Response to Anonymous Referee #1

We sincerely thank the reviewer for her/his effort and the very useful comments. We have revised our manuscript *Dynamic weighted ensemble of geoscientific models via automated machine learning-based classification* and have addressed all points raised by the reviewer.

Below, we provide a point-by-point response to all the comments. Text by the reviewer is in blue and indented. Our response is in black. *New text is green, italic*. *Existing (unchanged) manuscript text is black, italic*.

> This manuscript demonstrates the merits of automatic ML (AutoML) for two geoscience use cases. In general, the paper is well written. The authors developed an ML workflow to find the best combination of models or the optimal model. They used the term ML classifier. It took me a while to understand this is different from the conventional classification problem for which the goal is to identification class labels for each sample. Instead, the goal in this work is to find the weights for combining the physics-based model ensemble.

We greatly appreciate your positive feedback. Your encouragement has significantly boosted our confidence to continue our research in this field. To highlight the innovative aspects of our research, we have made relevant modifications to the relevant sections as follows. We hope that these revisions will help to provide a clearer depiction of the concept of mapping dynamic weights to probabilities in ensembles.

Section 2.1

"the ML classifier is trained to find the optimal models labeled as those that produce predictions with specific criteria (e.g., the least absolute error compared against observations for each sample of spatial/temporal predictions) under a specific environmental condition."

Section 3.1.1

"Additionally, the least absolute error between the predicted and observed moisture content was selected to label the optimal PTF for each sample in the workflow"

> My main question is whether it is necessary to use the ensemble-based AutoML in your use cases. Can you simply use a single ML model, e.g., XGBoost, to find the model weights/probabilities? Your workflow sounds like an ensemble of ML

models for an ensemble physics models. Is this right? If so, the computational burden may be overwhelming.

In response to your main question, the first author of the manuscript, Hao Chen, provided you with a preliminary reply by way of community comment (mainly including three aspects, please refer to https://doi.org/10.5194/egusphere-2022-1326-CC1), which we hope has addressed the primary concern to some extent. Further, we would like to add some other evidence (The results are shown in a new ***Table S3*** in our *Supporting Information*):

*Specifically, we selected the case of assembling 13 PTFs and compared 7 different classifier configurations. These classifiers were evaluated based on their computational time and the accuracy of their ensemble predictions. Here, we utilized the H2O-AutoML platform and made use of its scenario (parameter) settings, particularly the include_algos or exclude_algos parameters (refer to the provided link: https://docs.h2o.ai/h2o/latest-stable/h2o-docs/parameters.html), to train the first 5 classifiers: 1) Original classifier (**Original_withSE**): This refers to the original classifier used in our study (consist of 6 different ML algorithms, 32 models including 2 stacked ensemble models). 2) Balanced classifier (**Balanced_withSE**): In this configuration, we enabled the "balance_classes" parameter in the original classifiers to handle the potential class imbalance issue. 3) Balanced classifier without the StackedEnsemble algorithm (**Balanced_noSE**): Here, we excluded the StackedEnsemble algorithm from the balanced classifier, meaning that no further ensemble of ML models was performed. 4) Original classifiers with only the XGBoost algorithm (**XGBoost_withSE**): Based on the original classifiers, we eliminated other algorithms except for the XGBoost and the StackedEnsemble algorithms. 5) XGBoost classifier without the StackedEnsemble algorithm (**XGBoost_noSE**): In this case, we considered only the XGBoost algorithm in the original classifier, without utilizing the StackedEnsemble algorithm. Throughout the H2O-AutoML training process, we still set the total number of models (max_models, https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/max_models.html) to 30. For the 6th and 7th classifier, we opted to train the model in a Python environment using a combination of the state-of-the-art LightGBM algorithm (Ke et al., 2017) along with the efficient Optuna tool for accelerated hyperparameter optimization (Akiba et al., 2019). Note that a parameter called "n_trials", which represent the number of trials for each process in optimizing an objective function (https://optuna.readthedocs.io/en/stable/reference/generated/optuna.study.Study.html), were set to 30 and 300, respectively, thus, we obtained the other two classifiers,*

*namely **LGBM_noSE_30** and **LGBM_noSE_300**. The configuration details can be found in the code we have shared and updated recently (https://doi.org/10.6084/m9.figshare.21547134.v2).*

***Table S3**. Computational demands and accuracy of ensemble predictions of designed machine learning classification models.*

Classifier	*Computational time (minutes)*	*R^2 (-)*	*RMSE (m^3/m^3)*
Original_withSE	*84.23*	*0.8629*	*0.0444*
Balanced_withSE	*84.37*	*0.8654*	*0.0440*
Balanced_noSE	*44.54*	*0.8480*	*0.0467*
XGBoost_withSE	*80.28*	*0.8600*	*0.0449*
XGBoost_noSE	*52.40*	*0.8480*	*0.0467*
LGBM_noSE_30	*12.07*	*0.8465*	*0.0472*
LGBM_noSE_300	*300.33*	*0.8505*	*0.0466*

Specifically, ***Table S3*** provides several noteworthy findings:

1) In terms of computational demands, training classifiers with an ensemble of ML models (i.e., **with_SE**) does require more time compared to **no_SE** (e.g., **Balanced_withSE** takes 47% more time than **Balanced_noSE**). However, the absolute amount of time expended remains within an acceptable range. Comparing **LGBM_noSE_30** and **LGBM_noSE_300**, even though only one model is trained, the average computational time per model still surpasses that of training 30 models using the H2O-AutoML platform.

2) Regarding accuracy, the **withSE** classifiers generally outperform the **no_SE** ones, which further supports our hypothesis that it is challenging to determine whether a ML algorithm in isolation represents the optimal solution for a given problem.

3) The issue of class imbalance has minimal impact in this example, primarily due to the modest ratio between the maximum and minimum number of classes, which is approximately 2.19 (see the new ***Table S2***). Nonetheless, **Balanced_withSE** exhibits slight superiority over **Original_withSE**, underscoring the significance of considering class imbalance in the analysis.

Reference

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M.: Optuna: A Next-generation Hyperparameter Optimization Framework, Proceedings of the 25th ACM SIGKDD

International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 10.1145/3292500.3330701, 2019.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y.: LightGBM: a highly efficient gradient boosting decision tree, Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA2017.

Others minor comments:

1) Figure 3 (d)-(j). It seems all models fall outside the gray uncertainty envelope related to the 17 models. AutoML also represents an ensemble of ML models. In addition to plotting the ensemble mean from AutoML, can you develop an uncertainty envelope based on the AutoML ensemble.

Regarding Figure 3, as reply in Hao Chen's comments: we would first like to clarify that the gray bands represent the predictions of 13 PTF models, which explains why the ensemble class of models, and AutoML-Ens in particular, does not fall within this range of bands.

Here, we appreciate your suggestion, which we find to be a very good idea. However, we have made a slight adjustment to it. In addition to the 17 existing predictions, we also have included the ensemble predictions of a single ML algorithm for evaluation. Note that a particular class of ML algorithms encompasses several different variants (such as models listed in ***Table 1***), and we have selected the one that demonstrates relatively good classification accuracy to represent this specific ML algorithm family. Based on this, instead update ***Figure 3***, we created a new ***Figure S1*** in our *Supporting Information* by introducing a new band that represents the performance range (mean±standard deviation) of 6 individual ML algorithms. Note that for consistency throughout the study, we employ the classifiers derived from **Original_withSE**, as previously mentioned.

The results depicted in ***Figure S1*** demonstrate substantial variation in the ensemble prediction accuracy of individual ML models across specific environmental gradients, as evidenced by a wide range of R^2 or RMSE values. This again highlights the importance of carefully selecting the appropriate ML model for specific targets. Moreover, AutoML-Ens, as an ensemble of these ML models, exhibits prediction accuracy that, although falling within the range of ML-based ensemble accuracy, remains relatively high. This underscores the advantages of employing an ensemble ML approach in this particular case.

"Figure S1 presents a detailed prediction comparison of 13 individual PTFs and 6 individual ML algorithms along the environmental gradients."

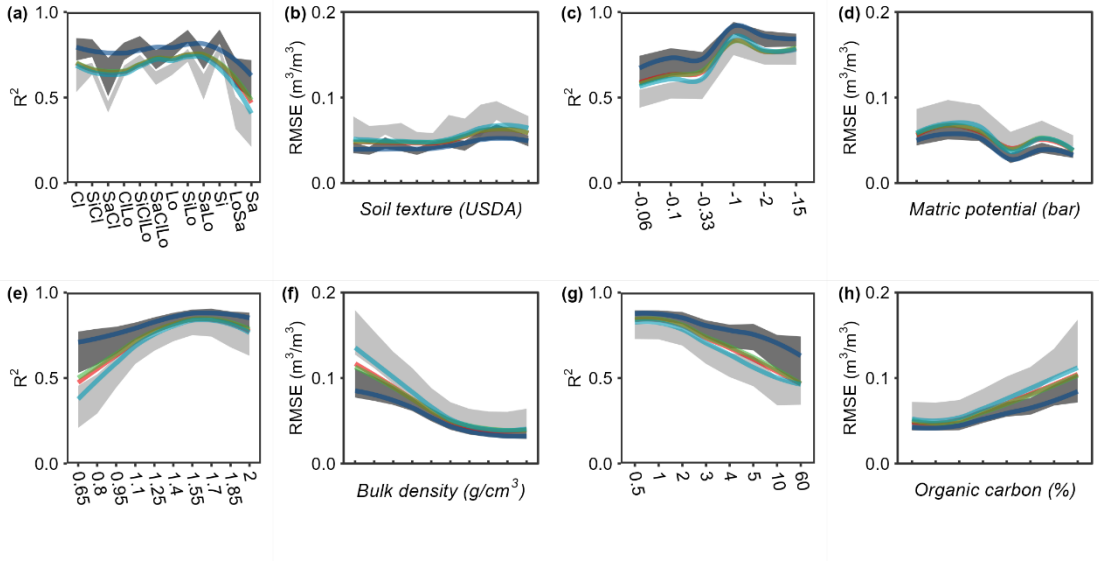

Figure S1. *R^2 (**a**, **c**, **e**, **g**) and RMSE (**b**, **d**, **f**, **h**) when the moisture content estimates of different ensemble approaches were compared with observations (including all training and testing data) under various environmental conditions (6 variables, among which, the content of sand, silt, and clay was expressed together in terms of USDA soil texture classes) that were represented by predictors for AutoML-Ens. The light gray band denotes the uncertainties calculated as the mean±standard deviation of the R^2 (or RMSE) values of the 13 selected PTFs. The dark gray band denotes the uncertainties of the 6 individual ML algorithms.*

2) Figure 7. Both AutoML-Ens and STIC use very similar reddish color. Can you make a stronger contrast?

Modified ***Figure 7***:

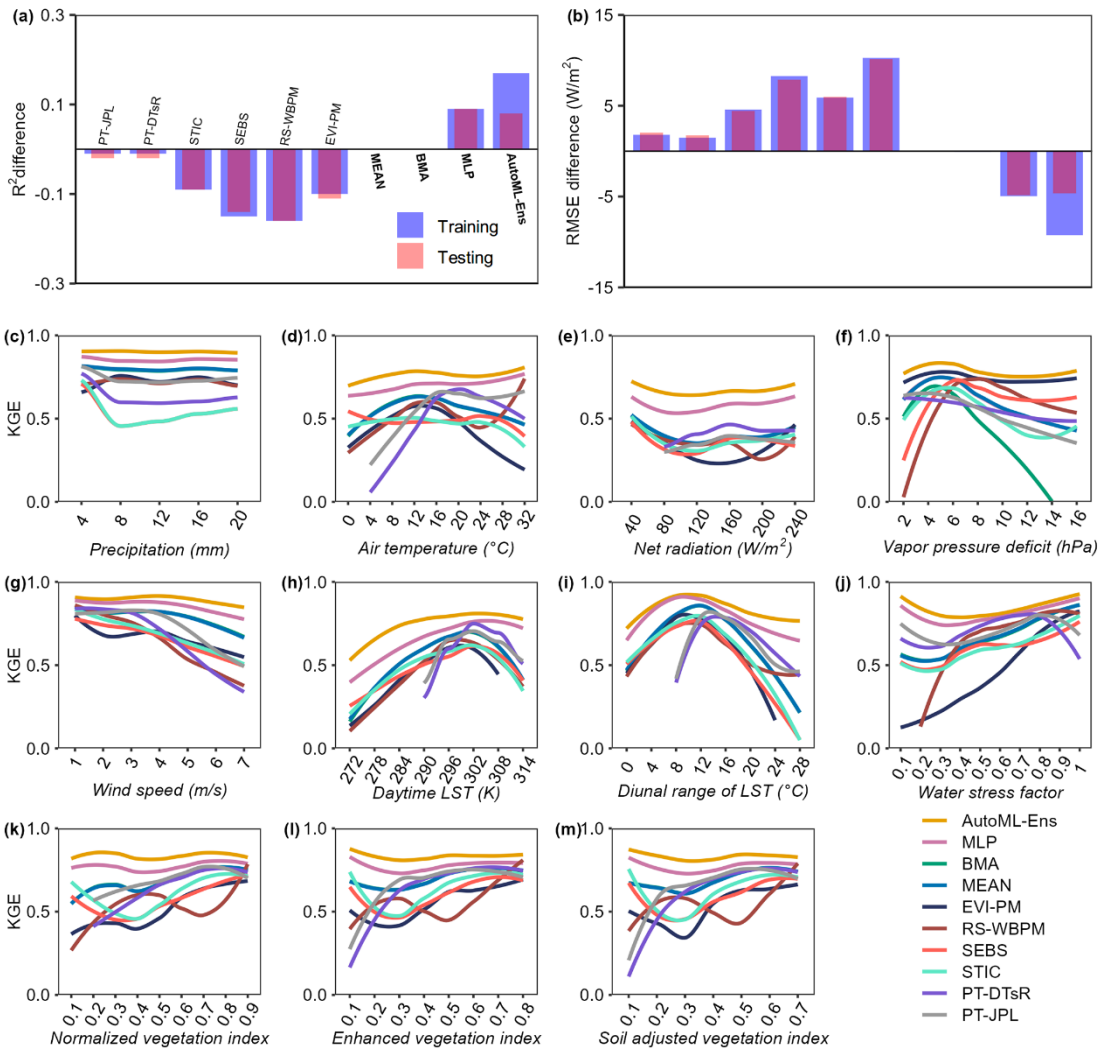

Figure 7. *Difference in performance metrics (R^2 (**a**) and RMSE (**b**)) between MEAN and all 10 models, including six physically-based ET models and four ensembles (in bold font) for training and testing data. A positive R^2 or negative RMSE difference means that the model yields a larger R^2 or smaller RMSE, indicating the better performance of the model than MEAN (considered as the benchmark). KGE (**c**-**m**) when ET estimates from the 10 models were compared against observations (including all training and testing data) under various environmental conditions (11 variables) that were represented by predictors for AutoML-Ens.*

Once again, we appreciate your hard work earnestly and hope that the explanations and modifications will meet with approval. If you have any other questions about this paper, please don't hesitate to let us know.

In the name of all co-authors, with kind regards.