Referee Report

Title: Spatialize v1.0: A Python/C++ Library for Ensemble Spatial Interpolation

Author(s): Felipe Navarro et al.

MS No.: egusphere-2025-3272

MS type: Development and technical paper

The authors present a Python library, `Spatialize`, which implements several spatial interpolation methods with automated hyperparameter calibration. The paper addresses an important need in the geosciences community for accessible spatial interpolation tools. However, the manuscript requires substantial revisions before it can be considered for publication.

The authors claim that the package is designed for experts and non-experts with minimal geostatistical knowledge. However, as an economist with an interest in climate data, I think the implementation still requires a fair level of understanding of the underlying model and basic parameters especially if you plan to do parameter calibration. An initialization is required for the library to conduct a grid search.

In addition, the paper does not clearly articulate what `Spatialize` can do that existing libraries (`SciPy`, `PyKrige`, and `scikit-learn`) cannot. The authors should clearly state which capabilities are unique to `Spatialize`, a table of performance comparison would be appreciated.

Major comments:

The flow of the paper is chaotic and fragmented. The authors present a series of simulation and validation, but they lack a coherent framework of how the examples are related, or build upon each other.

The performance evaluation relies mostly (if not solely) on graphical presentations, lacking numerical support. When performance is similar, it is difficult to identify the differences between figures, such as Figures 20 and 22. A table of quantitative metrics should be presented.

The validation is solely based on simulation data. A real world application would help a lot for demonstrating how the library can be applied in empirical studies.

The library supports high dimension interpolation, such as space-time variation, this is theoretically interesting as it can capture the dynamic special dependencies if they exist. But if this makes sense in practice remains unknow. If high-dimensional interpolation is

a key feature of the library, a real-world example demonstrating its necessity and showing how the library improves performance would be helpful.

It is not clear how ensembling multiple models outperforms the predictions of a single model, nor how the ensembling function is defined.

Specific comments:

Given that the stated target users include non-experts, it would be helpful to provide intuitive explanations of what each algorithm does in the algorithm descriptions.

In line 11, the period before the parenthesis citation should be removed. "...point locations. (Li and Heap, 2014)." should be "... point locations (Li and Heap, 2014)." The same applies to line 77.

Figures are not sufficiently discussed. For example, Figure 8 (a) is only mentioned in terms of the name, no explanation why the errors are clustered in low and high levels, but fewer observations have middle level errors. Also according to Figure 8 (b), it seems index 600 is lower than index 302, contrary to line 284, which states that the lowest error is located at index 302?

The function in Code snippet 1 has wrong indentation. Line 2 should be indented.