General comments:

This manuscript takes on the important task of improving N2O emissions predictions from grasslands using machine learning, crucially through the use of a large multi-site dataset. This is significant development from the earlier work on this topic. However, several aspects of the configuration of LSTMs and the preprocessing of data require clarification or improvement in order to provide a fair and transparent evaluation of the model's performance. Specifically, I have concerns regarding potential data leakage in the train/test splitting strategy, and the ambiguity regarding the use of past target variable values as inputs.

These details may seem esoteric, but without them, it is impossible to discern whether the RMSE achieved here is impressive or disappointing. That being said, I feel that these changes can be implemented—or at least clarified—relatively easily, and are necessary to validate the study's conclusions. I also strongly encourage the authors to make their model code and data available via a public repository. This would resolve the interpretability issues raised in this review and ensure the reproducibility of the study.

I believe that with these methodological issues resolved, this study could make valuable contribution to the field of environmental modeling, with broad potential impact.

AC: Thank you for acknowledging the study as significant development and for your constructive comments and suggestions. We addressed your concerns regarding potential data leakage in the train/test splitting strategy. Your concern that any past target variable values would have been used as inputs are a misunderstanding, this was and is not the case which we clarify below. In order to ensure transparency, we now make the model and specifications available via a public repository given in Code and data availability section.

Specific comments:

L105-114: Equation 1 suggests that you have manually included four additional time-lagged features of temporal drivers for each of the prior four days. If I have correctly interpreted this, it is unclear to me why you have manually created these time-lagged features given that you are using an LSTM model. The LSTM model already contains memory cells, forget-gates, an attention layer, and other architectural features which remember past events and learn which are important. As an LSTM model already has access to previous time-steps, forcing the model to use manually-created ones may interfere with the model's attempt to learn and forget appropriately. Please clarify whether the 't-1' etc indicate manually included time-lagged features, and if so, justify your use of them. It would be helpful to explicitly state the shape of the 3D tensor used in the LSTM (N_Samples, Time_Steps, N_Features).

AC: We did not manually include time-lagged features to interfere with the LSTMs internal operations, indeed we had to do preprocessing. The Equation 1 is only explaining how the LSTM walks on the data sequences and what is needed to predict one days of N2O emission: When the sequence length is set to 5, the LSTM processes the input time series using sliding windows of five consecutive time steps, provided that the total sequence length exceeds this value. For each input window, the LSTM processes one shot of five time steps **independently** and updates its hidden state and cell state through the recurrent computations of the network. The input, forget, and output gates also function within a window of length 5 and regulate how information from the current input and the previous hidden state is incorporated, retained, or discarded in the cell state. The model could also be trained in batches. In this case, the model receives several input sequences (each of length five) in **parallel** during each training step. The forward pass is performed **independently** for each sequence within the batch, producing a set of predictions. The loss is then computed across all sequences in the batch and averaged (or aggregated) to obtain the batch loss. So, if you want to evaluate the model with new data for 1 day, you will need to enter the values mentioned in Eq. 1.

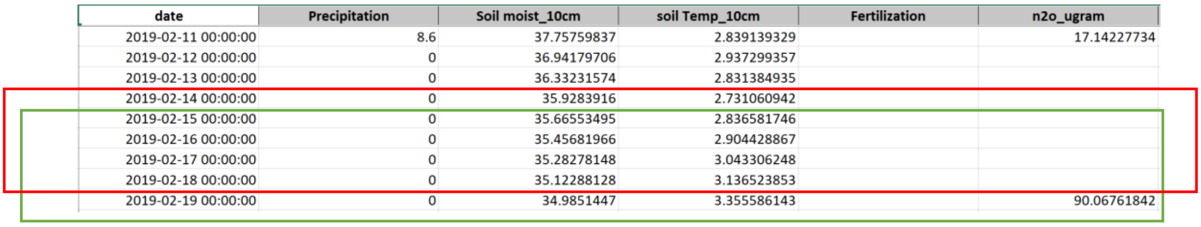
(N_Samples, Time_Steps, N_Features) in our data: (14500,5,4)→ in which 4 is the number of temporal features --> How LSTM looks into this regardless of the N-Samples: a length 5 time shot at each round.

we improve the description L109:L110 to provide more clarity: „To estimate the daily N_2O-N emission, the sequential LSTM model walks through the data sequences and in each round processes a subsequence of length 5 which include the information as Eq. 1 for estimating average daily fluxes: "

On the other hand, detailed preprocessing had to be applied on the daily data of each site to ensure we have continuous sequences of given daily value, which we explain further.

Firstly, we had to preprocess the large sequence of daily data for each site to remove the days including gaps in measurements of input variables such as soil moisture. This leads to smaller uncontentious subsequences. For instance, only 300 days of data rows for one year might be have measurements for all the input features, causing gap days in between.

Secondly, we had to consider the fact that there are also many days with gaps in N_2O emissions value. For instance, only for 100 days in one year the N_2O observation were available for a site. For each of these days, we needed the 5 subsequent days to be available after gap removal in first stage. Therefore, the sequences had to be preprocessed again to extract the sequences of length 5 for which we have one N_2O emission measurement and the input drivers for 5 subsequent days. Please see a part of the data in Fig. 2 below as an example of how the gaps in N2O emission dataset looked like:



date	Precipitation	Soil moist_10cm	soil Temp_10cm	Fertilization	n2o_ugram
2019-02-11 00:00:00	8.6	37.75759837	2.839139329		17.14227734
2019-02-12 00:00:00	0	36.94179706	2.937299357		
2019-02-13 00:00:00	0	36.33231574	2.831384935		
2019-02-14 00:00:00	0	35.9283916	2.731060942		
2019-02-15 00:00:00	0	35.66553495	2.836581746		
2019-02-16 00:00:00	0	35.45681966	2.904428867		
2019-02-17 00:00:00	0	35.28278148	3.043306248		
2019-02-18 00:00:00	0	35.12288128	3.136523853		
2019-02-19 00:00:00	0	34.9851447	3.355586143		90.06761842

Figure 2. a snapshot of dataset showing the gaps in the data. The green part of the sequence could be considered and used for LSTM, while the red subsequence is useless due to unavailability of the N_2O value.

Finally, after gap removal the smaller subsequences created for all sites had to be shuffled together, to ensure at each batch the LSTM sees the data from different sites. If you don't do this shuffle and just stack the sequences of all sites together, you'll see a much lower performance. This was a very important aspect that we discovered after investigations.

In conclusion, we didn't create individual time lags, we just had to shorten the big sequences of data to smaller length of 5 because of removing gaps and shuffling all sites.

L120-130: After a careful read, I have interpreted the authors' data splitting as follows. There are two subsets, one training and one testing subset. All of the Rottenbuch-in-Fendt data was siloed to the testing dataset, while all Neustift-in-Neustift data was used for training. All data from a single year of the Chamau site was withheld for the testing dataset, while all previous years were included in training. The rest of the data appears to be split at the treatment level, with all three replications and experiment years of a given treatment either sent to training or testing. This is somewhat different from the description given in the abstract, as the test set included not just a holdout site and withheld years as described, but also parallel years of separate treatments: "The trained LSTM model showed strong predictive performance (RMSE of 18 $\mu gm{-2}\ h\ {-1}$, and Relative RMSE of 270%) when evaluated on a test set that included both data from an independent soil and withheld years from training procedures." Importantly, withheld years and withheld sites make up only 320 and 259 data points of the total 14591 measurements in the testing set (~2% each).

I firstly want to commend the authors for clearly taking data handling seriously and devising a more rigorous method of splitting than a simple randomized split. I also appreciate that authors have provided a loose description of criteria and thought that went into dividing datasets into testing and training and their desire to represent a variety of soil, environmental, and management properties in both groups. That being said, I do still have concerns about the potential for data leakage given the methods described. Most crucially, I believe it generally best practice when working with sequential data to avoid having data from the same locations and time periods represented in both testing and training pools, even if there was a difference in management technique (e.g. extensive/intensive). As most experiments use one-factor-at-a-time, separate treatments typically experience mostly the same conditions at the same time, including weather and other factors which may affect biogeochemical systems and which may not be captured by model inputs (Zhu et al., 2023). Thus there is often a similarity in flux patterns over time, where factor differences may lead to a difference in peak height etc, but the overall shape of flux over time is similar. Thus when split across testing and training datasets, information about flux patterns and their drivers may leak from a parallel treatment, giving the model an unfair advantage during predictions.

In my view, how the authors have handled data from the Chamau site and Rottenbuch-in-Fendt are both strong examples of data splitting. From the Chamau site, the model had the opportunity to learn from previous years' data during training but was then evaluated on a separate year, with no opportunity to 'cheat' and gain information about events and effects that occurred during that year from other treatments. The Rottenbuch-in-Fendt is the strongest example of a true holdout testing the model's generalizability, as the authors point out. Arguments about the model's generalizability thus would be stronger based upon the model's performance on these two subsets. Aggregate RMSE and other metrics could be reported independently for withheld sites and withheld years, without the overwhelming influence of additional data included in the test set where parallel treatments are represented in training data. Authors may also wish to consider adding additional sites/years to these pools, given the small quantity of data handled in these manners.

AC: Considering your details discussion, also a similar concern about data leakage from other reviewers, we changed the strategy to apply your feedback. The train/test split is now ensuring to assess temporal generalizability using a hold-out set from Fendt-in-Fendt and Graswang-in-Fendt, in which all intensive and extensive treatments of them in 2020 are hold out from training, similar to Chamau. Cross-site test set to assess the spatial generalizability is Rottenbuch in Fendt as it was before. We corrected this in the paper in L126:L139

We now report error metrics separately on both the temporal and the spatial test sets in section 3.2 and table 5, and report it like this also in the abstract.

L140-142, L179-185: The use of log-transform and the loss function selected is an interesting combination of model techniques. I think it would be useful to provide a greater discussion on the effect of this combination on model behavior, given a non-statistical audience and a highly non-traditional loss term. The log transformation compresses the data distribution, forcing the model to minimize relative prediction errors on a relative, rather than absolute, scale. Then, the loss function penalizes prediction errors *exponentially,* placing a massive (approaching infinite) influence on any large errors. Thus the cost of a single large error will easily outweigh a thousand small errors. The result is a highly risk-averse model (avoids large errors at all cost), and which prioritizes consistency (always predicting within the same order of magnitude), in contrast to precision and accuracy on the bulk of data with poor predictions on outliers. It also likely forces the model to pay close attention to hot moments. It's an interesting and potentially effective strategy depending on the goals, but I feel the biogeochemist audience would benefit from more explicit discussion of these effects.

AC:

We chose this approach due to the following reasons:

(1) Applying the Log-transformation on highly right-skewed data is a crucial step before applying a model: The log transformation reduces skewness and stabilizes variance, resulting in better-behaved gradients, faster convergence, and more stable and effective training (West 2021; Bishop, 2006, Ch. 5; Hastie et al., 2009, Sec. 2.4; Box & Cox, 1964). In highly skewed data, rare extreme values dominate the gradient in gradient-based models such as neural networks, which is known to impair optimization and slow convergence in deep neural networks (Goodfellow et al., 2016, Ch. 8). Variance-stabilizing transformations such as logarithmic scaling are commonly used in recurrent neural forecasting models to address strong skewness and scale heterogeneity (Salinas et al., 2020; Smyl 2020). Without log-transformation, the extreme values create high gradients, dominate the training, and cause the model even not learn the relationship in the data but focus on the rare 5% of the extreme events only. Our goals is not to only predict large extremes, but learn the N2O emission pattern, shown in both low and high fluxes. We briefly mentioned the importance of log-transform in L150:L153

(2) Loss function: The log-transform helps training stability which is a must in this problem as discussed, but we still needed a proper loss function for the specific problem. Our goal is to have a model optimized to learn both high and low emissions, rather than only extremely high values. A weighted loss function ensures a trade-off between the target transformation for better convergence, and high performance of the model on high emissions. In addition, the exponential loss function is executed on log-transformed data, making it not only penalize large errors on high values, but also on low fluxes, which ensures the model learns both.

To ensure more details about this we add extra description in methods section as well a discussion: L339:L342

L164-175: LSTM models can be configured in three ways: (1) using ground-truth target values from previous time-steps as inputs (teacher-forcing or closed-loop), (2) using the model's own past predictions as inputs (autoregressive or open-loop), or (3) relying solely on environmental drivers without including any version of the past target variable as an input feature in the 3D tensor. I could not find any mention of whether some version of previous flux time-steps were included in the 3D tensor. This is a crucial detail, as flux is often highly autocorrelated, yet in practical scenarios, actual past flux values will not be known. Using teacher-forcing is thus appropriate during training to accelerate convergence, but inappropriate during model evaluation on testing datasets, as this would provide highly useful but practically impossible information. A discussion of these distinctions and how precisely the LSTM model was set up in both training and testing phases is immensely needed in order to assess the model's true predictive power.

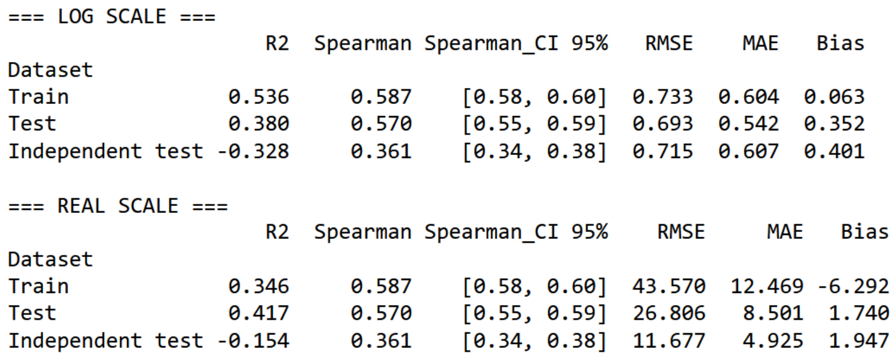
AC: Thanks for spotting the lack of clarity in our description. Option 3 was used for this paper, relying solely on the environmental drivers in both training and test phases. We add this in lines L118-L119 „The implemented sequential LSTM models in this study rely solely on the temporal and static input features shown in Eq. 1 in both training and testing phases, and no information from the N_2O emissions on previous days is used for modelling the N_2O emission of the current day.“

L222-224: I interpret this as you have evaluated two different hyperparameters for the n_steps/timesteps parameter of input_shape argument of the LSTM layer, those being 5 and 10. This sets a hard limit on the number of days your model can "look backward" at previous events. I am curious why such short timeframes were considered, especially given that a fertilization input may have an impact on soil nitrogen content and thus denitrification rates for many weeks, particularly if the fertilization happens to occur during a dry period. I note that you explain in the discussion that emissions nearly always peak shortly after fertilization, thus reducing the benefit of longer sequence inputs. However, I also notice from Figure 4 that moderate peaks often occur long after fertilization- much longer than the 10 days maximum evaluated. I am curious whether longer input sequences may improve predictions on such moderate peaks, for example if soil nitrogen is still moderately elevated a month after fertilization and a significant precipitation event occurs.

This also compounds my confusion regarding the aforementioned time-lagged features. The input_shape argument of an LSTM layer already defines the ceiling of how many previous time-steps the model is capable of considering, and attention and forget gates are used to determine which data among this pool to actually focus on or forget.

Thank you for the valid point. Initially, we started with sequence length 5, then we checked if longer sequences had an effect on model performance. Several points led to the decision to remain with shorter sequences in the publication:

(1) Changing the sequence length to 10 (and hyperparameter tuning) or 20 didn't bring significant improvements to the model so we didn't investigat longer sequences. We added one line to Table 4, so that the audience knows sequence length 20 was also tested. Please look at the Table 1 below showing how the model acts on samples of test set when we include a longer sequence of length 20:



```
=== LOG SCALE ===
                     R2  Spearman Spearman_CI 95%   RMSE    MAE   Bias
Dataset
Train             0.536     0.587    [0.58, 0.60]  0.733  0.604  0.063
Test              0.380     0.570    [0.55, 0.59]  0.693  0.542  0.352
Independent test -0.328     0.361    [0.34, 0.38]  0.715  0.607  0.401

=== REAL SCALE ===
                     R2  Spearman Spearman_CI 95%    RMSE     MAE   Bias
Dataset
Train             0.346     0.587    [0.58, 0.60]  43.570  12.469 -6.292
Test              0.417     0.570    [0.55, 0.59]  26.806   8.501  1.740
Independent test -0.154     0.361    [0.34, 0.38]  11.677   4.925  1.947
```

Table 1. Statistical metrics on a model trained on sequence length 20.

(2) Longer input sequences (length **X**) increase the amount of data required for the model to function effectively, which makes evaluation at a new site more challenging. This is because the LSTM requires **X consecutive days of input** to estimate the **N_2O flux for a given day**. As the required sequence length increases, the amount of usable data decreases due to gaps in input variables such as soil moisture and temperature. Consequently, we chose to keep the sequence length—and thus the model's data requirements—as small as possible to ensure robust performance. The potential benefits of longer sequences (e.g., **30 days or more**) will be investigated in future work using an expanded dataset.

we also added a statement in line L401:L403 in the Implications and limitations section to ensure using a longer sequence is known as a potential future work: „Firstly, expanding the dataset to encompass more geographically and climatically diverse sites, along with a wider range of management practices and legacy effects on N_2O emissions, would allow for the development of more generalized N_2O emission models for grasslands across different agroecosystems".

L235-236: I see that RMSE for the holdout set are given individually for timeseries in Fig 4, but I would appreciate an aggregate performance evaluation for the Rottenbuch-in-Fendt holdout listed here for comparison's sake.

AC: We separated now according to spatial and temporal test set and provide the performance evaluation in our updated table 5.

L269-271: Clarification is again needed here as to whether time-lagged features were directly represented as model features. If the importance of previous time-steps of features were determined by using standard PFI, which then varied the values of e.g. feature "soil_temperature_t-1" and ultimately assigned it an importance value, then this method would be invalid. In the case of using explicit time-lagged features, previous values of features are represented in the model in two places: the engineered feature, as well as the internal Hidden State of the LSTM. Even if the time-lagged feature were altered by PMI, the model would still have access to the true value in its hidden state, and thus the model could simply ignore the shuffling by PFI.

If this is a misinterpretation and time-lagged features are not explicitly represented, then I would like to see more detail on how PFI was adapted to assess the importance of previous time-steps given the LSTM architecture.

AC: We addressed this misinterpretation in the reply to your other comment above, time-lagged features are not explicitly represented to LSTM.

To calculate PFI for individual time steps: we got each sequence of 5 days generated with preprocessing, and shuffled the features only in that sequences. For instance, in each sequence of length 5 we only shuffled the precipitation within that single sequence, which reflects shuffling the time steps. So, if you had precipitation only in first day for 18mm and not in other days, then this value will move to another day randomly, for instance day 3 and in this way we could see how the models estimation on this sequence change and get the relative importance from PFI.

After getting comments about the PFI for time lags, and considering different opinions on this topic between data scientists which doubts whether this is a proper approach or might cause misunderstanding about model's results, we decided to remove this approach and solely rely on Fig. 6 as an standard way which the PFI is calculated in other scientific publications. We believe that such a method needs very wide discussion about all the aspects and the implementation method, which its out of the scope of this paper.

L349-353: Given that no process-based model was tested against the LSTM model, I question the need for this whole section of comparing with process model, which is not the scope of this study. I am skeptical that any meaningful comparison of predictive performance metrics can be made across different datasets, especially given the extreme temporal and spatial variability of N2O which translates to major variability in data distributions. Some datasets are simply easier to predict due to lesser influence of hot moments, and thus comparisons of models evaluated on different datasets should not be made without at least acknowledging their tenuousness.

AC: We removed the section 4.4 according to your feedback and the feedback from other reviewers.

L361-365: Again, considering that data exists for process-based model predictions at the same site (and year?) as one of the modeled sites in this study (Chamau), I would prefer that authors limit their comparisons of LSTM vs process-based simulations where these comparisons can be made fairly.

AC: We removed the section 4.4 according to your feedback and the feedback from other reviewers.

Technical corrections:

L24. Spell out full word ‘approximately’ Done.

L30. ‘2’ subscript Done.

L237: “Having relative RMSE higher than 1 is a known effect of the imbalanced dataset.” Citation needed.

AC: We removed RRMSE, as we got many feedbacks and investigated further that its not a proper metric for this problem with highly skewed data. We instead use a set of metrics in Table 5 to better interpreted the performance: section 3.2.

Other minor comments:

Table 1 and 2: Dataset size - clarify if this is daily flux observation numbers

Done.

L81: Better to say “flux” and not “flux rate” Done.

Fig 1: It would be helpful to include flux distribution for the test data dataset too. Done, please see Fig 1.

Table 5 and associated discussion: The table does not need to be in the main manuscript and can be presented as supplementary data.

AC: We actually provide the details to those related literature there to avoid huge amount of text. One would need the paper to follow the studies.

L297-303 sounding like the authors are criticizing the previous studies and can be shortened. These are some of the initial studies on this topic and should be acknowledged accordingly with limitations that the current study is addressing.

We improved this section, L316:L325

Fig 6: While feature importance was assessed separately for each site in Fig 6, the discussion on any variation in variable importance across sites is weak and contributes little to the understanding of differential N2O dynamics. Also, why time-lagged features were not included in Fig 6 analysis.

Please look into section 4.3 feature importance analysis, we added few more lines on this. Time lagged features were not included in Fig. 6 as this is the standard and general approach for PFI to be implemented on the features, regardless of time-lags. Fig. 7 which used to include time-lags due to the reasons mentioned in previous comments.

References:

Goodfellow, I., Bengio, Y., & Courville, A. (2016).
Deep learning. MIT Press.

Ghosh, T., Anand, S., Nannewar, R. G., & Nagaraj, N. (2025). Deep Learning for Short-Term Precipitation Prediction in Four Major Indian Cities: A ConvLSTM Approach with Explainable AI. *arXiv preprint arXiv:2511.11152.*

Hastie, T., Tibshirani, R., & Friedman, J. (2009).
The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Box, G. E. P., & Cox, D. R. (1964).
An analysis of transformations. *Journal of the Royal Statistical Society: Series B*, **26**(2), 211–243.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020).
DeepAR: Probabilistic forecasting with autoregressive recurrent networks.
International Journal of Forecasting, 36(3), 1181–1191.
https://doi.org/10.1016/j.ijforecast.2019.07.001

Smyl, S. (2020).
A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting.
International Journal of Forecasting, 36(1), 75–85.
https://doi.org/10.1016/j.ijforecast.2019.03.017

Raiaan, M. A. K., Sakib, S., Fahad, N. M., Mamun, A. A., Rahman, M. A., Shatabda, S., & Mukta, M. S. H. (2024). A systematic review of hyperparameter optimization techniques in Convolutional Neural Networks. *Decision Analytics Journal, 11*, 100470.
https://doi.org/10.1016/j.dajour.2024.100470

West, R.M. (2021) Best Practice in Statistics: The Use of Log Transformation. *Annals of Clinical Biochemistry: International Journal of Laboratory Medicine*, 59, 162-165. DOI: 10.1177/00045632211050531.