Regarding the response to the reviewers' comment on our manuscript entitled "**pyESDv1.0.1: An open-source Python framework for empirical-statistical downscaling of climate information**" by Daniel Boateng and Sebastian G. Mutz.

Dear Dr. Charles Onyutha,

We would like to thank you for agreeing to be the handling editor of our manuscript. We also thank the anonymous reviewers for their valuable comments and suggestions. We addressed each comment and suggestion by the reviewers and think it substantially improved the quality of our manuscript and made it more useful to prospective readers. We hope the revised manuscript also meets the referees' expectations and the high standards of GMD. The most important changes are summarized below:

1. We have improved and extended the package documentation website (https://dan-boat.github.io/PyESD/) to make it easy for new users to get started. We have added practical examples that cover all the modelling routines of PP-ESD and used Jupyter Notebooks, making it easy for users to adapt for the full end-to-end downscaling exercise. Most importantly, we provide further explanations of the methods included in the package, describe the installation process (which has been resolved based on issues raised by the reviewers), and showcase some projects where the package has been successfully used.

2. We have modified the outline figure to indicate what is included in the framework of pyESD. This resolves potential confusion regarding the capabilities of pyESD. Moreover, we have made modifications to Figure 6 to make it more friendly for colorblind individuals, as suggested by the reviewers.

We have provided more details on these and other points in the point-by-point response to the referees' comments. We deeply appreciate your and all referees' efforts to help us improve our manuscript.

The submission file consists of our point-by-point response to the referees' comments, and the revised manuscript (with tracked changes) specifying all the modifications made in accordance with the referees' comments.

Please contact us if further clarifications are required.

Sincerely,
Daniel Boateng and S. G. Mutz

Reviewers' comments are repeated in black. Authors' replies are highlighted in blue font, and the revised texts in the manuscript are in quotation marks with blue italics font.

Response to Reviewer 1 (Anonymous)

Global climate models often have difficulty accurately estimating climate change at the local scale due to their coarse resolution. To bridge this gap, statistical downscaling employs various statistical models that link large-scale predictors with local climate data. These models can range from simple linear regression to more advanced machine learning algorithms. The manuscript being presented describes a Python package that incorporates all the necessary steps for statistical downscaling, along with two examples—one in Europe and one in Africa.

Overall, the manuscript is very well written with a clear structure and relevant examples. The software package includes all the necessary steps for a downscaling application and, due to the widespread application of statistical downscaling, it can be very useful for a wide readership. Therefore, I recommend publishing the manuscript. However, I do have a few recommendations that the authors may want to consider.

We thank the reviewers for their constructive comments and time for highlighting parts of the manuscript that require further changes and improvement.

1) In many applications of statistical downscaling, where the statistical model is calibrated with observations, users often encounter missing values. In my interpretation, the software package does not consider this possibility. I am aware that missing values can be replaced by zero anomalies, or the climatological mean, but this is not an optimal solution. The estimation for future climate change should not be affected by missing values since climate simulations typically do not contain them. Nevertheless, the software package should include the possibility of missing values in observations. Missing values should be considered as non-existent; that is, time steps with missing values should not be included in the calculation of, for instance, principal components.

We thank the reviewer for raising such an important point, and we agree. Generally, before the start of the model training process, the time indices of missing values, if they exist in the predictand data, are dropped, which means they are not included in the optimization of the learning model. However, if users are sure that replacing missing values in their predictand data is merited for their particular problem, they can now choose to do so with methods ranging from simple interpolation to using seasonal means, as well as some advanced imputation techniques, as well as alternative steps. We have also included these details in the package documentation (https://dan-boat.github.io/PyESD/methods.html) to highlight the potential implications or dangers of replacing missing values to inform users.

2) I have downloaded the package and attempted to install it. I was unsuccessful using the pip3 command - it installed just the description of the package but not the actual libraries. I was successful, however, using a conda environment, dowloading the github folder, applied pip install -e pyESD on that local folder copying the generated libraries to my local conda python library folder.

Thus, after some manipulation, I I also attempted to run one of the tests included in the package, which was also successful. Therefore, I assume that the package can generally be run by other readers without problems. However, I was unable to conduct a full test of the entire package, which should be left to the wider readership interested in downscaling.

We thank the reviewer for raising the issues concerning the pip installation of pyESD and the extra efforts used to install the package after further modifications. Our initial attempt to upload the source files to PyPi failed to include the package modules. We have fixed this bug, tested its installation, and it was successful. Moreover, to facilitate an easy start for users with the package, and to enable them to conduct a comprehensive test using actual weather station datasets, we have expanded the package documentation website (https://dan-boat.github.io/PyESD/tutorials.html) to include practical tutorials that cover all the downscaling modelling routines. We used Jupyter Notebooks for these tutorials, which can be easily followed and adapted to users' interests. All the datasets, scripts, and notebooks are publicly available for users to adapt to their weather stations.

Apart from these two main and minor points, I would like to congratulate the authors on creating such a comprehensive software package and a very well-written manuscript.

We thank the reviewers for their encouraging feedback and for recognizing the importance of the pyESD tool in downscaling climate information.

Response to Reviewer 2 (Anonymous)

General Comments

To begin with the conclusion. I was very impressed with this manuscript. I believe it highlights a great effort towards modernizing and democratizing of tools for geoscience, in this context of downscaling. I believe it can be accepted with very minor changes detailed below, as it appears to be in an advanced stage already.

To support my conclusion, we can begin with the journal scope. This manuscript was submitted to GMD under the category of Model description papers; I believe this paper falls comfortably within the scope, under the specific context of "frameworks and utility tools used to build practical modelling systems". The paper presents a comprehensive framework for empirical downscaling with interfaces (e.g. CSV/NetCDF) compatible with many geoscience datasets. In terms of formalities, the paper title contains the utilities name and number, and the code for this is clearly linked to both a living github repository and a zenodo archive. The manuscript outlines the full context of applicability, noting where users must make executive choices, and illustrates the process in the context of a detailed tutorial based on a concrete Geoscientific case study of downscaling in a regional scale, the Neckar catchment. This is a well-motivated case study, due to its strong topography, and sensitivity to nonlocal features such as teleconnections all of which make it challenging for prediction.

Additionally to the context of scope, I thought the introduction gave a thorough and honest review of the process of empirical downscaling, its limitations, the currently available software, and their limitations. Such care was repeated in the package description (where details such as the data formats, and cost of data processing, the importance of predictor and regression/learning model selection, were appreciated) and the case study (where the benefit of trying a multitude of approaches for predictor selection, and learning models gave robustness to the approach, and explored the ease and benefit of having a library of tools within reach). The literature review was consistently comprehensive throughout the manuscript

The case study was detailed, including the sources of all data. The predictor selection and model evaluations were also well explained and visualized, leading to robust findings. The results were novel for the weather stations and have clear value for downstream impact assessment.

I was less impressed by the Documentation in the repository and believe it was overly represented in the text.

We thank the reviewer for raising such important concerns and for their overall positive feedback on the manuscript. We agree that the package documentation website requires further improvement. We have enhanced the documentation website (https://dan-boat.github.io/PyESD/methods.html) to include details about the methods of PP-ESD, demonstrated its use through a series of tutorials with weather stations, and provided additional information to help users get started more easily. We believe that the best way to improve the documentation further is to make changes in response to the experiences and suggestions of future users.

Minor recommendations

Figure 1.

I would recommend additions to the figure to highlight a clean demarcation of what is and what is not PyESD, particularly at the interfaces. From what I gather in the manuscript text, the Station/Reanalysis/GCMs inputs have different availability/level of interfacing/processing. It would be good to see what is exactly the boundary of this package which is what the users will most likely interact with. I presume this may have been loosely hinted at with colours, though this was not described in the caption either.

We thank the reviewer for pointing this out. In fact, the colour scheme provides details on what is within the framework of pyESD, including data preprocessing, ESD modelling pipelines, data analysis and visualization tools, while the input datasets are provided by users and the example of impact assessment studies that the downscaled outputs can be used for outside the pyESD framework. We modified the outline to highlight this and include details in the caption to resolve the package interface:

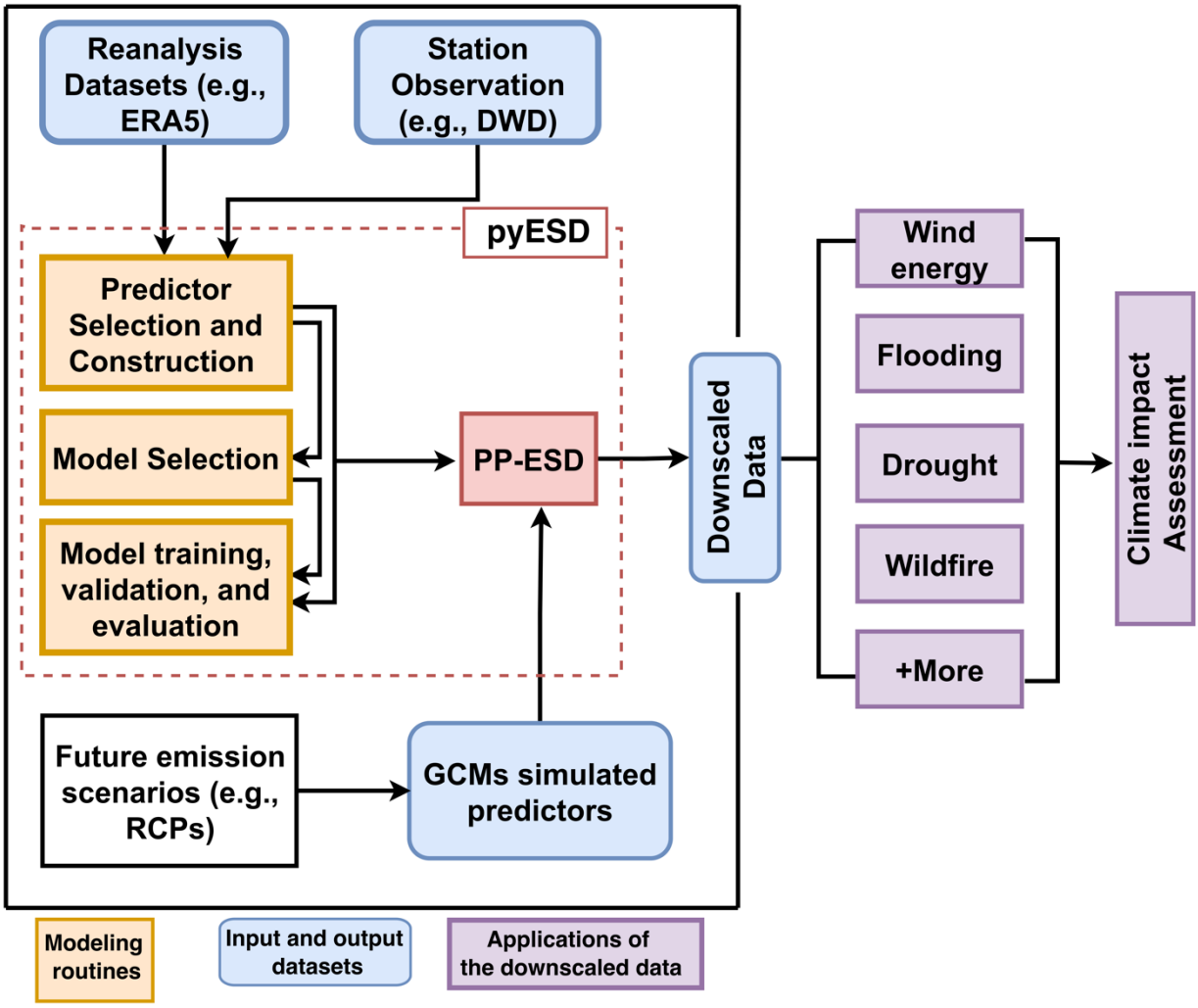

Fig. R1: The main features and workflow of PP-ESD implemented in the pyESD package (highlighted in red dash line box). The weather station and reanalysis datasets are used to select the robust predictors for model training and validation. The trained PP-ESD model is then coupled to GCMs simulations forced with different scenarios to predict the local-scale future estimates that can be used for climate change impact assessment (not included in the pyESD package).

Pseudocode description and formatting:

This may depend on what is permitted within the GMD templates, but I found the code quite hard to parse when dispersed through the prose, and written in italicized format. I would highly recommend code/pseudo code to be formatted properly (using latex packages such as \usepackage{listings}). It would be more readable in larger blocks, and listings allows things such as line numbering, colour coding, and commenting which will greatly aid in readability.

We agree with the reviewer. However, the ESD template doesn't explicitly highlight how code snippets should be formatted. We have improved this throughout the manuscript but are willing to make further changes based on the editor's suggestions regarding the accepted format (e.g.,):

```
1  from pyESD.Weatherstation import read_station_csv
2  variable = "Temperature" #or 'Precipitation'
3  SO = read_station_csv(filename, variable)
```

Uncertainty from Bayesian models:

In terms of the future directions the authors mention deep ML emulators etc. Have the authors also considered the direction of statistical emulators? For example including the data uncertainties correctly into a regression or Bayesian framework will give some parametric uncertainty in the forecasting model. These can be propagated to aid with prediction robustness. If possible please comment in the Bayesian models section and/or the future work section

We thank the reviewer for highlighting this. Statistical emulators within the framework of Bayesian inference for uncertainty quantification are well-established in machine learning and climate modelling communities. Often, these techniques are employed for spatio-temporal statistical downscaling at the scale of Regional Climate Models due to their computational requirements (e.g., Barboza et al., 2023). Incorporating such tools into the PyESD framework in the future would first require extending the package's capabilities to handle spatio-temporal learning and then testing its integration through preliminary studies. Therefore, mentioning this as an immediate future development would be misleading, as it is likely beyond the scope of immediate future improvements. However, we highlight this as a more general way forward in the development of such models.

Typo: Line 571 references 3.2.2 within section 3.2.2?

This has been corrected.

Overstating Documentation:

Though it is nice to see some documentation for this package, and the flowcharts were visually instructive, most of the currently available docs pages found on the repository are under construction (https://dan-boat.github.io/PyESD/index.html), contained dead links (e.g. to the examples folder on the homepage readme), or were a cursory few sentences, and a very large proportion of the API completely without docstring/comments. It is obviously unfair to compare to the likes of SciKitLearn or MatPlotLib, or some excellently documented Julia packages e.g. here (https://clima.github.io/OceananigansDocumentation/stable/). But I think the documentation arguably should contain a lot of the details that were provided in this manuscript to be considered

good documentation. I would not say that this package was “well documented”, I’d recommend either removing statements like this, or hopefully, making some substantive changes to add information and code snippets to help users get started, as well as results and details tutorials, the flowcharts help conception but users really want to see the objects in the documentation.

We agree. We have enhanced the documentation website to incorporate all the explicit suggestions made by reviewer 1. These suggestions include providing further explanations of the methods of PP-ESD, showing how to use the package with practical examples, expanding the description of case studies, and improving the documentation of the pyESD modules. The tutorial demonstrating how to use the package will make it easier for new users to get started (please refer to RC1 comments).

Colourblind friendly palettes.

I mainly remark upon this due to the semi-transparent red/black pairing in Figure 6, which (I think) can look identical for some people. And if visualization is part of the toolkit then colorblind friendly palettes should be used.

This is important. We have changed the colours in Figure 6 to red and blue using distinct markers to ensure colourblind-friendliness. We emphasize that various colorblind-friendly palettes can be applied with the visualization utility tools, depending on the user's preference. The visualization tools do not impose a fixed colour palette and can thus be utilized with any colour combinations.

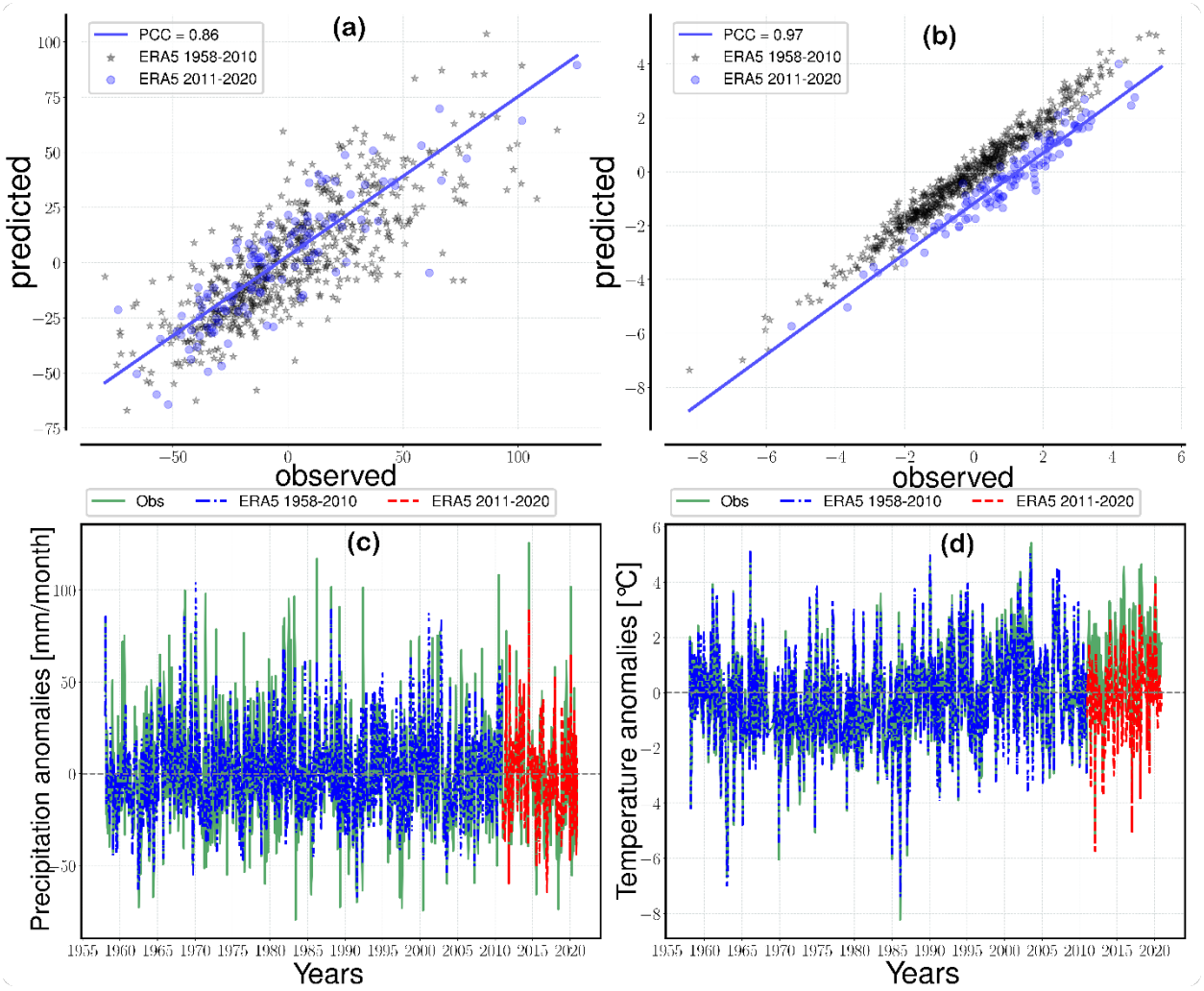

Fig. 6: Prediction example for the Hechingen station using the final regressor for precipitation (a, c) and temperature (b, d). The top panel (a-b) shows the linear relationship between the predictions and observed values, and the PCC (R value) for the testing data (blue-colored circles). The bottom panel (c-d) shows the 1-year moving average of the observed (green, solid) and ERA5-driven predictions for the training period (blue, dash-dotted) and the testing period (red, dashed).

Response to Reviewer 3 (Anonymous)

The manuscript presented here is a valuable contribution to the scientific literature on downscaling. The manuscript is very well-written, and the authors have clearly presented their work and done an impressive job of writing it up and making the tools they have developed usable by the climate impacts community.

Notwithstanding, I have a few overall comments that should be addressed before publication, and some minor comments throughout, both edits and comments where clarification or additional detail would be helpful.

Overall comments:

1. The authors use the term “coupling” throughout to discuss how the ESD models are used with GCMs. However, I take issue with this term given that it has a very specific meaning in terms of climate modeling. For example, one speaks of a land model (e.g. CLM) coupled to an atmosphere model (e.g. CAM, in the context of CESM) and information transfer occurs both ways, in other words, feedbacks occur. I would strongly suggest using a different term for this, but if you are insistent about using the “coupling” term, it needs to be deliberately clarified at first mention so that it is not misinterpreted by folks in the community who have worked on the actual coupling infrastructure of climate models and for whom this means something very specific. A suggestion rather than using the term “coupling” would be “pipeline” or something like “ingestion of climate model output”. Another suggestion would be to call it “weakly coupled” or something that also has a very specific and technical meaning.

   We agree with the reviewer that the term 'coupling' can be misleading. However, since this is a statistical downscaling model and not within the scope of climate models (whether fully coupled to different systems like the ocean or ice sheet), it should not be confused within the context of this manuscript. To make sure to avoid such misunderstandings, we have clarified this in our initial mention, explaining that the term 'coupling' refers to its literal meaning (e.g., combined or linked with) and that the calculations of ESD models are not fed back into the GCMs

2. To start off with, I think the authors have done an impressive job of documenting the package and trying to make it is as user-friendly as possible. However, when I compare the documentation in the body of the paper to what I find on Github, I think a great deal of the documentation in the paper should be on Github and reduced in the body of the manuscript. I would suggest only describing what the package does at a high level in the manuscript, and the specifics should be on Github. Moreover, they should be in a ReadtheDocs page not just on github pages. Currently, the documentation on Github is very rudimentary and should be fully fleshed out before the paper is accepted so that readers of the paper can immediately reference the documentation online when they want to start using the python package.

   We agree with the reviewer that the documentation needs improvement. We have expanded the documentation website to include most of the sections highlighted in the manuscript. Regarding hosting the website documentation, we prefer to use a single host, which means one link for the website. Therefore, we will continue to use GitHub Pages due to its ease of deployment after any updates in the documentation website.

Specific comments:

L11: As already mentioned, would not use the word coupling here, instead say application. Coupling means something very specific in the context of climate modeling

We now clarify what we mean by coupling in this particular manuscript, since replacing each occurrence with a more elaborate string of words/explanations would have compromised overall readability. We change the coupling in line 11 to application:

"This demand has led to an increase in the application of Empirical Statistical Downscaling (ESD) models to General Circulation Model (GCM) simulations of future climate."

L20: would not capitalize machine learning

L26: typo: downing should be downscaling

L42: take out accurate, instead say useful

L50: dynamical downscaling or statistical downscaling, ESD being one type of statistical downscaling

L58: not really coupled here

L95: I think the authors mean "suitable" instead of "suited"

L100: is there a citation available for the esd R package? If so, please include.

L111: Reconsider usage of "coupling" term.

L116: remove "by"

All the above suggestions have been corrected as suggested.

L100-L117: most of the packages discussed here are in R, but there are a few packages that the authors aren't mentioning available in Python. For example, the xclim package published by the OURANOS regional climate modeling consortium in Montreal. There are other examples as well. Since the pyESD package is python-based, would suggest condensing the discussion of R packages and also including some discussion of what is available in Python already.

We have mentioned these packages in our manuscript now (see lines 115-118). However, we would like to note that the other available Python-based downscaling tools (e.g., the xclim package) are primarily used for model output statistics and not for perfect prognosis statistical downscaling. Most of the publicly available PP-ESD models are developed in the R environment, which is why they dominate our discussion.

"Many of the Python-based tools currently available are primarily designed for bias correction in MOS downscaling, and extending these tools to the PP-ESD framework would diversify the publicly available downscaling tools (e.g., xclim (Bourgault et al., 2023), ibicus (Spuler et al., 2023), CCdowncaling (Polasky et al., 2023))"

L167: Why not store in zarr files instead of pickle files? You could have NaNs for gridcells not covered by the weather stations.

We opted for pickle files to store the extracted time series of the predictors due to their simple implementation at the early stage of the model development. Typically, the time series of the predictors consist of regional means, which do not contain NaNs, and the implementation and thorough testing would involve significant work. However, because of the overall advantages of Zarr, npy, and hdf5 files, we plan to include them in future development. We have highlighted their advantages and added them to the "to-do list" section of the package documentation website (https://dan-boat.github.io/PyESD/index.html).

L199: Could a polygon rather than a circle be defined?

The choice of a circle (i.e., radius) was purely a software development decision. The circle effectively selects the polygon grids within the defined range. For instance, if a radius of 5 km is specified around the weather station, and the large-scale data has a resolution of 80 km, this means that only the grid where the station is located would be selected. We find the use of radius as a parameter to be more intuitive than specifying polygon size, even though both approaches would result in the same area extent. We do acknowledge that polygons may be more suited for particular problems. However, implementing tools that allow the drawing of polygons would only be tailored to more specific needs, yet require significant development.

L274: Please include a definition of the term "feedforward"

We modify the text in lines 281-283 to clarify the meaning of 'feedforward,' which implies that the flow of data through the neural network is unidirectional (i.e., from the input layer to the output layer without recurrent connections).

"The MultiLayer Perceptron (MLP) is a classical example of a feedforward ANN, meaning that the flow of data through the neural network is unidirectional without recurrent connections between the layers (Gardner and Dorling, 1998; Pal and Mitra, 1992)."

L384-387: recommend adding numbers for the model training steps

This has been corrected.

L391: need a citation here

We have added citations about how cross-validation can prevent overfitting.

(e.g., Moore, 2001; Santos et al., 2018)

L401-403: remove extra parentheses around eqtn numbers

This has been corrected.

L495: 30a should be 30 years

This has been corrected.

L508: technically it is near-real-time, there is a 5-day lag

We have added the 5-day lag of the daily data availability to the text.

L517: Why not use CMIP6? This should be clarified.

We chose to use CMIP5 to maintain consistency when comparing with regional climate models that were forced with RCP scenarios in previous studies over the region of the case study. However, since the primary focus is to demonstrate the modelling framework, any GCM output could have been utilized. We have added a sentence to clarify the reason for choosing CMIP5 model output and to indicate that the package can be used with any GCM outputs (see lines 529-532).

"We highlight that CMIP5 model output was chosen in this illustrative study to enable consistent comparison with previous regional climate models over the region and any GCM outputs (e.g., CMIP6) can be combined with pyESD."

L595: are you using the typical D-statistic or the less typical p-value from the KS testing as a measure of significance? I found this information later, but it should be moved up and mentioned when KS testing is first discussed.

We thank the reviewer for addressing this issue. We have modified the text (see lines 626-629) to provide clarification at its initial mention, resolving the differences in the KS testing metrics. We computed KS D-statistics for all grid cells and displayed only those areas with a p-value less than 0.05, indicating a significant difference in distribution between the reanalysis predictors and the simulated predictors.

The KS statistic lies within the 0-1 range, with lower values indicating greater distribution similarity. For our 2-sided tests, we reject the null hypothesis (H_0 = the datasets have identical underlying distributions) for in the case of p-values being smaller than 0.05.

Package documentation:

Practical examples section needs more detail, would recommend adding end-to-end Jupyter notebooks with full examples of how to run the package along with example datasets (synthetic is fine).
We agree. The package documentation site (https://dan-boat.github.io/PyESD/index.html) has been expanded to include practical examples for all PP-ESD modelling routines. We used actual weather stations to demonstrate these through a series of tutorials. All the tutorials can be easily adapted with datasets, and the notebooks are publicly available.