Response to Reviewers

Reviewer #1

We thank the reviewer for the careful read of the manuscript.

Before addressing Reviewer #1's comments we note that we have reorganized the manuscript subtantially in order to incorporate Reviewer #1's detailed suggestions. We have also informed the editorial team and wish to state here that we have added Timothy A. Smith, NOAA Physical Sciences Lab, Boulder, CO, USA, to the team of authors.

In the following we address the reviewer's comments (reviewer's comments in red, our response in black).

Major comments:

The list of architectures given in section 2.2 should be revised. On the one hand, considering "blocks" or components of the network, it is not really a comprehensive list since it ignores

- Graph Neural Networks (i.e., the backbone of GraphCast, one of the leading atmospheric emulators (Lam et al., 2023))
- Transformers, which have been revolutionary in other ML/AI fields like natural language processing and image recognition/generation, and serves as the backbone for some of the leading atmospheric emulators like Pangu Weather (Bi et al., 2023), FuXi (Chen et al., 2023), FengWu (Chen et al., 2023), and in a sense FourCastNet (although FNOs/AFNOs/SFNOs tend to "feel" different than other transformers; Pathak et al., 2022)
- Regarding Convolutional Networks, at least some of the various works from Dale Durran's group should be listed, especially since the papers led by Weyn helped kick off the ML weather emulation generally. For example (Weyn et al., 2019; Weyn et al., 2020; Weyn et al., 2021; Karlbauer et al., 2023; Wang et al., 2024).

We thank the reviewer for their detailed suggestions. We have conducted a major restructuring and extension of the manuscript to accommodate all of the reviewer's comments. As a result, we have also extended the list of work cited.

The architectures above have proven skill in emulating medium range weather, whereas two of the architectures listed (LSTMs and Reservoir Computing) have not. Given that the authors state that the ocean prediction workflow mirrors that of NWP, I think it is therefore natural to make this comparison to Medium Range Weather. Moreover, for a more generic list like what is shown in this paper one could put LSTMs and Reservoir Computers under the same architecture umbrella, since they are both Recurrent Neural Networks, and therefore share the same inductive biases as outlined by Battaglia et al., 2016. As a final note on the RNNs, if Reservoir Computing is included in this list, then it may be useful to include references that focus on GFD related emulation rather than just

Lorenz-like systems. For example Arcomano et al., 2020 & Smith et al., 2023 might be useful to some readers.
We concur. In our revised and restructured manuscript, we have incorporated this comment and the related references. We have also accounted for the reviewer's comments (next paragraph of their review) regarding GAN's and the rise of diffusion models in our revised version.

This is somewhat subjective, but I strongly oppose the "Hard AI" and "Soft AI" terminology that is used. [+ rest of paragraph].
Although Chantry et al. (2021) attribute a different meaning to the terms "soft" vs. "hard", we concur with the reviewer that these terms are non-descriptive and can mean different things to different people. Following the reviewer's suggestions, we have made the following replacements:

- "soft AI" to "Hybrid physics-ML models"
- "hard AI" to "Purely data-driven models"

We've also added the following note in the revised manuscript:

> "Chantry et al. (2021) have used the terms "soft AI" versus "hard AI". We avoid the somewhat non-descriptive or ambiguous terminology in order not to give a false sense of which of these approaches is "harder" to realize."

Minor comments:

Line 30: "prerequisite to for" -> "prerequisite for"
Corrected.

Line 71: I would also include the following work in the list of hybrid dynamics/ML models: Arcomano et al., 2023
Added.

Line 78: PDE -> PDEs.
Done.

Line 83: I think the "i.e." should actually be "e.g." since MSE based loss (i.e., L2 norm loss) is only one example. Another popular choice is an L1 norm loss, although this has similar detrimental effects like producing overly blurred output. In generative applications, though, more generic loss functions are being used.
Thank you, we have modified the text to reflect this.

Line 138: Since the positive side of FNOs is listed, and since this is for an ocean audience, I would also list their main drawback for ocean applications - that they will be challenging (and maybe infeasible) to use in the ocean due to non periodicity and continental boundaries. This can create artifacts at the boundaries, which would limit their stability, and overall attractiveness, in comparison to atmosphere applications.

A very good point, which we have adopted in the revised manuscript. We have added the following sentence:

> “A drawback of FNOs applied to ocean (unlike atmospheric) modelling is the existence of land-covered portions of the domain, which renders challenging the use of periodic basis functions and may create artifacts near land-ocean boundaries.”