stanford University, Stanford, CA 94305, USA*
[2] *SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA*
[3] *University of Washington*

The scale of the LSST dataset will be such that, when considering the problem of finding lensed quasars, we should anticipate extracting as much information out of the the catalogs as possible before turning to the pixel-level data. In this work we explore the use of simple, low multiplicity Gaussian mixture models for realizing gravitational lens systems in LSST catalog space, to enable both large-scale data emulation and fast initial lens-or-not classification. We demonstrate the generation of toy `Source` and `Object` catalogs, and carry out a simple machine learning classification using them.

This LSST DESC Note was generated on: March 21, 2018

1. Introduction

We anticipate being able to detect around 8000 strongly lensed quasar systems, that will provide useful information on lens mass distributions and cosmological time delay distances (Treu & Marshall 2016; LSST Science Collaboration 2017). Finding these lensed systems among the billions of objects detected and measured by LSST (LSST Science Collaboration 2009) is a key challenge. Pixel-level searches (Gavazzi et al. 2014, e.g.) may be unfeasible, unless the targets are efficiently pre-selected. We can imagine doing an initial lens classification on *catalog-level* data using machine learning techniques, in order to make this pre-selection.

Machine learning to detect gravitational lensed systems is an active area of research, with most of the focus to data being on galaxy-galaxy "Einstein Ring" systems, where morphological classification using Convolutional Neural Networks (CNN) should be effective (Petrillo et al. 2017). Early experiments show some promising results (Pourrahmani et al. 2017; Lanusse et al. 2018; Jacobs et al. 2017). The LSST catalog can be thought of as a database of pre-extracted low-level image features, which can be used as inputs to machine learning techniques. How much lensing information do these features contain? Do we need to form higher-level features (somehow) before feeding them to a machine classifier? How can we best train a machine to classify the LSST `Object`'s as lenses or nots, without requesting the images?

To answer these questions, we construct a mock LSST dataset, emulating the action of the LSST data management software stack in generating the data release catalog. Our simple emulator is called `SLRealizer`: we explain the assumptions it encodes in section 2 below, and present a small toy emulated LSST dataset made with `SLRealizer` in section 3. We then carry out a simple demonstration machine classification, training and testing the machine on a our toy `Object` table, in section 4. We draw some conclusions about future work in section 5.

2. `SLRealizer`

2.1. Model assumptions

`SLRealizer` takes as input an extragalactic catalog of mock lensed quasar systems, and emulates the LSST data release catalog measurements of those lenses. It's assumptions are that the `Object`'s and `Source`'s in the catalog tables can be simply represented as mixtures of Gaussians, and measurements of them derived from those Gaussian mixtures.

Specifically, we assume that a lensed quasar system is composed of 2 or 4 point sources (for doubles and quads respectively), plus a lens galaxy that can be represented with an elliptically-symmetric Gaussian surface brightness distribution. The seeing FWHM in each visit is used to define a circularly-symmetric Gaussian PSF.

SLRealizer models the action of the LSST DM stack deblender as returning a single Object for each galaxy-scale lensed quasar system. Its "null deblender" yields predictions of the flux, position, size and ellipticity of each measured Source calculated by realizing the surface brightness of the PSF-convolved system on a pixelated "pseudo-image" grid, and then numerically integrating this image to obtain its zeroth, first and second moments. We use the python GalSim package to carry out the pseudo-image manipulations, and choose a pixel scale of 0.2 arcseconds (the same as the LSST detectors).

Gaussian noise is added to each measurement, based on a simple error model.

The Object table is then emulated by simply averaging the available Source flux, position, size and ellipticity measurements in each filter.

2.2. Emulator Inputs

Twinkles, a simulated LSST sky with observed with six filters for ten years, used ten years of mock observation history from the LSST Project's baseline cadence simulation, minion_1016. We use this history file to define an MJD date, filter, seeing FWHM and 5-sigma limiting depth for each visit in the history, and select just the first three years of observations, which yields 263 observation epochs.

We use the OM10 mock lens catalog (Oguri & Marshall 2010) to define the properties of the lens galaxy and lensed quasar images. We selected 2234 LSST-like systems by querying with a magnitude cut of 22.5. Colors were computed using the OM10 package, which makes use of the LensPop code (Collett 2015) for estimating galaxy and quasar colors.

3. Results: Toy Emulated LSST Data

A snippet from our toy Source catalog is shown below, in Table 2. As you can see, for now we have not calculated the positions (RA and DEC). The full catalog has 473596 rows, corresponding to approximately 212 observations of 2234 lenses.

obsHistID	expMJD	filter	FWHMeff	fiveSigmaDepth
183767	59823.286523	g	1.093153	24.377204
183811	59823.307264	g	1.23193	24.289872
184047	59823.418685	z	0.908511	21.923566
185595	59825.256044	r	0.949096	24.128617
185736	59825.325979	g	1.242407	24.316968
185785	59825.352519	g	1.139232	24.436879
187493	59827.2603	z	0.807941	22.896684
187525	59827.278039	z	0.789221	22.990253
187546	59827.287816	z	0.748829	23.078407
187589	59827.307705	z	0.78313	23.152559
187603	59827.314787	z	0.737639	23.278169

Table 1. A few entries of the Twinkles mock observation history data. The full dataset can be accessed here.

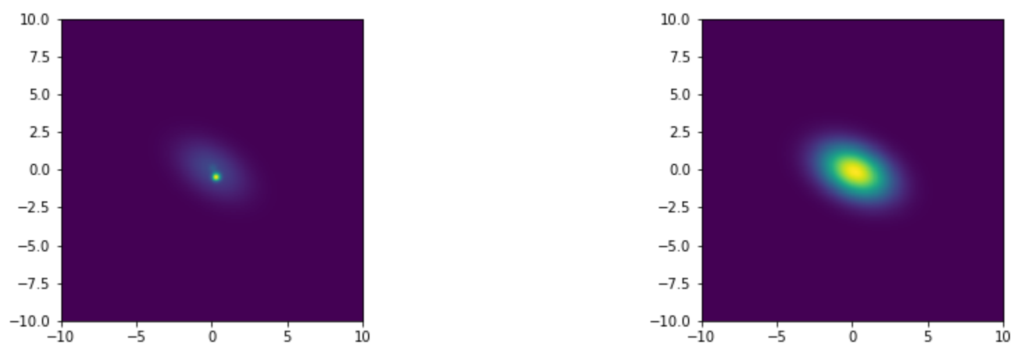

Figure 1. Example null-deblending in OM10 lens system 4898214. Image axes show offsets from the center of the lensed system in arcsec. Left: Realization of the lens system with zero-width PSF. The brightest source is the lensing galaxy, and there are two dimmer quasar images near the galaxy. There is only one quasar image that is obvious; this makes the blended object appear elliptical. Right: Realization of the lens system with realistic PSF. All the components of the system appear blended together. The color bar shows rescaled surface brightness: overall, flux is conserved between the two images.

Table 3 shows an excerpt from our toy `Object` catalog. The full `Object` catalog has 2234 rows.

4. Catalog-level Machine Learning Lens Classification

lensid	MJD	filter	RA	RA_err	DEC	DEC_err	x	x_com_err	y	y_com_err
710960	59823.286523	g	0	0	0	0	2.1350	0	1.2151	0
17432684	59823.286523	g	0	0	0	0	0.1226	0	0.7593	0
50310149	59823.286523	g	0	0	0	0	0.2527	0	0.4665	0
52812164	59823.286523	g	0	0	0	0	0.3874	0	-0.3413	0

flux	flux_err	size	size_err	e1	e2	e	phi	psf_sigma	sky
21.9127	0.03549	1.4501	0	0.2386	0.3360	0.4121	0.4766	1.093153	24.377204
18.2072	0.03549	1.1802	0	-0.0550	-0.004712	0.05525	0.04270	1.093153	24.377204
5.9831	0.03549	1.2253	0	-0.05931	0.02588	0.06471	-0.2057	1.093153	24.377204
6.2727	0.03549	1.2102	0	-0.03114	-0.05654	0.06455	0.5336	1.093153	24.377204

Table 2. A few sample entrees of the toy `Source` catalog. The full toy object catalog can be viewed here

lensid	u_flux	u_x	u_y	u_size	u_flux_err	u_x_com_err	u_y_com_err	u_size_err	u_e1
710960.0	37.0846	2.2817	1.2996	1.4151	0.2511	0.0	0.0	0.0	0.1399
17432684.0	26.7018	0.1211	0.7633	0.971	0.2516	0.0	0.0	0.0	-0.0968

g_flux	g_x	g_y	g_size	g_flux_err	g_x_com_err	g_y_com_err	g_size_err	g_e1	g_e2	g_e
19.9485	2.1555	1.2328	1.4608	0.1244	0.0	0.0	0.0	0.1967	0.2768	0.3
17.5991	0.1221	0.7518	1.2413	0.1244	0.0	0.0	0.0	-0.0532	-0.0045	0.0

r_flux	r_x	r_y	r_size	r_flux_err	r_x_com_err	r_y_com_err	r_size_err	r_e1	r_e2	r_e
31.0886	2.27	1.2928	1.2608	0.0923	0.0	0.0	0.0	0.1693	0.2395	0.29
25.2258	0.1215	0.7617	0.9958	0.0923	0.0	0.0	0.0	-0.0867	-0.0078	0.08

i_flux	i_x	i_y	i_size	i_flux_err	i_x_com_err	i_y_com_err	i_size_err	i_e1	i_e2	i_e
26.2547	2.3012	1.3075	1.2154	0.0433	0.0	0.0	0.0	0.1521	0.2146	0.263
22.747	0.1217	0.7612	1.0063	0.0433	0.0	0.0	0.0	-0.0813	-0.0071	0.081

z_flux	z_x	z_y	z_size	z_flux_err	z_x_com_err	z_y_com_err	z_size_err	z_e1	z_e2	z_e
19.7955	2.264	1.2879	1.2545	0.0322	0.0	0.0	0.0	0.1595	0.2247	0.2
18.0387	0.1216	0.7587	1.0622	0.0315	0.0	0.0	0.0	-0.0751	-0.0064	0.0

Table 3. A few sample entries of our toy `Object` catalog. The full toy object catalog can be viewed here

We now do a simple demonstration of a machine learning lens finder operating on the realized LSST tables. We first derive some higher-level features from the `Object` table measurements, and then test several different off-the-shelf classification algorithms, using the SLRealizer data in both the training and test sets. For non-lenses, we use a sample of SDSS galaxies that have the same measurements as SLRealizer predicts, and whose brightnesses approximately match the realized OM10 Object's.

4.1. Feature Selection

We expect that the quasar images will be brighter in the shorter wavelength filters, while the lens galaxies will be brighter in the longer wavelength filters. Thus, when we observe a lensed system through a u filter (the shortest wavelength filter that LSST has), the Object will appear more elongated because of the contribution from the quasar images. However, in the z band, we will see a rounder Object dominated by the contribution from the lens. When comparing the features in the u filter and the z filter, we expect to to see the biggest differences between lensed systems and SDSS galaxies.

The features that we derive from the Object table are as follows:[1]

- Changes in the first moment along the x-axis (reference to the r filter)
- Changes in the first moment along the y-axis (reference to the r filter)
- Changes in the position (reference to the r filter), ellipticities, rotation angles, fluxes, and sizes.

We can derive the same features from the catalog of SDSS galaxies: we approximate the magnitude system in SDSS to be the same in LSST, and the units are scaled to be the same. The main difference is in the sizes. SDSS's definition of size was $I_{xx} + I_{yy}$. GalSim calculates the size of a system by calculating the determinant of the second moment ($M = I_{xx}I_{yy} - I_{xy}I_{xy}$) and applying the fourth root on it ($\sqrt[4]{M}$). **In order to solve the problem by scaling the SDSS sizes, we multiplied the power of pixel-to-arcsec ratio to change the unit to arcseconds, multiplied two to convert the half size to the full size, and applied the square root to the value to get a right dimension.**[2]

Using these values, we computed various additional features. We plotted SDSS galaxies and OM10 lensed systems onto the corner plot subsection 4.1, and chose

[1] These quantities do not seem to be the ones shown in the corner plot figure.
[2] This text needs clarifying. Which quantity in the LSST Object table is being used? Which quantity in the SDSS catalog is being used? Why does the conversion from one to another involve a factor of 2?

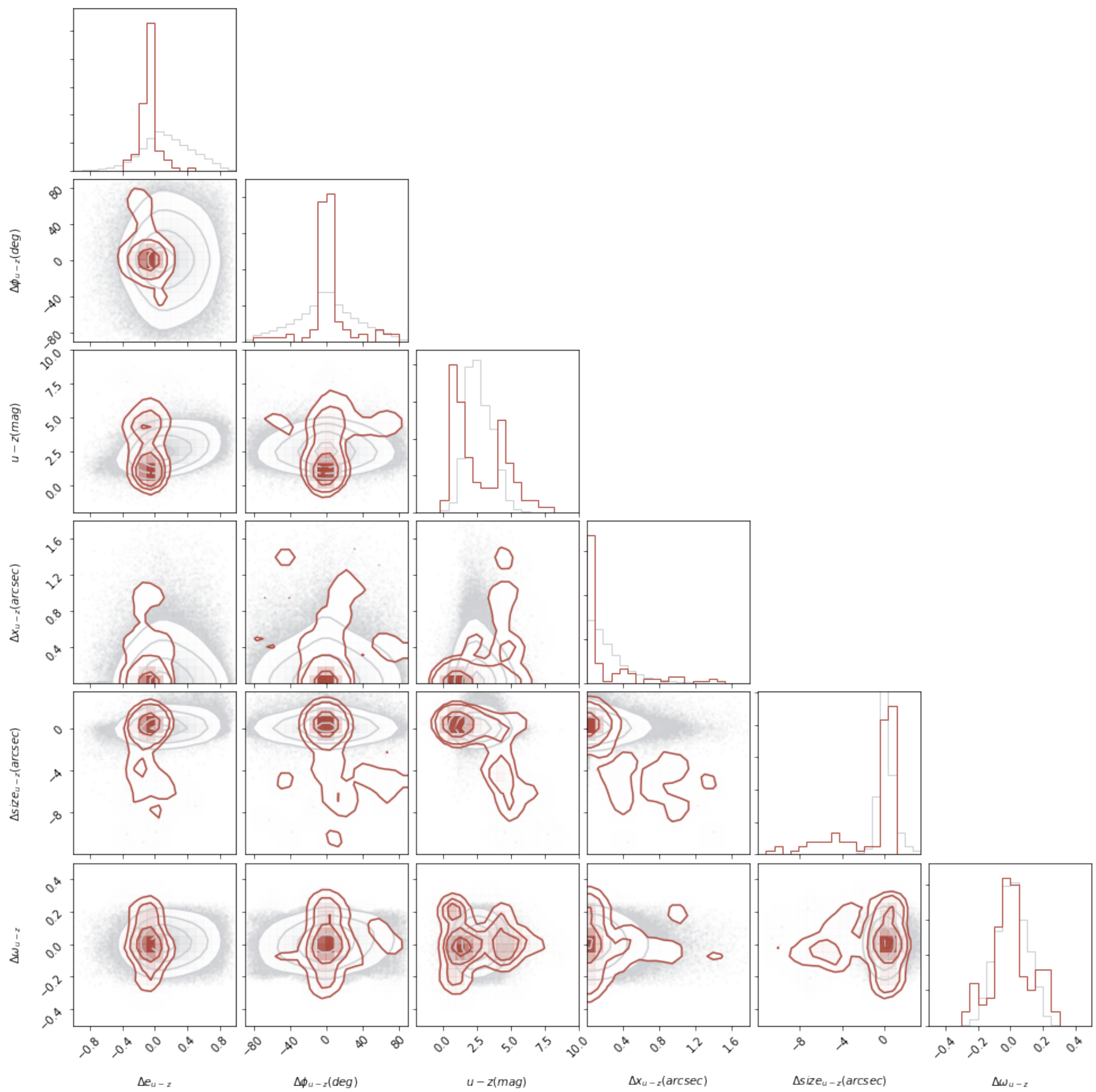

Figure 2. The cornerplot with six features.

the features that differentiated OM10 lensed systems from SDSS galaxies the most. We chose six main features that we thought would change dramatically between the filters for the realized lenses – sizes, ellipticities (e), rotation angles of galaxies (ϕ), magnitudes, positions(Δ x), and the angle between ellipticity vector and the

rotation vector ($\omega = \frac{e \cdot \phi}{|e||\phi|}$)**.**[3] The distributions of these properties, in both the OM10 lenses and SDSS galaxies, are plotted in Figure 2.[4]

In this figure, we see that the distributions of gray points (SDSS galaxies) and red points (OM10 systems) are most significantly different in the size feature.

4.2. Machine Classification

We have **2323**[5] realized OM10 lensed systems, and 16000 SDSS galaxies. In order to make a balanced data set, we randomly selected an equal number of SDSS galaxies. We shuffled the order of those two samples so that there will be a roughly same number of each OM10 and SDSS samples in both the test and the training data, and, using the `scikit-learn` `train_test_split` method, selected 75% of the data to be the training set and performed the test on the remaining 25%.

According to `scikit-learn`'s flowchart for choosing the right estimator, we identified three different algorithms for the classification. We did have more than 50 samples, we were predicting a category, we did have a labeled data, and we had less than 100K samples in a text data. This yields Linear SVC, KNeighbors Classifier, and Ensemble classifiers such as Random Forest. Figure 3 shows a receiver operating characteristic (ROC) curve comparison of these algorithms' performance.

Random forest showed the best performance among the three different algorithms. The greater the number of estimators, the better the algorithm performed. For the best algorithm, we were able to achieve a true positive rate (TPR) of 98$\%$ and a false positive rate (FPR) of 0.04$\%$.

Even though we have high accuracy, because we expect to have much more non-lensed systems than the lensed systems, we will have more contaminants in the truly-classified lensed systems than the actual lensed systems. For instance, we expect to find 10,000 times more non-lensed systems than the lensed ones. Thus, with 98% of the TPR and 0.042% of the FPR, we will have ~ 430 contaminants for every truly classified lensed

[3] This paragraph needs to be combined with the description of the itemized list above to make a simpler presentation. Why do we need to mention a reference to the r-band, when we are looking at differences between u and z?
[4] Full corner plots can be viewed in the `SLRealizer` GitHub repository notebook folder.
[5] There are 2234 `Object`'s in the toy object catalog, not 2323. What's going on here?

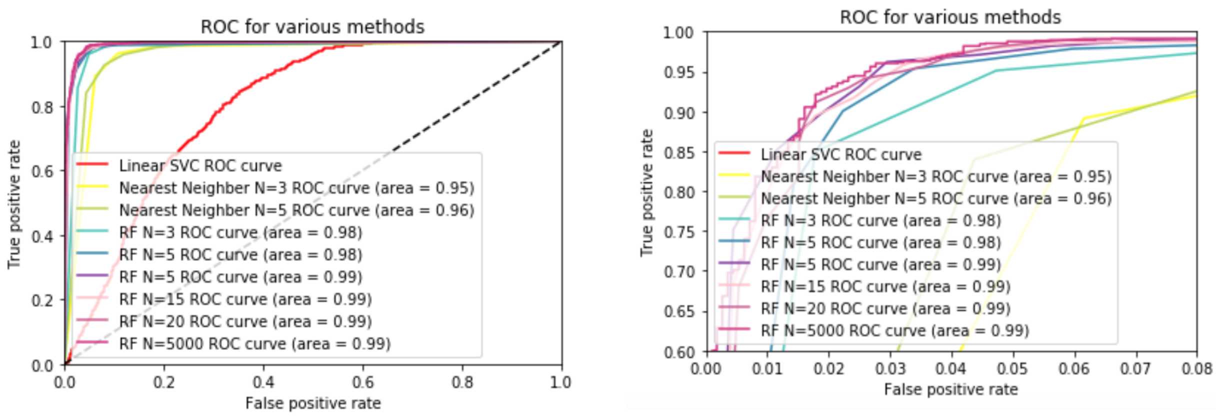

Figure 3. ROC curves for lens-or-not machine classification on `SLRealizer`-emulated LSST `Object` data. Left: full view. Right: zoomed-in view over the axes ranges FPR=[0:0.08] and TPR=[0.6:1.0].

system. This would translate to about 1 million LSST candidates for 3000 LSST lensed quasars.

We can also use the trained Random Forest classifier to quantify the importance of the input features. We find that, as seen in Figure 2, the size feature contains the most information about the `Object` classification. A bar chart of the feature importances is given in Figure 4.

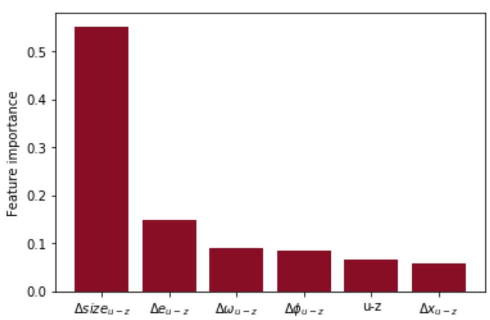

Figure 4. Feature importance calculated with the Random Forest algorithm.

5. Discussion and Conclusions

We implemented a simple model for realizing mock lensed quasar systems, and used it to emulate a small set of LSST `Object` and `Source` measurements, assuming that the LSST deblender cannot resolve the individual components of the lens systems. We then investigated some off-the-shelf machine learning classifiers with this simulated data, finding that the differences in `Object` brightness, position, ellipticity and in particular size between the u and z bands do seem to enable discrimination between lensed quasars and bright galaxies, albeit it at sub-percent purity.

We draw the following conclusions, and use them to provide pointers to further work:

- `SLRealizer` provides a framework for emulating LSST measurements of lensed quasar systems. Its initial model assumptions are simplistic but these can be refined; most importantly, the `SLRealizer` model needs to be validated against a set of LSST catalogs generated by the DM stack. In the first instance, this could be done using the OM10 lenses simulated in DC2.
- The difference in observed `Object` size between the u and z bands contains information useful for lens classification. The differences in ellipticity and centroid position are also informative, but less so.
- Predicting purity requires a realistic non-lens population as well as a realistic lens population: further work should include refinement of the non-lens population model, and investigation of specific problem cases such as physical quasar pairs, star-galaxy alignments, star clusters, and so on.
- For cosmology, gravitationally lensed systems with four images (quads) are more useful than systems with two images (doubles). We have not yet looked at the relative classification performance for quads and doubles, but we should.
- The time domain should contain much more information about the lens classification: further work should include extending `SLRealizer` to include emulation of the `DIASource` and `DIAObject` tables, and consideration of the time series of all relevant features. Since we will have measurements of the image quality and photometric depth for each visit, we will be able to (and will need to) fold this information in as well.
- Feature extraction could be improved through fitting a physical lens model to the catalog data. This could be done by realizing both a lens model and the catalog measurements onto matchin pseudo-image grids and computing a misft statistic, and then optimizing the model in the usual way. An alternative approach could be

to use deep learning networks to bypass the feature extraction step, instead just providing all available catalog data, suitably-packaged.

- If the LSST deblender does separate the components of lensed quasar systems, this should provide significantly more information about lens classification. Future work should include emulating the action of such a deblender: this will involve a more complex `Object` table, including meaningful “neighbor” linkages.

Acknowledgments

This research was partially supported by a Stanford University Physics Department summer research grant, awarded to JK. The work of PJM and SK was supported by the U.S. Department of Energy under contract number DE-AC02-76SF00515.

Author contributions are listed below.
Jenny Kim: Led algorithm and code development, wrote paper.
Phil Marshall: Initiated project, advised on motivation, model construction and testing.
Mike Baumer: Advised on LSST data characteristics, model construction and testing.
Steve Kahn: Advised on LSST data characteristics, model construction and testing.
Rahul Biswas: Advised on LSST observing cadence, catalog characteristics, error model.

References

Collett, T. E. 2015, ApJ, 811, 20

Gavazzi, R., Marshall, P. J., Treu, T., & Sonnenfeld, A. 2014, ApJ, 785, 144

Jacobs, C., Glazebrook, K., Collett, T., More, A., & McCarthy, C. 2017, MNRAS, 471, 167

Lanusse, F., Ma, Q., Li, N., et al. 2018, MNRAS, 473, 3895

LSST Science Collaboration. 2009, ArXiv e-prints, arXiv:0912.0201

—. 2017, https://github.com/LSSTDESC/Twinkles

Oguri, M., & Marshall, P. J. 2010, MNRAS, 405, 2579

Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2017, ArXiv e-prints, arXiv:1702.07675

Pourrahmani, M., Nayyeri, H., & Cooray, A. 2017, ArXiv e-prints, arXiv:1705.05857

Treu, T., & Marshall, P. J. 2016, A&A Reviews, 24, 11