The study presents a new open-source Python library for ensemble estimation of geospatial earth system variables. The new library (GPEP) is based on an existing one (GMET) that is programmed in Fortran. The authors aim to increase the flexibility that GMET provides, increasing the amount of variables that can be analyzed, the number of spatial interpolation schemes and other important characteristics. They apply the library to three demonstration experiments where they compare their results to those provided by GMET. They conclude by remarking the advantages of the new library and some of its drawbacks.

The main contribution of this work is to have translated a model in FORTRAN to Python. FORTRAN is an old-school programming language, fast and expressive, that has been extensively used for numerical model programming and other intensive tasks. It is less popular than it used to be and compiling it may be complicated. On the other hand, Python is an interpreted language that does not require to compile its codes, is very portable and with a plethora of libraries around it that automatically creates synergies with every new library, like it could be the case with GPEP. Indeed, the authors mention the possibility to interoperate GPEP with Scikit-Learn, a Python-based machine learning library, which may give access to many machine learning and artificial intelligence algorithms. Translating GMET to Python may make the software more accessible to many young researchers, so I believe that this work constitutes an interesting contribution to GMD.

I have, nonetheless, some comments and concerns about the current incarnation of the document that I would like to present.

Response: Thank you for the comprehensive review and insightful feedback on our manuscript. We have addressed each of your comments in detail below and revised the manuscript accordingly.

1. One of the main drawbacks of the new library, with respect to GMET, is its efficiency. It is a well-known drawback of Python, so it could have been expected. However, I wonder why, GMET being open source, the authors did not use the existing GMET FORTRAN code and wrap it using Cython, for instance, to get the best of both worlds. Many performant Python libraries, like SciPy, are just wrapping legacy FORTRAN code -for linear algebra, for instance-, making fast code available to Pythonistas. This approach would have saved time, since most of the routines were already programmed, and the effort would have been directed toward making the fast code accessible through Python. The total effort would have been smaller and the result equal on functional terms but better from a performance perspective.

Response: Thank you for your insightful feedback. We recognize the inherent performance advantages of Fortran, and the potential benefits of wrapping existing Fortran codes using tools like Cython. However, our primary goal with the development of the new package was not just to achieve a direct translation of GMET but to harness the extensive ecosystem, flexibility, and the rapid development pace that Python offers, especially with its growing capabilities in machine learning and data science.

While wrapping the Fortran codes would have indeed provided a performance boost, it would also come with its own set of challenges, such as increased complexity in maintenance, potential issues

with cross-platform compatibility, and limitations in seamlessly integrating with newer Python-based tools and libraries. By building directly in Python, we aimed to create a more accessible platform for the broader community, facilitating easier contributions, extensions, and integrations with other Python libraries. We do appreciate the value of performance, especially for large-scale applications, and will consider hybrid approaches cautiously in future versions, balancing both performance and development flexibility.

2. I miss some manual, or at the very least a small tutorial, to start to work with the library. Jupyter notebook is one of the standard ways to provide tutorials for Python libraries, which helps the user to familiarize with how to work with a new library in a graphical environment. I have checked the test cases in Zenodo, but they are quite a dry way to approach a new library. On top of showing its performance, I believe more developed examples would be very appreciated. Indeed, I believe the manual would be especially important to learn how to use the library in conjunction with sklearn since library interaction is not always an easy topic to grasp.

Response: The ./README.md and the ./docs/How_to_create_config_files.md serve as the manual for using the GPEP. We do recognize that the documents may not be so easy to follow for beginners. We have added a notebook, i.e., ./docs/GPEP_demo.ipynb, which streamlines downloading test cases from Zeono, run test cases, and visualize ensemble outputs. We have added clearer descriptions in Section 3.3. We thank the reviewer for taking the time to check the GMET cases.

3. A comment similar to the previous one, but related to the problem being solved. Existing users of GMET may already know the problem that the library solves, but new users could benefit from a small schematic definition of the problem. A small sketch or diagram could help the users to understand what problem the library solves and how it is solved. I have seen in the GitHub repo that there are many links to different papers, but a user, which may not necessarily be a scientist, may really appreciate a small theoretical introduction to the problem. In a sense, it would be making the library accessible for those outside academia that may use it to solve more practical problems.

With this comment in mind, I would say that section 2 could be extended so that readers may get a better idea of the library's inner workings. I believe that papers should provide enough detail to be self-contained, and I am not sure that all the most relevant details have been included in the document.

Response: Thank you for your suggestion. We agree that a more detailed introduction to the methodology will greatly benefit readers by providing all requisite information within the GPEP manuscript, minimizing the need to consult external literature. To this end, we have expanded Section 2 to include a more detailed exposition of the methodology. We have also segmented it into three distinct sub-sections for clearer understanding. While this provides a robust overview, we still advise readers seeking intricate technical details to refer to the original papers cited in Section 2, as we cannot introduce all nuances within this manuscript.

Besides, we also added a schematic in the manuscript, i.e., the new Figure 1, which shows how GPEP performs geospatial estimation.

Some minor comments:

Line 117, equation 1. I believe it should be two equations. It is just a problem with the organization of the equations.

Response: Thank you for pointing out this problem. Equation 1 should have two lines but the format was corrupted because of an unknown reason. We have fixed it.

Line 149: I think the 's' should be removed

Response: Yes. We have removed it.