

We thank the Reviewer for the helpful comments and suggestions. We hereby address them individually. In this document we indicate the Reviewer's comments in *italic dark grey*, while text that was changed in the paper in *blue*.

This paper presents a multi-scale, hydraulic-based graph neural network model (mSWE-GNN) designed to enhance flood modelling. Building on previous models like SWE-GNN, the authors introduce novel elements such as rotation-invariant inputs, ghost cells for boundary conditions, and a multi-scale architecture to simulate flood events over variable topographies and unseen meshes. The paper includes extensive theoretical explanations and benchmarking evidence that demonstrate the model's effectiveness across different flood scenarios. Overall, this research is timely and relevant, contributing meaningfully to the field of flood modeling with machine learning. I support the publication of this paper following some revisions.

General comments

The model presented in this paper, mSWE-GNN, demonstrates impressive capabilities in simulating dike-breach flood scenarios specifically. However, its applicability appears limited to this particular type of flood. Given the diversity of flood types—such as fluvial and pluvial floods—extending the model's capabilities or discussing potential adaptations for these other scenarios would significantly enhance its relevance and utility. A broader application across flood types would open up further opportunities for practical deployment and highlight the robustness of the proposed multi-scale architecture in different hydraulic contexts. Including a discussion on how the model might be adapted for other flood types would be a valuable addition.

However, the paper limits its application to dike-breach floods and does not explore other flood scenarios. Adding a section discussing potential adaptations for other flood types, such as fluvial or pluvial floods, would strengthen the paper's relevance.

We thank the Reviewer for the valuable point addressed here. We added a paragraph in the discussion section that indicates how to deal with other types of floods. This is written in lines 390-394 as:

“While the current model framework can work for dike-breach floods, we did not evaluate it for other types of floods. For river and coastal floods, the model should also perform well without modifications since the inputs are of the same type as dike breach floods, e.g., upstream discharge hydrograph or sea water levels. On the other hand, pluvial floods require precipitation as a further input. Assuming rainfall as a spatially distributed variable, it could be added as a dynamic forcing; this could work in a similar way as for static features, but changing at each time step, independently of the predicted output. For urban floods, the drainage system should also be included.

Similarly to numerical methods, this could be achieved by coupling the overland flow predicted by the mSWE-GNN with a 1D model for the sewers, potentially using another learned GNN as demonstrated in Garzon et al., 2024.”

References:

Garzón, A., Kapelan, Z., Langeveld, J. and Taormina, R., 2024. Transferable and data efficient metamodeling of storm water system nodal depths using auto-regressive graph neural networks. *Water Research*, p.122396.

**Additionally, the inclusion of a mass conservation term in the loss function, though discussed in the appendix, warrants further investigation, particularly regarding its influence on model stability.*

Discussion on Physical Constraints (Appendix B): The addition of mass conservation loss is an interesting consideration. However, it would be helpful if the authors could provide further analysis or reasoning on why this term, although theoretically sound, did not yield a consistent improvement in testing. This section could benefit from a comparison of other physical constraint approaches in neural networks.

We thank the Reviewer for the feedback. We gave an explanation of why we believe that the additional loss term did not improve our performance in lines 449-452:

“The reason why losses slightly improve might be because the added loss term depends only on the predicted water depth, so it enforces that value to be more precise. However, the conservation loss acts globally for each time step, instead of locally. So, the model cannot correctly improve the spreading of the flood but only the absolute values of total water depth.”

and in lines 456-458, in comparison with PINNs:

“One motivation is that the loss we employ does not rely on auto-differentiation in the same way that PINNs do. We also evaluate it globally, rather than at individual points as in PINNs.”

We believe that there are no additional main points that could cause justify the absence of improvements, as we were also expecting the loss term to improve our results.

Regarding other physical constraint approaches in neural networks, there are three main ways to add them: 1) using PINNs, which rely on predicting the variables (and their derivatives, by means of auto-differentiation) present in the underlying partial differential equation (PDE) and minimize the corresponding loss in several collocation points; 2) using a regularization term in the loss function that works similarly to the

PINN's loss, but without using the original PDE; 3) embedding physics in the model's architecture.

The mSWE-GNN, thanks to the computational structure similar to numerical hydraulic models, enforces locality of flood propagation, thus providing the third type of physical constraint defined before. We avoided PINNs and decided to explore softer physical constraints since we would have had to change the model's inputs and outputs so that the auto-differentiation could provide us with the estimates of the derivatives with respect to the predicted target variables, needed for the shallow-water equations. Because of this, adding PINNs would go out of the scope of the paper, which was to show that adding multi-scale information improves the model performance, that we can include a wide range of boundary conditions via ghost cells, and that rotational-invariant inputs improve generalization. Moreover, PINNs are typically designed to solve a given physical problem for a single set of boundary and initial conditions. This approach does not inherently generalize across varying conditions, which is a prerogative of our work. We included this justification in lines 470-475 as:

“This in part contradicts the idea of physics-informed neural networks (PINN), according to which adding a physical loss term improves performance (Raissi et al., 2019). One motivation is that the loss we employ does not rely on auto-differentiation in the same way that PINNs do. We also evaluate it globally, rather than at individual points as in PINNs. [Implementing a PINN loss would require adjustments to the model's inputs and outputs to allow auto-differentiation to estimate the derivatives of the predicted target variables. Moreover, PINNs are typically designed to solve a given physical problem for a single set of boundary and initial conditions, thus limiting the model's capacity to generalize across varying conditions, which is a prerogative of our work. Although we did not adopt this approach here, it could be explored in future studies. Notably, our loss term is independent of ground-truth data, making it a possible self-supervised loss that could be explored in future works.](#)”

Specific comment

L89-92: For clarity and reproducibility, it would be beneficial for the authors to provide additional detail on the criteria for mesh size. Specifically, a brief explanation of how mesh resolution is chosen and classified as "too small" would enhance reproducibility.

We thank the Reviewer for pointing out this concern. While addressing it, we also noticed a writing mistake, since “mesh edge” should be “flow edge”, which are defined in the MeshKernel library as “the edges connecting faces circumcenters”. Hence we corrected the corresponding sentence and also expanded on the definition of small mesh and mesh resolution.

Lines 83-86 now read as:

“For the same numerical constraints, after the orthogonalization, all elongated elements get removed, resulting in a mixture of triangular and quadrilateral elements. We define elongated elements as those whose line connecting barycentre and edge middle points is 0.1 times smaller than the other lines in the same element.”

Eq. 3 & Eq. 4: It may be helpful to specify definitions for h_{di} and h_{si} , as these variables are key in understanding the propagation rule and its application.

We added in line 114 a clarification of what these embedding mean:

“The encoded variables H_s , H_d , and E' represent a higher-dimensional version of the original inputs that is more expressive.”

We also clarified what h_{di} and h_{si} are in lines 135-136 as:

“where $\psi(\cdot) : \mathbb{R}^{5G} \rightarrow \mathbb{R}^G$ is an MLP, \odot is the Hadamard (element-wise) product, $h^{(\ell)}_{di}$ is the embedding of the dynamic inputs at node i and layer ℓ , h_{si} is the embedding of the static inputs at node i , and $W^{(\ell)} \in \mathbb{R}^{G \times G}$ are learnable parameter matrices.”

Boundary Condition Section (2.3): This section introduces ghost cells effectively but lacks explicit examples or visual representation of how they are implemented for various boundary types (e.g., inflows and outflows). Including this could aid in comprehending the technique's generalizability to different boundary configurations.

We clarified figure 3 to better visualize how the type of boundary condition affects the direction of the edge between the ghost cell and the boundary cell. Figure 3 now shows as:

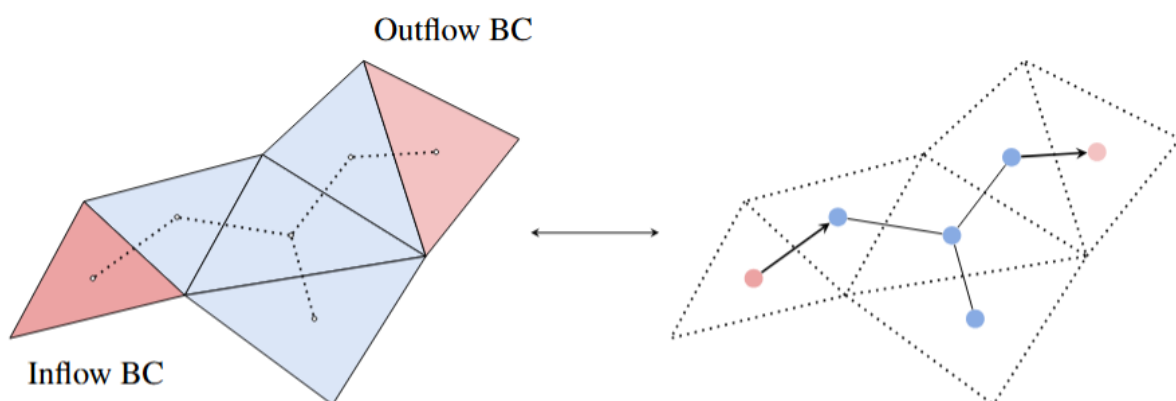


Figure 3

We also changed the caption to better describe this difference in type of boundary conditions as:

“Schematic representation of an arbitrary triangular volume mesh (left) with two ghost cells for inflow and outflow boundary conditions (BC). The ghost cells (red) are added in correspondence of a boundary cell which receives a given boundary condition. In the dual graph (right), a directed edge is added from the ghost cell to the domain cell, or vice-versa, depending on whether the boundary condition is an inflow or an outflow, respectively.”

Fine-Tuning Evaluation (Section 4.2): While the fine-tuning approach is well-motivated, it would be helpful to elaborate on the potential for overfitting given that a single simulation is used for fine-tuning and testing. The authors might explore this limitation and offer suggestions for future validation.

We motivated why we selected only one simulation for training and validation in lines 315-317 as:

“We trained and validated on the same simulation since we wanted to minimize the amount of data needed to fine tune the model. While in principle this might lead to overfitting, it was not the case here. This is probably due to the inductive biases of the model which constrain the model to learning only local dynamics.”

We added another motivation of why we argue this procedure does not lead to overfitting. In fact, the training process considers only a limited number of predicted steps ahead ($H=6$ in our experiments, as reported in line 255), while the full simulation has eight times more. This effect, combined with the previously-mentioned learning of local dynamics, makes it so that the model is forced to learn different dynamics in time rather than overfitting on a single pattern, even if we are using a single simulation.

A simple solution to the overfitting issue could also be to increase the size of the fine-tuning dataset, so that we further limit possible overfitting effects. In case of availability of more simulations at a low computational cost, this is surely a valuable alternative that should also improve the model’s performance.

Accordingly, we further expanded lines 315-319 as:

“We trained and validated on the same simulation since we wanted to minimize the amount of data needed to fine tune the model. While in principle this might lead to overfitting, it was not the case here. This is probably due to the inductive biases of the model which constrain the model to learning only local dynamics. Additionally, the training process considers only a limited number of predicted steps ahead, while the full simulation has many more. Consequently, the model is forced to learn different

dynamics in time rather than overfitting on a single temporal pattern, even if we are training on a single simulation.”

We also modified lines 374-376:

“The mSWE-GNN generalizes well to realistic case studies with as little as one fine-tuning simulation. We expect the model to further improve performance and reduce risk of overfitting by increasing the number of fine-tuning simulations. This result is in line with a similar finding for pluvial flooding where one fine-tuning simulation was enough to help generalization to diverse case studies”

Table 2: To support interpretability, clarify which mesh resolution is referred to in this table. Also, in Table 3, presenting units consistently as $[10^{-2} \text{ m}]$ and $[10^{-2} \text{ m}^2/\text{s}]$ improves clarity and aligns with standard formatting practices.

We clarified in Table 2 that all metrics refer only to the finest mesh by adding the following line to the caption “All metrics refer only to the finest mesh.”.

We homogenized all units in the paper to have a consistent format.

Appendix Figure A1: This figure is valuable in understanding unit discharge modeling, but it may benefit from a clearer representation of errors by using varying scales or alternative visual indicators for over- and under-predictions.

The scale we employed in Figure A1 was automatically generated based on the minimum and the maximum errors found throughout the simulation. We changed these values and corrected accordingly the legend, to better show the errors distribution over time. We then removed “The legends refer to the maximum values throughout the whole simulation.” From the caption. Figure A1 now shows as:

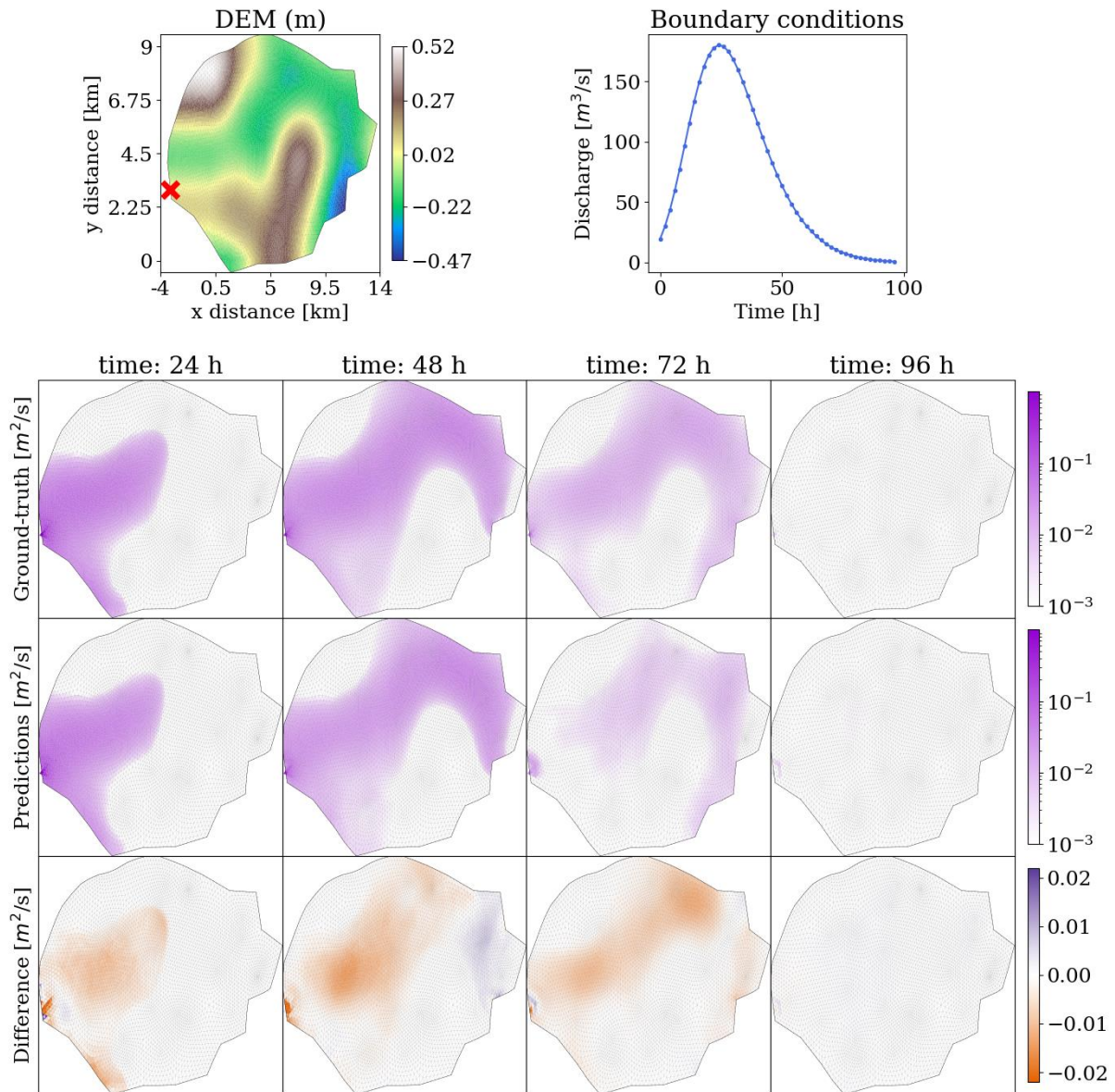


Figure A1. mSWE-GNN's predictions for unit discharges a test simulation from the synthetic dataset. The evolution over time for ground-truth output of the numerical simulation (top row) with the predictions (middle row) are represented using a logarithmic scale to better appreciate the values' distribution. The difference (bottom row) is evaluated as the predicted value minus the ground-truth one and is kept with a standard scale to highlight the use of the logarithmic scale; positive values correspond to model over-predictions while negative values correspond to under-predictions.

Technical corrections

Throughout manuscript: Maintain consistent formatting of units, particularly with spacing between numbers and "m" for meters.

We formatted all units in a consistent way, checking also for the spacing between numbers and units.

L466: “comes from the an increase” should be corrected to remove the redundant article “the.”

We thank the Reviewer for spotting the typo, which we corrected accordingly.

L302 & L305: The phrase “in correspondence of” could be replaced with more direct alternatives such as “occur at” or “align with” for clarity.

We modified the two instances of “in correspondence of” respectively with “occur at” and “at the end”.

Conclusion

In summary, the paper presents an interesting deep learning model with clear advancements over existing approaches in flood modeling. The mSWE-GNN’s ability to generalize and efficiently process multi-scale data is convincingly demonstrated. Nevertheless, a limitation of the current model is its focus on simulating dike-breach floods alone. Given the range of flood types—such as fluvial and pluvial floods—future work extending this model’s application to other types would significantly increase its value and utility. Including a discussion on potential adaptations for different flood scenarios would further enrich the paper, offering insights into the generalizability and versatility of the multi-scale approach.