Reviewer #2

Review of Manuscript

**'A GNN Routing Module Is All You Need for LSTM Rainfall–Runoff Models'**

by H. Mosaffa et al.

Dear Editor,

I have reviewed the manuscript. My conclusions and comments are as follows:

## 1. Scope

The article is within the scope of HESS.

## 2. Summary

In their manuscript, the authors test a series of data-based hydrological models combining an LSTM for catchment rainfall-runoff with a Graph Neural Network (GNN) for river routing. They test four GNN alternatives: Graph Convolutional Network (GCN), Graph Attention Network (GAT), Graph SAmple and aggreGatE (GraphSAGE), and Chebyshev Spectral Graph Convolutional Network (ChebNet). As a benchmark, they use a standard LSTM-only model. Neither of the models receives any observed streamflow as input. They apply their models to a 30-year daily data set of meteorological drivers (precipitation, air temperature, soil moisture), static attributes and river gauge observations of 530 catchments in the upper Danube catchments taken from the LamaH-CE data set. The catchment sizes range from tens to thousands of square kilometers. Analyzing the time-wise out-of-sample testing results, the authors find that all LSTM-GNN models outperform the LSTM- only, that the GAT is the best among the GNN alternatives, and that performance improvements by the GNN addition increase with catchment size and number of upstream connecting nodes.

## 3. Evaluation

Overall, this is a well-designed study on a relevant topic, with conclusions that are supported by the data and that that will have an impact on data-based hydrological modeling. There are only a few adjustments necessary for clarification and to increase readability of the paper.

We sincerely thank Dr. Ehret for the careful reading of the manuscript and for the constructive and insightful comments. Below, we address each comment in detail.

- In the LSTM-GNN approaches, each subcatchment has a unique place in the river network graph. Is each catchment LSTM then jointly trained with all catchments, but each as a single-catchment LSTM, or are they jointly trained as regional LSTM? Please clarify. Also, the LSTM-only, is it trained as a regional model? Please clarify.

Thank you for this clarification request. We clarify as follows:

In the LSTM-GNN framework, each subcatchment has its own dedicated LSTM unit that processes local inputs (precipitation, temperature, soil moisture). However, these individual LSTMs are not trained independently. Instead, they are jointly trained end-to-end as part of the complete LSTM-GNN architecture. Specifically,

• Shared parameters: All 530 subcatchment LSTMs share the same weight matrices and bias vectors (i.e., there is a single set of LSTM parameters applied to all nodes)

• Joint optimization: During training, the loss is computed across all stations simultaneously, and backpropagation updates the shared LSTM parameters based on the collective error across the entire network. • End-to-end learning: The LSTM encoder and GNN routing module are trained together in a unified framework.

Similarly, the LSTM-only model is also trained as a regional model with shared parameters across all catchments. The key difference between the two approaches is that the baseline LSTM predicts discharge at each gauge independently, without any spatial information exchange, whereas the LSTM–GNN incorporates explicit routing via the graph structure.

We added the following clarification to Section 3.1.1:

*"Our proposed model consists of two primary components: an LSTM module for local runoff generation and a GNN module for spatial runoff routing. Each subbasin is represented as a node in the river network and is associated with a local LSTM that processes catchment-averaged meteorological inputs. Importantly, these subbasin-level LSTMs are not trained independently; instead, all LSTMs share a single set of parameters and are trained jointly as a regional model. The GNN component then enables information exchange between subbasins according to the river network topology, explicitly modelling runoff routing. The entire framework is trained end-to-end across all subbasins simultaneously. The overall structure is visualized in Figure 2 and described in the following subsections."*

- Are the GNN nodes placed only at gauge locations, or also at river confluence points? Please clarify.

GNN nodes are placed only at gauged locations.

We have revised the following sentences *"We propose a novel LSTM–GNN model to predict the R–R process by jointly capturing local runoff generation and basin-scale flow routing within a unified framework. In contrast to traditional lumped models that treat the catchment as a single unit, our approach partitions the basin into multiple hydrologically connected subbasins, each represented as a node in a graph, with nodes corresponding exclusively to gauged subbasin outlets."*

- How are the catchment areas for each gauge determined for the LSTM-GNN and the LSTM-only? I assume for the LSTM-GNN it is either the upstream catchment (for a headwater gauge), or the intermediate catchment between the closest upstream gauge(s) and the current gauge, correct? And for the LSTM-only it is the total upstream catchment, independent from the presence of any upstream gauges. If my assumptions are correct, then the catchment-averaged drivers and static attributes will differ between the LSTM-GNN and the LSTM-only. Please clarify. Also, consider including two histograms or cdf's showing the distribution of catchment sizes for both the LSTM- GNN and LSTM-only case.

Both the LSTM–GNN and baseline LSTM-only models use exactly the same catchment delineation. For each gauge, all dynamic meteorological inputs and static attributes are spatially averaged over the total upstream contributing area of that gauge. As a result, the catchment-averaged drivers and static attributes are identical for both models. The only difference between the two approaches lies in the inclusion of explicit spatial routing via the GNN in the LSTM–GNN framework.

We have clarified this point in Section 2 (Study area and dataset):

"Crucially, these dynamic inputs are spatially averaged over the entire upstream catchment contributing to each gauge, providing a single representative value per basin per day."

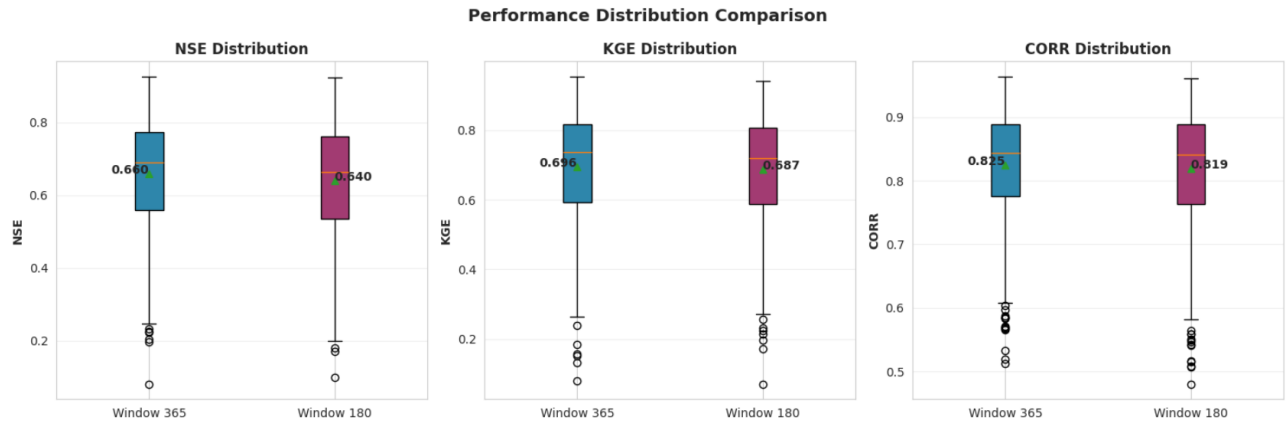
- Up to Vienna, the Danube catchment already covers more than 100.000 km², but the largest catchment included in this study is only 2.500 km² (Line 97). Why? The advantage of LSTM-GNNs over LSTM-only should be even more evident for very large catchments. Please clarify.

  Thank you for pointing this out. The value reported in the original manuscript referred to individual subbasin sizes, not the total upstream contributing area. We have corrected this for clarity.
  The maximum upstream contributing area included in the study is approximately 131,000 km². The manuscript has been revised accordingly to avoid confusion.

- Sequence length is 180 days (Line 130). Why not 1 year, especially if there are snow-dominated catchments included?

  We appreciate the reviewer's question regarding the choice of the 180-day input window. To address this systematically, we conducted a sensitivity analysis comparing window sizes of 180 and 365 days.
  For fair comparison, both models were trained with identical sample sizes (extracted from the 365-day window dataset to ensure no temporal gaps).
  Based on these results, we selected the 180-day window as optimal because the 365-day window provides only a marginal improvement in NSE (+0.02) but requires substantially more GPU memory (~2× increase), limiting batch size and training scalability. For operational deployment and future extensions, the 180-day window is more practical.

  Please note that the sample counts in this sensitivity analysis differ from those reported in the main paper results. This is because the sensitivity analysis required creating a no-gap dataset from the 365-day window to ensure fair comparison, whereas the main paper uses all available samples from the optimal 180-day window configuration.




- For the LSTM-GNN, is the LSTM-output of the individual catchments rescaled to [m³/s], before it is fed into the GNN part? In other words, how is it assured in the workflow that the

relative runoff contribution of each catchment is correctly represented? Please clarify.

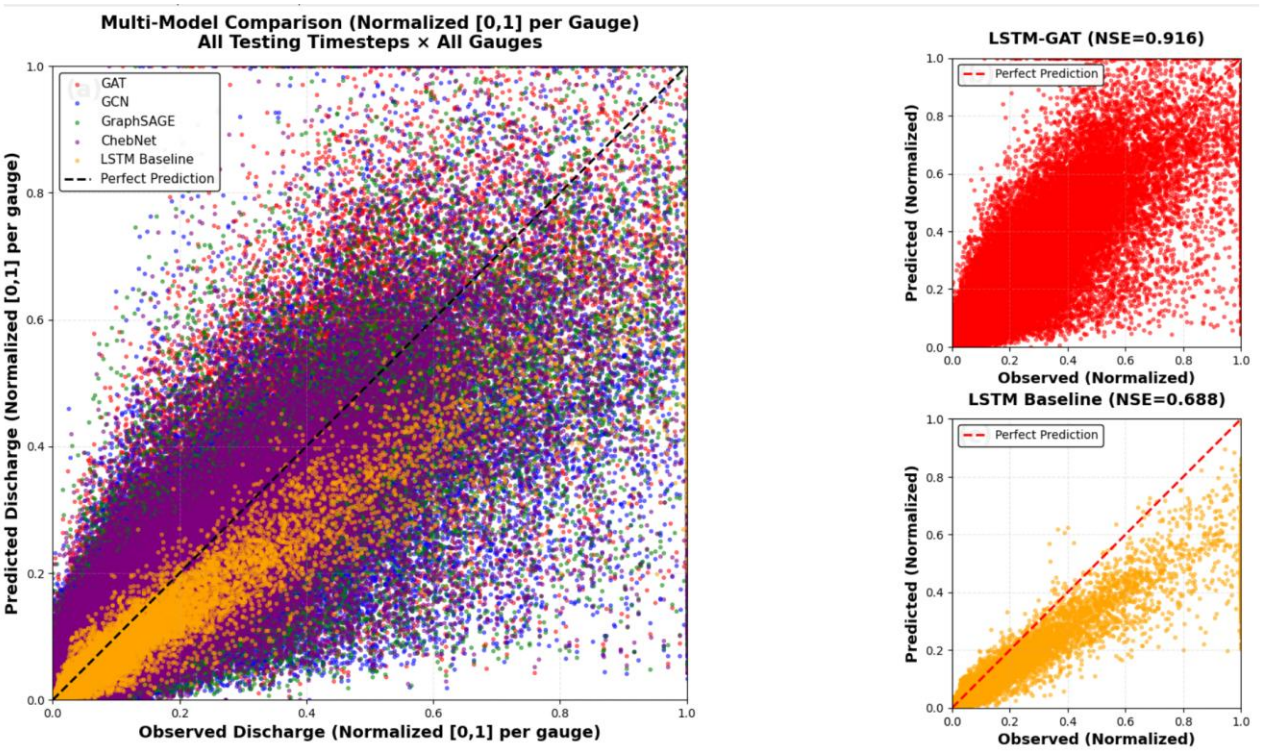
Thank you for this question. We clarify the data flow and scaling as follows.
The LSTM component does **not** directly output discharge values in physical units $[m^3/s]$. Instead, it produces latent feature embeddings of dimension 128 that encode the temporal hydrological state of each subbasin. Specifically, as described in Section 3.1.1, the final hidden state $h_i^{(L)}$represents a learned summary of the runoff-generating processes for subbasin $i$in latent space, rather than a physical discharge quantity. These temporal embeddings are then concatenated with encoded static catchment attributes (59 features) to form a combined latent representation. This combined embedding remains in latent space and serves as the input to the GNN. The GNN operates entirely on these latent representations, performing message passing and aggregation across the river network to model spatial routing. Only after the final GNN layer is the resulting embedding transformed into a discharge prediction through a linear output layer. The predicted values are produced in the same normalized space as the training targets (log-transformed discharge). During evaluation, these predictions are inverse-transformed using the inverse of the positive_robust_log transformation to obtain discharge values in physical units $[m^3/s]$.

We have added the following clarification to **Section 3.1**:
*"This combined representation h_i serves as the input to the GNN module and captures both the temporal runoff dynamics and static catchment characteristics of subbasin i; notably, routing is performed on these latent representations, and discharge values are predicted only after the GNN processing.."*

- Fig. 3: Unclear for which hop the results are shown. Also, in subplot d the color-coding is missing.

  The subtitle has revised as follows "*Boxplots comparing the baseline LSTM and four LSTM–GNN architectures (GAT, GCN, GraphSAGE, and ChebNet) across 530 stations using NSE, KGE, its components (α, β), correlation coefficient (CC), and RMSE (m3/s) for 3_hop. Green boxes indicate the best-performing model for each metric, while red boxes denote the lowest-performing model. And others in yellow*"
- All Figures: The dashed lines are hard to see. Use color-coding only.
  Done.
- Fig. 5: Is this really a plot of all testing timesteps and all gauges? Please clarify. Visually this is dominated by the floods in the few largest catchments. Maybe it is more illustrative to show plots of scaled streamflow, where separately for each gauge, streamflow is scaled [0,1].
  The original figure included all data points but was indeed visually dominated by high-discharge events from large catchments. Following your suggestion, we have replaced Figure 5 with normalized scatter plots where the discharge for each gauge is scaled to the range [0,1].

We have revised the result section:
*"Scatter plots of normalized predicted versus observed discharge (Figure 5), where flow values for each station are scaled to the [0,1] range, highlight the reduced bias and tighter clustering around the 1:1 line for LSTM-GNN models compared to the baseline. Among all models, LSTM–GAT predictions most closely align with the 1:1 line."*



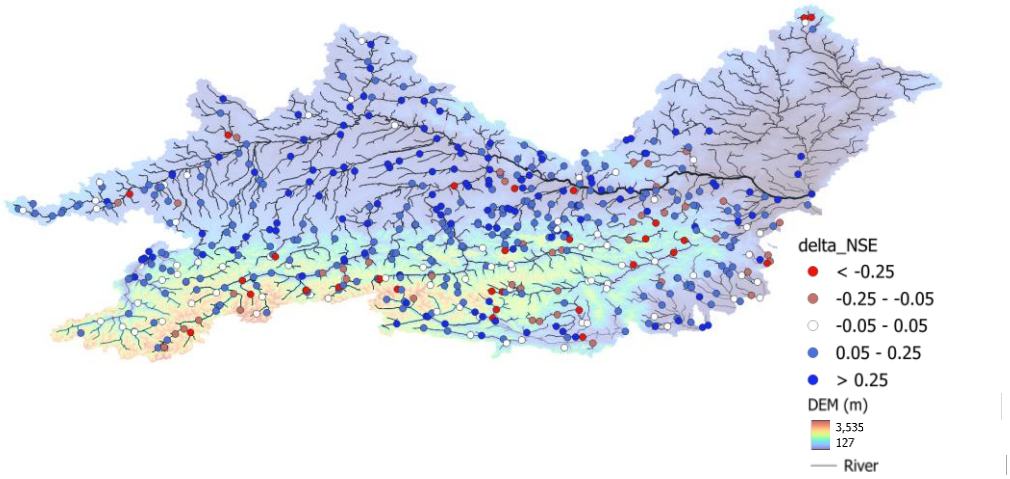

- Fig. 6: This is not very helpful as it does not reveal any distinct pattern. Consider replacing it with a plot of dNSE vs. catchment size (similar to Fig. 7d, but more detailed).

We thank the reviewer for this constructive suggestion. We have completely revised Figure 6 to provide much richer spatial information that reveals distinct geographic and hydrological patterns. The new Figure 6 now employs a multi-layer visualization approach that encodes multiple variables simultaneously: Point color: Five discrete categories of ΔNSE (from < -0.25 to > 0.25), River network thickness: Scaled by drainage area, and background elevation (DEM)

This enhanced visualization reveals several distinct spatial patterns that were not apparent in the original figure:

- Elevation-dependent performance: Stations with negative or minimal improvement are predominantly located in high-elevation.
- Downstream accumulation benefit: Stations with strong improvement (dark blue dots, ΔNSE > 0.25) cluster along major river reaches with large drainage areas (thick river lines), particularly in the lowland sections where flow routing and upstream contributions are most significant.

Also the text revised as follows “To further investigate where the LSTM–GAT model, the best-performing GNN architectures, offers improvements over the baseline LSTM, we conducted a spatial comparison of performance metrics across all gauged stations. The difference in NSE values (ΔNSE = NSELSTM-GAT– NSELSTM) was calculated for each of the 530 stations (Figure 6). Positive differences, shown in red, indicate locations where LSTM–GAT outperformed the baseline, while negative differences (blue) denote stations where the baseline LSTM achieved higher NSE. River network thickness is scaled by drainage area, and background colours show elevation from DEM. The analysis reveals that LSTM–GAT achieved higher NSE scores at 78% of stations. Stations showing strong improvement (ΔNSE > 0.25, dark blue) are predominantly located along major river reaches with large drainage areas. Conversely, stations with negative  changes (red dots) are concentrated in high-elevation headwaters.”




- Fig. 7 d: Here it seems that catchments > 100.000 km² are included, but only few. How does this match with the statements in Line 97?

  Revised.

- Fig. 9: Use same color-coding as in previous figures.

  Done.

- As the topic is closely related, the authors may wish to take a look at a recent preprint by Kraft et al. (2025).

We thank the reviewer for bringing this relevant recent work to our attention. We have reviewed Kraft et al. (2025) and recognize important parallels between DROP and our approach. Both studies demonstrate that incorporating explicit routing significantly improves deep learning-based hydrological predictions.

We have added to Discussion as follows.

*"Kraft et al. (2025) showed that incorporating routing into lumped LSTM baselines across Switzerland yielded substantial performance gains (KGE improvements of 24–62%). "*

Yours sincerely,
Uwe Ehret

References

Kraft, B., Kauzlaric, M., Aeberhard, W., Zappa, M., and Gudmundsson, L.: DROP: A scalable deep learning approach for runoff simulation and river routing, 10.22541/au.176410929.91946608/v1, 2025.