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

Response to reviewers

Dear reviewer,

We sincerely appreciate the time and effort that you have dedicated to providing valuable feedback on our manuscript. We are truly grateful for your insightful comments on our paper. We find them valuable and constructive. We have provided a point-by-point response to your comments and concerns (in blue). Additionally, certain figures have been adjusted, and we have prepared a new manuscript that incorporates all the changes.

Referee #1

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.

This is an important observation. The manuscript has been revised to clearly explain the underlying model and calibrate the parameters. We have reorganized the manuscript to better introduce each concept, facilitating an easy transition from simple to complex examples using real data.

Additionally, a new section has been incorporated under the title of "The Spatialize Library". The purpose of this section is to provide a concise introduction to the library, whilst also comparing it with existing libraries through a table of performance comparison (**Table 1**).

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.

We hope that incorporating the "The Spatialize Library" section results in a more seamless progression between the description of the ESI algorithm and the usage examples.

Moreover, the examples have been restructured into two distinct “case studies” to accurately reflect the intended manner of use of the library by its users, rather than the previous separation between gridded and non-gridded implementations with arbitrary examples.

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 I/ and II. A table of quantitative metrics should be presented.

We have incorporated tables with MAE, RMSE and MSE metrics in order to provide an explicit numerical performance evaluation. However, for the real-world example (copper grade dataset), we are only able to offer cross-validation metrics (see explanation in the next point).

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 copper grade drill holes dataset employed for the non-gridded example corresponds to a real-world application. We understand that our previous manuscript was not very clear in this aspect, which is why a thorough description of the datasets has been added (Section **4.1**). We expect that this is now clearer.

The choice to use a synthetic dataset –besides the real-world drill hole dataset– is because synthetic scenarios allow for better performance evaluation: in real-world applications, reference maps are not usually available, since measurements are taken at specific sampling locations. The decision to use simulation data is made so that performance metrics can be calculated across unmeasured locations. Due to the sparse nature of real-world data, we are only able to provide numerical evaluations for locations with available measurements and cross-validation methods.

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.

While high-dimensional interpolation is indeed a distinctive feature of the library, we have intentionally omitted such examples for two key reasons:

First, the primary objective of our case studies is to demonstrate the practical usage of the library's various tools, enabling readers to integrate them into their own analyses. Since the syntax remains consistent regardless of dimensionality, a high-dimensional example would not provide additional methodological insight beyond what is already presented.

Second, a comprehensive real-world spatio-temporal dataset would necessitate substantially more complex visualizations and extensive analysis—requiring a separate publication to do it justice. We have prioritized simpler, more focused applications to clearly illustrate each tool's functionality while maintaining a reasonable manuscript length.

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

ESI's ensemble approach outperforms single-model predictions by combining multiple local perspectives and reducing sensitivity to individual partition configurations.

In traditional interpolation, a single model uses all available data but applies uniform assumptions across the entire domain. This can lead to poor predictions in regions where local spatial structures differ from global patterns. ESI addresses this by creating multiple local estimates, each based on a random partition. While individual local estimates may be unstable (since they use fewer samples and depend on the specific partition configuration), aggregating many of them stabilizes the predictions while preserving sensitivity to local spatial patterns.

Specifically, the random partitioning and aggregation ensure that:

- Samples closer to the target location appear together in partition cells more frequently, naturally receiving higher effective weight in the final estimate without requiring explicit distance calculations or neighborhood definitions.
- Each partition captures different local spatial configurations. Aggregating across partitions averages out errors or biases from any single partition while reinforcing consistent local patterns that appear across multiple partitions.
- The resulting distribution of estimates enables uncertainty quantification, which single-model approaches cannot provide.

The ensemble function is simply the mean, median, or mode of the estimates across all partitions for each target point.

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.

Algorithm 1 is explained in lines **104-111**. If this explanation does not address your concern, we would appreciate further clarification on what additional information would be helpful.

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.

Thank you, this issue has been solved (lines **11** and **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?

This issue has been addressed in the revised manuscript through the clarification of examples and the reorganisation of the text. Additionally, we have enhanced the figure discussion and replaced the figures with colour-blind-compatible versions. The mentioned indexing issue has been resolved.

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

This issue has been solved (around line **274)**.