The Journal of Open Source Software

LightCurveLynx: Fast and Nimble Time-Domain Simulation for Astronomical Surveys

Jeremy Kubica [1], **Mi Dai** [2], **Olivia Lynn** [1], **Konstantin Malanchev** [1], **Alex I. Malz** [3], **Andrew Connolly** [4], **Melissa DeLucchi** [1], **Katarzyna Kruszyńska** [5], **Wanqing Liu** [1], **Rachel Mandelbaum** [1], **Nikhil Sarin** [6,7], **and Steve Schulze** [8]

1 McWilliams Center for Cosmology and Astrophysics, Department of Physics, Carnegie Mellon University, Pittsburgh, PA 15213, USA **2** Pittsburgh Particle Physics, Astrophysics, and Cosmology Center (PITT PACC). Physics and Astronomy Department, University of Pittsburgh, Pittsburgh, PA 15260, USA **3** Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, Maryland 21218, USA **4** DiRAC Institute and the Department of Astronomy, University of Washington, 3910 15th Ave NE, Seattle, WA 98195, USA **5** Las Cumbres Observatory, 6740 Cortona Drive, Suite 102, Goleta, CA 93117, USA **6** Kavli Institute for Cosmology, University of Cambridge, Madingley Road, CB3 0HA, UK **7** Institute of Astronomy, University of Cambridge, Madingley Road, CB3 0HA, UK **8** Department of Particle Physics and Astrophysics, Weizmann Institute of Science, 234 Herzl St, 76100 Rehovot, Israel

DOI: 10.21105/joss.10629

Software

- Review
- Repository
- Archive

Editor: Warrick Ball

Reviewers:

- @temuller
- @charlotteaward

Submitted: 28 March 2026
Published: 11 November 2026

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/1538-4357/aebc6c, published in the Astrophysical Journal.

Summary

`LightCurveLynx` is a Python-based forward simulation framework for catalog-level time-varying astronomical phenomena that brings together the community's many astronomical modeling packages into a single framework. Its goal is to allow the users to mix-and-match different models, effects, and surveys within a single simulation environment. It is optimized for scalability, modularity, and extensibility. `LightCurveLynx` integrates numerous popular packages, such as SNCosmo, PZFlow, and redback, along with different sampling packages and astronomical surveys.

Statement of need

To fully realize the value of the next generation of large-scale astronomical survey data, it will be necessary to analyze this data relative to expected observations from a range of underlying population models. Such analysis allows scientists to characterize the selection functions inherent in each survey, determine which objects could be observed/classified from the data, and to fit or refine models through simulation-based inference. Further, given these survey's duration and depth, it is increasingly vital that such simulations are efficient. Making such simulations available early can even help with optimization of the observing strategy for different science cases.

The astronomy community has invested significant resources into developing powerful software packages to simulate and model a wide range of physical phenomena. To provide only a few concrete examples, SNANA (Kessler et al., 2009), SNCosmo (Barbary et al., 2025), and skysurvey (Rigault et al., 2026), provide powerful libraries for realistic simulations of supernovae; VBMicrolensing (Bozza et al., 2025), PyLIMA (Bachelet et al., 2017), BAGLE (Lu et al., 2025) provide approaches for modeling and simulating microlensing events; and redback (Sarin et al., 2024) provides a comprehensive library of models for physical phenomena. Although this proliferation of software allows researchers to explore a large number of phenomena, it introduces a level of fragmentation and complexity. The feature set, such as supported surveys and noise effects, varies from package to package.

LightCurveLynx is a package for forward simulation of time-varying astronomical phenomena that brings together this extensive ecosystem of software into a consistent framework. The user can access models from different packages, a range of effects, and a variety of surveys and instrument types. In addition, the modular API makes LightCurveLynx highly extensible. Users can easily introduce new variability models or wrap other Python packages, expanding the system's capabilities to incorporate the community's latest developments.

State of the field

As noted in the previous section, the astronomy community has developed a range of powerful tools for forward simulation of transient and variable sources. Individual packages often focus on specific types of astronomical phenomena such as supernovae or microlensing. Further each package may support a subset of the existing physical effects or survey strategy. LightCurveLynx does not aim to duplicate those efforts, but rather to augment them by providing a common framework for users to combine the existing packages.

Software design

LightCurveLynx is designed to enable users to accurately and efficiently run simulations using a range of models and simulation packages. As such, it uses several core principles: (1) provide an extensible and flexible object-oriented API, (2) provide an interface to consistently sample from complex distributions, (3) allow users to save simulation state and replay all/part of the simulation for analysis and debugging, and (4) build in vectorization and parallelization for efficient runs. This modular structure (and the general program flow) are shown in Figure 1.

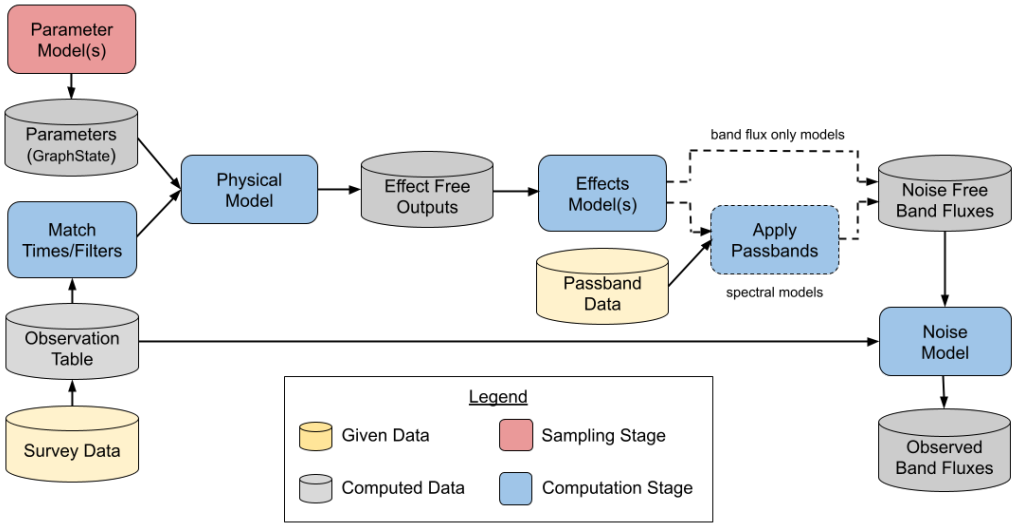


Figure 1: The basic simulation flow for LightCurveLynx.

The simulation starts by sampling the model's parameters from given statistical distributions. These parameters are combined with the information about where the survey is pointing to generate observed fluxes at either the spectral or band level. (We use "flux" as a synonym of "spectral flux density per unit frequency", while by "bandflux" we mean spectral flux density measured in the given photometric passband). LightCurveLynx uses the survey information, along with each object's position, to pre-filter evaluations to just those times when the object will be observed, saving significant computation over evaluating models at all times. The simulation then applies line of sight effects, such as dust extinction, to the fluxes. If the fluxes

were generated at the spectral level, they are integrated with the survey's filter to produce band fluxes. Finally, instrument and detector noise are sampled and added based on the survey's characteristics.

Models of physical phenomena are implemented as subclasses in a simple class hierarchy. The `BasePhysicalModel` class defines key astronomical parameters (`RA`, `Dec`, etc.) and an interface for all computations. Two subclasses provide further specializations for the different types of modeling. `SEDModel` serves as the parent class for any model that simulates output at the spectral level. `BandfluxModel` serves as the parent class for any model that simulates output at the bandflux level only. These classes handle common computations, such as correcting for redshift and extrapolating predictions. Creators of a new model can create a new subclass of one of these two types and focus on the core logic for computing fluxes.

Another advantage of this class hierarchy is that new features can be added in the parent classes and automatically applied to all models. One example of this is `LightCurveLynx`'s extrapolation functionality. Some common models, such as splines fit from data, only produce valid predictions over a finite range of times and wavelengths. By supporting bound checking and extrapolation in the parent classes, `LightCurveLynx` adds the ability to use an extrapolation function with any model.

The parameter distributions are specified using Pythonic syntax as a directed acyclic graph of parameter relations that can draw distributions from packages such as NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020), or PZFlow (Crenshaw et al., 2024, 2025). All parameter (and bookkeeping) information is handled through the `ParameterizedNode` base class and saved as a `GraphState` object, which allows the user to analyze or replay the simulations. Sampling is vectorized wherever possible and can be parallelized for efficiency. Where possible `LightCurveLynx` uses established scientific packages, such as Astropy (Astropy Collaboration et al., 2013, 2018, 2022) and SciPy, for computation.

Line of sight effects are wrapped subclasses of the `EffectModel` class and can be added to any `BasePhysicalModel` object. This separates the implementation of the model from the effects and allows a single effect type (e.g. dust extinction) to be consistently applied to any physical model. This approach ensures consistency and reduces code duplication.

Survey specific information is encapsulated in subclasses of the `ObsTable`, allowing users to simulate their models under different survey conditions. `LightCurveLynx` currently supports data from the Vera C. Rubin's LSST (Ivezić et al., 2019) in simulated and DP1/DP2 formats, data from the Zwicky Transient Facility (ZTF) surveys (Bellm et al., 2019), simulations of the Nancy Grace Roman telescope (Spergel et al., 2015), and simulations of the Argus Array (Law et al., 2022).

Parallelization is natively supported using Python's `ProcessPoolExecutor` as well as multi-machine parallelization such as Dask or Ray.

Research impact statement

The software was verified by simulating populations of Type Ia supernovae under the ZTF survey and comparing the simulated population statistics to actual data (Dai et al., 2026). The software is currently being used for multiple population studies under the proposed Rubin LSST cadence, including simulating RR Lyrae light curves and microlensing events.

Installation and usage

`LightCurveLynx` can be installed with `pip` or through `conda-forge`. See the instructions on our Read the Docs page. The project's tutorial notebooks documentation page provides a variety of usage examples and technical deep dives.

AI usage disclosure

All top level design, including control flow and class structure, was designed by humans. GitHub Copilot (using Claude Sonnet 4, GPT-5.3-Codex, and auto settings) was used during the development of the software for online suggestions and small changes. All code changes, regardless of author, were reviewed by humans. No generative AI tools were used in the writing of this manuscript.

Acknowledgements

This project is supported by Schmidt Sciences. The authors acknowledge support from the DiRAC Institute in the Department of Astronomy at the University of Washington. The DiRAC Institute is supported through generous gifts from the Charles and Lisa Simonyi Fund for Arts and Sciences, and the Washington Research Foundation.

References

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., N"othe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., … Astropy Project Contributors. (2022). The Astropy project: Sustaining and growing a community-oriented open-source project and the latest major release (v5.0) of the core package. *The Astrophysical Journal*, *935*(2), 167. https://doi.org/10.3847/1538-4357/ac7c74

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., … Astropy Contributors. (2018). The Astropy project: Building an open-science project and status of the v2.0 core package. *The Astronomical Journal*, *156*(3), 123. https://doi.org/10.3847/1538-3881/aabc4f

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

Bachelet, E., Norbury, M., Bozza, V., & Street, R. (2017). pyLIMA: An open-source package for microlensing modeling. I. Presentation of the software and analysis of single-lens models. *The Astronomical Journal*, *154*(5), 203. https://doi.org/10.3847/1538-3881/aa911c

Barbary, K., Bailey, S., Barentsen, G., Barclay, T., Biswas, R., Boone, K., Craig, M., Feindt, U., Friesen, B., Goldstein, D., Jha, S. W., Jones, D. O., Mondon, F., Papadogiannakis, S., Perrefort, D., Pierel, J., Rodney, S., Rose, B., Saunders, C., … Wood-Vasey, M. (2025). *SNCosmo* (Version v2.12.1). Zenodo. https://doi.org/10.5281/zenodo.15019859

Bellm, E. C., Kulkarni, S. R., Graham, M. J., Dekany, R., Smith, R. M., Riddle, R., Masci, F. J., Patterson, B., Rebbapragada, U., Walters, R., Ward, C., Ye, Q.-Z., Zolkower, J., & others. (2019). The Zwicky Transient Facility: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific*, *131*(995), 018002. https://doi.org/10.1088/1538-3873/aaecbe

Bozza, V., Saggese, V., Covone, G., Rota, P., & Zhang, J. (2025). VBMicroLensing: Three algorithms for multiple lensing with contour integration. *Astronomy & Astrophysics*, *694*, A219. https://doi.org/10.1051/0004-6361/202452648

Crenshaw, J. F., Kalmbach, J. B., Gagliano, A., Yan, Z., Connolly, A. J., Malz, A. I., Schmidt,

S. J., & The LSST Dark Energy Science Collaboration. (2024). Probabilistic forward modeling of galaxy catalogs with normalizing flows. *The Astronomical Journal*, *168*(2), 80. https://doi.org/10.3847/1538-3881/ad54bf

Crenshaw, J. F., Yan, Z., & Doster, V. (2025). *jfcrenshaw/pzflow: v3.3.0* (Version v3.3.0). Zenodo. https://doi.org/10.5281/zenodo.17809535

Dai, M., Kubica, J., Malanchev, K., Malz, A. I., Lynn, O., Connolly, A., Mandelbaum, R., & Wood-Vasey, W. M. (2026). *LightCurveLynx: Forward modeling of time-domain surveys with application to ZTF SN ia DR2*. https://arxiv.org/abs/2604.07134

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., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Ivezić, Ž., Kahn, S. M., Tyson, J. A., Abel, B., Acosta, E., Allsman, R., Alonso, D., Anderson, K., Andrew, J., Angel, J. R. P., & others. (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

Kessler, R., Bernstein, J. P., Cinabro, D., Dilday, B., Frieman, J. A., Jha, S., Kuhlmann, S., Miknaitis, G., Sako, M., Taylor, M., & Vanderplas, J. (2009). SNANA: A public software package for supernova analysis. *Publications of the Astronomical Society of the Pacific*, *121*(883), 1028. https://doi.org/10.1086/605984

Law, N. M., Corbett, H., Galliher, N. W., Gonzalez, R., Vasquez, A., Walters, G., Machia, L., Ratzloff, J., Ackley, K., Bizon, C., Clemens, C., Cox, S., Eikenberry, S., Howard, W. S., Glazier, A., Mann, A. W., Quimby, R., Reichart, D., & Trilling, D. (2022). Low-cost access to the deep, high-cadence sky: The Argus Optical Array. *Publications of the Astronomical Society of the Pacific*, *134*(1033), 035003. https://doi.org/10.1088/1538-3873/ac4811

Lu, J. R., Medford, M., Lam, C. Y., Bhadra, T. D., Huston, M. J., Abrams, N. S., Broadberry, E., Chen, J., Terry, S. K., Arredondo, N., & Scharf, A. (2025). *The BAGLE Python package for Bayesian analysis of gravitational lensing events*. https://doi.org/10.48550/arXiv.2512.03364

Rigault, M., Ginolin, M., Dellazzeri, L., Popovic, B., Hjortlund, J. O., Gilles-Lordet, A., Conseil, S., Coughlin, M., Ruppin, F., Smith, M., Townsend, A., Trigui, A., Barjou-Delayre, C., Kebadian, R., & Nordin, J. (2026). *skysurvey: A pure Python package to simulate the transient sky*. https://arxiv.org/abs/2605.25840

Sarin, N., Hübner, M., Omand, C. M. B., Setzer, C. N., Schulze, S., Adhikari, N., Sagués-Carracedo, A., Galaudage, S., Wallace, W. F., Lamb, G. P., & Lin, E.-T. (2024). redback: A Bayesian inference software package for electromagnetic transients. *Monthly Notices of the Royal Astronomical Society*, *531*(1), 1203–1227. https://doi.org/10.1093/mnras/stae1238

Spergel, D., Gehrels, N., Baltay, C., Bennett, D., & others. (2015). Wide-field InfraRed survey telescope (WFIRST) final report. *arXiv Preprint*. https://doi.org/10.48550/arXiv.1208.4012

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