Dear Joe Melton,

Thank you for the time and effort you have taken to review our work. We believe that your constructive feedback will help us improve the presentation of our findings in a revised version of the manuscript.

We are responding to your remarks in detail below. Your comments were left in black text, our replies in blue, old passages in red, and new passages in green.

Natel and coauthors are interested in developing emulators to allow easier integration of ecosystem models (like LPJ-GUESS) in broader frameworks that couple multiple models together. LPJ-GUESS is sufficiently computationally expensive that an emulator could be valuable for the multimodel frameworks (in particular LandSyMM). They use two machine learning based approaches: random forests (RF) and neural network (NN). Both emulators were trained using LPJ-GUESS outputs for some historical and future simulations. The emulators differed in both their performance and the main variables they were sensitive to but both were much faster than LPJ-GUESS itself.

The paper is generally well written and easy to follow. The work falls well within GMD's area of interest. I think the work is suitable for publication but have several questions that I would like to see answered beforehand.

Main comments:

1. I liked that the authors used two different ML-based approaches in their emulators and then attempted to understand/interpret what each emulator was sensitive to. This is valuable information but it felt like only half the story. What was missing was what LPJ-GUESS is sensitive to. If the point of the emulators was to allow cheaper approximation of LPJ-GUESS (the 'model') then the most important thing is that the emulator is responding in the same manner and to the same variables as LPJ-GUESS. There are many plots showing trajectories of pools and fluxes for LPJ-GUESS and the emulators (e.g. Fig 2 and 4) but no similar plots showing sensitivity of LPJ-GUESS as there is of the emulators (e.g. Fig 6). I realize this is more challenging with LPJ-GUESS since it is inherently a completely different kind of model, but I struggle to understand how one can trust either emulator without knowing if it is actually mimicking the model's sensitivities (which under this circumstance has to be assumed to be perfect).

We agree that matching the emulators to the original model sensitivities is valuable. However, as the reviewer correctly points out, a direct comparison between the ML explainability analysis (Fig. 6 and Fig. 7: SHAP values) and the LPJ-GUESS sensitivity is challenging due to structural differences between the models. Nevertheless, we believe that interpreting and discussing the SHAP values, especially for features related to the LPJ-GUESS forcing (climate variables and atmospheric CO2 concentration), in light of previous sensitivity studies of LPJ-GUESS (Ahlström et al., 2013, 2017; Piao et al., 2013), could hopefully address the reviewer's comment and provide more insight into how the emulation approach compares to the original model sensitivities. Please note that we refrain from discussing the SHAP values alongside LPJ-GUESS *parameter* sensitivity analysis studies, as they are not directly comparable or relevant to the emulation approach presented here.

In response to a relevant comment from the community (CC1), we have also recomputed the SHAP values for groups of correlated features, and we have amended the section below to reflect these changes.

Old passage:

4. Emulator explainability

[…]

4.2.1 Carbon stocks

The RF and NN emulators exhibited varying feature importance and feature value impacts on carbon stock predictions (Fig. 6). For VegC, the NN model ranked pre-disturbance VegC pool, growing degree-days, and annual mean temperature as the most influential features, while the RF model ranked initial VegC, time since disturbance, and CO_2. There was a discrepancy between the models in terms of the impact of annual mean temperature. While the NN model attributed a positive effect of lower temperatures on VegC, the RF model indicated a decrease in VegC under lower temperature values. Both models indicated a positive effect of the time since disturbance on VegC, which is consistent with the expected regrowth of vegetation carbon following a disturbance.

For SoilC, the NN model is primarily influenced by the initial SoilC state, temperature, and soil properties (sand, silt, and clay fractions), while the RF model relied mainly on the initial SoilC state. In NN model predictions, higher temperatures lead to an increase in soil carbon content. This result may seem counterintuitive, as higher temperatures usually increase Rh, reducing SoilC. However, in the warmer RCP scenarios where LPJ-GUESS simulates soil carbon increases, such as in tropical forests, NPP rises faster than Rh (Figs. 2 and 4). This might lead to a net carbon gain, suggesting that biomass input to the soil pool exceeds the increase in Rh.

For LitterC predictions, the NN model identified the initial LitterC state, growing degree-days, and soil sand fraction as the most significant features. In contrast, the RF model highlighted the initial carbon states of vegetation, soil, and litter, followed by time since disturbance and atmospheric CO_2 concentration, as more important. Both models concurred that in the initial years following a disturbance, lower time-since-disturbance values were associated with higher LitterC, which aligns with the observed peak in this pool after disturbance events.

Overall, variables associated with the carbon history of the stand before the disturbance are highly ranked in feature importance. Additionally, the time since disturbance, atmospheric CO_2 concentration, growing degree-days, and mean annual temperature also appeared to be crucial for predicting C stocks.

New passage:

4. Emulator explainability

[…]

For carbon stock predictions, the grouped SHAP analysis indicated similar feature importance rankings for both the RF and NN emulators, with only minor differences in the magnitude of feature group contributions to model outputs (Fig. 6). Precipitation and soil attributes emerged as the least influential factors, though these features exhibited slightly higher sensitivity in the NN emulators. Variables associated with the initial carbon state prior to disturbance emerged as the most influential across all carbon stock variables and emulators. Temporal features—such as atmospheric CO_2 concentration and time since disturbance—generally ranked second in importance, except for soil carbon predictions, where climate variables were the second most significant factor. Notably, for SoilC, variables beyond the initial carbon state contributed minimally to model outputs, in contrast to patterns observed for other carbon stock components.

For carbon flux predictions, the RF and NN emulators exhibited distinct feature importance patterns (Fig. 7). SHAP values from the RF emulator identified the initial ecosystem carbon state as the most influential feature group. In contrast, the NN emulator emphasized the joint contribution of atmospheric CO_2 concentration and time since disturbance for gross primary production (GPP), while climate variables were most influential for both net primary production (NPP) and heterotrophic respiration (Rh).

New figures:

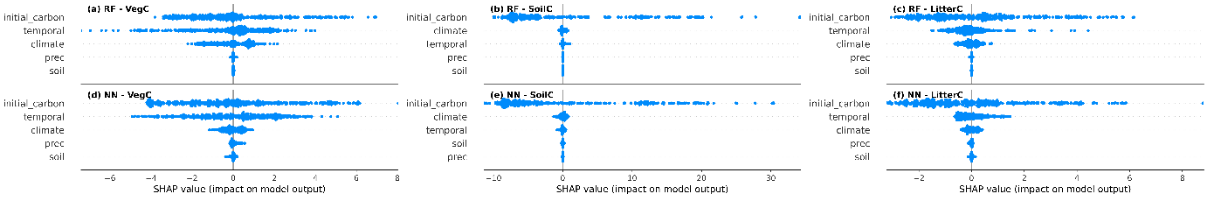

*Figure 6: SHAP values for grouped features in carbon stock predictions, including vegetation carbon (VegC), soil carbon (SoilC), and litter carbon (LitterC), using (a–c) Random Forest (RF) and (d–f) Neural Network (NN) emulators. The Y-axis lists feature groups ranked by importance, where correlated features were grouped as follows: initial carbon (*soilc_init, litterc_init, vegc_init*), climate (*temp, insol, temp_min, temp_max, mtemp_max, gdd0*), and soil (*clay, silt, sand*). Precipitation (*prec*) was not correlated with other features. The X-axis displays SHAP values, which represent the impact of each feature group on model predictions.*

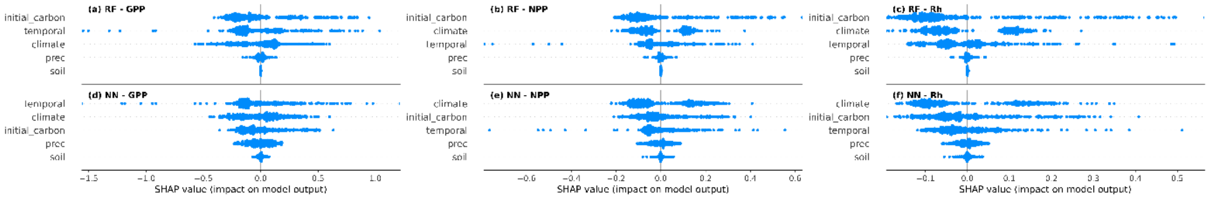

Figure 7: SHAP values for grouped features in carbon flux predictions, including vegetation carbon (GPP), soil carbon (NPP), and litter carbon (Rh), using (a–c) Random Forest (RF) and (d–f) Neural Network (NN) emulators. The Y-axis lists feature groups ranked by importance, where correlated features were grouped as follows: initial carbon (soilc_init, litterc_init, vegc_init)*, climate* (temp, insol, temp_min, temp_max, mtemp_max, gdd0)*, and soil* (clay, silt, sand)*. Precipitation* (prec) *was not correlated with other features. The X-axis displays SHAP values, which represent the impact of each feature group on model predictions.*

The section below has been amended to discuss the SHAP values in the light of previous LPJ-GUESS sensitivity analysis studies. Please note that the text in this section has changed significantly, mainly due to newly calculated SHAP values for groups of correlated features, instead of individual features.

Old passage:

5.2.1 Carbon stocks

In regard to VegC, both models concur that the initial state of VegC is the most significant predictor. However, they diverge on other influential factors. The NN model emphasizes growing degree-days and annual mean temperature, whereas the RF model emphasizes time since disturbance and CO_2 concentration. This suggests that the RF model is more sensitive to disturbance history and CO_2 levels, whereas the NN model captures temperature-related dynamics more strongly.

Both models indicate that the time elapsed since a disturbance positively impacts VegC, suggesting that the emulators effectively capture the dynamics of post-disturbance vegetation recovery. The rate of carbon accumulation is significantly influenced by the age of the forest stand. Younger forests, which have recently experienced disturbances, tend to absorb carbon at a much faster rate than older forests. In mature forests, carbon accumulation slows as trees approach their maximum growth potential (Cook-Patton et al., 2020; Pugh et al., 2019). Additionally, lower values of time since disturbance, representing the initial years following a disturbance, are associated with higher levels of LitterC. This feature helps capture the observed peak in the litter carbon pool from biomass killed during the disturbance (Zhang et al., 2024).

Initial carbon pool states were also important features, playing a central role in predicting both soil carbon and vegetation carbon pools. The initial VegC state might indicate how much carbon from organic matter is transferred to the litter and soil pools after a disturbance. Meanwhile, the initial SoilC pool may reflect the forest's carbon carrying capacity under prevailing environmental conditions. Without considering spatial proxies like coordinates in the emulation, these features may help differentiate biome-specific carbon dynamics. For instance, tropical forests store large amounts of carbon in aboveground biomass, while boreal forests store more carbon underground. We suppose that this bioclimatic variation is captured by the initial carbon pool features, offering insights into the potential carbon saturation of different ecosystems.

In the case of predicting SoilC, both emulators demonstrated excellent performance, with low errors and R^2 higher than 0.98. However, this may represent an overly optimistic result, partly attributed to the inclusion of the highly correlated initial SoilC pool as a feature. Since SoilC exhibits relatively small temporal variation, the inclusion of this feature might have exaggerated the model's performance by making the prediction task less challenging. The NN model incorporates a broader range of factors, such as temperature and soil attributes (sand, silt, clay fractions), suggesting it accounts for more complex interactions in soil dynamics. However, the RF model relies primarily on the initial SoilC state, implying that it gives less weight to environmental variables and might perform more conservatively in predicting soil carbon changes over time. That said, such a general prediction is beyond the scope of our stated objective, and we consider it legitimate to use this kind of information in surrogate models to speed up calculations needed for assessments. Nevertheless, to avoid this over-reliance on initial soil carbon state, future iterations of the emulator could apply more advanced regularization techniques to mitigate its influence in the overall output.

For LitterC, the NN model emphasized initial LitterC, gdd0, soil sand fraction, and annual mean temperature, reflecting sensitivity to regional soil conditions and environmental factors. In contrast, the RF model focused on initial carbon pool states (VegC, SoilC, and LitterC), disturbance history, and CO_2 concentration, indicating greater reliance on initial carbon conditions and disturbance-driven dynamics. The RF model seems to capture the indirect impact of atmospheric CO_2 on photosynthetic activity, which drives vegetation growth and ultimately influences litter carbon through increased biomass transfer to the litter pool.

Overall, the NN model appears to capture more complex ecological relationships, especially involving temperature and soil characteristics, which may make it better suited for understanding nuanced ecosystem processes, the RF model offers a more straightforward interpretation centered on disturbance and initial conditions. However, it tends to produce more conservative predictions and may overlook certain climatic variations across scenarios. Although ML explainability does not reveal the exact predictive value of each feature, it provides valuable insights into how individual features influence model behavior.

5.2.2 Carbon fluxes

Both models effectively capture the dominant role of atmospheric CO_2 in photosynthesis and the critical influence of initial vegetation carbon on potential carbon uptake and release. However, they diverge in their treatment of temperature variables. The RF model places greater emphasis on minimum temperature, suggesting a focus on colder temperature thresholds. In contrast, the NN model prioritizes growing degree-days and maximum mean temperature, indicating a more complex representation of how temperature extremes and accumulated warmth affect both photosynthesis and respiration. The NN model's greater emphasis on growing degree-days, in particular, suggests a stronger capability to integrate seasonal temperature dynamics and assess the cumulative impact of temperature on plant growth throughout the year.

Our results suggest that the NN model's decision process might align more closely with expected ecosystem carbon dynamics, while the RF model's predictions show a weaker alignment with underlying physical processes. This misalignment may negatively impact the RF model's predictions under warmer RCP scenarios over longer time periods, as shown in some disparities between LPJ-GUESS outputs and RF predictions toward the end of the 21st century for carbon fluxes, for example.

New passage (we have highlighted in bold text the discussion alongside the prior LPJ-GUESS studies)

5.2 Emulator explainability

[...]

Our results demonstrate that the initial ecosystem carbon state was the most important feature for predicting carbon stock variables in both the RF and NN emulators. Initial carbon pool sizes not only play a critical role in determining potential carbon loss under future warming scenarios (Todd-Brown et al., 2014), but may also serve as implicit indicators of forest biome characteristics in the context of emulation. Because spatial proxies such as geographic coordinates were excluded from the model inputs, baseline carbon pools may have indirectly captured biome-specific carbon allocation strategies. For example, tropical forests tend to store more carbon aboveground, whereas boreal forests allocate a larger fraction below ground.

Beyond initial carbon conditions, both models identified time since disturbance and atmospheric CO_2 concentration—grouped as temporal features—as key drivers of forest regrowth and carbon dynamics.

Time since the last stand-replacing disturbance likely served as a proxy for forest age, which plays a pivotal role in carbon accumulation. Younger forests typically act as stronger carbon sinks due to rapid growth, while older forests tend to exhibit slower carbon uptake as they reach maturity (Cook-Patton et al., 2020; Pugh et al., 2019). Additionally, elevated levels of LitterC may be predicted in the years immediately following disturbance—i.e., when time since disturbance is low—reflecting biomass transfer from vegetation to litter pools (Zhang et al., 2024).

Ahlström et al. (2017) demonstrated that climate biases influence LPJ-GUESS–simulated vegetation and soil carbon pools with comparable magnitude over time, although long-term soil carbon uptake tends to exhibit lower sensitivity. In line with this, our SHAP analysis indicates that VegC predictions are more responsive to climate variables than SoilC. The limited influence of climate on SoilC predictions may reflect the inherently slow turnover of this carbon pool. Within our simulation timeframe, the most influential predictor was the initial carbon stock after the spin-up period, with climate-driven changes contributing only marginally in absolute terms. The inclusion of the highly correlated initial SoilC pool likely simplified the learning task, potentially inflating emulator accuracy metrics for SoilC. This strong dependence on initial conditions may also signal reduced sensitivity to environmental drivers, thereby constraining the emulator's capacity to represent long-term soil carbon responses to climate change. Future work could address this limitation by incorporating regularization strategies aimed at mitigating over-reliance on initial carbon states.

Although precipitation ranked low in importance across most emulators and target variables, it did exhibit an influence on carbon flux predictions. **This finding is consistent with a prior study reporting a positive global relationship between GPP and precipitation in vegetation models such as LPJ-GUESS, particularly in tropical ecosystems (Piao et al., 2013). The observed sensitivity of carbon flux predictions to atmospheric CO_2 concentration and time since disturbance also reflects established experimental and modelling literature suggesting that elevated CO_2 can enhance NPP and forest carbon uptake (Ahlström et al., 2012; Piao et al., 2013).**

Other climate-related variables—such as temperature, gdd0, and insolation—also emerged as important drivers, particularly for VegC and carbon flux variables. In the NN emulator, climate features ranked highest in importance for NPP and Rh, and second—after temporal features—for GPP. **As noted in LPJ-GUESS sensitivity studies, longer and warmer growing season tends to enhance productivity in boreal and temperate regions with ample moisture (Ahlström et al., 2012). In temperate forests, however, the net impact of warming is more nuanced, balancing the benefits of an extended growing season against the drawbacks of increased summer soil moisture stress (Piao et al. 2013).** In our emulation framework, such seasonal dynamics could have been captured by the annually accumulated growing degree days. Closer inspection of the individual feature contributions (Fig. S2) confirms that gd00 and temperature are among the most influential predictors of carbon fluxes in the NN emulator.

NNs also showed a slightly greater sensitivity to soil properties - although still modest - in all predictions of carbon stocks and fluxes (except SoilC). This is consistent with expectations from LPJ-GUESS, which provides only a coarse representation of soil properties. Overall, our analysis suggests that the emulators capture key sensitivities present in LPJ-GUESS, albeit in different ways. However, interpreting the behaviour of ML models remains challenging when input features are highly correlated, as only coarse groups, rather than individual features, can be analysed. In addition, structural differences between LPJ-GUESS and the ML emulators make direct comparisons of feature importance difficult. Future emulator designs could benefit from the integration of physical constraints or process knowledge to better reflect plausible relationships between forcing and carbon dynamics, thereby improving the robustness and reliability of the emulation.

Although we believe the above discussion demonstrates that the emulators reasonably capture LPJ-GUESS model sensitivities, we acknowledge that they do not fully replicate the entirety of LPJ-GUESS. This limitation arises from the fact that the emulator has been trained on a limited dataset.

However, we also argue that achieving complete sensitivity matching is neither strictly necessary nor always practical in emulation applications. Emulators are typically designed to approximate specific input-output relationships of (parts of) a complex model under defined conditions, rather than replicate every internal process or parameter sensitivity of the original model. To clarify this point further, we have added the following in Section 5.4 Emulator Application:

New passage:

It's important to note that while the emulators were generally able to reproduce LPJ-GUESS's outputs related to forest carbon dynamics for the employed RCP scenarios, they should not be expected to capture all original model sensitivities, including both parameter sensitivities (e.g. parameters governing vegetation dynamics), and the original model' physical responses (e.g. the response of carbon dynamics to atmospheric CO_2 outside the bounds of training data)".

2. I am quite skeptical of the claim that (L 342) 'NNs excel at modeling continuous relationships, making them more capable of generalizing to unseen data, particularly when extrapolation is required'. My read of Muckley et al. (2023) does not support this contention. Muckley et al. test out the performance of linear regressions and black box RF and NN models. For the interpolation tasks, the linear model was poor but when it came to extrapolation it could out perform the black box models in some of the tests (~40%). The authors state the '{linear regressions}... may be desirable over complex algorithms in many extrapolation problems because of their superior interpretability...'. Lakshminarayanan et al. (2017) nicely demonstrate this for a toy example using a NN whereby the NN extrapolates poorly (their Fig 1 - left panel is bounds of 5 NNs). They show that the uncertainty bound via an ensemble technique can be created that encompasses the true function. So, getting to my main concern, given that NNs do not extrapolate well (same with decision tree-based methods), how can we trust the NN/RF models when they are forced to extrapolate? The approach here doesn't have any way to uncertainty bound the emulator results so it can extrapolate (poorly) blindly to the user. I don't expect the authors to fix this problem right now, but I would like to see more discussion about this difficulty and how it could be addressed for emulators as their use if becoming more common.

Thank you for this valuable feedback. We fully acknowledge that our original statement about NN extrapolation was not accurate. Additionally, we have not conducted a systematic experiment in this study to test this claim, therefore we sincerely apologize for this oversight.

Our argument in that specific passage should not have been about *extrapolation*, but rather *interpolation*, and we explain why here. During our preliminary experiments, we observed that NNs were better interpolators in the neighbourhood of feature values (while still within the bounds of the training data, and therefore not *extrapolation*) compared to RFs. One possible explanation (among several) for this is the difference in inductive biases and function approximation capabilities inherent to each ML technique. For example, RFs are based on CART (classification and regression trees), which partition the data using step functions, resulting in a non-smooth relationship between features and target. In contrast, NNs can learn smoother functions. Our preliminary observations align with what is illustrated in Fig.1a and Fig.1b below (not from our work).

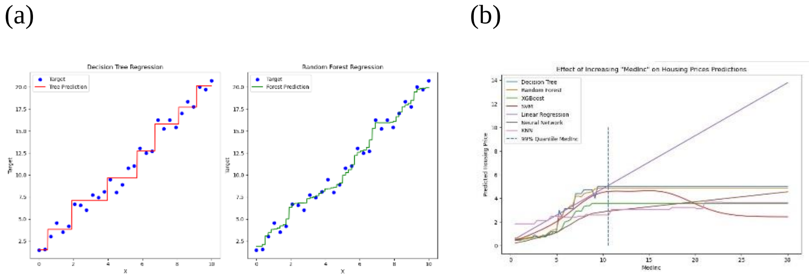

Figure 1. Inductive biases effect on predictions. (a) Decision Tree vs. Random Forest, (b) Extrapolation in the feature space by different machine learning models. Source: Christoph Molnar (2025)

While changing the term *extrapolation* to *interpolation*, and a few more details in the text could, to some extent, improve the text to better reflect our view, we have opted to remove the entire paragraph. This decision is based on the fact that our experiments do not systematically test inductive biases, nor have we reported our initial analysis in the manuscript. We believe that removing the passage below will ensure that the presentation of our findings is more accurate and better aligned with our experiments.

Old passage:

The generalization power of any learning algorithm depends on the inductive biases of that algorithm (Mitchell and Sheppard, 2019). RF models, for example, partition data through decision node splits using step functions that approximate relationships between inputs and outputs (Breiman et al., 2017). While this structure works well within the bounds of the training data, it limits the model's capacity to smoothly extrapolate outside the training data distribution. In contrast, NNs excel at modeling continuous relationships, making them more capable of generalizing to unseen data, particularly when extrapolation is required (Muckley et al., 2023). Given that our dataset contains less data for warmer climates near the end of the century, this limitation may affect the RF model's performance, particularly in extrapolating to these conditions. However, NNs are generally more robust in handling such extrapolation tasks due to their smoother function approximations. This suggests that RF may be more reliable when future projections remain close to the training data distribution or for interpolation tasks, while NNs could be better suited for scenarios where extrapolation is critical. This distinction is particularly relevant for climate assessments, where projecting across a wide range of future scenarios is essential for decision-making.

We have also added a paragraph in Section 5.4 to address the reviewer's comment on the extrapolation issue and uncertainty in ML emulations, as follows:

New passage:

Furthermore, ML-based emulators should not be assumed to reliably extrapolate beyond the training distribution without proper validation. In our study, both models were evaluated on scenarios that, while challenging, were within reasonable bounds of the training conditions. As demonstrated in Lakshminarayanan et al. (2017), NNs extrapolate poorly and uncertainty bounds using ensemble techniques may help to encompass the true function, an approach that could be explored in further development of emulation approaches.

Minor Comments:

Supplement - Can you explain more about the disturbance interval of 100 years and how that is applied? Also with fire off, the disturbance is then what? I see land use change is also not used.

We have now added the following paragraph to the Supplemental Materials to clarify this point in the section describing the LPJ-GUESS setup.

New passage:

In LPJ-GUESS, in addition to fire disturbances, we account for other external disturbances (e.g. windstorms, plant diseases etc) using a generic patch-destroying regime with a stochastic probability based on the expected return time. Disturbance return time varies substantially across the global forest area (Pugh et al., 2019), and the interval we have chosen is a simplification that has been adopted in several previous studies using LPJ-GUESS and other vegetation models, as reported by Zaehle et al., 2005.

L 84 - This sentence is a bit confusing as I found it less clear what the features were selected for.

This selection refers to the features used during emulation training. We have now revised the text as follows:

Old passage:

We selected 15 features, including variables related to climate, carbon states prior to a stand-replacing event, soil attributes, and a disturbance timer that tracks the time elapsed since the last stand-replacing disturbance (Table 1).

New passage:

We used 15 features as inputs to train the emulators. These included variables related to climate, carbon states prior to a stand-replacing event, soil attributes, and a disturbance timer that tracks the time elapsed since the last stand-replacing disturbance (Table 1).

L 112 - I wonder about the influence of this post-processing step for non-negative C stocks. How often did this come up? Were the instances where this came up for regions with very low stocks such that the inaccuracy would be small? (e.g. true value is 0.1 kg C/m2, so going negative is fairly reasonable but if it was really supposed to be 10 kg C/m2 then that is a big problem). This also demonstrates a problem with using an off-the-shelf NN whereby it has no knowledge of boundaries that one like a physics-informed ML model could.

The post-processing step was included as a safeguard against unrealistic values, which can arise when training NNs. However, we had not previously assessed its impact on our predictions. Thank you for raising this point. Upon analysis, we found that negative predictions occurred in only 0.09% of cases in our test dataset, primarily affecting vegetation carbon estimates. These errors were generally small, with a mean absolute error of 0.78 kg C/m² and a maximum absolute error of 2.89 kg C/m². Notably, 90% of these cases occurred when the true values were below 1.5 kg C/m². We will include the additional scripts for this analysis in our code repository, in a subfolder called tests/.

L 137 - sampled by grid cell, time, or ?

The random sampling was across grid cells and time steps. We have revised the text to clarify this as below:

Old passage:

To reduce computational time, the SHAP analysis was conducted on a randomly sampled subset of the test dataset (n=500) from the predictions made using the MPI-ESM1-2-HR forcing data for the historical period and climate scenarios (RCP2.6, RCP4.5, RCP7.0 and RCP8.5).

New passage:

To reduce computational time, the SHAP analysis was conducted on a randomly sampled subset of the test dataset (n=500), drawn across grid cells and time steps from the predictions made using the MPI-ESM1-2-HR forcing data for the historical period and climate scenarios (RCP2.6, RCP4.5, RCP7.0, and RCP8.5).

L 153 - with the meteorology/climate of MPI..., not the actual climate model itself.

Correct. Thank you! We have fixed this as below:

Old passage:

For our benchmark, we estimated the computational gain using predictions for a 165-year historical period (1850–2015) simulated with the MPI-ESM1-2-HR climate model.

New passage:

For our benchmark, we estimated the computational gain using predictions for a 165-year historical period (1850–2015) simulated with the climate of MPI-ESM1-2-HR climate model.

L 155 - Does LPJ-GUESS not have a dependence for computational cost on the number of PFTs present or the soil permeable depth?

Yes, indeed, if multiple woody PFTs can establish in a grid cell, each may form a cohort (a group of trees of the same PFT). More cohorts mean more computations (e.g., growth, competition, mortality), increasing runtime. The computational time is also dependent on the number of layers and soil permeable depth, which, in LPJ-GUESS, consists of 15 soil layers (each 10 cm deep) plus 5 additional layers for temperature padding.

Acknowledging that runtime may vary across grid cells due to these factors, we have now recomputed the computational gain by using a representative set of global grid cells (the same used for the validation period, n=344, Fig.1 of the manuscript). The recalculated value is presented in the revised text below.

Old passage 1:

We used a single grid cell (0.5° x 0.5°) for this comparison, noting that the gain would scale proportionally for larger areas.

New passage 1:

We used 344 grid cells from the validation set for this comparison, expecting that this representative set would capture runtime variations due to differences in the number of simulated woody PFTs and soil permeable depth across grid cells.

Old passage 2:

The emulators demonstrated a significant reduction in simulation execution time, with a 97% decrease observed when compared to the execution time of LPJ-GUESS.

New passage 2:

The emulators demonstrated a significant reduction in simulation execution time, with a 99% decrease observed compared to the execution time of LPJ-GUESS. The LPJ-GUESS runtime for the validation set was 5765.56 seconds, whereas the RF emulator ran in 1.26 seconds and the NN emulator in 2.77 seconds on a single-processor computer. However, since LPJ-GUESS grid cell simulations are typically run in parallel on high-performance computing systems, we also calculated the computational gains per grid cell by dividing the total runtime by 344 (number of grid cells used in this analysis) for both the original model and the emulators, resulting in an average 95% decrease in runtime with the emulators.

L 158 - If it takes 5000 sims to train the emulator but actually running the model that uses the emulator only happens 1000 times then you may end up with no net benefit. Also, sorry if I missed it, how many simulations did you need to train the emulator?

The emulator training required 25 simulations (4 RCPs + historical period x 5 GCMs) for each of the 3448 grid cells used in training, validation and test, resulting in a total of 86,200 samples, which is significantly fewer than the 574,650 predictions needed to run the full LPJ-GUESS model for all forested grid cells (n= 23,007), climate models and scenarios. While this upfront cost might seem substantial, the real benefit of the emulator lies in its integration with the LandSyMM model. Within this framework, the emulator needs to be called repeatedly to evaluate a wide range of management scenarios to supply e.g. timber demand or carbon sequestration options, which would be computationally infeasible using the full LPJ-GUESS model. Therefore, we are positive that the emulator will, in the long run, far outweigh the initial training cost when integrated into LandSyMM. The number of samples needed to develop the emulator is reported in Table 2.

L 207 - Could you give the real values in addition to the percent. It would be nice to see how much in clock time these cost (acknowledging it is system dependent).

Of course, please see the revised passage above with the recalculated computational gain, where we mention the clock times.

L 215 - I think the lack of decline in performance is simply due to training with the most extreme ends of the scenarios. This ensured that you were interpolating as much as possible. This is likely the only reasonable approach given the these techniques do not extrapolate well (see one of my main comments). But it means that the emulator always requires retraining for new scenarios and the scenarios always need be more extreme than what the actual system should realistically experience. I think some aspects of this bear mentioning.

Definitely, we appreciate your comment and are addressing this along with your main comment to clarify that extrapolation is a limitation of black-box models. The revised paragraph is presented below:

Old passage:

As shown in Table 3, the emulators were able to generalize to LPJ-GUESS outputs produced with climate projections not included in the training data without a significant decline in performance.

New passage:

As shown in Table 3, the emulators were able to generalize to LPJ-GUESS outputs generated with climate projections (RCP4.5 and RCP7.0) that were not included in the training data, without a significant decline in performance compared to the training scenarios (RCP2.6 and RCP8.5). This indicates that the emulators can generalize across intermediate emission scenarios, which fall within the range defined by the low (RCP2.6) and high (RCP8.5) extremes used during training. However, extrapolation beyond this range would require additional training and evaluation, as black-box models are not inherently robust in extrapolation tasks (Muckley et al., 2023).

L 218 - 'greatest accuracy' - by NN? Unclear as written.

Reply: No, we are actually referring to overall accuracy (for both NNs and RFs) compared to other targets. We have revised the sentence for clarity:

Old passage:

Among the carbon stock variables, SoilC was predicted with the greatest accuracy, exhibiting the lowest error and highest R² values.

New passage:

Among the carbon stock variables, SoilC was predicted with the highest accuracy by both NNs and RFs, showing the lowest error and highest R² values.

Fig 2 and 4 - What about adding new plots presenting these and the fluxes as cumulative plots so the impact of over/under predicting over time are visible? The fluxes is important as it has impact on how much C the land surface takes up/releases. The stocks as it changes how much C is emitted during disturbance or land use change. A cumulative plot can show the effect across the simulated period.

Reply: We have now generated the suggested plots (shown below) and will include them in the Supplemental Materials due to space constraints in the main manuscript. These plots illustrate cumulative biome-specific carbon (stock/flux) changes (1900–1930 vs. 2070–2100) across grid cells and time for each scenario, based on the test dataset. If this interpretation does not fully align with your suggestion, we are happy to refine the analysis further, please let us know any additional specifics you’d like included.

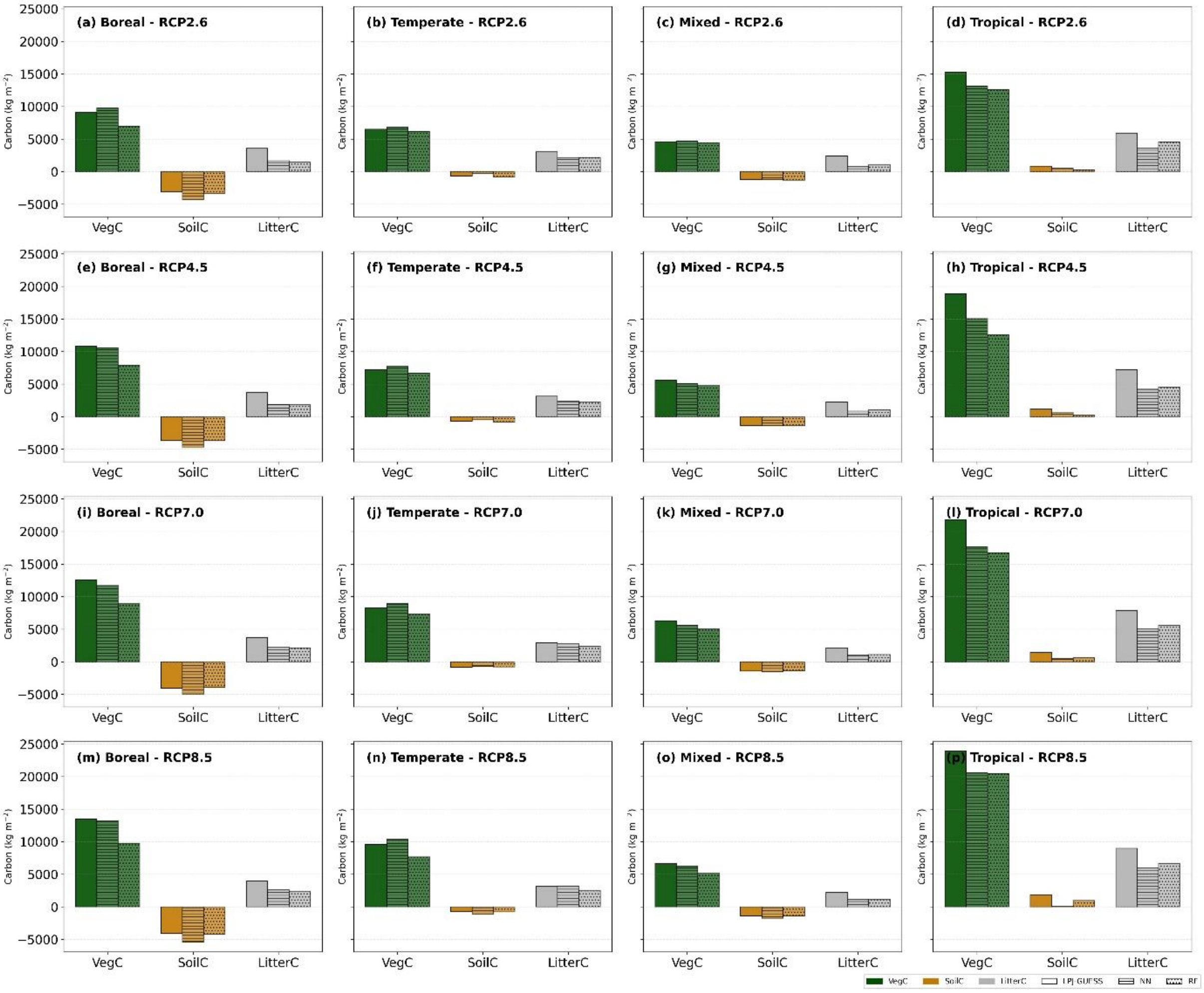

Figure 2. Biome-specific change in carbon stocks (1900 – 1930 to 2070 – 2100) for the test set. Values represent the cumulative stocks across time and grid cells for a range of climate change scenarios.

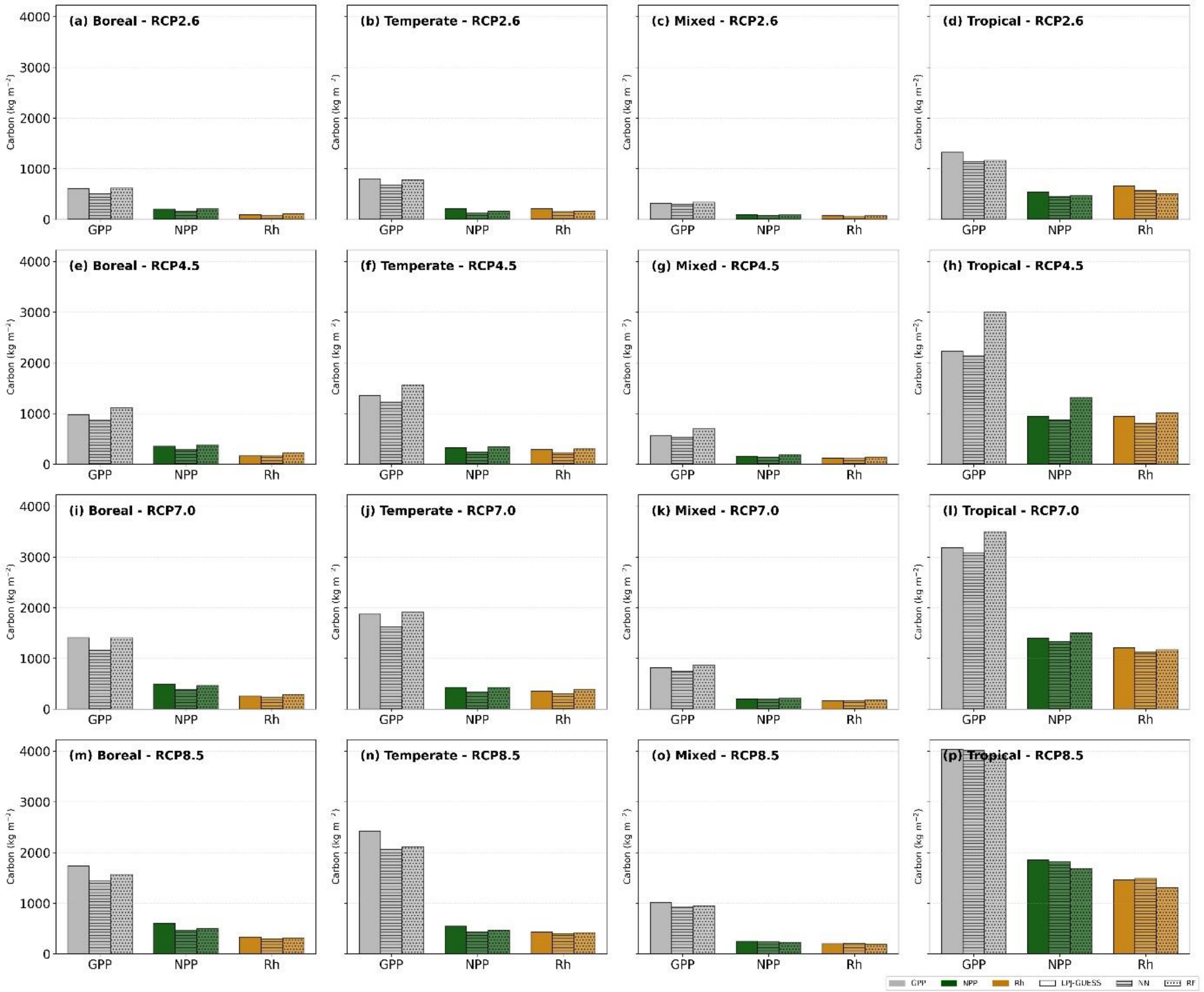

Figure 3. Biome-specific change in carbon fluxes (1900 – 1930 to 2070 – 2100) for the test set. Values represent the cumulative fluxes across time and grid cells for a range of climate change scenarios.

Lakshminarayanan, B., Pritzel, A., and Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles, arXiv [stat.ML],31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA. https://arxiv.org/pdf/1612.01474

Once again, we sincerely appreciate the reviewer's feedback, which will help strengthen the manuscript.

Best regards,

Carolina Natel, on behalf of all coauthors

References

Ahlström, A., Schurgers, G., & Smith, B. (2017). The large influence of climate model bias on terrestrial carbon cycle simulations. Environmental Research Letters, 12(1), 014004. https://doi.org/10.1088/1748-9326/12/1/014004

Ahlström, A., Smith, B., Lindström, J., Rummukainen, M., & Uvo, C. B. (2013). GCM characteristics explain the majority of uncertainty in projected 21st century terrestrial ecosystem carbon balance. Biogeosciences, 10(3), 1517–1528. https://doi.org/10.5194/bg-10-1517-2013

Cook-Patton, S. C., Leavitt, S. M., Gibbs, D., Harris, N. L., Lister, K., Anderson-Teixeira, K. J., Briggs, R. D., Chazdon, R. L., Crowther, T. W., Ellis, P. W., Griscom, H. P., Herrmann, V., Holl, K. D., Houghton, R. A., Larrosa, C., Lomax, G., Lucas, R., Madsen, P., Malhi, Y., … Griscom, B. W. (2020). Mapping carbon accumulation potential from global natural forest regrowth. Nature, 585(7826), 545–550. https://doi.org/10.1038/s41586-020-2686-x

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (n.d.). Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles.

Muckley, E. S., Saal, J. E., Meredig, B., Roper, C. S., & Martin, J. H. (2023). Interpretable models for extrapolation in scientific machine learning. Digital Discovery, 2(5), 1425–1435. https://doi.org/10.1039/D3DD00082F

Piao, S., Sitch, S., Ciais, P., Friedlingstein, P., Peylin, P., Wang, X., Ahlström, A., Anav, A., Canadell, J. G., Cong, N., Huntingford, C., Jung, M., Levis, S., Levy, P. E., Li, J., Lin, X., Lomas, M. R., Lu, M., Luo, Y., … Zeng, N. (2013). Evaluation of terrestrial carbon cycle models for their response to climate variability and to CO2 trends. Global Change Biology, 19(7), 2117–2132. https://doi.org/10.1111/gcb.12187

Pugh, T. A. M., Lindeskog, M., Smith, B., Poulter, B., Arneth, A., Haverd, V., & Calle, L. (2019). Role of forest regrowth in global carbon sink dynamics. Proceedings of the National Academy of Sciences, 116(10), 4382–4387. https://doi.org/10.1073/pnas.1810512116

Zaehle, S., Sitch, S., Smith, B., & Hatterman, F. (2005). Effects of parameter uncertainties on the modeling of terrestrial biosphere dynamics. Global Biogeochemical Cycles, 19(3), 2004GB002395. https://doi.org/10.1029/2004GB002395