The Journal of Open Source Software

UFig v1: The ultra-fast image generator

Silvan Fischbacher [1*], **Beatrice Moser** [1*], **Tomasz Kacprzak** [1,2*], **Luca Tortorelli** [1,3*], **Joerg Herbel**[1*], **Claudio Bruderer**[1*], **Uwe Schmitt** [1,4], **Alexandre Refregier** [1], **Joel Berge** [1,5], **Lukas Gamper**[1], **and Adam Amara** [1,6]

1 ETH Zurich, Institute for Particle Physics and Astrophysics, Wolfgang-Pauli-Strasse 27, 8093 Zurich, Switzerland **2** Swiss Data Science Center, Paul Scherrer Institute, Forschungsstrasse 111, 5232 Villigen, Switzerland **3** University Observatory, Faculty of Physics, Ludwig-Maximilian-Universität München, Scheinerstrasse 1, 81679 Munich, Germany **4** ETH Zurich, Scientific IT Services, Binzmühlestrasse 130, 8092 Zürich, Switzerland **5** DPHY, ONERA, Université Paris Saclay, F-92322 Châtillon, France **6** School of Mathematics and Physics, University of Surrey, Guildford, Surrey, GU2 7XH, UK * These authors contributed equally.

DOI: 10.21105/joss.08697

Software

- Review
- Repository
- Archive

Editor: Josh Borrow

Reviewers:

- @rmjarvis
- @ismael-mendoza

Submitted: 16 December 2024
Published: 16 September 2025

Summary

With the rise of simulation-based inference (SBI) methods (see e.g. Cranmer et al., 2020), simulations need to be fast as well as realistic. `UFig v1` is a public Python package that simulates astronomical images with exceptional speed, taking approximately the same time as source extraction. This makes it particularly well-suited for SBI methods where computational efficiency is crucial. To render an image, `UFig` requires a galaxy catalog, and a description of the point spread function (PSF). It can also add background noise, sample stars using the Besançon model of the Milky Way (Robin et al., 2003), and run `SExtractor` (Bertin & Arnouts, 1996) to extract sources from the rendered image. The extracted sources can be matched to the intrinsic catalog using the method described in Moser et al. (2024), flagged based on `SExtractor` output and survey masks. Emulators can be used to bypass the image simulation and extraction steps (Fischbacher et al., 2025). A first version of `UFig` was presented in Bergé et al. (2013) and the software has since been used and further developed in a variety of forward modelling applications (Bruderer et al., 2016; Chang et al., 2015; Fischbacher et al., 2025; Herbel et al., 2017; Kacprzak et al., 2020; Moser et al., 2024; Tortorelli et al., 2020, 2021).

Statement of need

`UFig` is a crucial part of the GalSBI framework. GalSBI is a galaxy population model that is used to generate mock galaxy catalogs for all kinds of cosmological applications such as photometric redshift estimation, shear and blending calibration or to forward model selection effects and measure galaxy population properties. This model is constrained by comparing simulated data to observed data. To accurately compare the simulations with the data, the simulations need to be as realistic as possible. We therefore need to include instrumental and observational effects such as the PSF and the background noise of the data, as well as the survey masks. This can be done by rendering images from the intrinsic GalSBI catalogs and extracting the sources from the images with the same method as for the data.

Since the dimensionality of the parameter space of the galaxy population model is high (around 50 parameters) and the numbers of simulations required to constrain the model is hence large, a fast image generator is crucial to make the inference feasible. `UFig`'s rendering implementation is based on a combination of pixel-based and photon-based rendering methods (see Bergé et al., 2013 for more details). This hybrid approach is one of the key factors behind `UFig`'s fast rendering speed, which is roughly comparable to the time required for source extraction from the images. For the simulations of the Hyper-Suprime-Cam (HSC, Aihara et al., 2018) deep fields presented in Moser et al. (2024) and Fischbacher et al. (2025), the rendering time is between 5 and 10 seconds for a typical image on a single CPU core.

`UFig` is optimized for wide-field galaxy surveys, where saturated bright objects or artifacts are typically masked, and detailed galaxy morphology (e.g., spiral structures) is not critical. As a result, the software includes only a simplified treatment of saturation, does not simulate artifacts such as cosmic rays, and models galaxy light profiles using Sérsic profiles.

At the same time, `UFig` is capable of accurately modelling background noise, including correlated noise from the coaddition process, and the point-spread function (PSF). This is essential for applying `UFig` to galaxy surveys, such as weak lensing, where precise shape measurements are required. This balance makes `UFig` unique in the field of image simulation compared to other software packages, such as `GalSim` (Rowe et al., 2015) and `GalSim`-based packages like `ImSim` (LSST Dark Energy Science Collaboration (DESC), 2024) and the GREAT3 simulations (Mandelbaum et al., 2015), as well as `PhoSim` (Peterson et al., 2020), `Skymaker` (Bertin, 2009), and other GREAT challenge simulations (Bridle et al., 2010; Kitching et al., 2012). To flexibly adapt to different use cases, `UFig` is based on the `ivy` workflow engine and provides plugins for the different steps of the image generation process. The full workflow can then be defined in a single configuration file, where the user can specify which plugins to use and how to configure them, e.g. by defining the PSF or background model, making the image generation process flexible and easy to use. Examples of configuration files can be found in the Advanced UFig tutorial in the `UFig` documentation.

Compared to the first version of `UFig` presented in Bergé et al. (2013), new features and improvements have been added. In Chang et al. (2015), `UFig` was used to model the transfer function of images of the Dark Energy Survey (DES, Dark Energy Survey Collaboration et al., 2016) from intrinsic galaxy catalogs to measured properties. Bruderer et al. (2016) used `UFig` to render DES-like images for which the PSF modeling and the background noise were adapted to the DES data. Furthermore, to ensure a realistic distribution of the stars in the images, a plugin to sample stars from the Besançon model of the Milky Way (Robin et al., 2003) was added (see also Bruderer, 2018 for a comprehensive overview of the `UFig` features at that time). Herbel et al. (2017) constrained a galaxy population model using `UFig`. This galaxy population model was then used to measure cosmic shear in Kacprzak et al. (2020). This effort required major improvements in the background and PSF modelling. The PSF modelling based on a convolutional neural network (CNN) was first presented in Herbel et al. (2018). Tortorelli et al. (2018) and Tortorelli et al. (2021) adapted `UFig` to render images for narrow-band filters in the context of the Physics of the Accelerating Universe (PAU, Benítez et al., 2009; Martí et al., 2014) Survey. Moser et al. (2024) used `UFig` to simulate deep fields of the HSC which required further adaptions for the PSF modelling and the matching of the extracted sources to the input catalog. Finally, Fischbacher et al. (2025) introduced emulators to bypass the image simulation and extraction steps.

A possible workflow using `UFig` could be the following:

1. Define an intrinsic galaxy catalog. An easy way to do this is to use the GalSBI galaxy population model and its catalog generator (Fischbacher et al., 2024). However, the catalog can also be generated manually without using the GalSBI model.
2. Sample stars from the Besançon model with the `UFig` plugin.
3. Add observational effects such as the PSF, background noise or saturation with the corresponding `UFig` plugins.
4. Obtain the measured catalog, either by rendering the image, running `SExtractor`, matching the extracted sources to the intrinsic catalog and flagging the sources based on the `SExtractor` output and survey masks, or by using emulators to bypass the image simulation and extraction steps.

5. Save the measured catalog and/or the rendered image.

Apart from the first step, all steps can be done with `UFig` plugins.

`UFig` images and catalogs

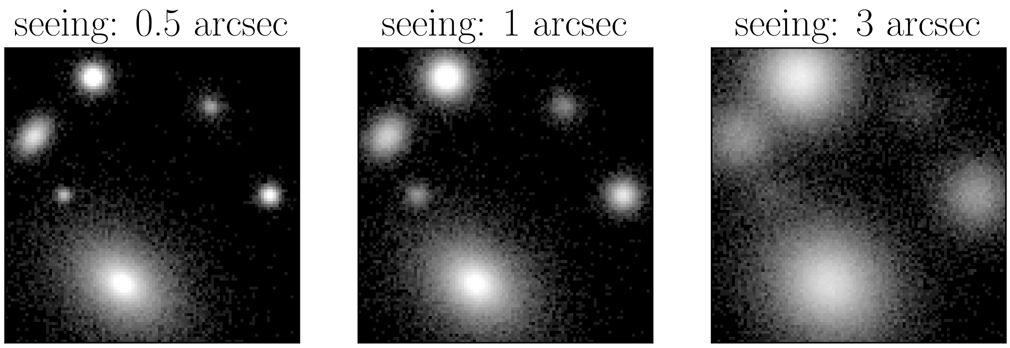

Figure 1: Rendered image with three galaxies (a large object at bottom center, an elliptical one at upper-left below a star, and a fuzzier one at upper-right) and three bright stars (round and bright at center-left, center-right, and upper-left). The PSF size varies with different seeing conditions and no background noise is added.

In the simplest case, `UFig` can render an image with a few predefined galaxies and stars without background noise. An example of such a rendered image is shown in Figure 1. From left to right, you see the same objects with different seeing conditions, which change the size of the PSF. The PSF is modelled as a mixture of one or two Moffat profiles $I_i(r)$ given by

$$I_i(r) = I_{0,i} \left(1 + \left(\frac{r}{\alpha_i}\right)^2\right)^{-\beta_i}, \tag{1}$$

with a constant base profile across the image. The ratio of $I_{0,1}$ and $I_{0,2}$ is a free parameter (in the case of a two-component Moffat) and the sum of the two profiles is determined by the number of photons of the object. The β_i parameter is free and α_i is chosen such that the half light radius of the profile is one pixel. This base profile is then distorted at each position of an object by three transformations accounting for the size of the PSF, the shape of the PSF (ellipticity, skewness, triangularity and kurtosis) and the position of the PSF (see Herbel et al., 2018 for more details). These distortions can be passed as a constant value across the image, as a map with varying values for each pixel or estimated using the CNN presented in Herbel et al. (2018). Additionally, an approximated brighter-fatter effect can be included by using a first-order linear correction that scales the PSF size and ellipticity components based on the source magnitude relative to a reference magnitude.

Figure 2 shows the same image as in Figure 1 but with added background noise. Background noise can be added as a Gaussian with constant mean and standard deviation across the image or as a map with varying mean and standard deviation for each pixel. Correlated noise is introduced by Lanczos resampling.

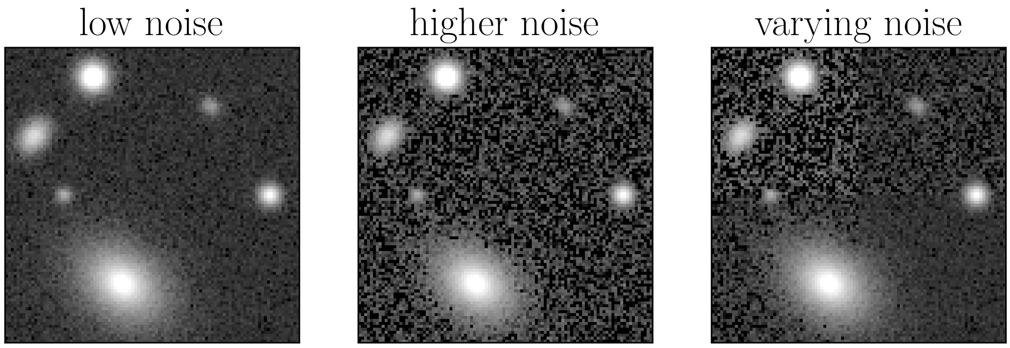

Figure 2: Rendered image with three galaxies (a large object at bottom center, an elliptical one at upper-left below a star, and a fuzzier one at upper-right) and three bright stars (round and bright at center-left, center-right, and upper-left). The PSF size is constant and the background level is varied. The left panel shows an image with low background noise, the middle panel higher noise and the right panel shows an image where each quarter has a different background noise.

Creating a more realistic galaxy catalog can be done by using the GalSBI galaxy population model and the corresponding galaxy sampling plugins of the `galsbi` Python package (Fischbacher et al., 2024). An example of rendered images for different bands with galaxies sampled from the GalSBI model presented in Fischbacher et al. (2025) is shown in Figure 3.

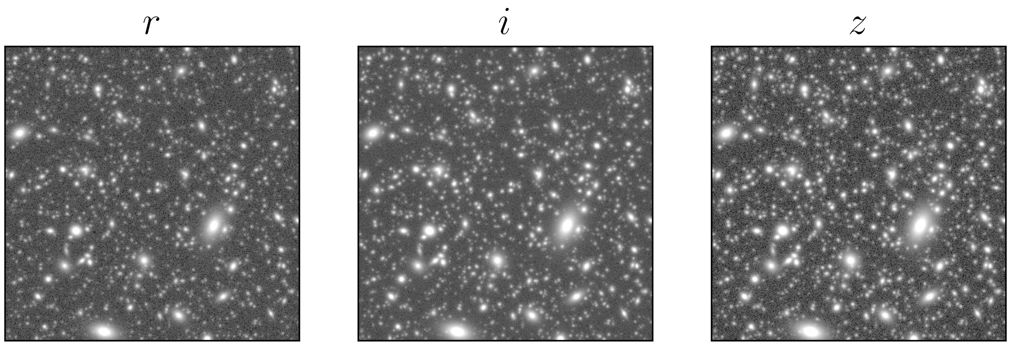

Figure 3: Rendered images with galaxies sampled from the GalSBI model for different bands. Background level and PSF estimation correspond to a typical HSC deep field image.

`UFig` also includes plugins to extract sources from the rendered images using `SExtractor` (Bertin & Arnouts, 1996), where the user can specify the `SExtractor` configuration file. This saves the detected objects in a catalog. Figure 4 shows an example of the extracted sources from a rendered image. Stars have a constant size corresponding to the PSF size and ellipticities close to zero as expected for a point source whereas galaxies have broader distributions of sizes and ellipticities.

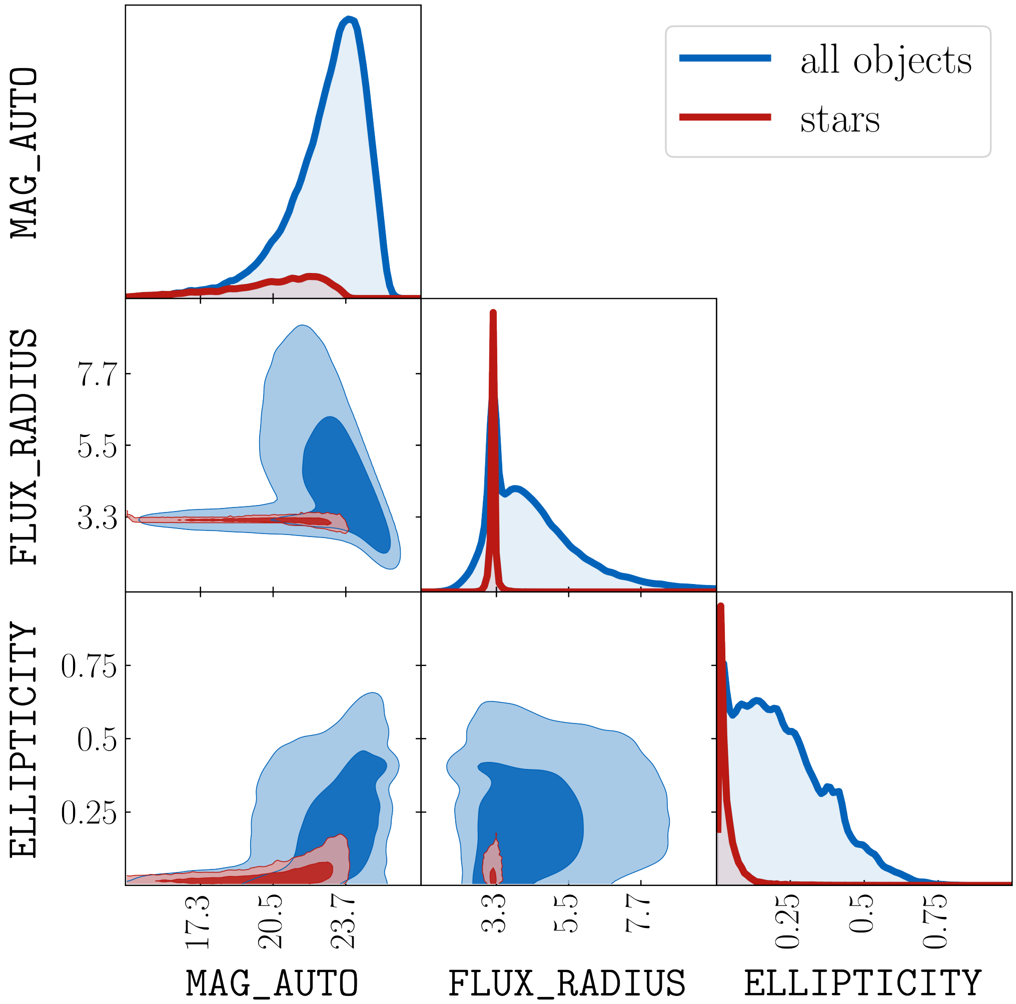

Figure 4: Catalog of sources extracted from a rendered image using `SExtractor`. The apparent magnitude (`MAG_AUTO`), angular size in pixel (`FLUX_RADIUS`) and the absolute ellipticity (`ELLIPTICITY`) are shown. All objects in the image are shown in blue, stars are shown in red.

Acknowledgments

This project was supported in part by grant 200021_143906, 200021_169130 and 200021_192243 from the Swiss National Science Foundation.

We acknowledge the use of the following software packages: NumPy (Van Der Walt et al., 2011), SciPy (Virtanen et al., 2020), Astropy (Astropy Collaboration et al., 2013), `healpy` (Zonca et al., 2019), Numba (Lam et al., 2015), `edelweiss` (Fischbacher et al., 2025), `scikit-learn` (Pedregosa et al., 2011). For the plots in this paper and the documentation, we used Matplotlib (Hunter, 2007), and `trianglechain` (Fischbacher et al., 2023; Kacprzak & Fluri, 2022). The authors with equal contribution are listed in inverse order of their main contribution.

References

Aihara, H., Arimoto, N., Armstrong, R., Arnouts, S., Bahcall, N. A., Bickerton, S., Bosch, J., Bundy, K., Capak, P. L., Chan, J. H. H., Chiba, M., Coupon, J., Egami, E., Enoki, M., Finet, F., Fujimori, H., Fujimoto, S., Furusawa, H., Furusawa, J., … Yuma, S. (2018). The Hyper Suprime-Cam SSP Survey: Overview and survey design. *Publications of the Astronomical Society of Japan*, *70*(SP1), S4. https://doi.org/10.1093/pasj/psx066

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 and Astrophysics*, *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Benítez, N., Gaztañaga, E., Miquel, R., Castander, F., Moles, M., Crocce, M., Fernández-Soto, A., Fosalba, P., Ballesteros, F., Campa, J., Cardiel-Sas, L., Castilla, J., Cristóbal-Hornillos, D., Delfino, M., Fernández, E., Fernández-Sopuerta, C., García-Bellido, J., Lobo, J. A., Martínez, V. J., … Vicente, J. F. de. (2009). Measuring baryon acoustic oscillations along the line of sight with photometric redshifts: The PAU survey. *The Astrophysical Journal*, *691*(1), 241. https://doi.org/10.1088/0004-637X/691/1/241

Bergé, J., Gamper, L., Réfrégier, A., & Amara, A. (2013). An ultra fast image generator (UFig) for wide-field astronomy. *Astronomy and Computing*, *1*, 23–32. https://doi.org/10.1016/j.ascom.2013.01.001

Bertin, E. (2009). SkyMaker: Astronomical image simulations made easy. *Memorie Della Societa Astronomica Italiana*, *80*, 422.

Bertin, E., & Arnouts, S. (1996). SExtractor: Software for source extraction. *Astronomy and Astrophysics Supplement Series*, *117*(22), 393–404. https://doi.org/10.1051/aas:1996164

Bridle, S., Balan, S. T., Bethge, M., Gentile, M., Harmeling, S., Heymans, C., Hirsch, M., Hosseini, R., Jarvis, M., Kirk, D., Kitching, T., Kuijken, K., Lewis, A., Paulin-Henriksson, S., Schölkopf, B., Velander, M., Voigt, L., Witherick, D., Amara, A., … Wittman, D. (2010). Results of the GREAT08 challenge: An image analysis competition for cosmological lensing. *Monthly Notices of the Royal Astronomical Society*, 2044–2061. https://doi.org/10.1111/j.1365-2966.2010.16598.x

Bruderer, C. (2018). *Forward modeling for gravitational lensing in cosmology* [PhD thesis, Eidgenossische Technische Hochschule, Zurich, Switzerland]. https://ui.adsabs.harvard.edu/abs/2018PhDT.......263B

Bruderer, C., Chang, C., Refregier, A., Amara, A., Berge, J., & Gamper, L. (2016). Calibrated ultra fast image simulations for the Dark Energy Survey. *The Astrophysical Journal*, *817*(1), 25. https://doi.org/10.3847/0004-637X/817/1/25

Chang, C., Busha, M. T., Wechsler, R. H., Refregier, A., Amara, A., Rykoff, E., Becker, M. R., Bruderer, C., Gamper, L., Leistedt, B., Peiris, H., Abbott, T., Abdalla, F. B., Balbinot, E., Banerji, M., Bernstein, R. A., Bertin, E., Brooks, D., Carnero, A., … Zuntz, J. (2015). Modeling the transfer function for the Dark Energy Survey. *The Astrophysical Journal*, *801*, 73. https://doi.org/10.1088/0004-637X/801/2/73

Cranmer, K., Brehmer, J., & Louppe, G. (2020). The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, *117*(48), 30055–30062. https://doi.org/10.1073/pnas.1912789117

Dark Energy Survey Collaboration, Abbott, T., Abdalla, F. B., Aleksić, J., Allam, S., Amara, A., Bacon, D., Balbinot, E., Banerji, M., Bechtol, K., Benoit-Lévy, A., Bernstein, G. M., Bertin, E., Blazek, J., Bonnett, C., Bridle, S., Brooks, D., Brunner, R. J., Buckley-Geer, E., … Zuntz, J. (2016). The Dark Energy Survey: More than dark energy - an overview. *Monthly Notices of the Royal Astronomical Society*, *460*, 1270–1299. https://doi.org/10.1093/mnras/stw641

Fischbacher, S., Kacprzak, T., Blazek, J., & Refregier, A. (2023). Redshift requirements for cosmic shear with intrinsic alignment. *Journal of Cosmology and Astroparticle Physics*, *2023*(01), 033. https://doi.org/10.1088/1475-7516/2023/01/033

Fischbacher, S., Kacprzak, T., Tortorelli, L., Moser, B., Refregier, A., Gebhardt, P., & Gruen, D. (2025). GalSBI: Phenomenological galaxy population model for cosmology using simulation-based inference. *Journal of Cosmology and Astroparticle Physics*, *2025*(06), 007. https://doi.org/10.1088/1475-7516/2025/06/007

Fischbacher, S., Moser, B., Kacprzak, T., Herbel, J., Tortorelli, L., Schmitt, U., Refregier, A., & Amara, A. (2024). *`galsbi`: A Python package for the GalSBI galaxy population model*. *arXiv:2412.08722*. https://doi.org/10.48550/arXiv.2412.08722

Herbel, J., Kacprzak, T., Amara, A., Refregier, A., Bruderer, C., & Nicola, A. (2017). The redshift distribution of cosmological samples: A forward modeling approach. *Journal of Cosmology and Astroparticle Physics*, *2017*(08), 035–035. https://doi.org/10.1088/1475-7516/2017/08/035

Herbel, J., Kacprzak, T., Amara, A., Refregier, A., & Lucchi, A. (2018). Fast point spread function modeling with deep learning. *Journal of Cosmology and Astroparticle Physics*, *2018*(07), 054–054. https://doi.org/10.1088/1475-7516/2018/07/054

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

Kacprzak, T., & Fluri, J. (2022). DeepLSS: Breaking parameter degeneracies in large-scale structure with deep-learning analysis of combined probes. *Physical Review X*, *12*, 031029. https://doi.org/10.1103/PhysRevX.12.031029

Kacprzak, T., Herbel, J., Nicola, A., Sgier, R., Tarsitano, F., Bruderer, C., Amara, A., Refregier, A., Bridle, S. L., Drlica-Wagner, A., Gruen, D., Hartley, W. G., Hoyle, B., Secco, L. F., Zuntz, J., Annis, J., Avila, S., Bertin, E., Brooks, D., … Weller, J. (2020). Monte Carlo control loops for cosmic shear cosmology with DES Year 1. *Physical Review D*, *101*(8), 082003. https://doi.org/10.1103/PhysRevD.101.082003

Kitching, T. D., Balan, S. T., Bridle, S., Cantale, N., Courbin, F., Eifler, T., Gentile, M., Gill, M. S. S., Harmeling, S., Heymans, C., Hirsch, M., Honscheid, K., Kacprzak, T., Kirkby, D., Margala, D., Massey, R. J., Melchior, P., Nurbaeva, G., Patton, K., … Zuntz, J. (2012). Image analysis for cosmology: Results from the GREAT10 galaxy challenge. *Monthly Notices of the Royal Astronomical Society*, *423*(4), 3163–3208. https://doi.org/10.1111/j.1365-2966.2012.21095.x

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6. https://doi.org/10.1145/2833157.2833162

LSST Dark Energy Science Collaboration (DESC). (2024). *imSim: GalSim-based Rubin Observatory image simulation package*. https://github.com/LSSTDESC/Imsim.

Mandelbaum, R., Rowe, B., Armstrong, R., Bard, D., Bertin, E., Bosch, J., Boutigny, D., Courbin, F., Dawson, W. A., Donnarumma, A., Fenech Conti, I., Gavazzi, R., Gentile, M., Gill, M. S. S., Hogg, D. W., Huff, E. M., Jee, M. J., Kacprzak, T., Kilbinger, M., … Zuntz, J. (2015). GREAT3 results – I. Systematic errors in shear estimation and the impact of real galaxy morphology. *Monthly Notices of the Royal Astronomical Society*, *450*(3), 2963–3007. https://doi.org/10.1093/mnras/stv781

Martí, P., Miquel, R., Castander, F. J., Gaztañaga, E., Eriksen, M., & Sánchez, C. (2014). Precise photometric redshifts with a narrow-band filter set: The PAU survey at the William Herschel Telescope. *Monthly Notices of the Royal Astronomical Society*, *442*(1), 92–109. https://doi.org/10.1093/mnras/stu801

Moser, B., Kacprzak, T., Fischbacher, S., Refregier, A., Grimm, D., & Tortorelli, L. (2024). Simulation-based inference of deep fields: Galaxy population model and redshift distributions.

Journal of Cosmology and Astroparticle Physics, *2024*(05), 049. https://doi.org/10.1088/1475-7516/2024/05/049

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

Peterson, J., Dutta, A., Jernigan, G., Remocaldo, C., & Sembroski, G. (2020). PhoSim: A tool to simulate astronomical images one photon at a time. *American Astronomical Society Meeting Abstracts #235*, *235*, 313.03.

Robin, A. C., Reylé, C., Derrière, S., & Picaud, S. (2003). A synthetic view on structure and evolution of the milky way. *Astronomy & Astrophysics*, *409*(22), 523–540. https://doi.org/10.1051/0004-6361:20031117

Rowe, B. T. P., Jarvis, M., Mandelbaum, R., Bernstein, G. M., Bosch, J., Simet, M., Meyers, J. E., Kacprzak, T., Nakajima, R., Zuntz, J., Miyatake, H., Dietrich, J. P., Armstrong, R., Melchior, P., & Gill, M. S. S. (2015). GalSim: The modular galaxy image simulation toolkit. *Astronomy and Computing*, *10*, 121–150. https://doi.org/10.1016/j.ascom.2015.02.002

Tortorelli, L., Della Bruna, L., Herbel, J., Amara, A., Refregier, A., Alarcon, A., Castander, F. J., De Vicente, J., Eriksen, M., Fernandez, E., García-Bellido, J., Gaztanaga, E., Miquel, R., Padilla, C., Sanchez, E., Serrano, S., Stothert, L., & Tonello, N. (2018). The PAU survey: A forward modeling approach for narrow-band imaging. *Journal of Cosmology and Astroparticle Physics*, *2018*(11), 035–035. https://doi.org/10.1088/1475-7516/2018/11/035

Tortorelli, L., Fagioli, M., Herbel, J., Amara, A., Kacprzak, T., & Refregier, A. (2020). Measurement of the B-band galaxy luminosity function with approximate Bayesian computation. *Journal of Cosmology and Astroparticle Physics*, *2020*(09), 048. https://doi.org/10.1088/1475-7516/2020/09/048

Tortorelli, L., Siudek, M., Moser, B., Kacprzak, T., Berner, P., Refregier, A., Amara, A., García-Bellido, J., Cabayol, L., Carretero, J., Castander, F. J., De Vicente, J., Eriksen, M., Fernandez, E., Gaztanaga, E., Hildebrandt, H., Joachimi, B., Miquel, R., Sevilla-Noarbe, I., … Wright, A. H. (2021). The PAU survey: Measurement of narrow-band galaxy properties with approximate Bayesian computation. *Journal of Cosmology and Astroparticle Physics*, *2021*(12), 013. https://doi.org/10.1088/1475-7516/2021/12/013

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Walt, S. J. van der, Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … Contributors, S. 1. 0. (2020). SciPy 1.0–fundamental algorithms for scientific computing in python. *Nature Methods*, *17*(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

Zonca, A., Singer, L. P., Lenz, D., Reinecke, M., Rosset, C., Hivon, E., & Gorski, K. M. (2019). Healpy: Equal area pixelization and spherical harmonics transforms for data on the sphere in Python. *Journal of Open Source Software*, *4*(35), 1298. https://doi.org/10.21105/joss.01298