Response to community comments CC2&CC3:

I would like to congratulate the authors for this original and well-structured contribution. The PLSTM-Reg model is an elegant way to embed physical consistency directly into the recurrent loop, and the results across the diverse set of reservoirs are compelling. I have several question regarding the model training and evaluation that I hope the authors can address:

Response: Dear Dr. Francois, thank you very much for your thorough review and encouraging comments. We have carefully addressed your comments, as detailed below.

Major Comments

1. Estimation of operational release limits Q_min and Q_max

Initial Comment1:

The Physical Knowledge Module (Equations S8–S10) requires per-reservoir estimates of minimum and maximum allowable release, Q_min and Q_max. The manuscript and supplement do not describe how these values were derived. Could the authors clarify: Were they computed from the observed outflow record (e.g., obs min and max values? empirical percentiles?) or from engineering/regulatory sources?

Follow-up Comment1:

After reviewing the code shared by the authors, I would like to ask a few follow-up questions regarding Equation S8 and the role of Q_min and Q_max in the Physical Knowledge Module.

In the released code (ReservoirLSTM.rnncell), the physical module applies Equations S9–S11 as described in the supplement, but Equation S8 does not appear to be implemented. Instead, the candidate release predicted by the linear output head is denormalized directly using the per-reservoir empirical mean and standard deviation of observed outflow:

release_denorm = release_factor * target_scale[0] + target_center[0]

This denormalization step does act as a soft, data-driven constraint: when the LSTM output is near zero, the predicted release is close to the observed mean, and the per-reservoir scaling discourages predictions far outside the observed distribution. However, this is a statistical prior derived from the training data rather than a hard operational bound. It cannot be directly interpreted as Q_min and Q_max, and it provides no guarantee that releases remain within physically or operationally

meaningful bounds, particularly for out-of-distribution conditions.

I understand that the storage clamp (Eq S10) combined with Eq S11 provides an implicit physical floor and ceiling on release (you cannot release more than the available water, and you must spill when the reservoir is full). This is a meaningful hard constraint. However, it does not serve the same purpose as Q_min (minimum environmental or operational flow requirement) and Q_max (maximum release capacity).

Could the authors clarify: (a) Was Equation S8 intentionally omitted from the implementation, with the normalization scheme serving as the intended proxy for release bounds? (b) If so, how are Q_min and Q_max defined in the manuscript, and are they used anywhere in the training or evaluation pipeline? (c) Does the absence of an explicit release clamp affect the physical interpretability of the model, particularly the claim that "physical knowledge" is enforced?

Response: Thank you for these questions regarding the implementation of Eq. (S8). Q_{min} and Q_{max} are intended to represent reservoir-specific operational limits (e.g., downstream environmental flow or navigation requirements for Q_{min}; turbine or spillway capacity for Q_{max}) rather than empirical percentiles derived from the observed outflow records. In our previous work (Yu et al., 2025), these constraints were explicitly enforced because reliable operational data were available for the two studied reservoirs. However, such data were not consistently available across the 256 reservoirs analyzed here. Therefore, although Eq. (S8) was retained in the manuscript for methodological completeness, it was intentionally not activated as a hard clamp in the present large-sample implementation. Instead, the statistical denormalization served as a soft data-driven prior.

We believe the omission of Eq. (S8) would not undermine the core physical interpretability of the PLSTM-Reg model. The primary physical knowledge enforced by the architecture is mass conservation. While Eq. (S8) reflects human-defined operational preferences, Eqs. (S9)-(S11) enforce fundamental physical constraints on storage and release dynamics. Specifically, the model cannot release more water than it is available in storage and must spill when storage capacity is exceeded. Therefore, the mass-balance consistency of the model remains preserved.

To clarify this point, we have added the following text to **Text S1** of the Supplement: "*It is important to note that, although Eq. (S8) is included for theoretical completeness, reliable estimates of the operational and infrastructure limits* Q_{min} *and* Q_{max} *are generally unavailable for large-sample reservoir datasets. Therefore, to avoid introducing potentially unrealistic operational constraints, Eq. (S8) was intentionally not enforced in the current large-sample implementation*"

Furthermore, to better align our published Zenodo repository (https://doi.org/10.5281/zenodo.18265198; Yu, 2026) with the theoretical framework, we have updated the reference code to include the Q_{min} and Q_{max} constraints. These constraints remain deactivated by default for the current large-sample experiments, but can be enabled when reliable reservoir-specific operational metadata are available.

2. LSTM warmup period and handling of the physical storage state

Initial Comment2:

Standard LSTM implementations use a warmup period to bring the hidden state from zero-initialization to a realistic operating regime before the loss is computed. In PLSTM-Reg, the physical storage state s_t is simultaneously updated at every timestep via the water-balance equation (Eq. S9), creating a tighter coupling between the recurrent state and the physical state than in a conventional LSTM. Could the authors clarify: (a) Was a warmup period used, and if so, what length? (b) During warmup, was the physical knowledge module (Eqs. S8–S11) active, or was the physical state held at the observed initial storage? Else?

Follow-up Comment3:

I also noticed that the initial physical storage state is hard-coded to 0.5 (i.e., half of the normalized capacity) regardless of the actual observed storage at the start of each sequence:

```
# initialize reservoir storage using 0.5
device = x_d_embedded.device
temp_vars = torch.ones((batch,2)).to(device=device,
dtype=x_d_embedded.dtype)*0.5
```

Could the authors comment on the sensitivity of the model to this initialization choice? In particular, for short sequences or reservoirs that are frequently at extreme storage levels (near empty or near full), a fixed 0.5 initialization may introduce a systematic spin-up bias. Was this choice evaluated against alternatives such as initializing from the observed storage at the start of each sequence, or using a learned initial state? Is this initial storage used as the start of the warm-up period during training only?

Response: We appreciate this important question regarding state initialization and warmup.

Yes, 1-year (365 days) warmup period was used for all long-term sequence simulations. During warmup, the physical knowledge module (Eqs. S8–S11)

remained fully active, allowing both the LSTM hidden state and the physical storage state to evolve towards a realistic mass-balanced regime before the loss computation and evaluation.

Regarding the initialization of the physical storage state, the strategy depended on the experimental setting. In data-rich settings (Training & Experiment II), the observed initial storage was used. In ungauged or data-scarce settings (Experiment IV & Regional Benchmarks), where initial storage was assumed unavailable, the model used 50% capacity (0.5) as a neutral prior.

To evaluate the sensitivity to the 0.5 initialization setting, we conducted an additional sensitivity analysis for the spatial generalization model (Experiment IV), initializing storage at 10%, 30%, 50%, 70%, and 90% of reservoir capacity. As shown in **Supplementary Fig. S2** (included below), model performance remained highly consistent across all initialization settings, indicating that the 1-year active warmup period effectively removes the influence of initial condition on long-term simulation.

We have clarified this point in the revised manuscript (**lines 245-249**): "*Furthermore, for all simulations in data-scarce settings (Experiment IV and benchmarks), storage was initialized at 50% of reservoir capacity, whereas observed initial storage was used during training. A 1-year warm-up period was applied to equilibrate model states. Sensitivity analysis confirming the robustness of model to this initial storage assumption is provided in Supplementary Fig. S2.*"

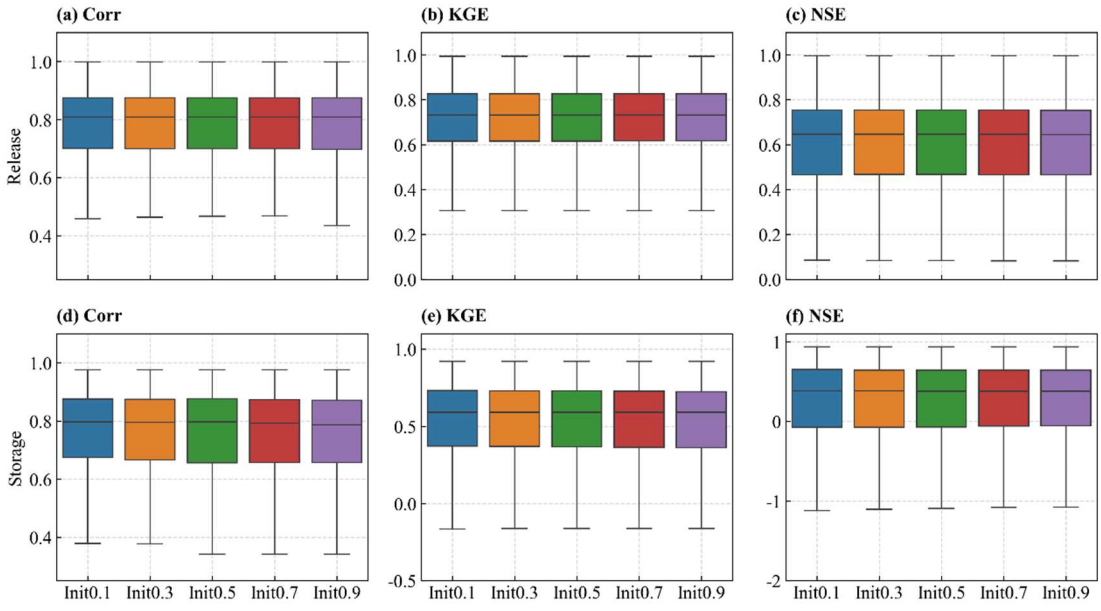

Figure S2. Sensitivity of spatial out-of-sample long-term simulation (Experiment IV) performance to initial storage conditions. Boxplots display evaluation metrics for release simulations (a–c) and storage simulations (d–f) when the physical storage state is initialized at 10%, 30%, 50%, 70%, and 90% of reservoir capacity. The similar performance distributions across initialization settings demonstrate that the 1-

year warmup period effectively minimizes sensitivity to initial storage conditions in long-term simulations.

Furthermore, we have updated the initialization logic in the public Zenodo repository (https://doi.org/10.5281/zenodo.18265198) so that the initialization state is automatically determined from the experimental setting, avoiding ambiguity associated with hard-coded priors.

3. Training sequence length

Initial Comment3:

The choice of sequence length is particularly important for PLSTM-Reg because the model performs a free-run simulation over the full sequence, meaning error in storage accumulates over time. Could the authors indicate the sequence length used during training and whether they observed sensitivity to this hyperparameter? Specifically, did longer sequences lead to better long-term simulation performance at the cost of slower training convergence?

Response: We used a sequence length of 1460 days (approximately 4 years) for both training and testing in long-term simulations. This choice was based on our previous study (Yu et al., 2025), which showed a 4-year sequence effectively capture inter-annual variability while maintaining long-term mass balance.

To further evaluate the sensitivity to sequence length, we conducted an additional sensitivity analysis using 2-, 4-, and 6-year training sequences. As shown in the newly added **Supplementary Fig. S1** (included below), extending the training sequence from 4 to 6 years did not improve simulation performance, despite increased computational cost.

We have clarified this point in the revised manuscript (**lines 226-229**): "*The sequence length for both model training and testing was set to 1460 days (4 years). A sensitivity analysis using 2-, 4-, and 6-year sequence lengths yielded similar performance across all tested settings (Supplementary Fig. S1), indicating that the results are not strongly sensitive to this choice.*"

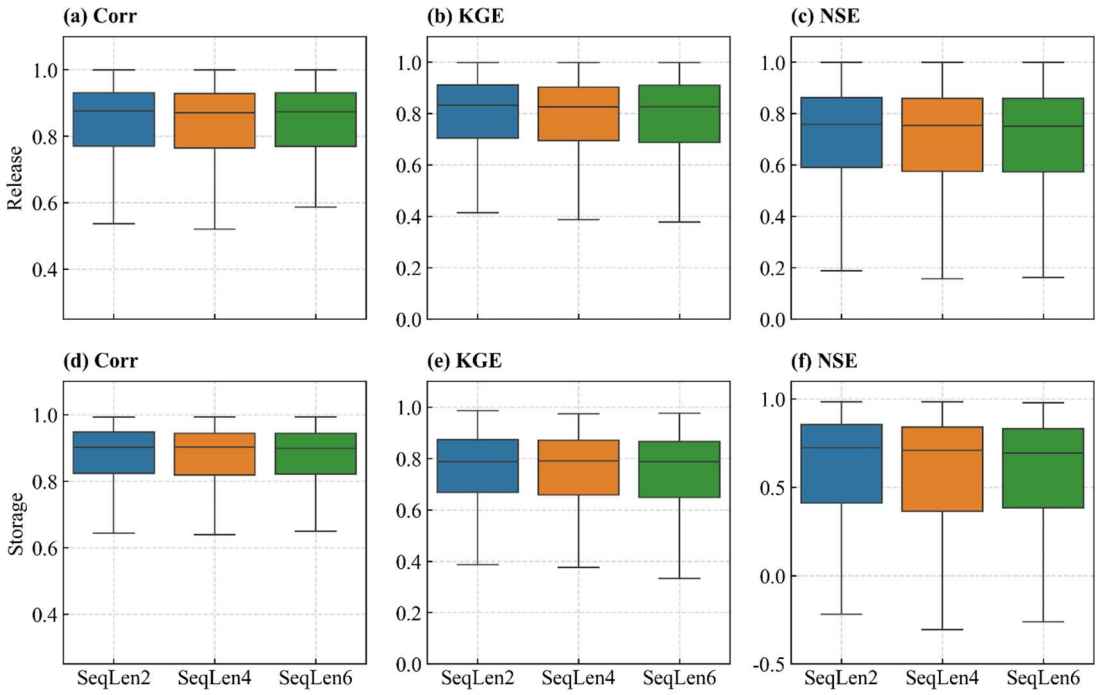

Figure S1. Sensitivity of long-term simulation performance to training sequence length. Boxplots compare release (a–c) and storage (d–f) simulation performance across training sequence lengths of 2, 4, and 6 years for all reservoirs. Evaluation metrics include Pearson Correlation (Corr), Kling-Gupta Efficiency (KGE), and Nash-Sutcliffe Efficiency (NSE). The largely consistent performance distributions across sequence lengths indicate that extending the training sequence beyond 4 years provides limited additional benefit for long-term simulations.

4. Long-term water balance consistency

Initial Comment4:

While the physical knowledge module enforces local mass balance at every timestep (Eq. S11) by construction, this does not guarantee that the long-term water balance is respected at the reservoir scale. We identify several potential sources of systematic bias:

First, the linear output head predicts a candidate release r~_t without any architectural constraint that enforces long-term mass conservation (i.e., there is no mechanism ensuring that the mean predicted release equals the mean inflow minus the long-term storage change). The LSTM may systematically over- or under-predict releases, creating a persistent bias in the physical storage trajectory.

Second, even if the candidate release were unbiased on average, the physical knowledge module can introduce a systematic offset by activating the storage and flow constraints. For instance, frequent clamping at S_max (spilling excess water via

Eq. S11) or at Q_max (capping release and accumulating storage) will alter the long-term mean release relative to what the LSTM predicted. The direction and magnitude of this effect depend on the distribution of reservoir states relative to the constraint bounds and is not self-correcting.

Third, and more fundamentally, the observed inflow and release records in datasets such as ResOpsUS may themselves not close the water balance, due to measurement uncertainty, unobserved fluxes (direct lake evaporation, groundwater exchange, water withdrawals from the reservoir), or data gaps. In such cases, the LSTM may partially learn to compensate for these residuals. This raises the question of whether the model is learning physically meaningful release dynamics or partly fitting an artifact of the input data.

Did the authors evaluate the long-term water balance of the PLSTM-Reg simulations, for example by comparing the multi-year mean simulated release to the mean inflow minus the observed long-term storage trend? A systematic evaluation of this property across the 259 reservoirs, and a discussion of how unobserved fluxes in the training data may affect the model's behavior, would significantly strengthen the physical interpretability of the results.

Response: We thank you for this insightful comment regarding long-term mass conservation and potential systematic drift in free-run simulations.

Regarding the concern about unobserved fluxes and residual water-balance errors in the training data, the reservoir dataset used in this study (Chen et al., 2025) provides net inflow rather than raw inflow. Specifically, net inflow is derived from the observed mass balance equation ($I_{net,t} = S_{t+1} - S_t + R_t$), such that unobserved gains and losses (e.g., evaporation, groundwater exchange, or withdrawals) are implicitly incorporated into the inputs. Therefore, the model is trained on a closed mass-balance system.

In addition, because the physical storage state is explicitly updated and fed back into the model at every timestep, persistent drift in release or storage cannot accumulate freely during long-term simulations.

To evaluate this empirically, we analyzed the long-term release bias across all 256 reservoirs using the absolute percent bias (|PBIAS|) over the testing period (Experiment II). As shown in the newly added **Supplementary Fig. S5** (included below), both the locally and regional PLSTM models maintain very low long-term volume bias, with median |PBIAS| values below 0.4%, indicating that systematic long-term drift is minimal.

To clarify the issues of data closure and long-term mass conservation, we have added the following text to the revised manuscript:

In Section 2.1 (**lines 114-117**): "*Notably, the dataset provides net inflow derived directly from the observed storage–release balance ($I_{net,t} = S_{t+1} - S_t + R_t$), providing an internally closed water balance formulation that implicitly incorporates unobserved fluxes such as evaporation and seepage.*"

In Section 3.1 (**lines 305-307**): "*Crucially, both models maintain long-term water balance with minimal systematic drift (median absolute release percent bias (|PBIAS|) below 0.4%; Supplementary Fig. S5).*"

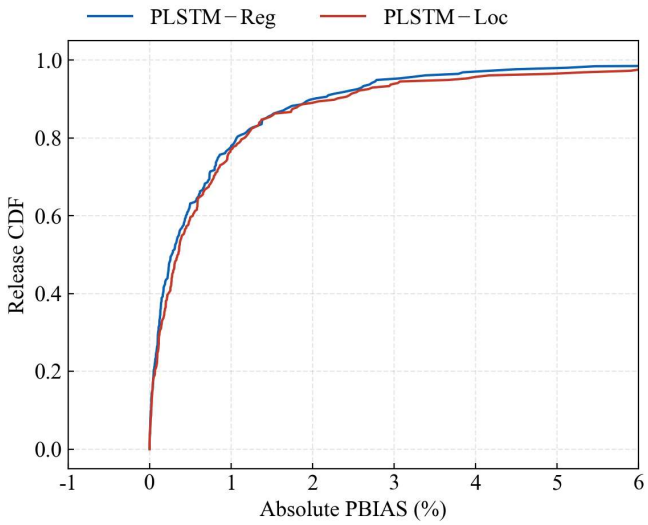

Figure S5. Empirical validation of long-term mass balance consistency across all reservoirs during the testing period. CDFs display the absolute Percent Bias (|PBIAS|) of simulated release for the regional (PLSTM-Reg) and local (PLSTM-Loc) models. Both models exhibit very low long-term release bias across reservoirs, indicating minimal systematic drift.

5. Spatial Out-of-Sample Generalization and Denormalization

Follow-up Comment2:

In addition, I am curious about the spatial out-of-sample evaluation. The model's linear output head predicts a normalized release factor that must be denormalized using the per-reservoir mean and standard deviation of observed release (target_center and target_scale in the code). For reservoirs that were withheld from training, this implies that observed release statistics are still required at inference time. If that is the case, the model is not fully independent of observed data for unseen reservoirs, which would partially undermine the out-of-sample spatial generalization claim. Could the authors clarify how denormalization is handled for reservoirs not seen during training — specifically, whether per-reservoir observed statistics are used?

Response: We appreciate your careful inspection of the codebase. To clarify, we do not use per-reservoir observed statistics for denormalization in any experiments, including the spatial out-of-sample evaluations.

Specifically, the target_center and target_scale parameters are regional statistics computed exclusively from the aggregate training set. By using these global scaling factors, no local observed outflow information from unseen reservoirs is required during inference. Therefore, the spatial out-of-sample evaluation remain fully independent of local observational data.

To clarify this point, we have added the following sentence to Section 2.2 (**lines 194-195**): "*Data normalization was performed using regional scaling factors computed from the training set, rather than per-reservoir statistics.*"

References

Chen, Y., Cai, X., & Li, D. (2025). *Historical Operation Data of 256 Reservoirs in Contiguous United States* [Dataset]. HydroShare. https://doi.org/10.4211/hs.092720588e2e4524bf2674235ff69d81

Yu, B. (2026). *PLSTM-Reg v1.0: A regional physics-encoded LSTM model for simulating reservoir operations under data scarcity* [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.18265198

Yu, B., Zheng, Y., He, S., Xiong, R., & Wang, C. (2025). Physics-encoded deep learning for integrated modeling of watershed hydrology and reservoir operations. *Journal of Hydrology*, *657*, 133052. https://doi.org/10.1016/j.jhydrol.2025.133052