Response to reviewer 2

The manuscript investigates the possibility of making an easy use of AI techniques in existing ESM. It addresses an interesting concept by focusing on coupling FORTRAN models with PYTHON functions via an existing coupler. The authors develop and evaluate a python tool called EOPHIS that adapts OASIS3-MCT coupling library for a new purpose. The manuscript demonstrates EOPHIS capacity to run python functions such as neural network inference or density bias correction within NEMO ocean model workflow in an efficient way. Authors show that it is possible to generalize the coupling of ESM to innovative python methods and modern infrastructure. EOPHIS usage sounds promising, it offers a way to test new methods developed with a modern, user-friendly language before integrating them into an ESM environment. It also allows to externalize treatments on GPU.

I recommend clarifying the structure of sections 2.2 and 3 to improve readability. The role of EOPHIS and its interface with the OASIS3-MCT framework remain unclear. EOPHIS polymorphic characteristics make it difficult to describe. Introducing briefly OASIS3-MCT (before explaining how it is used) could provide helpful context for presenting EOPHIS.

We agree that the original draft did not sufficiently introduce OASIS3-MCT before presenting Eophis, making their relation obscure. Section 2.2 now begins with a concise description of the OASIS3-MCT communication workflow and explains how coupled Fortran and Python components interact through the OASIS3-MCT APIs. This context clarifies that Eophis relies on the existing OASIS3-MCT infrastructure to exchange data with the ESM.

Related modifications on revised manuscript: paragraph at lines 134-139.

The motivation for developing a new Python library should be simplified.

The reviewer makes a good point here. The original motivation in Section 2 indeed lacked structure and presented the arguments in a scattered manner. We reorganized the discussion into three explicit points with intermediate concluding statements for a more structured and progressive reading:

1. Missing Python-side capabilities required to interact with the OASIS3-MCT API,

Related modifications on revised manuscript: lines 151-153.

2. The unnecessary complexity of exposing the full coupling-library feature set for ESM–ML workflows,

Related modifications on revised manuscript: line 154 and 160.

3. The need for flexible and automated configuration mechanisms

Related modifications on revised manuscript: Paragraph at lines 162-163.

The presentation of Eophis and its main features has been changed in accordance with this new argumentation structure.

Related modifications on revised manuscript: Paragraph at lines 169-173.

To improve clarity, authors might consider to draw a timeline illustrating where and how EOPHIS operates within the coupling process.

The captions of Figures 1 and 2 were rewritten to provide a more explicit description of how Eophis interacts with the OASIS3-MCT API and the ESM.

Related modifications on revised manuscript: Captions of Figure 1 and 2.

To improve readability I suggest to use short sentences. I also advise having a single designation for each component along the manuscript.

We have addressed this comment by shortening sentences where possible.

While a detailed comparison of CPU and GPU performance is beyond the scope of this manuscript, it may be worthwhile to briefly address their relative price and energy consumption.

We agree that this is an interesting point. However, a detailed discussion of relative CPU/GPU costs, price, and energy consumption involves broader considerations that extend beyond the scope of this manuscript. We therefore do not expand this aspect in the main text. Instead, we refer the reviewer to *Durand et al.* (2027, https://doi.org/10.31223/X5HB6S), which provides a dedicated analysis of these topics.

We also note that an additional Appendix (“Cost analysis”) has been added in response to a related comment from Reviewer 1, which partly addresses cost considerations without entering into detailed CPU/GPU market analysis.

Related modifications on revised manuscript: Appendix B at lines 503-543. References to Appendix B at lines 311 and 334.

I recommend publication with minor revisions. Please find below a list of minor specific comments in addition to the general suggestions that may be taken into consideration by the authors.

Figure 1 : The presence of two OASIS API boxes in the diagram is unclear. In accordance with L160 I suggest to illustrate EOPHIS positioning with two main boxes : ESM (including the OASIS Fortran API) and EOPHIS (including the OASIS Fortran API). Could you clarify the variables (a,b, F(a,b)) in the legend.

Figure 1 was revised to clarify the encapsulation of the OASIS3-MCT Fortran and Python APIs within the ESM and Eophis, respectively. The caption was expanded with the workflow description to describe how the exchanged variables are transferred between components using

OASIS3-MCT. The variables on the figure are explicitly described in the caption and have been modified to fit with the examples given in the other sections of the paper.

Related modifications on revised manuscript: Diagram on Figure 1. Caption of Figure 1.

Figure 2 : Please clarify the role of Tunnel box by explaining how it is linked to OASIS API boxes from figure 1. It appears that EOPHIS encapsulates the 2nd OASIS API box. This should be clearly illustrated in both the figure and the legend. This figure describes components of the workflow not the workflow itself. Please explain briefly the workflow of the ESM/ML and how EOPHIS gets involved in it in section 3.1. If possible, I recommend separating the configuration phase from the execution phase.

We revised Figure 2 as a decomposed view of Figure 1 to maintain continuity in the presentation of the role of the OASIS3-MCT in the global ESM-ML coupling workflow. Figure 2 is now introduced as a focus on the internal Eophis components and their respective roles. The caption also now states more clearly that the Tunnel abstraction acts as the interface between the simplified Eophis API and the underlying OASIS3-MCT layer.

Related modifications on revised manuscript: Diagram on Figure 2. Caption of Figure 2.

Section 3.1 was also rewritten to introduce the complete ESM–OASIS-Eophis workflow in a more sequential way. We believe that this clarifies the introduction of the Eophis abstractions.

Related modifications on revised manuscript:

- Paragraph at lines 187-191 for complete coupling workflow
- Paragraph at lines 195-198 for introduction of Eophis abstractions

The remainder of the section was also restructured to describe the Eophis abstractions one by one, with additional explanations regarding the OASIS3-MCT operations they encapsulate or automate. The presentation order was modified as much as possible to first introduce abstractions related to the configuration phase, followed by those involved during execution.

Related modifications on revised manuscript:

- Paragraph at lines 200-203, 208, 217-219, 221-224, 225-227 and 230-234.

Figure 3 : I suggest clarifying the role of EOPHIS in the data redistribution.

The data redistribution operations are handled directly by OASIS3-MCT. The role of Eophis here is to configure the OASIS3-MCT layer through a higher-level interface. This distinction should now be clearer thanks to the revised description of the Eophis components in Section 3.1.

Related modifications on revised manuscript: Lines 208.

Figure 4 : Please separate configuration from execution. Authors might mention that EOPHIS configures light gray cells exchanges through OASIS3 MCT framework and it builds halos necessary for python functions.

The corresponding text was revised accordingly. We now explicitly clarify that the halo reconstruction relies on two distinct steps. First, during the configuration phase, a specific OASIS3-MCT configuration is used to exchange the additional cells. Then, during execution, Eophis reconstructs the halo regions.

Related modifications on revised manuscript: Lines 216-219.

Figure 7 : Can you comment on the differences in SYPD between both reference experiment a) and reference experiment b) ?

A clarification was added with the following footnote:

“The reference speed differs between the 2D and 3D cases because the 2D configuration required the HPC center’s ML libraries and associated software stack, implying a different environment and a NEMO build with different compilers.”

Related modifications on revised manuscript: Line 306 and footnote at page 13.

L135 : « Although OASIS3-MCT and coupling libraries providing similar API diversity, such as YAC, have already been employed for hybrid modeling, the practical steps required to develop such applications are not entirely straightforward. » To be clarified. Authors should make it explicit that the current coupling ESM/ML is done quasi-manually. « being not straightforward » understates the complexity and technical challenges involved in this process.

The intermediate conclusions of the revised motivation for Eophis (Section 2.2) introduced in the previous comment also explicitly highlight the practical difficulties associated with each point.

Related modifications on revised manuscript: Lines 151-154, 159-161 and 166.

L154 : « Finally, even once OASIS3-MCT is properly configured on the Fortran side and the Python component equipped with the necessary features, additional work is needed to implement and configure the OASIS3-MCT interface within the Python script and to prepare the corresponding OASIS3-MCT namelist. » Please clarify what has to be configured or implemented. Please name clearly OASIS2-MCT Fortran API and OASIS2-MCT Python API.

The paragraph was revised to clarify this point, as the previous formulation could be misleading by suggesting one of the main difficulties lies in the implementation of OASIS3-MCT Python API. This is not the case, as we intended to focus on the configuration of the OASIS3-MCT Python API, rather than its implementation. We therefore rephrased the text accordingly to emphasize on the amount of work required for preparing the coupling setup. We also listed the main steps required for this operation.

Related modifications on revised manuscript: Paragraph at lines 162-165.

L160 : « We propose a Python library named Eophis, which acts as a generic coupled OASIS3-MCT Python component equipped with the functionalities required by the coupling library, and in which any function from an external Python script can be imported and called with the exchanged data…. » From this description, it is not clear to me how does EOPHIS interact with OASIS and ESM. I suggest to make compatible the EOPHIS presentation and Figure 1.

L183… 191 : I recommend reorganizing this paragraph with a simple and sequential description of the workflow.

We address these two comments together as they are closely related. The issues raised regarding the clarity of the interaction between Eophis, OASIS3-MCT, and the ESM, as well as the need for a more sequential workflow description, stem from the same aspect of the original presentation.

These points have been resolved through the revisions made in Sections 2.2 and 3.1, as described in the responses above. The addition of a description of the OASIS3-MCT workflow and the restructured argumentation in Section 2.2, together with the fully sequential description of the ESM–OASIS–Eophis workflow in Section 3.1, now provide a more explicit presentation of the interactions between the different components.

Related modifications on revised manuscript: Paragraph at lines 134-139, 169-173 and 187-191.

L269 : « Additionally, the 2D cases are doubled. » This is not clear.

This sentence was reformulated to state that all 2D benchmark cases are executed a second time under a different configuration.

Related modifications on revised manuscript: Line 292.

section 4 : I recommend explicitly stating the reasons why the 3D experiment did not utilize GPU resources.

The text has been revised to state in a footnote that the library used in the 3D experiments is not currently ported to GPU architectures, which prevents the use of GPU acceleration for these cases.

Related modifications on revised manuscript: Line 293 and footnote at page 12.

Response to reviewer 1

In the scalability discussion, I wonder if adding some cost analysis can enhance the understanding of the resulting curve seen in Figure 7? I provide what I meant as below, but it is up to authors to decide if such an extension is desirable. [...]

We thank the reviewer for this suggestion. We agree that adding a cost analysis improves the interpretation of the scalability curve and provides a useful way to estimate both computational cost and runtime.

We have therefore added this analysis in a new appendix, which holds as presented in the reviewer comments for the synchronous strategy. For the asynchronous strategy, the original formulation was incomplete. We have corrected and extended it accordingly. The revised manuscript now provides a consistent and accurate derivation that matches observed behaviour.

Related modifications on revised manuscript: Appendix B at lines 503-543. References to Appendix B at lines 311 and 334.

Following this addition, we also acknowledged the reviewer contribution in the acknowledgements section. We would like to ask whether the reviewer would be comfortable with removing anonymity so that its contribution can be explicitly named in the paper.

Related modifications on revised manuscript: Line 573

1. Line 80: “Another category”: I suggest “Second category” to enhance clarity.

We have modified the text accordingly.

Related modifications on revised manuscript: Line 80

2. Lines 221-222, Figures 5 and 6: I think “grids line” and “exchs line” are a bit awkward. Maybe adding line numbers to figures such that can directly refer as “code lines 6 and 7”.

We have revised the figure annotations to include explicit line numbers. Content of the revised manuscript now directly refers to the code lines as suggested by the reviewer, which improved readability.

Related modifications on revised manuscript: Figures 5 and 6. Lines 245 and 246.

3. Line 268-269: “the 2D cases are doubled”: needs clarification. What in 2D cases are doubled?

The intended meaning is that all 2D cases are executed twice: once using a pure CPU implementation and once using GPU-based inference. The text has been revised to explicitly state that the cases are “run twice” under these two execution modes.

Related modifications on revised manuscript: Lines 292.

4. For section 5.2 Future perspectives, I wonder if authors can provide some views on the recent effort to revive the usage of model hierarchy by Shaw et al (2025) which seems relevant?

We have extended Section 5.2 to reflect this perspective with the following paragraph:.

"In the long term, the need for bridging strategies (Section 2.1) may indeed be transitional. As discussed in Section 2.1, several components of Earth System Models (ESMs) are increasingly being implemented in high-level programming environments, sometimes assisted by AI-based coding tools. This trend suggests that entire ESMs may eventually be developed in environments such as JAX, Julia, or PyTorch.

In such a setting, cross-language bridging layers would no longer be necessary, as neural networks and numerical components could coexist within a unified runtime. We believe this direction aligns with the vision of hierarchical model integration discussed by Shaw et al. (2025), potentially enabling a broader paradigm shift where hierarchies of ESMs with varying levels of complexity can be more easily constructed and deployed."

Related modifications on revised manuscript: Paragraph at lines 393-398.

5. Figure 5: Is "ESM, = eophis.register_tunnels(…)" a typo? It looks like the comma on the left hand side is a bug. In python it should be "ESM, _ = …" or "ESM = …".

The reviewer points out a confusing notation. The function returns a list because it takes as input a list of Tunnel configurations. In the example, only one Tunnel is specified. The comma is used to unpack the first element of the returned list. The notation was indeed misleading. We now explicitly refer to the first element of this list for clarity in the code presentation.

Related modifications on revised manuscript: code line 10 on Figure 5.

6. Figure 1

- Since in the later example you are going to us variables u, v, maybe you can replace a, b with u, v to help the readers?
- I would suggest consistently use OASIS3-MCT as in the text throughout the manuscript, rather than "OASIS3" as if this is a different idea.
- I also think add text to differentiate two "OASIS API" boxes. Such as "OASIS3-MCT API (Python)" and "OASIS3-MCT API (C-Python adapter)". Current Figure 1 confuses me easily.

We thank the reviewer for these relevant suggestions aiming to improve both the consistency of the manuscript and clarity of Figure 1. We have modified Figure 1 to apply these recommendations directly. The caption has also been updated accordingly to better describe the workflow and the role of each component.

Related modifications on revised manuscript: figure 1 and caption of Figure 1.