Response to reviewer

R. El Montassir, O. Pannekoucke and C. Lapeyre

June 4, 2024

We thank the anonymous reviewer for his/her feedback on our manuscript. We appreciate the positive remarks and constructive suggestions. We are grateful for the time and effort that he/she invested in evaluating our work.

We copied the reviewer's commentary below, and we replied in teal to each point. We also provide the changes made in the manuscript for each comment.

1. *Introduction and positioning of the paper: I appreciate the authors' introduction and I think that it effectively sets the stage for the study. However, I believe additional refinement could be beneficial. For example, around line 65, the authors discuss residual modeling as a methodology to mitigate imperfections in physics-based models. From my understanding, residual modeling in this context means using machine learning to correct errors of physical models. I suggest to avoid using the term "residual modeling" here, as it could potentially be confused with residual networks. In the same context, the authors mention that model correction "does not have the ability to enforce physics-based constraint." Correcting model errors can be approached in various ways. For instance, one approach involves defining machine learning-based parameterizations of physical models, where deep learning doesn't model a residual but instead represents missing physical processes. Recent advancements in the field have explored incorporating various physical constraints into such deep learning-based corrections. I suspect that this is what the authors are referring to in lines 70 to 80. I suggest that the authors revise this section and include references to recent state-of-the-art research to provide further context and clarity.*

   - Even if the term is explained in the text, we agree that the term "residual modelling" can be misleading, we have removed it. For the second point, adding data driven parametrisations is not part of what we called residual modelling, but, as you mentioned, it is part of the next category of methods that is more general and includes residual modelling. Therefore, the statement "does not have the ability to enforce physics-based constraint" is still accurate. However, this sentence is misleading and not necessary; we removed it. A figure (Fig. 1) has been added to clarify this concept of residual modelling.

   "To address imperfections in physics-based models, a common strategy is ~~residual~~ error modelling. Here, an ML model learns to predict the errors (also called residuals) made by the physics-based model (Forssell and Lindskog, 1997). This approach leverages learned biases to correct predictions (see Fig. 1.). ~~However, it does not have the ability to enforce physics-based constraints, as it primarily deals with errors rather than physical states.~~

   ~~An advanced variation of residual modelling involves the integration of~~ A more general approach that does not deal only with errors is to create hybrid models merging physics-based models and ML models. ~~In scenarios~~ For example, in scenarios where the dynamics of ~~Physics are~~ physics is fully defined, ~~a straightforward method involves using~~ the output of a physics-based model can be used as input to an ML model."

2. *The majority of section two is not necessary and should be removed. It is quite standard to write numerical schemes in PyTorch. You can just mention in section three that you implemented your numerical advection scheme in PyTorch, and you can discuss the space and time discretizations you used in the appendix. Everything else including the discussion on automatic differentiation, gradient decent and the subsection 2.1 on combining neural nets with physics should be removed.*

   - Similar comments were made by the other reviewer. We are aware that the section contains standard information, however, we do not fully agree that these details are known to the majority of the readers, especially in our restricted community, meteorology. We believe that it is important to provide a clear and detailed explanation of the challenges encountered when combining neural networks with physics-based models, in order to provide a smooth reading experience for the reader. We have, however, moved the second section to the appendix and updated the introduction of the section 3 as follows:

"In this ~~section, we introduce our hybrid~~ work, we address applications involving dynamics with unknown variables that require estimation. For example, the cloud motion field is one of the unknown variables in the application considered. In such cases, as discussed in the introduction, a joint resolution approach is more appropriate. Here, the physical model uses the neural network outputs to compute predictions, integrating the two models as follows:

$$y = \phi \circ f_\theta(x),$$

where x is the input, f_θ represents the neural network, ϕ denotes the physical model, and y is the output. In this setup, ϕ implicitly imposes a hard constraint on the outputs, potentially accelerating the convergence of the neural network during training.

This method raises some trainability challenges as the physics-based model is involved in the training process, and it should be differentiable, in the sense of automatic differentiation, in order to allow the back-propagation of gradients (refer to Appendix B). We show in Appendix B how spatial derivatives of PDEs can be approximated within a neural network in a differentiable way using convolution operations. This allows us to compute gradients and back-propagate them during the training process. This fundamental knowledge serves as a foundation for our investigation of novel hybrid Physics-AI architectures. With these established principles, we present in this section the proposed hybrid architecture, which is applied to cloud cover nowcasting. In this section, we introduce our hybrid Physics-AI architecture, ~~denoted as HYPHAI (an abbreviation for Hybrid Physics-AI),~~ detailed in Sect. ~~3.1. Section 3.2~~ 2.1. Section 2.2 explains the different physical modelling approaches investigated in this study. Following that, Sect. ~~3.3~~2.3, Sect. ~~3.4~~ 2.4 and Sect. ~~3.5~~ 2.5 sequentially present the training procedure, evaluation metrics, and benchmarking procedure."

3. *I also suggest to trim down the text in the remaining sections for example: around line 330 you should remove "which rely on computing the loss gradients with respect to the model parameters. These gradients guide the update of the model's weights during the training process".*

   - Done.

   "The training of the model parameters is achieved through gradient-based methods~~, which rely on computing the loss gradients with respect to the model parameters. These gradients guide the update of the model's weights during the training process.~~. Here, Adam optimiser [Kingma and Ba, 2017] is used with a learning rate of 10^{-3} and a batch size of 4 with 16 accumulation steps, ~~which allows~~ allowing us to simulate a batch size of 64. The training was performed using a single Nvidia A100 GPU for 30 epochs."

4. *The authors developed 4 different versions of their model based on different parameterizations of the source term. However, in the experiments, only two configurations, the HyPhAI-1 and 2 are tested. If you plan to only test these two configurations, you should remove HyPhAI-3 and 4.*

   - The same point was raised by the other reviewer. The reason why these two versions were presented even if they didn't show any improvement is to show the flexibility of the model. We added a sentence to clarify this and we moved these two versions to the appendix.

   "The second version of the hybrid model, denoted ~~HyPhAI~~HyPhAICCast-2, adds this source term to the advection. This modelling is described in the following equations:

$$\partial_t P_j + \overrightarrow{V} \cdot \overrightarrow{\nabla} P_j = \tanh(S_j) \quad \forall j \in \{1, 2, \ldots, C\}, \tag{1}$$

   where S_j is estimated using a second U-Net model (see Fig. 4). While the previous modelling describes the missing physical process in the advection, it does not satisfy the probability conservation property. Thus, this modelling does not conserve the probabilistic nature of P over time. To ensure the appropriate dynamics of probability, a robust framework is provided by continuous-time

Markov processes across finite states [Pavliotis and Stuart, 2008, chap. 5]. In this framework, the probability trend is controlled by a linear dynamics, ensuring the bound preservation, positivity, and probability conservation. Two other models based on this framework, named HyPhAICCast-3 and HyPhAICCast-4, are presented in the Appendix A1 and Appendix A2. However, these models did not show any performance improvement compared to the simpler HyPhAICCast-1. Indeed, beyond the performance aspect, this hybridisation framework is flexible, not only limited to the advection, and can be extended to other physical processes"

References

[Kingma and Ba, 2017] Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs].

[Pavliotis and Stuart, 2008] Pavliotis, G. and Stuart, A. (2008). *Multiscale Methods: Averaging and Homogenization*, volume 53. Springer, New York, NY.