

# Response letter to reviewer comments on "geospaNN: A Python package for geospatial neural networks"

## General Response

We thank the editor and the reviewers for their positive feedback and thoughtful comments which have helped to improve the manuscript. We have revised the manuscript thoroughly. Please find the point-point responses below.

## Reviewer 1

**Reviewer summary:** Overall the package is a valuable and timely contribution, and the software itself is well implemented, well documented, and supported by a clear website and examples. The manuscript, however, would benefit from revisions to better align with typical JOSS expectations. In its current form, the manuscript reads like a hybrid between a methods paper and a JOSS software article. My comments below focus on aligning the paper with JOSS requirements more closely: a clear software overview, a focused statement of need, an accurate comparison to existing tools, and a software-centered description of functionality.

**Author response:** Thank you for the very positive feedback and the insightful suggestions, which helped us improve the manuscript. In this revision, we have addressed the reviewers' comments and better aligned the manuscript with JOSS expectations. We reduced redundant material in the methodology section and placed greater emphasis on the software by highlighting its functionality, significance, and potential future directions.

### 1: Reviewer comment:

The Summary section currently emphasizes background material (SPLMM, GPs, kriging, NN context) and only introduces the software after substantial methodological discussion. JOSS looks for a short, high-level summary that tells a general reader what the software does and why it matters. The current Summary leans too much on technical background before getting to the point. Putting the main capabilities of geospaNN up front, and trimming the technical detail, would make this section clearer and much more accessible to non-specialists. Some of the broad overview now in the Statement of Need could be moved here in a more concise form.

### Response:

Thanks for the comment. We have revised the Summary section with less background and more high-level descriptions. We also moved some broad overview from Statement of Need to here and leave technical details to the later sections:

Geostatistical models are essential for analyzing data with spatial structure across the geosciences, such as climate, ecology, and environmental science. At the same time,

modern machine learning methods, especially neural networks (NNs), offer powerful tools for capturing complex, nonlinear relationships. Our package geospaNN bridges these two worlds by providing a Python library that integrates NN modeling with scalable spatial statistics. The software enables users to fit flexible spatial regression models, estimate complex mean structures, and generate Gaussian process (GP)-based spatial predictions with uncertainty quantification. Built on the PyG library designed for efficient graph neural network (GNN) training, geospaNN supports efficient computation on large, irregular spatial datasets. To handle modern geospatial data sizes, geospaNN incorporates the Nearest Neighbor Gaussian Process (NNGP) approximation [@datta2016nearest] for fast covariance computations.

## 2: Reviewer comment:

The Statement of Need reads mainly as a description of the package rather than discussion on specific gap the software fills. JOSS expects this section to clearly explain the problem the software solves, who it is for, and why existing tools aren't enough. The current Statement of Need doesn't really spell out what gap in the Python spatial/GP ecosystem geospaNN is filling. Rewriting this section to directly state the need, and briefly explain why current libraries don't address it, would bring it in line with what JOSS is looking for.

### Response:

Thank you so much for the comment. We have revised the Statement of Need section thoroughly. We now point out the gaps in existing tools from different perspectives and describe when and how *geospaNN* can help the users:

Researchers in geoscience and related fields frequently need to model relationships among spatially distributed variables and generate reliable spatial predictions. Although many Python machine learning libraries can fit complex nonlinear regression models, they typically ignore spatial correlation, leading to biased estimates and misleading inference when applied to geospatial data. Existing spatial modeling tools in Python provide only partial solutions: some rely on complex neural architectures that sacrifice scientific interpretability, while others use full GP models whose computational demands make them impractical for large datasets.

geospaNN addresses these limitations by providing a spatial regression framework that combines the flexibility of NNs with the interpretability and statistical rigor of geo-statistical models. It is designed for geoscientists, environmental researchers, and machine learning practitioners who need scalable and principled spatial modeling tools in Python. geospaNN enables geometry-aware covariance estimation and spatial prediction at scales—tens of thousands of locations—that are feasible on a personal laptop. This makes advanced spatial analysis accessible to individual researchers without specialized computing infrastructure.

The NNGP implementation within geospaNN also fills a notable gap in the Python ecosystem. While widely used R packages such as spNNGP (Finley et al., 2019) and BRISC (Saha and Datta, 2018) provide efficient NNGP-based spatial models, no comparable Python implementation currently exists. geospaNN therefore offers the first Python-based

pathway for NNGP modeling in geospatial applications, meeting the growing demand for large-scale spatial analysis.

**3: Reviewer comment:**

The State of the Field discusses TorchGeo, geodl, and general GNN applications, but doesn't connect geospaNN to the main Python tools people already use for spatial modeling and Gaussian processes (like GPyTorch, PyKrig, or scikit-gstat). JOSS requires placing the software in its real ecosystem. Adding a short comparison to these commonly used packages (in terms of functionality) would make geospaNN's unique role much clearer.

**Response:**

Sorry that we missed the important comparison with those existing GP-based tools. We now added a new paragraph comparing GPyTorch and PyKrig in this section, and point out their difference with geospaNN:

GP-based tools provide another major category of spatial modeling software. PyKrig (Murphy, 2014) offers classical kriging prediction but is limited to predefined mean functions and lacks scalable covariance computation for large datasets. GPyTorch (Gardner et al., 2018) supports flexible mean modeling and GP inference within a mixed-model framework, but its functionality is highly modular and requires substantial custom implementation, making it difficult for general users to apply. Moreover, its covariance approximations are not explicitly designed to exploit spatial geometry, which can reduce efficiency and accuracy compared with approaches tailored to geostatistical structure.

There are also some slight modification on the original paragraphs. They are not attached here but can be found in the updated .md file.

**4: Reviewer comment:**

The software description currently spends a lot of time re-explaining the NN-GLS theory from a methods aspect, instead of giving a clear picture of how the package actually works. JOSS expects this section to be about the software itself: what modules it has, how the workflow looks, and how users interact with it. Shifting the focus toward the package structure and practical API, and cutting back the heavy theory, would make this section clearer.

**Response:**

Thank you so much for the suggestion. In the revised manuscript, we heavily cut the NN-GLS theory and added a section "Core features of geospaNN" that briefly goes through the workflow of geospaNN with slight explanation and code example. We hope it can give readers a better sense of the modules and functions in the package.

**5: Reviewer comment:**

The Discussion highlights methodological points and conceptual future directions. Right now, the Discussion doesn't really talk about the software itself; what it can do, where it struggles, or what features are coming next. JOSS usually expects a short wrap-up that highlights current capabilities, known limitations, and planned improvements. Shifting the Discussion toward these practical software points would make it more useful for readers.

### **Response:**

Thank you so much for the suggestion. We rephrased the Discussion section to focus more on geospaNN itself. The first paragraph below summarizes the current capability of the software. The next two paragraphs talk about the limitations and future improvement. The section now provide a more comprehensive overview and should be more helpful to the readers.

The geospaNN package provides a machine learning toolkit for geostatistical analysis. Built on an efficient implementation of the NN–GLS approach proposed in Zhan and Datta (2025), geospaNN supports a range of core statistical tasks, including nonlinear mean-function estimation, covariance parameter estimation, and spatial prediction with uncertainty quantification. Leveraging the sparsity of the NNGP approximation, the software integrates naturally into the GNN framework, enabling the use of graph-based operations and opening the door to more advanced neural architectures in geospatial modeling.

Despite these strengths, the current version of geospaNN has several limitations. At present, the package supports only a limited set of stationary, parametric covariance models and does not handle non-stationary or non-Gaussian spatial processes. The neural network component is designed and tested mainly for simple architectures such as multi-layer perceptrons, which work well for moderate-scale spatial data but limit applicability to more complex or high-dimensional input structures. One important future direction is to increase the flexibility of our model and add features to the main steps in geospaNN to adopt more general estimation and prediction tasks. In addition, geospaNN currently requires R-dependency and does not support GPU acceleration. In the future releases, we will address these key issues to further improve the performance of the software.

Conceptually, a longer-term direction for geospaNN is to evolve into a general framework for geospatially informed deep learning, where spatially structured message passing can be incorporated while maintaining statistical interpretability. We also plan to extend the methodology to additional data types and distributional settings beyond the current Gaussian framework.

## **Reviewer 2**

**Author response:** Thank you for the very positive feedback. We appreciate your comments on the software installation and dependency management in our repository. Please feel free to let us know if you have any further suggestions on the software.

## **Additional Changes**

- Minor typos throughout the manuscript were corrected.
- Install instruction updated for a more clear and concise workflow. Also catch up with the recent release of PyTorch and PyG libraries.

## References

- Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E. and Banerjee, S. (2019), ‘Efficient algorithms for bayesian nearest neighbor gaussian processes’, *Journal of Computational and Graphical Statistics* **28**(2), 401–414.
- Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D. and Wilson, A. G. (2018), ‘Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration’, *Advances in neural information processing systems* **31**.
- Murphy, B. S. (2014), Pykrige: development of a kriging toolkit for python, in ‘AGU fall meeting abstracts’, Vol. 2014, pp. H51K–0753.
- Saha, A. and Datta, A. (2018), ‘Brisc: bootstrap for rapid inference on spatial covariances’, *Stat* **7**(1), e184.
- Zhan, W. and Datta, A. (2025), ‘Neural networks for geospatial data’, *Journal of the American Statistical Association* **120**(549), 535–547.