Dear Anonymous Referee：

We are grateful for your thoughtful and constructive comments. Your feedback has significantly strengthened the manuscript in terms of clarity, methodological soundness, and presentation. We have provided detailed responses to each comment.

**Overall comment**

This paper addresses the problem of daily streamflow forecasting of multi-station by applying CEEMDAN decomposition to extract multiscale dynamic features, and an adaptive graph recurrent network to capture spatiotemporal dependencies. The authors also discuss the advantages of uncertainty-based interpretation. Overall, the manuscript is clearly written and generally well organized. However, the proposed approach mainly combines existing techniques rather than introducing a truly novel methodology. The framework—decomposing the time series, modeling each component separately, and then reconstructing the final output—is quite common in the literature. Moreover, the model appears to focus on only one-step-ahead forecasting, which limits its practical value for real-world hydrological applications. Therefore, I don't recommend the publication of the manuscript in HESS in its present form.

*Response:*

*We sincerely thank the reviewer for the positive assessment regarding the clarity and organization of the manuscript, as well as for the constructive concerns about methodological novelty and forecasting horizon. We agree that CEEMDAN and the logarithmic transform are established techniques, and that "decompose-model-recompose" frameworks have been widely adopted in hydrological forecasting studies. In the revised manuscript, we have adjusted the Introduction and contribution statements to avoid overstating the novelty of these individual components.*

*Our intention is not to claim that CEEMDAN or log-based preprocessing as methodological innovations. Instead, our focus is on how multiscale decomposition and graph neural networks interact in a multi-station setting where both spatial connectivity and non-stationary dynamics play central roles. To the best of our knowledge, most existing CEEMDAN-based hydrological studies couple CEEMDAN with LSTM/GRU/CNN at the single-station level, while GNN-based hydrological studies often relies on static graphs when representing spatial relationships among multiple stations. In this respect, our contribution lies at the level of an integrated framework and its systematic evaluation. Specifically, the proposed model couples CEEMDAN's multi-scale temporal decomposition with an adaptive graph*

*recurrent architecture, enabling the extraction of the intrinsic temporal characteristics of runoff at different frequencies and their spatial propagation patterns across the watershed network.*

*In addition to proposing an integrated forecasting pipeline, the manuscript also aims to provide interpretive insights into spatial dependencies and predictive uncertainty. First, we analyze the adaptive adjacency matrix learned by ASGGRU on the original series (Section 4.4, Fig. 7), and show that many strong directed edges coincide with known upstream-downstream relationships, while others reflect cross-basin connections plausibly driven by shared meteorological forcing. This indicates that learned graph captures hydrologically meaningful connectivity without explicitly encoding river topology. Second, each deterministic model is coupled with an HMM-GMR post-processor to quantify forecast uncertainty and evaluate coverage, sharpness and CRPS across models (Section 4.6). This provides a systematic assessment of uncertainty for the LCEEMDAN-ASGGRU hybrid framework at the basin scale and helps interpret performance gains.*

*Regarding the reviewer's concern that "one-step-ahead forecasting limits model's practical value for real-world hydrological applications", we acknowledge that the main experiments in the current version focus on one-day-ahead daily prediction (Section 4.1). To address this concern, we have conducted additional experiments where separate instances of the proposed LCEEMDAN-ASGGRU model were trained for lead times from 2 to 7 days ahead, using the same 12-day input window. The results are included in the revised manuscript. Figure R1 in this response summarizes the mean NSE over the 14 stations for lead times from 1 to 7 days, the mean NSE decreases smoothly from 0.888 at lead-1 to 0.832, 0.767, 0.762, 0.728, 0.695 and 0.662 at lead-2 to lead-7, respectively. This behavior is consistent with the expected degradation of forecast skill as the lead time increases, and indicates that the proposed architecture still provides useful predictive information several days in advance.*

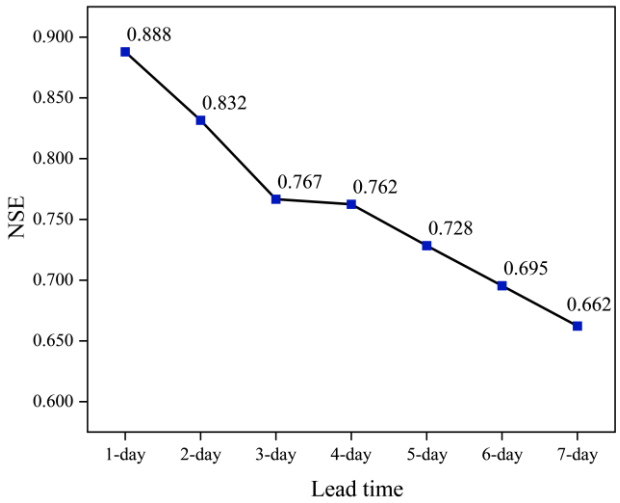


*Figure R1: The NSE of different lead time.*

*In summary, we have revised the manuscript to:*

*(i) temper the novelty claims around CEEMDAN and log-transform preprocessing;*

*(ii) more clearly articulate the contribution of combining multiscale decomposition, adaptive graph learning and uncertainty quantification for multi-station streamflow forecasting, including an illustrative spatial analysis of the adaptive graph learned by ASGGRU;*

*(iii) address the original focus on one-day-ahead prediction by incorporating comprehensive multi-day (2-7 day) lead-time experiments.*

**Specific comments**

L120: Figure 1 is not clear, and this issue persists throughout the manuscript. Figures in general need to be improved in clarity and readability.

*Response:*

*Thank you for the comment. We revised Figure 1 by adding an elevation color scale with units (m) and explicitly labelling the five major tributaries of the Poyang Lake Basin (Ganjiang, Fuhe, Raohe, Xinjiang, and Xiushui). In addition, we have improved the clarity of all figures throughout the manuscript by increasing resolution, enhancing line and marker weights, and enlarging font sizes. These adjustments collectively make the figures clearer and more readable.*

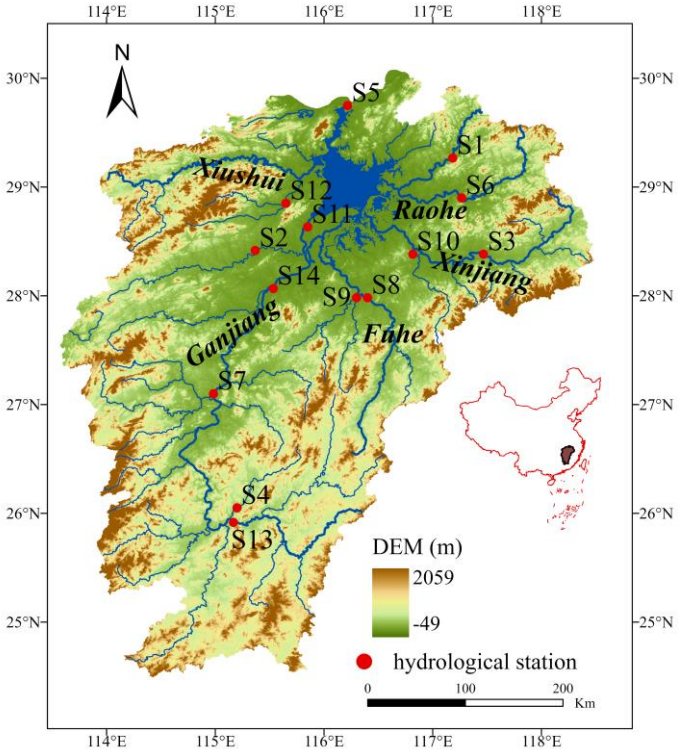


*Figure 1 : Location of hydrological stations.*

L202: The abbreviation Aadp is not defined.

*Response:*

*Thank you for pointing this out. We have supplemented the definition of* $A_{adp}$ *in Section 3.2.*

*L203 below Eq. (6) has been revised to read:*

*"where* $A_{adp} \in R^{N\times N}$ *denotes the adaptive adjacency matrix,* $E_1, E_2 \in R^{N\times e}$ *are two trainable node-embedding matrices,* $N$ *is the number of stations and* $e$ *the embedding dimension."*

L208: The meaning of arrow in the lower right corner of Figure 2 is unclear.

*Response:*

*Thank you. We improved all arrows for readability.*

L212: Is SGGRU proposed by the authors' team, or is it based on Zhao et al. (2020)? Please clarify.

*Response:*

*Thank you for the question. Our spatial graph gated recurrent unit (SGGRU) follows the same principle as the T-GCN cell proposed by Zhao et al. (2020), in which graph convolution is embedded within a*

*GRU to jointly model spatial and temporal dependencies. Zhao et al. (2020) implement their model using an undirected road-network graph, whereas in our study the framework is instantiated with directed and potentially asymmetric graphs.*

*To eliminate ambiguity, we revised the description of SGGRU in Section 3.3:*

*Line 210: "Following the temporal graph convolutional unit of Zhao et al. (2020), we adopt a GRU cell in which the affine transformations are replaced by graph-convolution operators defined on either fixed or learned adjacency matrices. We refer to this generic graph-convolutional GRU as the spatial graph gated recurrent unit (SGGRU) in this study."*

L257: Corresponding to the logarithmic and standardization steps in Step 1 and 3, shouldn't inverse standardization and inverse log-transformation be applied before combining the final results? (Figure 3 indicates such steps are needed.)

*Response:*

*We thank the reviewer for raising this point. The reviewer is correct that the preprocessing in Steps 1 and 3 (logarithmic transform and standardization) must be inverted when reconstructing the final streamflow series. This is indeed how the model is implemented, but our description in Section 3.4 and Figure 3 was not sufficiently explicit. We have therefore rewritten this part of the text and clarified the full pipeline as follows , with Figure 3 revised to reflect these clarifications:*

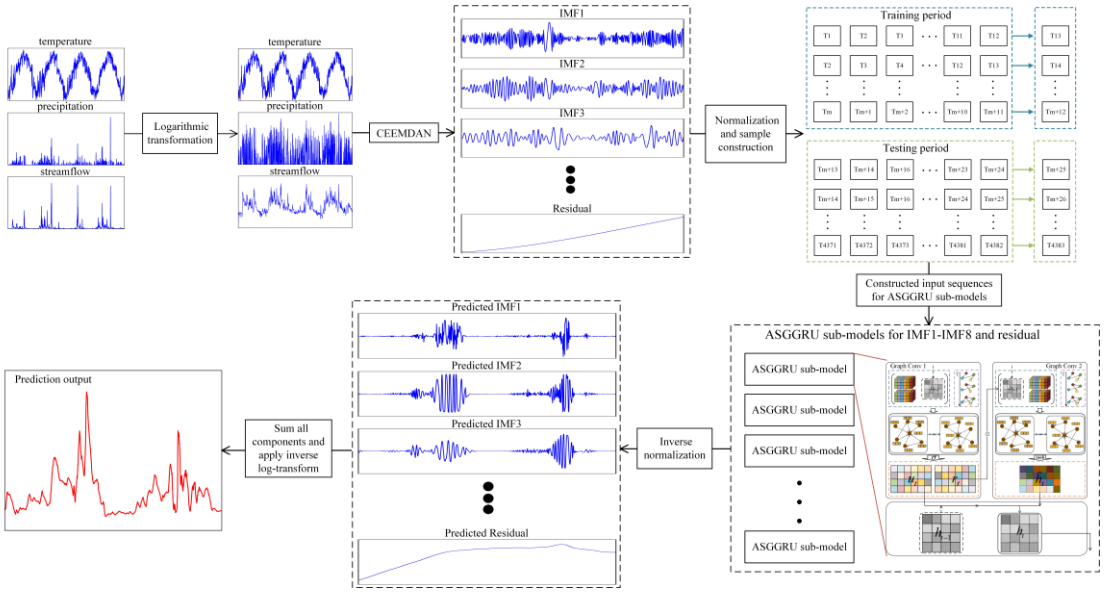


Figure 3: Framework of LCEEMDAN-ASGGRU. The notation T1-T4383 represents the sequence of daily time steps in the full dataset (from day 1 to day 4383).

*Line235:*

*(1) To stabilize variance and mitigate the influence of extreme values observed in streamflow and precipitation time series, we apply a logarithmic transformation prior to decomposition. Specifically, we adopt the natural logarithm with the transformation defined as:*

$$X_{\log}(t) = \log(1 + X(t)) \tag{10}$$

*This transformation ensures numerical stability while reducing the influence of large outliers, thereby enhancing the subsequent CEEMDAN decomposition and improving the separability of IMFs across scales.*

*(2) The CEEMDAN decomposition was performed using the PyEMD library with default parameters, including a noise standard deviation of 0.2 and 250 noise-assisted trials. Each variable (streamflow, precipitation, and temperature) at each station was decomposed into eight IMFs and one residual component.*

*(3) Each IMF and residual component is standardized using z-score normalization before forecasting to ensure they remain on the same scale:*

$$X_k^{'}(t) = \frac{X_{\log,k}(t) - \bar{X}_{\log,k}}{\sigma_k} \tag{11}$$

*where $X_k^{'}(t)$ denotes the normalized series of the k-th component, $X_{\log,k}(t)$ denotes the k-th IMF in log domain. $\bar{X}_{\log,k}$ and $\sigma_k$ denote the mean and standard deviation of this component. These two corresponding normalization parameters were stored and applied during postprocessing to enable accurate inverse transformation of the model predictions back to the original scale.*

*(4) Each of the nine decomposed components from the CEEMDAN process is treated as an independent prediction subtask. For each component, a separate instance of the ASGGRU model is trained independently, allowing the model to specialize in capturing the temporal dynamics unique to that frequency scale. Notably, each IMF and the residual has its own set of optimal hyperparameters, reflecting the varying statistical characteristics and predictive complexities across components. This design provides additional flexibility, enabling the model to allocate capacity appropriately.*

*(5) During inference, each trained submodel outputs a predicted sequence corresponding to its specific component. These component outputs are first mapped back to the log domain via inverse z-score normalization, then linearly aggregated across all components to reconstruct the log-transformed*

*streamflow series, and finally converted to the original streamflow scale using the inverse log-transform. Mathematically, the reconstruction can be expressed as:*

$$y'_{\log,k}(t) = y'_k(t)\sigma_k + \bar{X}_{\log,k}(t) \quad (12)$$

$$\hat{y}_{\log}(t) = \sum_{k=1}^{K} y'_{\log,k}(t) \quad (13)$$

$$\hat{y}(t) = \exp(\hat{y}_{\log}(t)) - 1 \quad (14)$$

*where $y'_k(t)$ denotes the prediction of the k-th IMF or residual submodel in the ASGGRU submodel. $y'_{\log,k}(t)$ is the inverse-normalized value. $\hat{y}_{\log}(t)$ denotes the reconstructed streamflow series in log domain, and $\hat{y}(t)$ is the final streamflow prediction value after applying the inverse logarithmic transformation.*

L360: What does the learned graph structure of ASGGRU look like after training? Please provide a visualization. Which also makes me confused in L436.

*Response：*

*Thank you for raising this point. In the submitted manuscript, the adaptive adjacency matrix learned by ASGGRU is already visualized as a heat map in Section 4.4 (Figure 7). However, we realize that this figure was only introduced later in Section 4.4, the discussion around L360 and L436 may have been difficult to follow on first reading.*

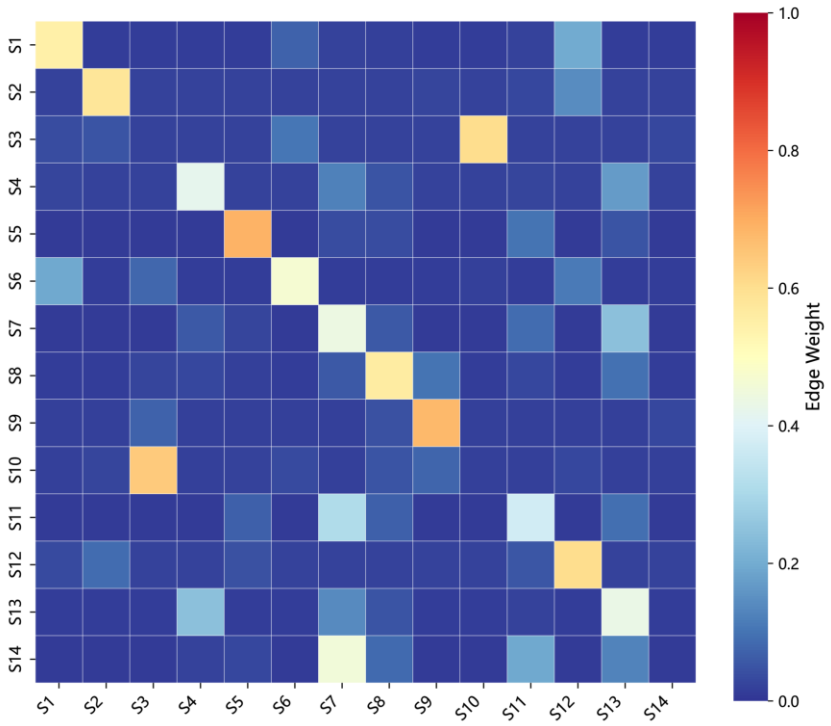


Figure 7: Learned adaptive adjacency matrix obtained from ASGGRU.

*In the revised manuscript, we have clarified connection. When introducing ASGGRU and its adaptive graph mechanism in Section 4.3.2, we now explicitly refer the reader to the spatial analysis in Section 4.4 and Figure 7, where the learned adaptive graph is visualized and interpreted.*

*Specifically, Line 370 has been revised to:*

*"In contrast, the ASGGRU model learns an adaptive spatial graph via node embeddings that are updated jointly with the model parameters, the spatial patterns of the learned adaptive graph are further analyzed in Section 4.4 (see Figure 7)."*

L421: From Figure 6, the ASGGRU results exhibit notable under- and over-estimations of flood peaks, and perform no better than DTWSGGRU and FDSGGRU. This appears inconsistent with the performance metrics reported in the text. Could the authors provide additional metrics, specifically for high-flow and low-flow conditions, based on Figure 4?

*Response:*

*Thank you for the comment. Figure 6 displays time series at two representative stations (S4 and S5), whereas the performance metrics summarized in Figure 4 are aggregated over all 14 stations and over the whole flow range. We agree that when focusing specifically on the flood peaks in Figure 6, the*

*ASGGRU-based curves still display instances of underestimation and overestimation during some extreme events. This may appear inconsistent with the overall NSE values.*

*To examine this more systematically, we conducted an additional evaluation in which model performance is stratified by flow condition. For each station, the empirical Q75 of the observed daily streamflow was used as the threshold. Days with $Q \geq Q75$ were classified as high flow and days with $Q < Q75$ as low-to-moderate flow. NSE and RMSE were then computed separately for the two subsets and averaged over the 14 stations and 10 runs. The resulting performance metrics for high flow and low-to-moderate flow NSE and RMSE are summarized in Figure R2 of this response. This figure and corresponding analyzes have been provided in the revised manuscript.*

*Considering both conditions together, the Q75-based analysis shows a consistent ranking of model performance. Under high-flow conditions ($Q \geq Q75$), LCEEMDAN-ASGGRU attains the best skill among all models, with higher NSE and lower RMSE than CEEMDAN-ASGGRU, ASGGRU with the learned adaptive graph, the two static-graph (DTWSGGRU and FDSGGRU), and LSTM. In this condition, the ordering of high flow NSE aligns well with the basin-wide NSE reported in Figure 4, indicating that the overall metrics already reflects relatively high-flow performance.*

*For low-to-moderate flow condition ($Q < Q75$), the differences between models are even more pronounced. LCEEMDAN-ASGGRU again shows the strongest performance, while the static-graph models and the CEEMDAN-ASGGRU (without log transform) exhibit substantially lower. ASGGRU maintains positive NSE in this condition and outperforms DTWSGGRU, FDSGGRU, and LSTM. Importantly, introducing the logarithmic transform in LCEEMDAN-ASGGRU yields a substantial improvement in low-to-moderate performance, highlighting its effectiveness in stabilizing errors under low-to-moderate conditions.*

*In summarize，because NSE is more sensitive to errors during high-flow periods, the mean NSE in Figure 4 already captures much of the high-flow model skill. The Q75-stratified analysis confirms the overall performance ranking remains robust across flow conditions, and the apparent peak mismatches in Figure 6 represent local examples rather than contradictions to the basin-wide metrics.*

*We also identified and corrected a labelling error in the caption of Figure 6 in the original submission (the order of S4 and S5 was inadvertently swapped). The revised caption now reads:*

*"Figure 6: Observed and predicted streamflow at two stations: (a) S5, situated at the northernmost point and exhibiting frequent and extreme high-flow events; (b) S4, located at the southernmost point of the basin and characterized by persistently low-flow conditions."*

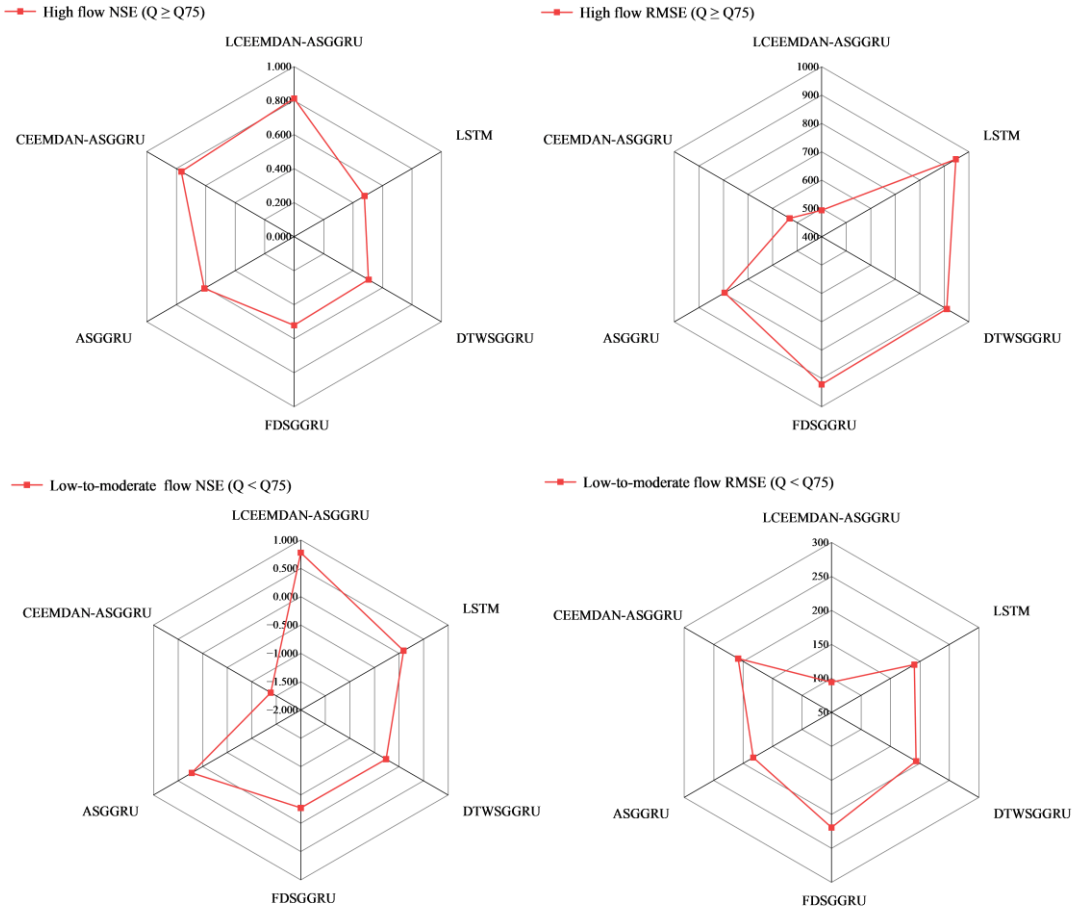


*Figure R2: Model performance comparison in high-flow and low-to-moderate conditions across different forecasting models.*

L446: The overlap ratio between the ASG-based graph and the flow-direction-based graph (FD) has been provided. Accordingly, the overlap ratio between the DTW-based graph (DTW) and the FD should be reported for comparison. The overlap ratio between the DTW-based graph (DTW) and the flow-direction-based graph (FD) should be quantified and reported.

*Response:*

*Thank you for this helpful suggestion. In the revised manuscript, we have extended the overlap analysis to explicitly include the DTW-based graph (DTW) and the flow-direction-based graph (FD).*

*We now report directional overlap ratios among the learned adaptive graph $A_{adp}$, the DTW graph and the FD graph. For two edge sets X and Y, we define the directional overlap X→Y as the fraction of edges in Y that also appear in X, that is, "X→Y = (number of edges shared by X and Y) / (number of edges in Y)".*

*For the adaptive graph $A_{adp}$, we focus on $A_{adp}$→FD and $A_{adp}$→DTW, which quantify how much of the physical-flow and similarity-based prior information is recovered by the learned graph. For the two static graphs, we additionally report DTW→FD and FD→DTW to characterize the extent to which flow-direction structures are embedded within the broader DTW similarity network.*

*During this recomputation, we identified a minor preprocessing error in our original script: a header row had been mistakenly interpreted as an additional node when constructing the adjacency matrices for overlap analysis. This issue affected only the previously reported numerical values and did not influence any part of the model training or prediction. We have corrected the error, recalculated all overlap ratios, and updated the relevant text and Table 4 in Section 4.4. The qualitative conclusions remain unchanged.*

*The updated results can be summarized as follows (see Table 4).*

*Table 4: Learned adaptive adjacency matrix with the flow-direction (FD) and DTW-based graphs under varying threshold levels.*

| Threshold | $A_{adp}$ edge counts | FD edge counts | DTW edge counts | $A_{adp}$→FD | $A_{adp}$→DTW | DTW→FD | FD→DTW | $A_{adp}$-only | DTW-only | FD-only |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.05 | 33 | 13 | 182 | 38.5% | 18.1% | 100% | 7.1% | 0 | 119 | 0 |
| 0.1 | 18 | 13 | 182 | 30.8% | 9.9% | 100% | 7.1% | 0 | 132 | 0 |
| 0.2 | 6 | 13 | 182 | 7.7% | 3.3% | 100% | 7.1% | 0 | 139 | 0 |

*For all thresholds, the overlap between DTW and FD is 100 %, which means that every flow-direction edge is contained in the DTW graph. In other words, the physical flow-direction network is fully embedded within the DTW-based similarity graph, while FD links account for only a small fraction of all DTW edges (FD→DTW = 7.1%). The adaptive graph $A_{adp}$ can be viewed as a sparse, task-oriented subgraph extracted from this dense similarity network. It retains only a small proportion of DTW edges, and among those, it tends to preserve a larger share of flow-direction consistent links. At the 0.05 threshold, for example, A contains 33 effective edges, recovering 38.5 % of the FD but only 18.1 % of the DTW links. This pattern suggests that the learned graph selectively preserves temporally similar connections that are also hydrologically plausible, rather than inheriting the majority of DTW-only correlations.*

*The corresponding paragraph in Section 4.4 has been revised as follows:*

*"To further quantify the relationship between the learned adaptive graph and the two static priors, we conducted an overlap analysis under varying edge-weight thresholds (0.05, 0.10, and 0.20). For two edge sets X and Y, the directional overlap X→Y is defined as the fraction of edges in Y that also appear in X. For the adaptive graph $A_{adp}$, we report $A_{adp}$→FD and $A_{adp}$→DTW, which indicate how much of the physical-flow and similarity-based prior information is recovered by the learned graph. For the two prior graphs themselves (i.e., the static graphs), we report DTW→FD and FD→DTW to characterize how the FD structure is embedded within the broader DTW similarity network. The resulting overlap ratios and edge counts are summarized in Table 4.*

*The results reveal two main patterns. First, DTW→FD is 100 % for all thresholds, which means that every flow-direction edge is contained in the DTW graph. FD→DTW is 7.1% for all thresholds, which means that FD links account for only a small fraction of all DTW edges. The adaptive graph $A_{adp}$ can be viewed as a sparse, task-oriented subgraph extracted from this dense similarity network. At the 0.05 threshold, $A_{adp}$ contains 33 effective edges, which together recover 38.5 % of the FD links but only 18.1 % of the DTW links. As the edge-weight threshold is increased from 0.05 to 0.20, the number of retained edges in A decreases from 33 to 18 and 6, and the coverage of both FD and DTW links is reduced ($A_{adp}$→FD from 38.5% to 7.7 %, $A_{adp}$→DTW from 18.1 % to 3.3 %). Across all examined thresholds, the fraction of FD links recovered by $A_{adp}$ remains larger than that of DTW links, confirming that the learned graph is more strongly aligned with the flow-direction prior than with the full DTW similarity network. Importantly, no $A_{adp}$-only or FD-only edges are observed for any threshold, whereas a large number of DTW-only edges remain. This suggests that the adaptive graph learning primarily operates within the joint subspace spanned by the flow-direction and DTW priors, refining and re-weighting hydrologically and temporally meaningful spatial relationships rather than inventing entirely new connections."*

L460: According to the setup (3 × 2 × 2 = 12 models + baseline = 13), only six models are presented, please clarify the rationale.

*Response:*

*Thank you. In the original text, the* design space *was described as involving three spatial graphs (flow-direction, DTW, adaptive graph), CEEMDAN, and logarithmic transform, along with an LSTM*

*baseline. We agree that this wording may have unintentionally suggested that a full 3 × 2 × 2 factorial experiment was both implemented and fully reported. This was not our intention.*

*As state in L338-L341, our study follows a* ***hierarchical ablation strategy****, rather than an exhaustive enumeration of all 12 combinations:*

*"To validate the effectiveness of the proposed LCEEMDAN-ASGGRU, this section presents a comprehensive performance analysis through comparative experiments. Instead of evaluating each model in isolation, we adopt an ablation approach, progressively breaking down the proposed model into its key components: the adaptive graph learning module, the multi-scale CEEMDAN decomposition, and the logarithmic transformation."*

***(i) Graph structure without decomposition or log transform.***

*We first fix the preprocessing (no CEEMDAN, no logarithmic transform) and compare four architectures: LSTM, DTWSGGRU, FDSGGRU, and ASGGRU. This isolates the effect of spatial graph construction under the same temporal pipeline (SGGRU). As reported in Fig. 4 and Tables 2-3, all graph-based models outperform LSTM, and ASGGRU yields the best performance;*

***(ii) Decomposition and log transform, conditional on the best graph.***

*Next, we fix the spatial graph to the best-performing configuration (ASGGRU) and evaluate the contribution of CEEMDAN and the logarithmic transform. To this end, we report CEEMDAN-ASGGRU and LCEEMDAN-ASGGRU. With in this comparison, CEEMDAN-ASGGRU improves upon ASGGRU, and LCEEMDAN-ASGGRU further improves upon CEEMDAN-ASGGRU. The six models shown in the main text are correspond to this stepwise ablation and are validate the performance advantages of the proposed LCEEMDAN-ASGGRU.*

*Thus, the hierarchical ablation design clearly demonstrates the incremental contribution of each component and avoids unnecessary repetition.*