Summary

I appreciate the authors' revisions and responses to both my and the other reviewer's comments. I think the structure and focus of the manuscript have improved since the initial submission. I have a few remaining minor comments / suggestions, but overall I think this manuscript can be accepted subject to minor revisions.

Response to responses

> Reply: Sorry for the confusion. Each bar in Figure 10 represents the performance for the hybrid system that couples the ML methods into the ESM, not just the ML module alone [….].

Indeed I understood this, but was wondering about how relevant this discussion was in a paper focused on a new coupling approach. Nevertheless, I think what is in the manuscript is OK. It is useful as a case study on the impact of size of representative ML model on computational performance relative to a representative GCM, and is facilitated by the flexibility of the coupling approach. I agree it is also useful as a point of reference to compare to previous work, assuming it is roughly an apples-to-apples comparison.

Reply: Thank you for your comments. In Wang et al. (2022), a ResNet deep learning model was used. To enable a fair comparison, we also tested the overhead of ResNet in our study. This allows for a direct comparison of the computational overhead associated with different coupling approaches.

> Reply: This is because NE4 uses 128 cores for parallel computation, whereas SCM only uses a single core.

Would it be possible to increase the single column model performance by running with more cores? Again it is surprising that a simulation with multiple interacting columns is faster than a simulation with a single column, but maybe that is just due to differences in the typical processor layouts (no need to make any changes).
Reply: Yes, it is possible to use more cores for a single column model by decomposing the vertical layers. However, most climate models, including E3SM, use 2D horizontal decomposition. For a single column model, the computational cost is significantly lower than that of a global model, so vertical decomposition is typically unnecessary.

I understand your concern that the NE4 configuration might appear faster than the single column model, even though it uses a very coarse resolution and more cores. The reason could be that the single column model involves additional overhead, such as the higher frequency of output.

Specific comments

Lines 76-79: "The hard-coding has limitations. Such hard-coding approach restricts the ML algorithm’s ability to adapt to changes in the model dynamics over time, as the ‘online’ updating requires a two-way coupling between the dominantly Fortran-based ESMs and Python ML libraries.” I am still not sure I fully understand what is meant by "two-way coupling" in this context. Other than flexibility, how is the coupling used in O'Gorman and Dwyer (2018) conceptually different than that offered by FKB, CFFI, or the authors' new approach? In all of those cases, the ML model is typically trained offline, and is in a sense frozen when coupled online—like any parameterization, the only way its outputs change is as a result of its inputs changing. Are the authors envisioning some sort of "online" training approach that does not depend on the differentiability of the GCM?

Replay: In this study, the term ‘two-way’ means that the offline-trained ML model not only affects the GCM when coupled to it but also allows the GCM, particularly a potentially differentiable dynamical core, such as NeuralGCM, to ‘online’ update the trained ML model. O’Gorman and Dwyer (2018) saved the trained ML model in a NetCDF file. Then the GCM loads it. In their approach, the ML model is frozen and can not update during runtime.

In contrast, methods such as FKB, CFFI, and our approach establish an interface between Fortran and Python, where the ML model is loaded on the Python side. This design provides the flexibility to update the ML model with a differentiable dynamical core during runtime, which enables the possibility of online learning.

We have added the following discussion in the manuscript “When a trained ML model is incorporated into ESMs, it is frozen and cannot be updated during runtime. Recently, Kochkov et al.(2024) introduced the NeuralGCM, an innovative approach that enables the ML model to be updated during runtime with a differentiable dynamical core. This allows for end-to-end training and optimization of the interactions with large-scale dynamics. However, the hard-coding coupling method does not support such capability. ”

Figure 2: this is great—thanks for adding it, as well as the details about compilation. Maybe mention in the caption that a fleshed-out, compilable version of this toy example exists in the linked GitHub repository as well.
Replay: Thanks for the suggestion. We have added this in the caption.

Lines 512-513: "In contrast, Wang et al. (2022) reported a 100% overhead in their interface, which transfers parameters via files." What kind of model was used in Wang et al. (2022)? Is it comparable to the one tested here?
Replay: In Wang et al. (2022), a ResNet model was used to train the ML parameterization. To ensure a fair comparison, we also tested the ResNet ML model in our study. We have clarified this in the manuscript.

Line 519: "predicts the computational ratio relative to the CNTL run by taking the number of ML parameters as input" is somewhat vague. I might suggest using something like: "predicts the

ratio of the simulated years per day of the ML-augmented run to that of the CNTL run as a function of the number of ML parameters"
Replay: Thanks for the suggestion. Revised it.

Figure 9b: this is very minor, but it might make things a little more intuitive to read if the y-scale for the ratio exactly corresponded with the y-scale for the simulated years per day (i.e. ran from something like 0.222 to 1.111). This way the "truth" line would run through the top of each bar.
Replay: Thanks for the suggestion. Revised it.

Technical corrections

Line 132 "only minimal disrupting" -> "only minimal disruption to"
Replay: Revised it.

Line 161: "using toy code example" -> "using a toy code example"
Replay: Revised it.

Line 169: "real model" -> "fortran model"
Replay: Revised it.

Line 276: "[...] (1.5 km grid spacing). Met Office [...]" -> "[...] (1.5 km grid spacing) Met Office [...]" (i.e. remove the period).
Replay: Revised it.

Figure 10 caption: "Compassion" -> "Comparison"
Replay: Revised it.