We sincerely thank the reviewer for the thoughtful and constructive feedback. Below, we provide detailed responses to each comment, along with explanations of how and where the corresponding revisions have been incorporated into the manuscript with track changes. Line numbers may vary slightly depending on formatting.

Anonymous Referee #3' s comments:

I want to thank the authors for addressing my comments and making the necessary changes to the manuscript. However, I have one more comment that needs more clarification from the authors and therefore suggest a minor revision.

Authors' answer: Thank you for your thoughtful review and kind acknowledgment of our earlier responses. We appreciate your additional comment and provide the requested clarifications below.

1: *It remains unclear to me how the multi-quantile loss is implemented. Given that all three models are deterministic, how would each model generate the quantile prediction \hat{Q}^q_\tau in Eq.(29)? The authors argue that "the uncertainty arises from the MQL formulation, which estimates conditional quantiles of discharge Qt+h|Xt" in the response letter. The description is too general to capture the details fully. Please provide a mathematical explanation of how a given deterministic model generates the quantile used in Eq. (29) during the model optimization process*

Authors' answer: Thank you for asking for a precise formulation. Let $\mathcal{D} = \{(X_t, y_{t+h})\}_{t=1}^{N}$ denote the training pairs, where X_t is the input context (past 24 h of discharge in our setup) and y_{t+h} is the discharge h hours ahead. For a fixed horizon h and a set of quantile levels $\{\tau_k\}_{k=1}^{K}$, each model f_θ (LSTM, N-HiTS, N-BEATS) is trained to output the vector of conditional quantiles directly:

$$\widehat{\mathbf{Q}}_{t+h} \;=\; f_\theta(X_t) \;=\; (\hat{Q}_{t+h}^{\tau_1}, \dots, \hat{Q}_{t+h}^{\tau_K}) \in \mathbb{R}^K.$$

Training minimizes the multi-quantile (pinball) loss, summed over times and quantile levels:

$$\mathcal{L}(\theta) \;=\; \frac{1}{NK}\sum_{t=1}^{N}\sum_{k=1}^{K} \rho_{\tau_k}\left(y_{t+h} - \hat{Q}_{t+h}^{\tau_k}\right), \qquad \rho_\tau(u) = \max\left(\tau u, (\tau-1)u\right)$$

Equivalently, with the indicator form,

$$\rho_\tau(u) = (\tau - \mathbb{1}_{\{u<0\}})\, u.$$

Because ρ_τ is convex and piecewise linear, its (sub)gradient with respect to the prediction $\hat{Q}_{t+h}^{\tau}$ is:

$$\frac{\partial \rho_\tau(y - \hat{Q}^\tau)}{\partial \hat{Q}^\tau} = \begin{cases} -(1-\tau), & y - \hat{Q}^\tau < 0, \\ -\tau, & y - \hat{Q}^\tau > 0, \end{cases}$$

This yields standard backpropagation updates under Adam optimizer. No sampling is involved: the quantile $\hat{Q}_{t+h}^{\tau}$ is the model's direct output, learned by minimizing the pinball loss at level τ.

Uncertainty bands are then formed from these quantile outputs. For a 95% interval, we use the MQL-trained $\tau = 0.025$ and $\tau = 0.975$ predictions, i.e., $[\hat{Q}_{t+h}^{0.025}, \hat{Q}_{t+h}^{0.975}]$. This captures aleatoric uncertainty conditional on X_t.

We expanded and clarified this method in the revised version of the manuscript. See Lines 444 - 452.

#We thank the reviewer for the insightful and constructive comments.