

JAX-bandflux: differentiable supernovae SALT modelling for cosmological analysis on GPUs

Samuel Alan Kossoff Leeney ^{1,2}

¹ Astrophysics Group, Cavendish Laboratory, J. J. Thomson Avenue, Cambridge CB3 0HE, UK ² Kavli Institute for Cosmology, Madingley Road, Cambridge CB3 0HA, UK

DOI: [10.21105/joss.09441](https://doi.org/10.21105/joss.09441)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Josh Borrow](#)  

Reviewers:

- [@Samreay](#)
- [@wbenoit26](#)

Submitted: 10 April 2025

Published: 17 February 2026

License

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

Summary

JAX-bandflux is a JAX ([Bradbury et al., 2018](#)) implementation of critical supernova modelling functionality for cosmological analysis. The codebase implements key components of the established library SNCosmo ([Barbary et al., 2016](#)) in a differentiable framework, offering efficient parallelisation and gradient-based optimisation capabilities through GPU acceleration. The package facilitates differentiable computation of supernova light curve measurements, supporting the inference of SALT ([Kenworthy et al., 2021](#); [Pierel et al., 2022](#)) parameters necessary for cosmological analysis. When parameters are evaluated in batches on GPU (as is standard in JAX-based samplers), the bandflux kernels deliver order-of-magnitude speedups ($\approx 100\times$ per parameter set compared with SNCosmo) while matching fluxes to within 0.001% ([Leeney et al., 2025](#)), enabling fast evaluation of large, fused likelihoods.

Statement of need

Accurate estimation of supernova flux is essential in cosmological studies. These measurements are fundamental to the calibration of standard candles and subsequent distance determinations, which are used to answer cosmological questions (e.g., the rate of expansion of the Universe). Existing packages such as SNCosmo ([Barbary et al., 2016](#)) are widely used for analysing supernova data. However, traditional implementations are not designed to run on GPUs and they lack differentiability. A differentiable approach enables efficient gradient propagation during parameter optimisation. Additionally, JAX's compilation model supports large-scale parallel computations on modern hardware such as GPUs. This JAX implementation addresses these requirements by providing differentiable, parallelisable routines for SALT parameter extraction.

Implementation

The package is structured into several modules and example scripts that demonstrate various aspects of the supernova modelling workflow. Two primary example scripts, `fmin_bfgs.py` and `ns.py`, illustrate optimisation via L-BFGS-B and nested sampling respectively. These scripts utilise core routines from the JAX modules, following a structure similar to SNCosmo while enabling differentiability and GPU acceleration. The central computation is contained in the file `salt3.py`, which implements the SALT3 model.

The SALT model is of the form:

$$F(p, \lambda) = x_0 [M_0(p, \lambda) + x_1 M_1(p, \lambda) + \dots] \times \exp[c \times CL(\lambda)]$$

where free parameters are: x_0 , x_1 , t_0 , and c . Model surface parameters are: $M_0(p, \lambda)$ and

$M_1(p, \lambda)$ are functions that describe the underlying flux surfaces, and p is a function of redshift and $t - 2$.

The computation of the bandflux is achieved by integrating the model flux across the applied bandpass filters. Combining multiple bands, the bandflux is defined as:

$$\text{bandflux} = \int_{\lambda_{\min}}^{\lambda_{\max}} F(\lambda) \cdot T(\lambda) \cdot \frac{\lambda}{hc} d\lambda$$

Here, $T(\lambda)$ is the transmission function specific to the bandpass filter used; h and c are the Planck constant and the speed of light respectively.

Within `salt3.py`, the implementation computes the rest-frame model flux by combining the base spectral surface $M_0(p, \lambda)$ with the stretch-modulated variation $M_1(p, \lambda)$, each scaled by their respective SALT parameters. These operations utilise JAX's vectorised array manipulations, which are JIT-compiled for efficient, parallel execution on GPUs. The resulting flux is computed in a fully differentiable manner. The computed flux is then multiplied by the instrument's transmission function $T(\lambda)$ and by the wavelength factor $\lambda/(hc)$, followed by trapezoidal integration along the wavelength dimension using JAX's numerical integration capabilities. These operations are also JIT-compiled and can be parallelised across multiple data instances via `vmap`.

The package includes comprehensive bandpass filter handling through the `bandpasses.py` module, which provides a `Bandpass` class to represent filter transmission functions. A set of commonly used astronomical filters is pre-integrated into the system, whilst additional custom bandpasses can be registered as needed through functions such as `register_bandpass` and `load_bandpass_from_file`. The system also facilitates the creation of bandpass objects from the Spanish Virtual Observatory (SVO) filter service. For data handling, the `data.py` module offers utilities for loading and processing supernova observations from various formats, including functions to handle redshift data and prepare it for model fitting. The package currently supports both SALT3 and SALT3-NIR models through dedicated interpolation routines found in `salt3.py`.

This architecture allows gradient propagation through the entire analysis pipeline, enabling techniques that benefit from JAX's differentiable, parallelisable programming paradigm. The implementation maintains functional parity with `SNCosmo` whilst providing an enhanced computational efficiency and scalability for contemporary cosmological analyses.

References

- Barbary, K., Barclay, T., Biswas, R., Craig, M., Feindt, U., Friesen, B., Goldstein, D., Jha, S., Rodney, S., Sofiatti, C., Thomas, R. C., & Wood-Vasey, M. (2016). *SNCosmo: Python library for supernova cosmology*. Astrophysics Source Code Library, record ascl:1611.017.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., & Zhang, Q. (2018). *JAX: Composable transformations of Python+NumPy programs* (Version 0.3.13). <http://github.com/google/jax>
- Kenworthy, W. D., Jones, D. O., Dai, M., Kessler, R., Scolnic, D., Brout, D., Siebert, M. R., Pierel, J. D. R., Dettman, K. G., Dimitriadis, G., Foley, R. J., Jha, S. W., Pan, Y.-C., Riess, A., Rodney, S., & Rojas-Bravo, C. (2021). SALT3: An improved Type Ia supernova model for measuring cosmic distances. *923*(2), 265. <https://doi.org/10.3847/1538-4357/ac30d8>
- Leeney, S. A. K., Handley, W. J., Bevins, H. T. J., & de Lera Acedo, E. (2025). Bayesian anomaly detection for Ia cosmology: Automating SALT3 data curation. *arXiv Preprint arXiv:2509.13394*. <https://doi.org/10.48550/arXiv.2509.13394>

Pierel, J. D. R., Jones, D. O., Kenworthy, W. D., Dai, M., Kessler, R., Ashall, C., Do, A., Peterson, E. R., Shappee, B. J., Siebert, M. R., Barna, T., Brink, T. G., Burke, J., Calamida, A., Camacho-Neves, Y., de Jaeger, T., Filippenko, A. V., Foley, R. J., Galbany, L., ... Wang, Q. (2022). SALT3-NIR: Taking the open-source Type Ia supernova model to longer wavelengths for next-generation cosmological measurements. *939*(1), 11. <https://doi.org/10.3847/1538-4357/ac93f9>