The authors compare four different strategies for incorporating static features into global deep learning models. They show that the repetition method generally achieves the best performance but they are less computationally efficient. They also state that the selection of static features is more important than the choice of integration strategies. The manuscript is generally well-presented.

We thank the reviewer for the positive assessment of the manuscript and for the constructive comments, which we believe help to further improve the quality and clarity of the paper. In the following, we address each point raised by the reviewer in detail.

However, I still have the following concerns.

In the Table 1 of Heudorfer et al. (2023), it shows that the time series features were derived from past groundwater level time series until 2011. The training, validation, and test periods for this study are 1991-2007, 2008-2012, 2013-2022, respectively. If the authors directly adopt the time series features from Heudorfer et al. (2023), there might be data leaking issue during validation.

We thank the reviewer for pointing this out. Although the feature definitions follow Heudorfer et al. (2023), all time-series-based static features were recomputed in this study using only information available up to the respective training period. In particular, no information from the validation or test periods was used in the feature derivation, thus preventing any data leakage. We will clarify this explicitly in the revised version of the manuscript.

Also, while the authors have provided references for the time series features, it is nice to list the time series features in the main text or appendix for readability.

Thank you for that constructive suggestion! For improved readability, we will add a complete list of the time-series-based static features to the Appendix.

For the conditional model, it is unclear to me how the output of this layer is split and used to initialize the hidden and cell states of the first LSTM layer in the dynamic branch (Line 167-168). Does it mean that the output is directly used as the initial condition of the hidden and cell states? Please clarify.

Thank you for the comment. We agree that this was not sufficiently clear. Yes— the static branch outputs a vector of length $2H$, which is split into two tensors of length $H$and directly provided as the initial hidden and cell states (initial_state=[h0, c0]) of the first LSTM layer in the dynamic branch. We will clarify this in the revised manuscript.

In Line 176, the authors mentioned that there are 10 model initializations. Does it mean that the authors train 10 global models with different initial conditions?

Yes, correct. We will clarify that the ensemble consists of 10 independently trained global models with identical architecture, data splits, and hyperparameters, differing only in their random weight initialization.

For the results, the authors that the repetition model performs the best, but they show the results for the 256-neuron model for the repetition model and the 128-neuron model for the other models. It is hard to identify whether the better performance is due to the integration strategy or the more hidden neurons.

We thank the reviewer for the comment. As described in the manuscript, all models were implemented with a baseline LSTM size of 128 units. For the repetition strategy, we additionally tested a configuration with 256 units to account for the increased input dimensionality resulting from the replication of static features at each time step. Since this larger configuration yielded slightly better performance, only the results of the 256-unit repetition model are reported in the Results section. We hope that this description sufficiently clarifies this point.

Specific comments:

Explain labels/legends in Figures 2 and 3.

Thank you for pointing this out. We will improve the figure captions and legends to explicitly explain all labels and symbols in Figures 2 and 3 in the revised manuscript.