The Journal of Open Source Software

JaxNRSur - Numerical Relativity Surrogates in JAX

Kaze W. K. Wong [1], **Maximiliano Isi** [2], **Ethan Payne** [3], **and Vijay Varma** [4]

1 Data Science and AI institute, Johns Hopkins University, Baltimore, MD 21218, US **2** Center for Computational Astrophysics, Flatiron Institute, New York, NY 10010, US **3** Department of Physics, California Institute of Technology, Pasadena, California 91125, USA **4** Department of Mathematics, Center for Scientific Computing and Data Science Research, University of Massachusetts, Dartmouth, MA 02747, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

The detection and analysis of gravitational waves relies on accurate and efficient modeling of waveforms from compact binary coalescences. Numerical relativity (NR) surrogate models are a family of data-driven models that are directly trained on NR simulations to provide fast and accurate approximations of these waveforms without many assumptions about the form of the model.

`JaxNRSur` is a Python library that implements NR waveform models in `JAX` (Bradbury et al., 2018), a high-performance numerical computing library that supports automatic differentiation, just-in-time (JIT) compilation, and hardware acceleration (GPU/TPU). By using JAX, `JaxNRSur` provides the following key features:

Key features

- **Accelerator support:** The same code runs efficiently on CPUs, GPUs, and TPUs, with no user intervention required. This allows for fast waveform evaluations in large-scale analyses.
- **Simple vectorization:** JAX's vectorization capabilities allow users to efficiently evaluate waveforms for multiple source parameters or times in parallel.
- **Automatic differentiation:** Compute gradients of waveforms with respect to time, source parameters, or model parameters (i.e., the data in the surrogate model) automatically, enabling advanced inference techniques such as gradient-based sampling or optimization.
- **Unified interface:** A consistent and user-friendly interface for accessing different NR surrogate models, making it easy to switch between models.

`JaxNRSur` currently includes implementations of several NR surrogate models, including:

- `NRHybSur3dq8`: A hybridized surrogate model for non-precessing binary black hole mergers (Varma, Field, Scheel, Blackman, Kidder, et al., 2019), valid for mass ratios up to 8.
- `NRSur7dq4`: A surrogate model for precessing binary black hole mergers (Varma, Field, Scheel, Blackman, Gerosa, et al., 2019), valid for mass ratios up to 4.

We have validated our implementation against the original implementations of these models in `gwsurrogate` (Field et al., 2025) to ensure correctness and consistency. The package also includes utilities for downloading and caching the required data files from Zenodo (European Organization For Nuclear Research & OpenAIRE, 2013), making it easy to set up and use.

Statement of need

The original implementations of NR surrogate models in `gwsurrogate` (Field et al., 2025) are mainly implemented in `numpy` and `scipy` with additional binding from `c`, whcih gives the package a reasonable performance. However, this implementation does not take advantage of more modern computing paradigms such as the use of accelerators (GPUs/TPUs) and automatic differentiation, which have been widely adopted in the machine learning and high performance computing communities. This presents a few challenges when using and developing NR surrogate models, and this package aims to address these challenges:

A main challenge in using NR surrogate models for downstream tasks such as parameter estimation is the relatively high computational cost compared to other waveform approximants such as IMRPhenomXPHM (Pratten & others, 2021). NR surrogate models involve a lot of dense matrix multiplications and other linear algebra operations, which can be efficiently parallelized on accelerators such as GPUs and TPUs. This `JAX` implementation allows users to leverage accelerators to speed up waveform evaluations. For example, in a publicly available benchmark on an NVIDIA T4 GPU in a Google Colab environment, the `NRSur7dq4` waveform can be evaluated for 100 parameters in approximately 65 ms, which is significantly faster than the original implementation in `gwsurrogate`[1]. This speedup is further compounded by the fact that `JAX` supports vectorization of functions that is accelerator-aware, providing the performance the NR surrogate model family needs for downstream tasks.

Another feature this package provides is the ability to compute gradients of the waveform with respect to the source parameters through automatic differentiation. This is useful for tasks such as template bank generation (Coogan et al., 2022) or gradient-based Markov chain Monte Carlo (MCMC) sampling (Betancourt, 2017) (Wong et al., 2023) (Cabezas et al., 2024). Parallel to the improvement in waveform evaluation throughput, being able to leverage gradient information often speeds up the convergence of these downstream tasks, which further improves the performance of a complete pipeline when compared to non-gradient-based counterparts. On top of performance gains, higher order derivatives such as Hessians can be used as a natural metric for understanding uncertainties in the pipeline and help in sensitivity analysis.

Finally, this package also supports differentiating against the model parameters in a fashion similar to how neural networks are trained. This allows users to fine-tune the model parameters to better fit their data (Lam et al., 2024). Another research avenue our package opens is

Acknowledgements

We thank Tousif Islam, Will Farr, Keefe Mitman, and Colm Talbot for their contributions to the JaxNRSur codebase and discussions that improved the package. V.V.~acknowledges support from NSF Grant No. PHY-2309301 and UMass Dartmouth's Marine and Undersea Technology (MUST) Research Program funded by the Office of Naval Research (ONR) under Grant No. N00014-23-1–2141.

References

Betancourt, M. (2017). A conceptual introduction to hamiltonian monte carlo. *arXiv: Methodology*. https://api.semanticscholar.org/CorpusID:88514713

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/jax-ml/jax

[1] Benchmarking results on Google Colab may vary depending on the specific hardware and software environment that are provisioned.

Cabezas, A., Corenflos, A., Lao, J., & Louf, R. (2024). *BlackJAX: Composable Bayesian inference in JAX*. https://arxiv.org/abs/2402.10797

Coogan, A., Edwards, T. D. P., Chia, H. S., George, R. N., Freese, K., Messick, C., Setzer, C. N., Weniger, C., & Zimmerman, A. (2022). Efficient gravitational wave template bank generation with differentiable waveforms. *Phys. Rev. D*, *106*(12), 122001. https://doi.org/10.1103/PhysRevD.106.122001

European Organization For Nuclear Research, & OpenAIRE. (2013). *Zenodo*. CERN. https://doi.org/10.25495/7GXK-RD71

Field, S. E., Varma, V., Blackman, J., Gadre, B., Galley, C. R., Islam, T., Mitman, K., Pürrer, M., Ravichandran, A., Scheel, M. A., Stein, L. C., & Yoo, J. (2025). GWSurrogate: A Python package for gravitational wave surrogate models. *J. Open Source Softw.*, *10*(107), 7073. https://doi.org/10.21105/joss.07073

Lam, K. K. H., Wong, K. W. K., & Edwards, T. D. P. (2024). Recalibrating a gravitational wave phenomenological waveform model. *Phys. Rev. D*, *109*(12), 124009. https://doi.org/10.1103/PhysRevD.109.124009

Pratten, G., & others. (2021). Computationally efficient models for the dominant and subdominant harmonic modes of precessing binary black holes. *Phys. Rev. D*, *103*(10), 104056. https://doi.org/10.1103/PhysRevD.103.104056

Varma, V., Field, S. E., Scheel, M. A., Blackman, J., Gerosa, D., Stein, L. C., Kidder, L. E., & Pfeiffer, H. P. (2019). Surrogate models for precessing binary black hole simulations with unequal masses. *Phys. Rev. Research*, *1*, 033015. https://doi.org/10.1103/PhysRevResearch.1.033015

Varma, V., Field, S. E., Scheel, M. A., Blackman, J., Kidder, L. E., & Pfeiffer, H. P. (2019). Surrogate model of hybridized numerical relativity binary black hole waveforms. *Phys. Rev. D*, *99*(6), 064045. https://doi.org/10.1103/PhysRevD.99.064045

Wong, K. W. k., Gabrié, M., & Foreman-Mackey, D. (2023). flowMC: Normalizing flow enhanced sampling package for probabilistic inference in JAX. *J. Open Source Softw.*, *8*(83), 5021. https://doi.org/10.21105/joss.05021