General remarks

We are grateful for the evaluation of the manuscript and have now addressed the issues and questions that will certainly improve the comprehensiveness of the text in a point to point manner.

Specific answers to the remarks (EC: Reviewer's Comment; AC: Authors' Response)

RC: The validation workflow remains somewhat unclear for me. The manuscript states that an 80/20 train-test split was used, but later refers to five-fold cross-validation on the full dataset. It is still not fully clear whether the cross-validation was performed only on the training subset or on the entire dataset after the initial split. The authors should describe the evaluation workflow more precisely.

AC: We thank the reviewer for this comment. The manuscript has been revised to describe the validation workflow more precisely (Section 2.3.1).

The 80/20 split and the five-fold cross-validation served distinct purposes at different stages. The 80/20 split was used exclusively during benchmarking: the three candidate algorithms were trained on the 80% partition and evaluated on the withheld 20% (serving as the actual space) to identify the best-performing algorithm which is the performance reported in Table 1 and Figure 2. This split was not carried forward into subsequent steps. Once the best algorithm was selected and hyperparameters tuned, five-fold cross-validation was applied to the full dataset, i.e., the original 80% training partition plus the 20% benchmarking test set. This was done to confirm that the final model generalised stably across the full data before the refit.

The following lines have now been updated to help in describing the workflow better.

L334-337

"First, each candidate model was initially trained using its default hyperparameters (Table S.1), and baseline performance was evaluated on two test sets: the withheld 20% of the synthetic training data (possibility space) and an independent real-world dataset (actual space)."

L350-352

"Once the hyperparameter search converged, the selected configuration was used to train the selected model using five-fold cross-validation on the full dataset, i.e., the original 80% training partition plus the 20% test set used in the benchmarking stage (Section 2.3.1.1)."

RC: The term "mechanistic fidelity" still appears in the introduction (L102). I suggest making this terminology consistent throughout the manuscript, as is used in the Abstract.

AC: The reviewer is correct that "mechanistic fidelity" still appears in the introduction. We have now reviewed the manuscript and updated it to be consistent as used in the abstract. It can be seen below.

L101-103
"Therefore, we introduce mLDNDC, a machine-learning surrogate of LandscapeDNDC that maintains high fidelity to the parent process-based model's input-output behaviour,"

RC: A few smaller points could also be checked. The abstract states that XGBoost delivered the most accurate and stable predictions, but Table 1 shows that LightGBM is equal to or slightly better on some metrics (and I don't understand why XGBoost's results are in bold)

AC: Thank you for your important observation. The bold formatting on XGBoost results in Table 1 was intended to indicate the selected model, not superior performance. This has been removed to avoid implying XGBoost outperformed LightGBM across all metrics.

RC: I find the reported inference-time difference between XGBoost and LightGBM somewhat surprising. Since this may depend on implementation details, hardware settings, batching, model size, or parameter choices, I would avoid presenting it as a general advantage of XGBoost over LightGBM. This does not affect the main conclusions, as both models appear to be reasonable surrogate options. However, the authors could soften the wording and make clear that XGBoost was selected based on the performance and computational cost observed in this particular implementation. This would also be helpful for readers who may wish to choose among established tree-based surrogate models for similar applications.

AC: We appreciate the reviewer for this observation. The wording has been softened to make clear that the inference-time difference was specific to this implementation rather than a general property of the two algorithms. The revised text shown below now states that the selection was based on performance and computational cost observed in this particular setup, notes that the difference likely reflects implementation-specific factors, and acknowledges LightGBM as a viable alternative for readers working under different computational constraints.

L483-490

“Although XGBoost and LightGBM achieved very similar predictive performance across all target variables, XGBoost was selected as the final model due to its substantially faster inference time as observed in this implementation. On average, XGBoost took approximately three minutes to generate predictions for all four variables in this setup, whereas LightGBM took nearly one hour to produce the same outputs under identical computational conditions. This difference likely reflects implementation-specific factors such as hardware, batching, and model size, and may not generalise across all settings. For applications with different setups and computational constraints, LightGBM remains a viable surrogate option.”