Latent data assimilation with non-explicit observation operator in hydrology

Received: 9 July 2024 | Revised: 13 March 2025 | Accepted: 22 April 2025

DOI: 10.1002/qj.5009

Quarterly Journal of the Royal Meteorological Society RMetS

RESEARCH ARTICLE

Latent data assimilation with non-explicit observation operator in hydrology

Kun Wang[1,2] | **Sibo Cheng**[3] | **Matthew D. Piggott**[2] | **Sarah L. Dance**[4,5,6] | **Yanghua Wang**[1,2] | **Rossella Arcucci**[2,7]

[1]Resource Geophysics Academy, Imperial College London, London, UK

[2]Department of Earth Science and Engineering, Imperial College London, London, UK

[3]CEREA, ENPC and EDF R&D, Institut Polytechnique de Paris, Île-de-France, France

[4]Department of Meteorology, University of Reading, Reading, UK

[5]Department of Mathematics and Statistics, University of Reading, Reading, UK

[6]National Centre for Earth Observation, University of Reading, Reading, UK

[7]Data Science Institute, Imperial College London, London, UK

Correspondence
Rossella Arcucci, Department of Earth Science and Engineering, Imperial College London, South Kensington, London, SW7 2AZ, UK.
Email: r.arcucci@imperial.ac.uk

Funding information
Resource Geophysics Academy; NERC National Centre for Earth Observation

Abstract

Natural hazards can cause significant damage to human life and property. Among them, floods are one of the most severe and frequent natural disasters, making flood prediction crucial. River discharge is an essential factor in causing floods, so accurate and fast prediction of river discharge is crucial for flood mitigation. Data assimilation (DA) as a method of combining different sources of data (e.g., state field and observations) has the ability to estimate the possible states of river discharge. However, DA on high-dimensional data such as river discharge can be computationally expensive. Furthermore, when the DA process lacks explicit mappings from the state field to the observations, DA cannot be conducted effectively. In this work, we design a latent neural mapping (LNM) in the form of a neural network (NN) as the observation operator and integrate this within a three-dimensional variational data assimilation (3D-Var) framework. By operating within latent space, the resulting approach helps mitigate computational costs and allows us to run DA within seconds despite the high-dimensional data. In addition, several alternative NNs are employed to build mapping functions, which map data from the state space to the observation space (and vice versa), and benchmarked against the latent-space-based LNM approach. We test the LNM with real river discharge data from the UK and Ireland. The National River Flow Archive (NRFA) dataset provides the observations, and the data provided by a surrogate model from the European Flood Awareness System (EFAS) dataset serves as the state field. LNM outperforms the alternative methods in terms of accuracy and efficiency. The LNM developed can be applied to areas other than hydrology to integrate data efficiently with models.

KEYWORDS

data assimilation, latent neural mapping, neural network

1 | INTRODUCTION

In recent decades, global climate change has intensified, combined with an intensification in human activities, leading to an elevated occurrence and severity of natural hazards. Among the various natural hazards, floods, being a relatively frequent phenomenon, have in recent years inflicted significant economic losses and human casualties (Merz *et al.*, 2021). Therefore, delivering timely and precise flood predictions is essential for implementing effective flood mitigation measures and facilitating prompt evacuations, thereby reducing economic losses and safeguarding human life and safety. In the prediction of floods, it is crucial to take into account the factors contributing to flooding, with a primary focus on river discharge and precipitation (Dung *et al.*, 2022). Hence, this research centers on river discharge as the focal point of investigation and applies a methodology that integrates machine learning and data assimilation (DA), which together can help the rapid and accurate prediction of river discharge.

1.1 | Related work

DA is a technique that integrates multiple pieces of information to estimate a system's state and has been employed extensively for enhancing model prediction accuracy in dynamic simulations (Asch *et al.*, 2016; Carrassi *et al.*, 2018; Cheng *et al.*, 2023b; Gong *et al.*, 2020, 2021; Melinc & Zaplotnik, 2024). In this process, observations are typically fused with the state simulated by the predictive model to achieve a more accurate representation of the system's state. Since both the observations and the state are uncertain, DA serves the purpose of reducing the uncertainty and providing a state closer to the real values. Therefore, DA has been used for diverse applications in the domain of hydrology, particularly for predictive purposes (El Gharamti *et al.*, 2021; Ziliani *et al.*, 2019). In the prediction, the uncertainties of the predicted state are influenced by the initial conditions, meteorological, and hydrological factors. Post-processing is widely used as a method to improve the reliability of hydrological predictions (Matthews *et al.*, 2023). Compared with post-processing, DA can reduce the uncertainty and has the ability to evaluate the uncertainty (Evensen, 2003). Also, DA can integrate observations from multiple sources, such as closed-circuit television (Vandaele *et al.*, 2021, 2023). Variational DA is one type of DA for handling nonlinear problems, of which two common methods are three-dimensional variational data assimilation (3D-Var: Courtier *et al.*, 1998) and four-dimensional variational data assimilation (4D-Var: Bannister, 2017). Variational DA is accomplished by minimizing a loss function to obtain an analysis state that aligns closely with the true values. In recent years, there has been a growing body of research employing variational DA in hydrology (Larnier & Monnier, 2023; Li *et al.*, 2022; Liu *et al.*, 2020; Nearing *et al.*, 2021; Penny *et al.*, 2022; Pujol *et al.*, 2022; Zanchetta & Coulibaly, 2020). Given the prevalence of nonlinear relationships within the hydrological domain, the utilization of variational DA proves to be a more fitting approach for their resolution. However, in the current application of variational DA methods in hydrology, several challenges persist that prove to be difficult to resolve.

The first challenge involves the computational cost associated with the conventional DA and hydrology simulation process (Buizza *et al.*, 2022; Cintra *et al.*, 2016; Kurtz *et al.*, 2017; Li *et al.*, 2014). Machine learning, as a data-driven approach, offers a promising avenue to enhance simulation speed compared with conventional hydrological models (Kratzert *et al.*, 2018, 2022; Lees *et al.*, 2021; Li *et al.*, 2023; Penny *et al.*, 2022; Sikorska-Senoner & Quilty, 2021). Hence, employing machine learning as a surrogate model for simulation is a method to boost simulation speed. The machine-learning models here are offline models, meaning they do not require frequent retraining. Therefore, training neural networks is typically not considered part of the computational cost. Moreover, in temporal terms, river discharge prediction is a time-series prediction problem. This underscores the importance of analysing historical data to inform future predictions. Long short-term memory (LSTM: Hochreiter & Schmidhuber, 1997), as a machine-learning method capable of addressing long-sequence dependence issues effectively, and its variants are widely employed as surrogate models for hydrological models. In recent research (Chen *et al.*, 2023; Li *et al.*, 2022; Zhang *et al.*, 2023), it is evident that contemporary machine-learning models exhibit accuracy and efficiency in predicting river discharge. Zhang *et al.* (2023) proposes the Attention mechanism-based long short-term memory with double time sliding windows (ALSTM-DW) model, which integrates attention mechanisms, double-time sliding windows, and a weighted mean-square error loss function to enhance urban flood forecasting accuracy. The results demonstrate that the model reduces peak flow and time-to-peak errors significantly, showing strong predictive capabilities in addressing flood risks under urbanization and climate change. Li *et al.* (2022) proposed a method to calculate watershed runoff directly from two-dimensional rainfall radar maps. This approach utilizes a convolutional neural network (CNN) to capture spatial information and LSTM to model temporal information. The method demonstrates superior performance in comparison with state-of-the-art models in several hydrological metrics,

such as the Nash–Sutcliffe model efficiency coefficient (NSE) and Kling–Gupta efficiency. Chen *et al.* (2023) develop a rapid urban flood prediction model using a combination of CNN and LSTM networks, driven by hydrodynamic model simulations. The model predicts urban flooding depth quickly in under 10 seconds, with an average error of less than 6.50% between the predicted and measured depths, demonstrating its effectiveness for emergency flood control and reducing potential losses. To implement DA for high volumes of high-dimensional data, reduction methods (e.g., empirical orthogonal functions [EOFs], truncated singular value decomposition [TSVD], or Principal Component Analysis [PCA]: Asch *et al.*, 2016; Arcucci *et al.*, 2019) or compression methods (e.g., AutoEncoder [AE] and similar: Amendola *et al.*, 2020; Cheng *et al.*, 2021a; Ghorbanidehno *et al.*, 2020; Hernández & Liang, 2018, 2019; Pasetto, 2013) are commonly used. When employing a reduced or compression model in DA, known as latent DA (Buizza *et al.*, 2022), the model can learn the spatial feature representation of data in latent space, a process also referred to as representation learning (Bengio *et al.*, 2013). Nevertheless, DA in latent space necessitates an explicit observation operator to represent the relationship between the latent-state vector and the observations.

The second challenge is the lack of an explicit observation operator, which is essential in the cost function of variational DA. When dealing with compressed data that do not have the usual physical meaning, this situation implies that establishing a direct relationship between the state field and the observations is not feasible. If the relationship from the state field to the observations cannot be represented explicitly, thereby posing a challenge in constructing a cost function for variational DA, the assimilation process cannot be executed (Courtier *et al.*, 1998). Cheng *et al.* (2023a) proposed a method to build the observation operator by compressing both the state field and the observations into the same space through use of an encoder. However, the predictive model of this method necessitates surrogate function construction online, incurring substantial computational costs. Hence, a viable solution for addressing the issue of missing explicit expressions remains elusive. The current methodologies are characterized by time-consuming procedures and inevitably introduce additional errors that impact the accuracy of predicted outcomes significantly. In the domain of Earth system modeling, Geer (2024) proposed an empirical state method, which used unsupervised learning to build the connection between the observations of satellite microwave radiance and the state of sea ice. Additionally, several studies (Gettelman *et al.*, 2022; Lam *et al.*, 2022; Schneider *et al.*, 2017) have demonstrated that neural networks (NN) can effectively capture the relationship between observations and states, as well as modeling the dynamics of the Earth system.

The third challenge is local correction, induced by the sparse spatial distribution of observations. While it is not a universal problem, there are specific ways of setting up a system that can cause problems. Setting error covariance matrices could be helpful for handling sparse observations. However, estimating these error covariance matrices is very challenging in real-world DA problems due to the lack of true states. The observations are usually sparse in hydrology (Barthold *et al.*, 2010), which can result in a scenario where the background field, post DA, is such that corrections are localized to the vicinity of the locations of the observations, leaving the remainder of the area unchanged (Cheng *et al.*, 2024). Recent studies have tackled the challenge posed by sparse observations by employing field reconstruction techniques to fill in the gaps where observations are lacking (Brajard *et al.*, 2020; Chen *et al.*, 2021; Cheng *et al.*, 2024; Wang *et al.*, 2019). Field reconstruction can learn from historical observations and the state field to build the state field directly from the observations, enabling the reconstruction of unobserved locations. Nevertheless, these methods exhibit certain issues that may affect prediction accuracy adversely. These methods introduce errors during the reconstruction process, and these errors propagate persistently through the predictive model, leading to an accumulation of inaccuracies. Also, some methods (Cheng *et al.*, 2024; Wang *et al.*, 2019) incorporate additional variables as auxiliary information during the reconstruction process, resulting in heightened computational demands and extended processing times for reconstruction. Additionally, the field reconstruction operation constructs the state field from sparse observations, thereby increasing the volume of data substantially. Consequently, utilizing field reconstruction technique to assist DA amplifies the computational effort and extends the time required for the DA process.

1.2 | Contribution of the present work

In this research, to address the limitations of current methods for predicting river discharge mentioned above, a 3D-Var method in latent space for river discharge, employing NNs as the predictive model and the observation operator, is applied. This method gives predictions of river discharge for each day in the UK and Ireland region. The block diagram is shown in Figure 1. In Figure 1, firstly, the full-model state field is processed through the encoder and compressed into latent space. Then, the latent state vector is utilized to train the predictive model. Meanwhile, the observation operator of the latent neural mapping (LNM) is trained by inputting both the latent state

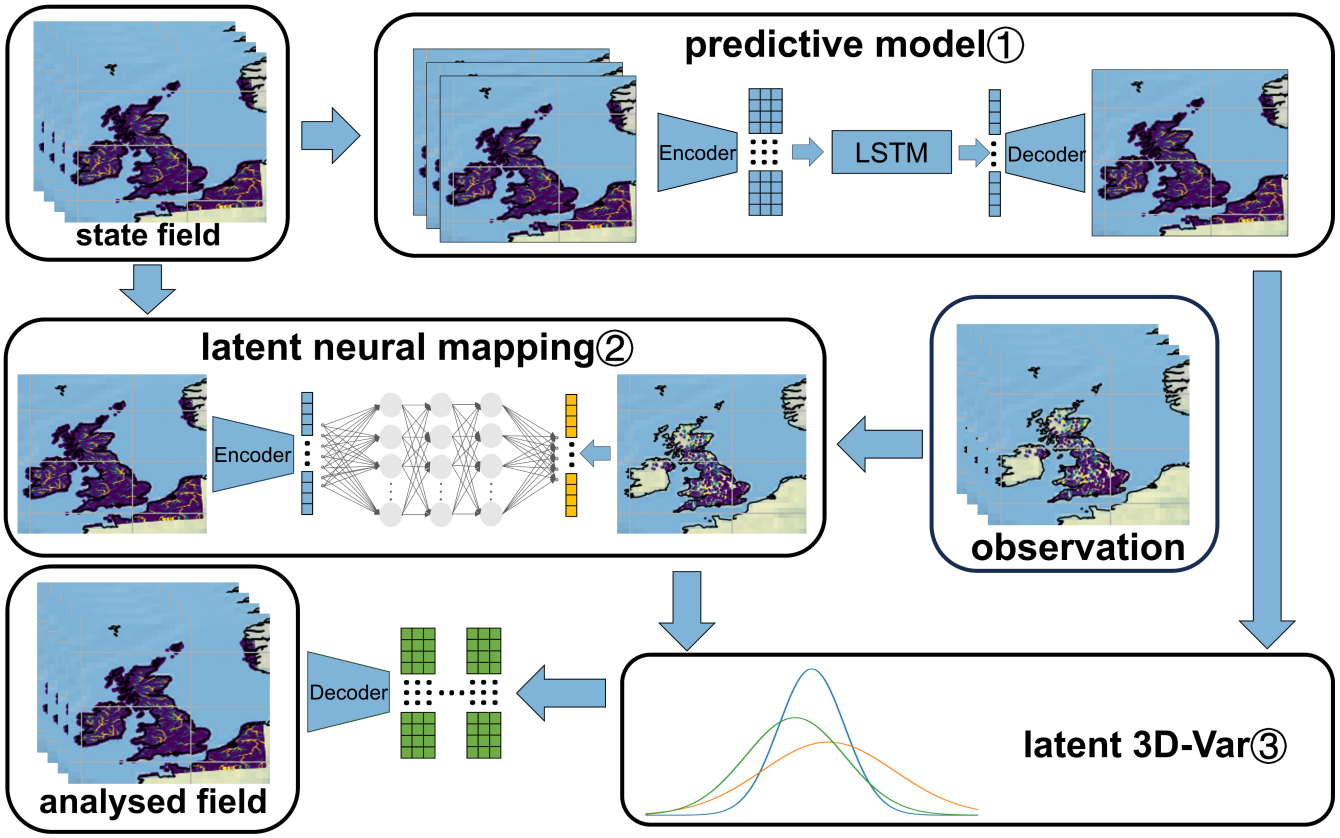

FIGURE 1 Structure of latent 3D-Var (sub-block 1 represents the predictive model, sub-block 2 represents the LNM, and sub-block 3 represents the latent 3D-Var; blue cubes represent the state field, orange cubes represent the observations, and green cubes represent the analysed state). [Colour figure can be viewed at wileyonlinelibrary.com]

vector and the observations. Once the predictive model and observation operator are trained, the process of latent 3D-Var is conducted. Subsequently, the latent analysed vector is obtained. Following the decoding process, the full-model analysed field is retrieved.

This method tackles the computational resource-intensive nature of processing kilometer-level river discharge data and the simulation process through a hydrological model. Additionally, this method employs a NN mapping to address challenges arising from the absence of an explicit observation operator from the latent state vector to the observations. Simultaneously, this method achieves the global correction of the full-model state field through representation learning.

In the proposed method, a convolutional autoencoder (CAE: Rifai *et al.*, 2011) is employed as a compression model, compressing the full-model state field into latent space. This not only reduces computational resources significantly but also, importantly, harnesses the spatial information extraction capability of the convolutional layers within the encoder structure of CAE. This allows learning of the representation of the full-model state field. Following 3D-Var on the representation in latent space, the full-model state field will be corrected globally by the decoder.

Moreover, this method employs LSTM in latent space as a surrogate model for the hydrological model, enabling forward simulations on the latent state vector by block 1 in Figure 1. LSTM is proficient in handling time-series data, and its efficiency is particularly advantageous given the abundance of historical data. Furthermore, when compared with hydrological models, lsoLSTM reduces computational resource requirements significantly.

The pivotal aspect of this methodology hinges on employing a NN as the observation operator within the DA framework, named LNM, denoted by block 2 in Figure 1. This represents the initial incorporation of a NN as the observation operator in hydrology. Current DA tools, such as OpenDA (Ridler *et al.*, 2014) and ADAO (Goeury *et al.*, 2017), rely on derivatives of the observation operator to minimize the cost function. However, these existing open-access DA packages in Python do not support a NN as observation operator or forward function in the minimization loop. Hence, this method employs open-slsocode that supports performing DA in latent space, with derivatives calculated using the PyTorch

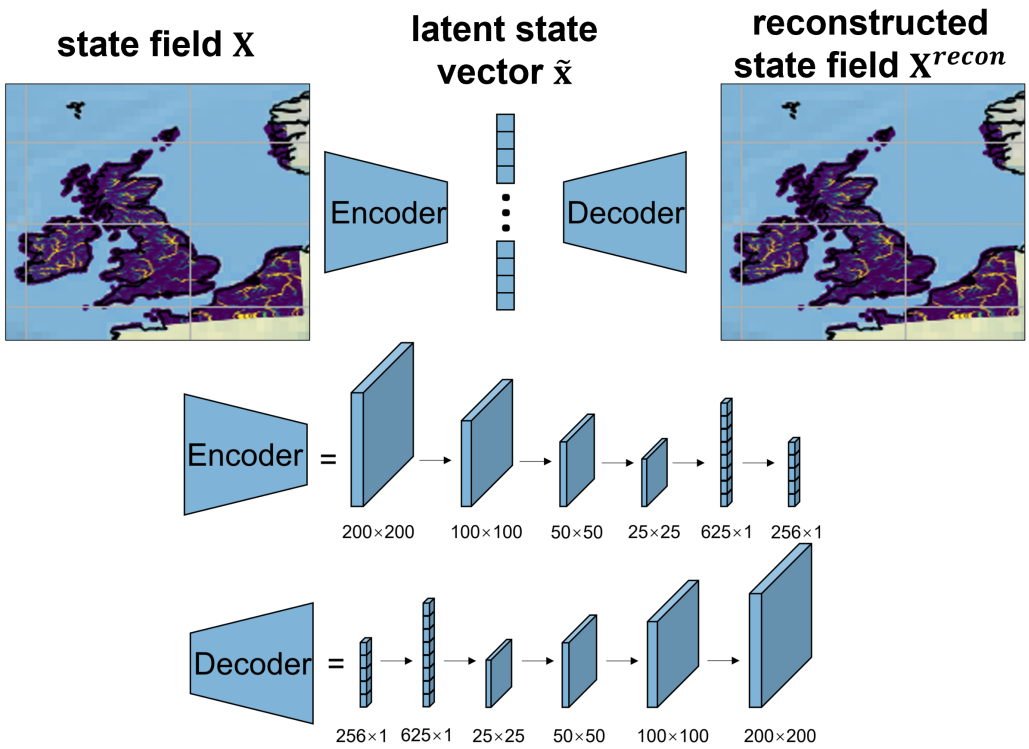

FIGURE 2 Structure of the compression model. (The input and output of the CAE is the full-model state field. The encoder and decoder consist of five layers, with the parameters of each layer shown.) [Colour figure can be viewed at wileyonlinelibrary.com]

library (Paszke *et al.*, 2019). This capability enables the NN to function as the observation operator in the DA process. Leveraging a NN as the observation operator addresses issues where there is no explicit observation operator from the latent state vector to the observations.

In summary, this article develops an integrated system for predicting river discharge. This research employs a CAE as the compression model to learn the latent representation of the full-model state field, enabling the 3D-Var process to achieve faster computation. Also, spatial correlation information captured by CAE enhances the accuracy of the 3D-Var results. Additionally, the research applies a 3D-Var method for river discharge in latent space, utilizing NNs as the predictive model to improve the prediction efficiency. Most importantly, LNM is designed to address challenges in DA arising from the absence of an observation operator that maps the latent state vector to the observations.

The article is organized as follows. Section 2 explains the methodology proposed in this study. Section 3 describes the study area, dataset, and data preprocessing methods employed. Section 4 discusses the training strategy employed for the experiment and outlines the alternative methodological designs utilized to evaluate our proposed method. Section 5 presents the experimental results and their analysis. Lastly, Section 6 provides a summary of the findings and proposes future work.

2 | METHODOLOGY

This section introduces the latent 3D-Var method for river discharge, employing NNs as the predictive model (block 1 in Figure 1) and the observation operator (block 2 in Figure 1).

2.1 | Compression model

In this research, due to the characteristics of the sample from the European Flood Awareness System (EFAS) dataset, each sample contains 14,120 non-zero values, representing areas containing rivers. With only 35.3% of the elements containing non-zero values and the rest representing background regions, the utilization of a compression model is both feasible and advantageous. Each sample is a 200×200 matrix denoted by $\mathbf{X}_{i,i=1,2,\dots,t}$, where t represents the number of samples in the dataset. The compression model is employed to compress the sample into a k-dimensional vector denoted as $\tilde{\mathbf{x}}_{i,i=1,2,\dots,t}$.

The CAE, a variant of autoencoder (AE: Hinton & Salakhutdinov, 2006), serves as a NN employed in the capacity of a compression model. The convolutional structure of CAE can extract spatial information and reduce the number of network parameters by replacing the fully connected layer structure of AE. Compared with nonlinear

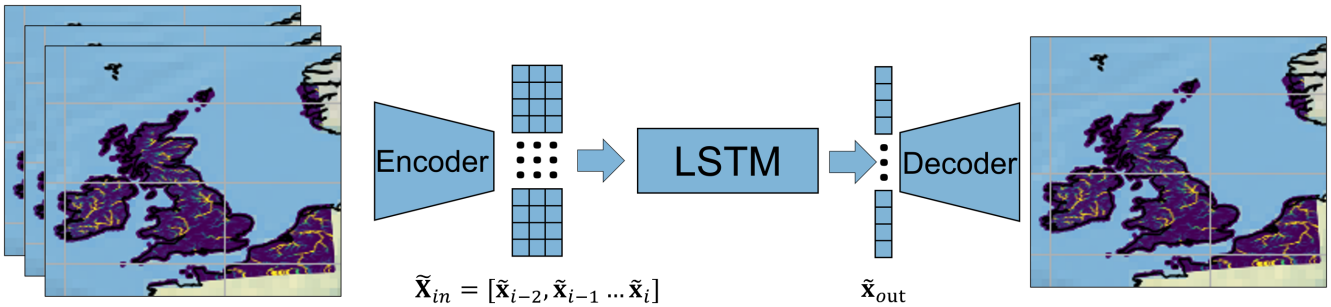

FIGURE 3 Structure of the predictive model. (The input to the LSTM comprises three time steps of the latent state vector, while the output corresponds to one time step of the latent state vector.) [Colour figure can be viewed at wileyonlinelibrary.com]

reduced-order models like EOF, CAE has better ability to handle nonlinear compression problems, offering more efficient feature extraction and data representation. CAE facilitates the training of network parameters by minimizing the disparity between the inputs and outputs of the model.

In this research, the data sourced from the EFAS dataset undergo compression into latent space through CAE. This compressed representation is then employed to build an efficient surrogate model. The architecture of the CAE consists of two primary components: an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$,

$$\begin{aligned} \mathbf{X}_i^{\text{recon}} &= \mathcal{D}(\mathcal{E}(\mathbf{X}_i)), \\ \mathcal{L} &= \frac{1}{n}\sum_{i=1}^{n} ||\mathbf{X}_i - \mathbf{X}_i^{\text{recon}}||^2, \\ \tilde{\mathbf{x}}_i &= \mathcal{E}(\mathbf{X}_i), \end{aligned} \tag{1}$$

where $\mathbf{X}_i$ represents the ith time step of the sample, $\mathbf{X}_i^{\text{recon}}$ represents the ith time step of the reconstructed sample through CAE, $\mathcal{E}$ represents the encoder, $\mathcal{D}$ represents the decoder, $\mathcal{L}$ represents the loss function between the sample and the reconstructed sample, and $\tilde{\mathbf{x}}_i$ represents the ith time step of the representation of the sample in latent space. The dimension of $\tilde{\mathbf{x}}_i$ is much smaller than that of $\mathbf{X}_i$.

In the structure of the CAE, the encoder is constructed with four convolutional layers, four downsampling layers, and two fully connected layers. Conversely, the decoder is structured with two fully connected layers, four convolutional layers, and four upsampling layers. The input to the CAE is represented by a 200×200 matrix. This matrix undergoes encoding followed by decoding processes within the CAE, resulting in the output being the reconstructed sample, denoted by another 200×200 matrix. During the training of the CAE, the primary objective is to minimize the loss function $\mathcal{L}$ that quantifies the disparity between the input and output. After training the parameters of the CAE successfully, the encoder is employed to compress the input into latent space, wherein the output of the encoder manifests as a vector with dimensions of 256×1, shown in Figure 2.

TABLE 1 The network parameters of LSTM.

Type of parameter	Parameters
Input size	256×3
Hidden size	1024×1
Cell size	1024×1
Number of layers of LSTM	2
Output size	256×1

After processing the sample in the EFAS dataset through CAE, the sample in latent space will be utilized to train the predictive model.

2.2 | Predictive model in latent space

Following the processing of data through the CAE, leading to its representation in latent space, this subsection will concentrate on formulating the predictive model for the latent state vector. In the research, the predictive model relies on LSTM for time-series prediction, a variant of recurrent NN (RNN: Hochreiter & Schmidhuber, 1997). In the prediction of river discharge, the physics model entails inputting pertinent physical quantities and forecasting future river discharge by leveraging hydrological physics formulas (Cheng *et al.*, 2021b). In the research, LSTM is utilized to emulate the physics model, harnessing its capability to address time-series prediction tasks, and designed specifically for implementation within latent space for the predictive model. The designed LSTM is an offline model, meaning that once the LSTM parameters are well-trained there is no need to reinitialize the model at each step.

The structure of LSTM is shown in Figure 3. The LSTM network parameters used in this research are presented in Table 1.

In the process of the predictive model, a sequence-to-sequence prediction is implemented, wherein three

consecutive time steps are inputted and the model outputs a single time step. Each time step corresponds to a one-day interval. The input sequence is $\tilde{\mathbf{X}}_{\text{in}}=[\tilde{\mathbf{x}}_{i-2}, \tilde{\mathbf{x}}_{i-1}, \tilde{\mathbf{x}}_i]$, denoted by a matrix of 256×3, and the output sequence is $\tilde{\mathbf{x}}_{i+1}^{\text{pred}}$, denoted by a vector of 256×1. In the training process, the input of the predictive model is derived by shifting the initial time of each time step sequentially, shown in Equation (2),

$$\begin{aligned}
\tilde{\mathbf{x}}_4^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3], \\
\tilde{\mathbf{x}}_5^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3, \tilde{\mathbf{x}}_4], \\
&\cdots, \\
\tilde{\mathbf{x}}_{i+1}^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_{i-2}, \tilde{\mathbf{x}}_{i-1}, \tilde{\mathbf{x}}_i], \\
\mathcal{L} &= \frac{1}{n}\sum_{i=1}^{n} ||\tilde{\mathbf{x}}_i^{\text{pred}} - \tilde{\mathbf{x}}_i||^2,
\end{aligned} \tag{2}$$

where $\mathcal{M}_{\text{LSTM}}$ represents the LSTM model, and $\tilde{\mathbf{x}}_i^{\text{pred}}$ represents the predicted latent state vector of the ith time step. When the predictive model is employed, the initial three time steps $[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3]$ are utilized as inputs, and each output is iterated as the input for the subsequent time step, shown in Equation (3),

$$\begin{aligned}
\tilde{\mathbf{x}}_4^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3], \\
\tilde{\mathbf{x}}_5^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_2, \tilde{\mathbf{x}}_3, \tilde{\mathbf{x}}_4^{\text{pred}}], \\
\tilde{\mathbf{x}}_6^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_3, \tilde{\mathbf{x}}_4^{\text{pred}}, \tilde{\mathbf{x}}_5^{\text{pred}}], \\
&\cdots \\
\tilde{\mathbf{x}}_{i+1}^{\text{pred}} &= \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_{i-2}^{\text{pred}}, \tilde{\mathbf{x}}_{i-1}^{\text{pred}}, \tilde{\mathbf{x}}_i^{\text{pred}}].
\end{aligned} \tag{3}$$

Upon completion of the modeling of the predictive model for the latent state vector, the next steps of the process will describe how to relate the latent state vector to the observations.

2.3 | Latent neural mapping

The latent background vector provided by the prediction of LSTM is depicted as a vector with dimensions of 256×1, whereas the observations are presented as a vector with dimensions of 924×1. As an explicit expression describing the relationship from the latent state vector to the observations is unavailable, a gap exists in the observation operator required for performing DA. This subsection presents a method aimed at addressing the issue arising from the lack of an explicit formula delineating the relationship from the latent state vector to the observations.

In this research, LNM is proposed as a novel approach for utilizing NN as the observation operator in hydrological DA, building on prior efforts in this area (Liang *et al.*, 2023; Storto *et al.*, 2021). A DA approach based on PyTorch (Paszke *et al.*, 2019) is designed to facilitate the integration of NN as the observation operator. This implies that a NN can function as an observation operator when it can capture the relationship from the latent state vector to the observations effectively. This enhancement broadens the array of choices available for assimilating the observations into the latent state vector.

Given the vectorial nature of the latent state vector and the observations (due to the sparse distribution and subsequent vectorization), and recognizing the suitability of multi-layer perceptron (MLP: Rumelhart *et al.*, 1986) for mapping between vectors, LNM incorporates an MLP as the observation operator. This MLP is utilized to map from the latent state vector to the observations,

$$\mathbf{y}_i = \mathcal{H}_{\text{LNM}}(\tilde{\mathbf{x}}_i), \tag{4}$$

where $\mathbf{y}_i$ represents the ith time step of the observations and $\mathcal{H}_{\text{LNM}}$ represents the model of LNM.

In the design of LNM, the input to the MLP network is the ith time step of the latent state vector $\tilde{\mathbf{x}}_i$, denoted by a 256×1 vector. The MLP model $\mathcal{H}_{\text{LNM}}$ comprises fully connected layers, layer normalization layers, and dropout layers, shown in Table 2.

TABLE 2 The network parameters of LNM.

Layer	Output size	Activation
Fully connected layer	384×1	LeakyReLU
LayerNorm		
Dropout = 0.1		
Fully connected layer	512×1	LeakyReLU
LayerNorm		
Dropout = 0.1		
Fully connected layer	768×1	LeakyReLU
LayerNorm		
Dropout = 0.1		
Fully connected layer	924×1	LeakyReLU
LayerNorm		
Dropout = 0.1		
Fully connected layer	1024×1	LeakyReLU
LayerNorm		
Dropout = 0.1		
Fully connected layer	2048×1	LeakyReLU
LayerNorm		
Dropout = 0.1		
Fully connected layer	924×1	

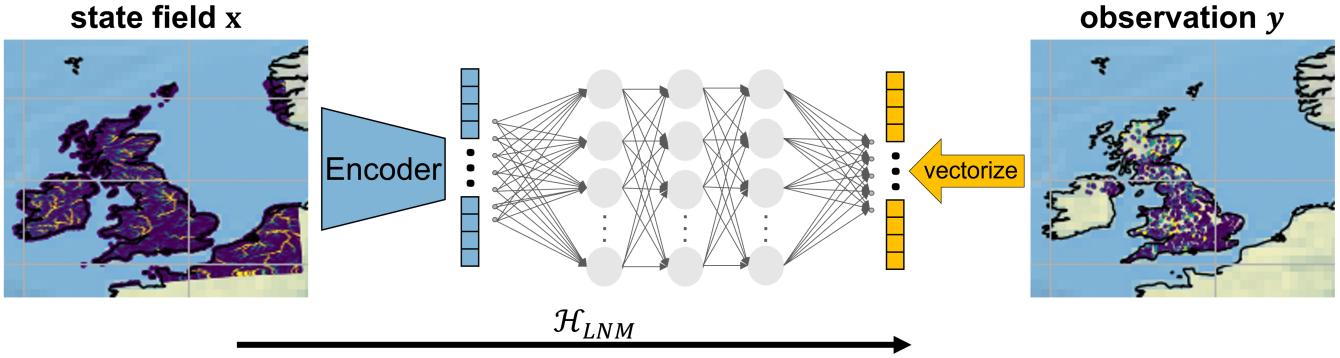

FIGURE 4 Structure of latent neural mapping. (The input of the MLP is the latent state vector and the output is the vectorized observations.) [Colour figure can be viewed at wileyonlinelibrary.com]

The output of the MLP network is the ith time step of observations $\mathbf{y}_i$, denoted by a 924×1 vector. The structure of LNM is shown in Figure 4.

After training the observation operator, the subsequent subsection elucidates how to employ LNM for 3D-Var, referred to as LNM-DA.

2.4 | Latent data assimilation

When making predictions about the state field, uncertainties are inherent in both the prediction and the observations. DA is the process of incorporating the observations into the background field, considering both of these uncertainties. This adjustment is aimed at enhancing the accuracy of the prediction. In this research, the DA method employed is 3D-Var, and it is applied to assimilate data on river discharge in latent space. Specifically, data provided by LSTM prediction are utilized as the latent background vector, while data from the National River Flow Archive (NRFA) dataset serve as the observations. The process of latent 3D-Var is shown in Equation (5),

$$\begin{aligned} J(\tilde{\mathbf{x}}) &= \frac{1}{2}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_\mathrm{b})^\mathrm{T}\tilde{\mathbf{B}}^{-1}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_\mathrm{b}) \\ &\quad + \frac{1}{2}(\mathbf{y} - \mathcal{H}(\tilde{\mathbf{x}}))^\mathrm{T}\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\tilde{\mathbf{x}})), \\ \tilde{\mathbf{x}}^\mathrm{a} &= \mathrm{argmin}(J(\tilde{\mathbf{x}})), \end{aligned} \tag{5}$$

where $\tilde{\mathbf{x}}_\mathrm{b}$ represents the latent background vector that is the output of the LSTM prediction, $\tilde{\mathbf{B}}$ and $\mathbf{R}$ represent the background-error covariance matrix in latent space and the observation-error covariance matrix, $\mathcal{H}$ represents the observation operator, $\mathbf{y}$ represents observations, and $\tilde{\mathbf{x}}_\mathrm{a}$ represents the latent analysed vector. When conducting 3D-Var in latent space, the computational workload is significantly smaller compared with 3D-Var in full space. This is because the dimension of the error covariance matrix of the latent background vector $\tilde{\mathbf{B}}$ is much smaller than the dimension of $\mathbf{B}$, which represents the error covariance matrix of the full-model background field, resulting in higher computational efficiency. More importantly, in the process of DA in latent space, the observations are utilized to correct the latent background vector. The latent background vector is derived from the full-model background field after undergoing representation learning through an encoder. Therefore, DA in latent space involves correcting the spatial features of the full-model background field, rather than correcting the entire full-model background field. Consequently, to have that correction in full space, we use decoder to achieve the reconstruction from the latent representation.

From Equation (5), in the process of minimizing the cost function it is imperative to compute the gradient of the cost function. Hence, it is essential to calculate the derivative of the observation operator. As mentioned in Section 2.3, the conventional DA methods face limitations in differentiating NN. Consequently, this study utilizes NN to train the observation operator, offering a novel contribution to the integration of NN in hydrology DA. This research utilized the "TorchDA" package,[1] a DA package based on PyTorch. PyTorch proves its ability in addressing the challenges associated with differentiating NN. Therefore, the implementation of NN as the observation operator in this research, facilitated by PyTorch, is executed seamlessly.

In the DA algorithm employed in this research, the latent representation of the full-model state field is achieved through an encoder, producing features of the state field in latent space. The features of three consecutive time steps of the latent state vector are subsequently inputted into a predictive model to predict the next time step of the latent background vector. Subsequently, LNM is employed as the observation operator to map from the latent state vector to the observations. This enables the

utilization of 3D-Var for the correction of the latent background vector. Finally, the corrected features in latent space are processed through a decoder to obtain the full-model analysed field. The whole process is shown in Algorithm 1, where $k_{\max}$ represents the number of iterations and $\mathbf{X}_{a,i}$ represents the ith time step of the full-model analysed state. AdamW (Loshchilov, 2017) is used to minimize the cost function of 3D-Var, which combines weight decay and an adaptive learning rate to optimize complex nonlinear objective functions more consistently, thus improving accuracy and convergence speed.

After the LNM-DA process, the analysed state is obtained. The following section provides an overview of the study region and dataset used in the research, along with the data preprocessing procedures.

3 | DATA

In this section, we describe the chosen study area and dataset employed in this research. The rationale behind choosing this specific research area is elaborated upon and comprehensive information regarding the content of the dataset is presented.

3.1 | Study region and dataset

The scope of this research encompasses two countries: the United Kingdom and Ireland. River discharge impacts the incidence of flooding profoundly, particularly in this region prone to floods (Sayers *et al.*, 2018). It is crucial to investigate river discharge in the region for the purpose of flood warning. Furthermore, this region is characterized by a multitude of high-flow rivers, including the Severn, the Thames, and the Trent (Macdonald & Sangster, 2017). Moreover, the geographical location of this area renders it especially vulnerable to flooding (Sayers *et al.*, 2018). Therefore, the selection of this region as the focal point for this research is appropriate, considering its susceptibility to flooding, and the abundance of available data facilitates rigorous experimentation.

This research utilizes two datasets. The first dataset comprises gridded modeled daily river discharge time series provided by River Discharge and Related Historical Data from EFAS (Mazzetti *et al.*, 2019) Version 3.5, hereafter called the EFAS dataset. The second dataset comprises river-gauge station data provided by NRFA.[2] The EFAS dataset supplies data for both the UK and Ireland, while the NRFA dataset offers data for the UK exclusively.

Firstly, EFAS is a service of the Copernicus Emergency Management Service (CEMS).[3] The system is designed specifically to identify river systems across Europe and provide predictions and early warnings for potential hydrological events in the European region. The EFAS dataset provides gridded modeled daily hydrological time series forced with meteorological observations provided by the European Meteorological Observations (EMO) dataset (Gomes *et al.*, 2020). Also, this dataset is produced by forcing the open-source LISFLOOD hydrological model with gridded observational data of precipitation and temperature at a $5\,\text{km} \times 5\,\text{km}$ resolution across the EFAS domain. Therefore, the EFAS dataset provides river discharge simulation data without assimilating the gauged river discharge observations. The dataset offers daily modeled hydrologic data, with the main variables including mean discharge in the last 24 hours, land binary mask, Lambert azimuthal equal area, latitude, and longitude. In this research, only river discharge data are used.

The NRFA dataset, provided by the UK Centre for Ecology and Hydrology, serves as the official repository for river flow data in the United Kingdom. This dataset encompasses daily hydrological data collected from 1600 hydrological gauge stations distributed across the United Kingdom, shown in Figure 5. The orange dots in the figure indicate the locations of gauge stations, while the blue lines correspond to rivers. Several stations within the dataset have been recording data since the 1800s, with the majority commencing data recording in the 1960s. Consequently, the dataset encapsulates a significant volume of historical data. Its extensive data coverage supports studies related to river hydrology and facilitates assessments of natural events within river basins.

Algorithm 1. LNM-DA.

Inputs: $\mathbf{X}, \mathbf{y}, \mathcal{E}, \mathcal{D}, \mathcal{H}_{\text{LNM}}, \mathcal{M}_{\text{LSTM}}$
Parameters: $k_{\max} = 500, \tilde{\mathbf{B}}, \mathbf{R}$
$k = 0, \tilde{\mathbf{x}}_i = \mathcal{E}(\mathbf{X}_i)$
while $k < k_{\max}$ **do**
  if $\left\lfloor \frac{i}{3} \right\rfloor == 0$ **then**
    $J(\tilde{\mathbf{x}}_i) = \frac{1}{2}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{bi})^{\mathrm{T}}\tilde{\mathbf{B}}^{-1}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_{bi}) + \frac{1}{2}$
    $(\mathbf{y}_i - \mathcal{H}_{\text{LNM}}(\tilde{\mathbf{x}}_i))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_i - \mathcal{H}_{\text{LNM}}(\tilde{\mathbf{x}}_i))$
    $\tilde{\mathbf{x}}_i^{\mathrm{a}} = \text{argmin}(J(\tilde{\mathbf{x}}_i))$
    $\tilde{\mathbf{x}}_{bi+1} = \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_{bi-2}, \tilde{\mathbf{x}}_{bi-1}, \tilde{\mathbf{x}}_i^{\mathrm{a}}]$
  end if
  $\tilde{\mathbf{x}}_{bi+1} = \mathcal{M}_{\text{LSTM}}[\tilde{\mathbf{x}}_{bi-2}, \tilde{\mathbf{x}}_{bi-1}, \tilde{\mathbf{x}}_{bi}]$
  $k = k + 1$
end while
$\mathbf{X}_i^{\mathrm{a}} = \mathcal{D}(\tilde{\mathbf{x}}_i^{\mathrm{a}})$
output: Assimilated field $\mathbf{X}_i^{\mathrm{a}}$

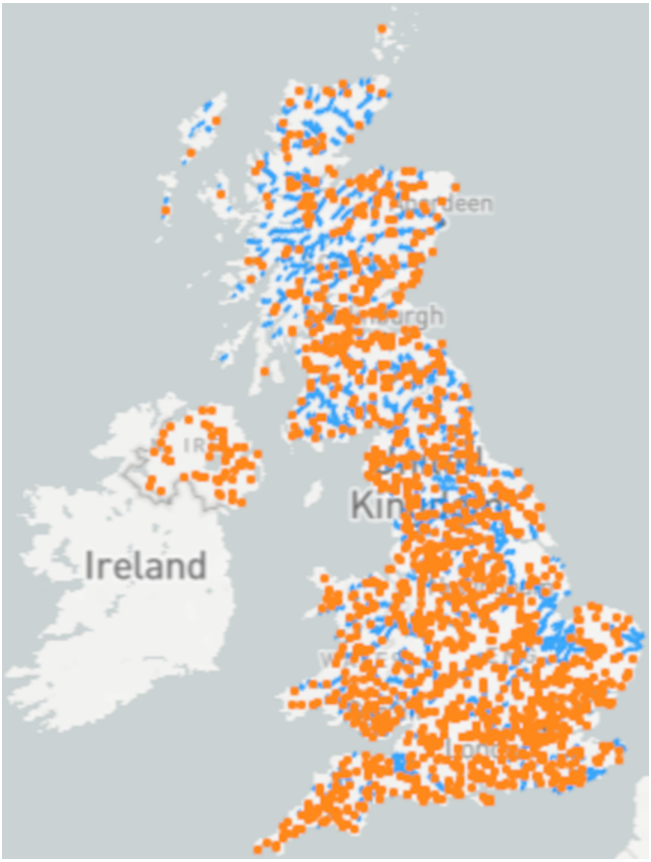

FIGURE 5 Distribution of gauge stations from the NRFA dataset. (The orange dots in the figure indicate the locations of gauge stations, while the blue lines correspond to rivers.) [Colour figure can be viewed at wileyonlinelibrary.com]

The basic information of these two datasets is presented in Table 3. The rationale for utilizing the EFAS dataset in this experiment is due to its provision of abundant historical discharge data. Consequently, these data can be employed to train the predictive model in this experiment. Additionally, the reason for incorporating the NRFA dataset is that it provides observational data from hydrological gauge stations. These data can be utilized for DA with the predictions generated by the predictive model, thereby enhancing the accuracy of prediction results.

3.2 | Data preprocessing

The data necessitate preprocessing to delineate the appropriate spatial and temporal ranges for the research. Firstly, the EFAS dataset covers Europe, encompassing a territory that extends from North Africa to the northern tip of Scandinavia, reaching to the Atlantic Ocean in the west and to the Caspian Sea in the east (Matthews *et al.*, 2022).

TABLE 3 Basic information of the EFAS and NRFA datasets.

Name	EFAS	NRFA
Study region	UK & Ireland	UK
Variables	River discharge	River discharge
Temporal scale	1991–present	1800s (most 1960s) –present
Number of stations		1600
Spatial resolution	5 km × 5 km	
Temporal resolution	Daily	Daily

TABLE 4 The data after preprocessing.

Name	Type	Temporal coverage	Number of time steps	Dimension of data
EFAS	State field	1991–2021	11,323	200 × 200
NRFA	Observation	1991–2021	11,323	924 × 1

The study region for this research is the United Kingdom and Ireland and, based on the latitude and longitude information of this area, it ranges from 49°–59°N and 11°W–2°E. Furthermore, the EFAS dataset comprises river discharge data from 1991 to the present. Data with a temporal resolution of one day, spanning from 1991 to 2021, have been selected, totaling 11,323 days.

Secondly, the NRFA dataset comprises 1600 gauge stations. Given that the timeframe of the EFAS dataset in this research is selected from 1991 to 2021, it is reasonable to choose the corresponding data within the same timeframe. Therefore, the total number of gauge stations that can cover this timeframe is 924. The data provided by these gauge stations contain some missing values. Since we require a fixed number of observations in each time step, interpolation in time is necessary for those time steps where observations are missing. Consequently, acknowledging the inherent temporal characteristics of the data, an autoregressive integrated moving average (ARIMA) (Makridakis & Hibon, 1997) approach is employed for the imputation of missing values.

The data employed for the research are presented in Table 4. The following section will elucidate the methodologies applied in the research. In the NN training phase, 70% of the data are allocated randomly to the training set, 10% to the validation set, and the remaining 20% to the test set. In the DA process, the results from the predictive model based on the EFAS dataset serve as the background

field, while the data from the NRFA dataset are treated as observations. During testing, none of the data used in training is utilized; all test data remain entirely unseen. This ensures the fairness of the experiment and provides an unbiased basis for comparison.

3.3 | Estimation of error covariance matrix

In 3D-Var, the error covariance matrix is a fundamental component that plays a crucial role. The background-error covariance matrix $\mathbf{B}$ characterizes the statistical properties of errors in the background field (Bannister, 2008). Its primary functions include representing the magnitude and spatial correlations of background errors, as well as spreading and filtering observation information. Similarly, the observation-error covariance matrix $\mathbf{R}$ encapsulates the statistical properties of errors in observational data (Janjić *et al.*, 2018). Its key function is to represent the magnitude and dependence of observation errors and to balance the weighting between observational and background information. Since our experiment is designed as a twin experiment to present a proof of concept, we treat the simulation results from the EFAS dataset as the ground truth for validating our approach.

Since the proposed LNM-DA is conducted in latent space, the formula for calculating the background-error covariance matrix in latent space $\tilde{\mathbf{B}}$ (Melinc & Zaplotnik, 2024) is shown in Equation (6):

$$\tilde{\mathbf{B}} = \langle (\tilde{\mathbf{x}}_{\mathrm{b}}^{i} - \tilde{\mathbf{x}}_{\mathrm{gt}}^{i})(\tilde{\mathbf{x}}_{\mathrm{b}}^{i} - \tilde{\mathbf{x}}_{\mathrm{gt}}^{i})^{\mathrm{T}} \rangle, \tag{6}$$

where $\tilde{\mathbf{x}}_{\mathrm{b}}^{i}$ represents the latent background vector at the ith time step, $\tilde{\mathbf{x}}_{\mathrm{gt}}^{i}$ represents the latent ground-truth vector at the ith time step, and $\langle \cdot \rangle$ denotes the average over these pairs. In this research, the background-error covariance matrix is estimated using data from both the training and validation datasets. The ground truth refers to the full-model state field from the EFAS dataset, while the latent ground-truth vector is derived from it after passing through the decoder. The latent background vector represents the prediction result of the LSTM model. The dimension of $\tilde{\mathbf{B}}$ is 256×256.

For the observation-error covariance matrix $\mathbf{R}$, we estimated it from the covariance matrix of the observation error (Janjić *et al.*, 2018), shown in Equation (7):

$$\mathbf{R} = \langle (\mathbf{y}^{i} - \mathcal{H}_{\mathrm{LNM}}(\tilde{\mathbf{x}}_{\mathrm{gt}}^{i}))(\mathbf{y}^{i} - \mathcal{H}_{\mathrm{LNM}}(\tilde{\mathbf{x}}_{\mathrm{gt}}^{i}))^{\mathrm{T}} \rangle. \tag{7}$$

The observation-error covariance matrix is estimated using data from both the training and validation datasets and is then approximated by retaining only the diagonal elements. We further assumed that each observation was independent of the others. Since the number of missing observations in the dataset is much smaller than the total number of observations, applying ARIMA for interpolation does not change the value of the observation-error covariance matrix significantly. Therefore, the observations after interpolation are used to estimate the observation-error covariance matrix in this method. The dimension of $\mathbf{R}$ is 924×924.

After estimating the latent background and observation-error covariance matrices using the training and validation datasets, these two matrices are fixed and remain unchanged throughout the 3D-Var process.

4 | TRAINING AND ALTERNATIVE METHODS

4.1 | Training strategy

This subsection delineates the strategies employed in training the NN models as discussed previously. In this research, during the training of both the predictive model and LNM, 70% of the dataset is allocated as the training set, while 10% is assigned to the validation set, and the remaining 20% is designated as the test set. The optimization strategy employed in the training process is stochastic gradient descent (SGD) with mini-batch processing (Stich, 2018). When training the NN, the loss function employed is the mean-square error (MSE) loss function for both the predictive model and LNM.

During both the network training and the DA process, the hardware used was an NVIDIA A100 Tensor Core GPU.

4.2 | Alternative methods

Currently, there is no widely recognized state-of-the-art implementation designed specifically to address the challenge posed by the lack of observational operators. In light of this, we propose two alternative methods based on our understanding of the problem for comparative analysis. While these methods do not serve as definitive benchmarks, they provide reasonable alternatives to illustrate the strengths and limitations of the proposed LNM method within this context. In the first alternative method, an MLP is employed as the observation operator from the full-model state field to the observations, termed full-space neural mapping (FNM) (Quilodrán-Casas *et al.*, 2021), shown in Equation (8), where $\mathcal{H}_{\mathrm{FNM}}$ represents the

observation operator of FNM-DA, which conducts 3D-Var in the full space.

The FNM-DA proposes a observation operator $\mathcal{H}_{\mathrm{FNM}}$ where the input is derived by flattening the non-zero values of the full-model state field, which is initially a 200×200 matrix, into a $14{,}120 \times 1$ vector, donated as $\mathbf{x}^{\mathrm{f}}$. After processing through the MLP model, the output is the observations of a 924×1 vector. The formula of the observation operator of FNM-DA is shown in Equation (8):

$$\begin{aligned} \mathbf{x}_i^{\mathrm{f}} &= \mathcal{F}(\mathbf{X}_i), \\ \mathbf{y}_i &= \mathcal{H}_{\mathrm{FNM}}(\mathbf{x}_i^{\mathrm{f}}), \end{aligned} \tag{8}$$

where $\mathcal{F}$ represents the transformation function that maps the full space to a vector space by flattening the non-zero values, $\mathbf{x}^{\mathrm{f}}$ represents the state vector after flattening, and $\mathcal{H}_{\mathrm{FNM}}$ denotes the observation operator of FNM-DA.

The structure of FNM-DA is shown in Figure 6.

The cost function of the FNM-DA is shown in Equation (9):

$$\begin{aligned} J(\tilde{\mathbf{x}}) = &\frac{1}{2}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\mathrm{b}})^{\mathrm{T}}\tilde{\mathbf{B}}^{-1}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\mathrm{b}}) \\ &+ \frac{1}{2}[\mathbf{y} - \mathcal{H}_{\mathrm{FNM}}(\mathcal{F}[\mathcal{D}(\tilde{\mathbf{x}})])]^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y} - \mathcal{H}_{\mathrm{FNM}}(\mathcal{F}[\mathcal{D}(\tilde{\mathbf{x}})])]. \end{aligned} \tag{9}$$

In the second alternative method, a CNN-based mapping strategy is proposed to map from the observations to the full-model state field, termed inverse neural mapping (INM: Wang *et al.*, 2022). This mapping strategy is rooted in the concept of field reconstruction, which goes beyond simple interpolation. The observations $\mathbf{y}_i$ serve as the input to a CNN (LeCun *et al.*, 1998), while the output of the CNN is represented by $\breve{\mathbf{X}}_i$, which implies that inputting the observation into the CNN results in a representation that resides in the same spatial domain as the full-model state field. This representation $\breve{\mathbf{X}}_i$, serving as a reconstructed field, establishes a connection between the full-model state field and the observations, shown in Equation (10):

$$\begin{aligned} \breve{\mathbf{X}}_i &= \mathcal{G}_{\mathrm{INM}}(\mathbf{y}_i), \\ \mathcal{L} &= \frac{1}{n}\sum_{i=1}^{n} ||\breve{\mathbf{X}}_i - \mathbf{X}_i||^2, \end{aligned} \tag{10}$$

where $\breve{\mathbf{X}}_i$ represents the ith time step of the reconstructed field from the observations, $\mathcal{G}_{\mathrm{INM}}$ represents the reconstruction operator from the observations to the full-model state field by CNN, and $\mathcal{L}$ represents the loss function between the reconstructed field from the observations and the full-model state field.

Considering that the observations are represented as a 924×1 vector and the full-model state field as a 200×200 matrix, a reconstruction operator based on a CNN with a structure resembling that of a decoder is employed to facilitate the mapping, shown in Figure 7.

The cost function of the INM-DA is shown in Equation (11):

$$\begin{aligned} J(\tilde{\mathbf{x}}) = &\frac{1}{2}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\mathrm{b}})^{\mathrm{T}}\tilde{\mathbf{B}}^{-1}(\tilde{\mathbf{x}} - \tilde{\mathbf{x}}_{\mathrm{b}}) \\ &+ \frac{1}{2}(\mathcal{E}[\mathcal{G}_{\mathrm{INM}}(\mathbf{y})] - \tilde{\mathbf{x}})^{\mathrm{T}}\mathbf{R}_{\mathrm{INM}}^{-1}(\mathcal{E}[\mathcal{G}_{\mathrm{INM}}(\mathbf{y})] - \tilde{\mathbf{x}}), \end{aligned} \tag{11}$$

where $\mathbf{R}_{\mathrm{INM}}$ represent the error covariance matrix of observations after being mapped by $\mathcal{G}_{\mathrm{INM}}$ and processed through $\mathcal{E}$, with a size of 256×256.

To facilitate a comparison between LNM and the two alternative methods, essential information is summarized in Table 5.

Compared with the two alternative methods, LNM exhibits a significantly smaller parameter count. This reduction is attributed to the fact that LNM maps from

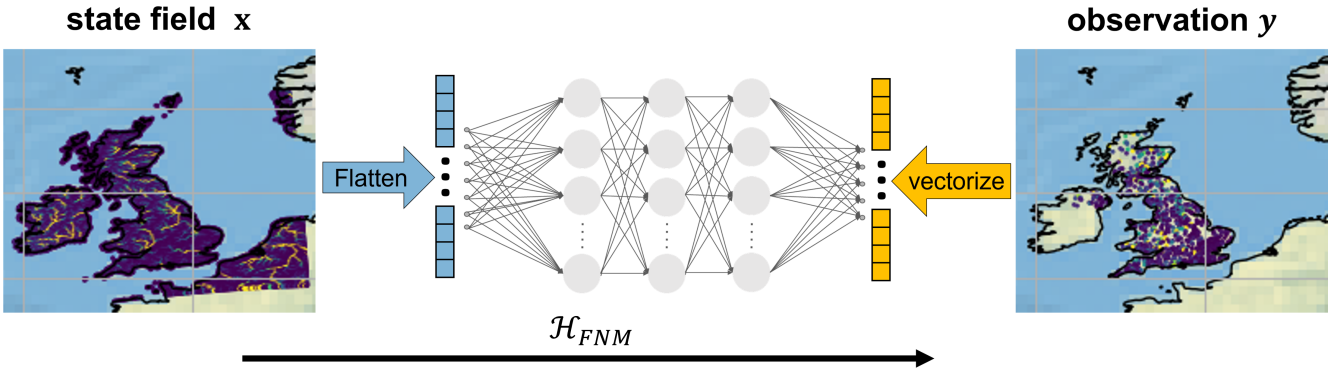

FIGURE 6 Structure of the FNM. (The input of the MLP is the flattened state vector and the output is the vectorized observations.) [Colour figure can be viewed at wileyonlinelibrary.com]

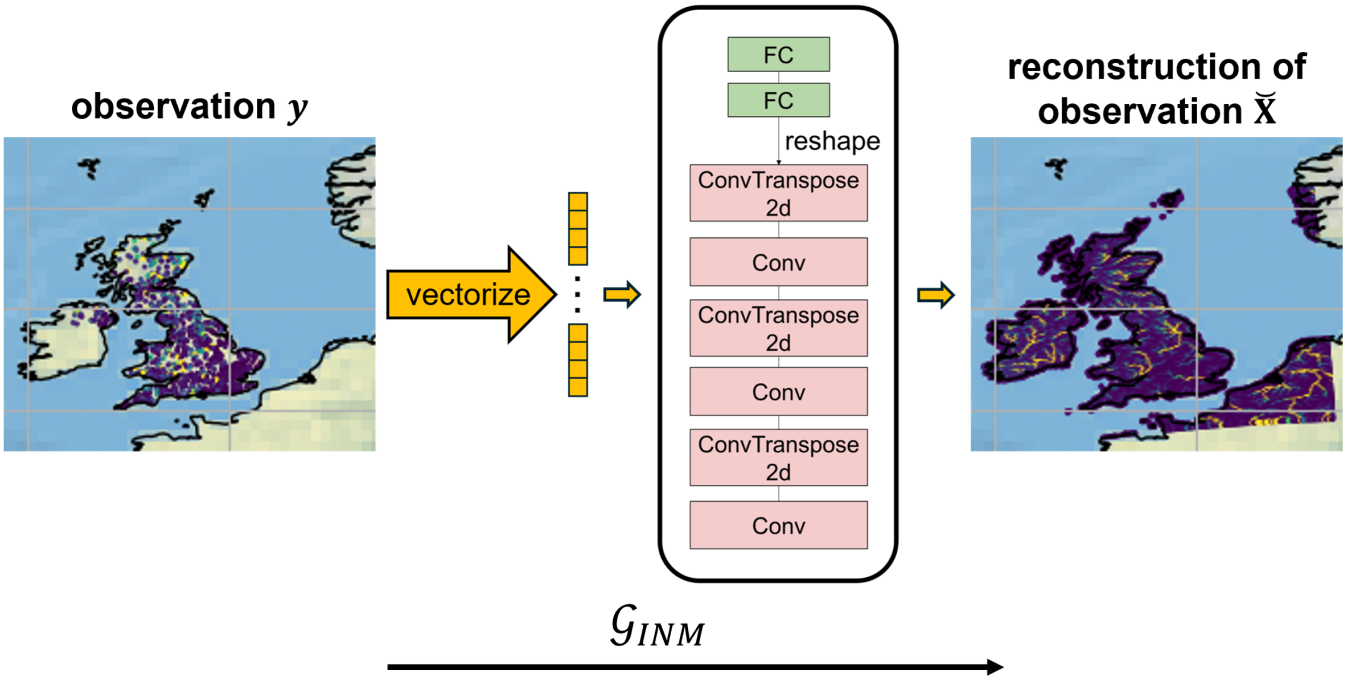

FIGURE 7 Structure of the INM. (The input of the CNN is the vectorized observation and the output is the full-model state field.) [Colour figure can be viewed at wileyonlinelibrary.com]

TABLE 5 Comparison between three mapping strategies.

Mapping strategy	LNM	FNM	INM
Input dimensions	256×1 (latent state vector)	$14{,}120 \times 1$ (flattened state vector)	924×1 (observation)
Output dimensions	924×1 (observations)	924×1 (observations)	200×200 (full-model state field)
Model architecture	MLP with FC, LayerNorm, Dropout	MLP with FC, LayerNorm, Dropout	CNN decoder-like structure
Number of parameters	$\approx$ 11.7 m	$\approx$ 204.2 m	323.9 m
Training loss (MSE)	$\approx$ 0.1	$\approx$ 0.1	$\approx$ 0.3

the latent state vector to the observations, with an input size of a 256×1 vector. In contrast, FNM takes the flattened state vector, which is a $14{,}120 \times 1$ vector, as input and the output of INM is the full-model state field, which is a matrix of 200×200. Consequently, LNM provides computational resource savings and enhanced processing speed due to its smaller input size. Compared with FNM, both LNM and INM employ convolutional layers. LNM incorporates convolutional layers when employing CAE for representation learning. The advantage of utilizing convolutional operations lies in their ability to extract spatial features. Additionally, the similar and smaller training loss of LNM and FNM compared with INM can be attributed to the fact that the training error of INM is higher. This is because INM evaluates the training error by calculating the MSE of the full-model state field, whereas LNM and FNM calculate the MSE based on the observations.

5 | RESULTS AND DISCUSSION

This experiment implements LNM-DA on two datasets, EFAS and NRFA, and compares the results of LNM-DA with those generated by two alternative methods and LSTM (without DA). The ground truth is the simulation result provided by the EFAS dataset. In the experiment, 3D-Var is conducted every day. Additionally, regularization is not applied to the alternative methods or our model. Implicit regularization is already incorporated in the background-error covariance matrix and the observation-error covariance matrix. The optimization problem exhibits good convergence and numerical stability, making additional regularization unnecessary. In this section, the results of four randomly selected time intervals, each comprising 30 sequential time steps, are employed as a demonstration to assess the performance of our approach.

5.1 | Comparison of results

Firstly, MSE is utilized as the criterion for comparing the accuracy between LNM-DA, LSTM (without DA), and two alternative methods (FNM-DA and INM-DA), shown in Equation (12):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{X}_{\mathrm{a}}^{i} - \mathbf{X}_{\mathrm{gt}}^{i})^{2}, \tag{12}$$

where i represents the time step, $\mathbf{X}_{\mathrm{a}}^{i}$ represents the full-model analysis state at the ith time step, and $\mathbf{X}_{\mathrm{gt}}^{i}$ represents the ground truth at the ith time step. The results of MSE for LNM-DA, LSTM (without DA), and two alternative methods (FNM-DA and INM-DA) are depicted in Figure 8. The ground truth is represented by the simulation data derived from the EFAS dataset.

In Figure 8, the red, blue, orange, and green symbols represent the error between the results predicted by LSTM and the ground truth, the result of LNM-DA and the ground truth, the result of FNM-DA and the ground truth, and the result of INM-DA and the ground truth, respectively. As evident from the figure, the error of LNM-DA has decreased significantly, exhibiting an average reduction of 55.9% compared with the LSTM prediction. The most substantial decrease recorded is 60.1%. In FNM-DA, the average reduction in error is 34.1%, with the most significant decrease being 40.7%. In INM-DA, the average reduction in error is 18.7%, with the most substantial decrease recorded at 21.1%. None of these methods performs as well as the ground truth. In LNM-DA, the incorporation of the CAE facilitates a structural arrangement comprising convolutional layers, thereby accommodating the positional information of both the full-model state field and the corresponding observations. This methodology stands in contrast to the two alternative methods, consequently yielding a diminished assimilation error within LNM-DA. As shown in Figure 8, the error of LNM-DA is smaller than the error of the two alternative methods. From the mean and maximum values of error, it can be concluded that LNM-DA is more accurate than the two alternative methods proposed in Section 4.2.

The rationale behind this result is that, when LNM-DA performs 3D-Var in latent space, the correction is applied to the latent representation of the background field. As a result, the latent representation of the background field is rectified. After the decoder process, the background field's representation is reconstructed into full space, indicating that the full-model analysed field is also corrected. The structure of the CAE captures the spatial correlations of the full-model state field, providing additional information to improve the correction of the field. Furthermore, the CAE acts as a denoising method (Ashfahani *et al.*, 2020; Gondara, 2016; Saad & Chen, 2020; Vincent *et al.*, 2010) and can effectively remove errors caused by noisy information that accumulate during the 3D-Var process. However, when conducting INM-DA in full space, the sparse distribution of observations across full space makes it challenging to reconstruct the full-model state field accurately. Similarly, when FNM-DA performs 3D-Var, it needs to pass through the decoder to return to full space and then use the FNM as the observation operator. This process also leads to error accumulation.

Additionally, to ensure a comprehensive comparison, we selected one month of observations from NRFA to evaluate these methods and calculate the MSE

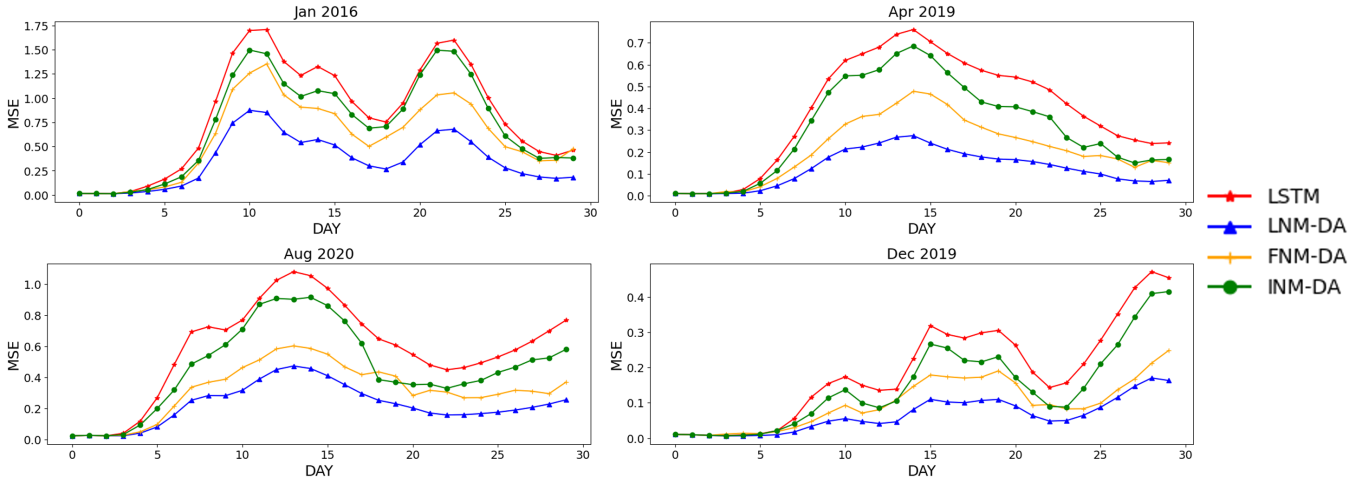

FIGURE 8 The MSE for the test data over a span of four months is compared for LNM-DA, LSTM, and two alternative methods (FNM-DA and INM-DA). [Colour figure can be viewed at wileyonlinelibrary.com]

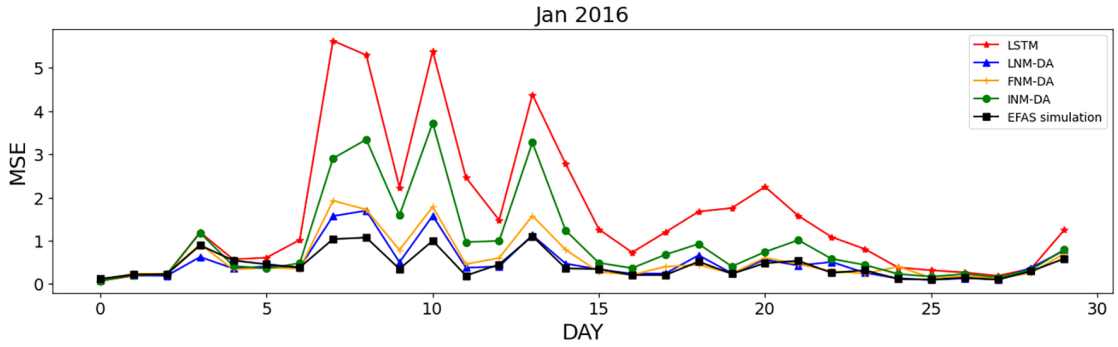

FIGURE 9 MSE against the observations from the NRFA dataset over the test data over one month is compared for LNM-DA, LSTM, two alternative methods (FNM-DA and INM-DA), and the simulation data from the EFAS dataset. [Colour figure can be viewed at wileyonlinelibrary.com]

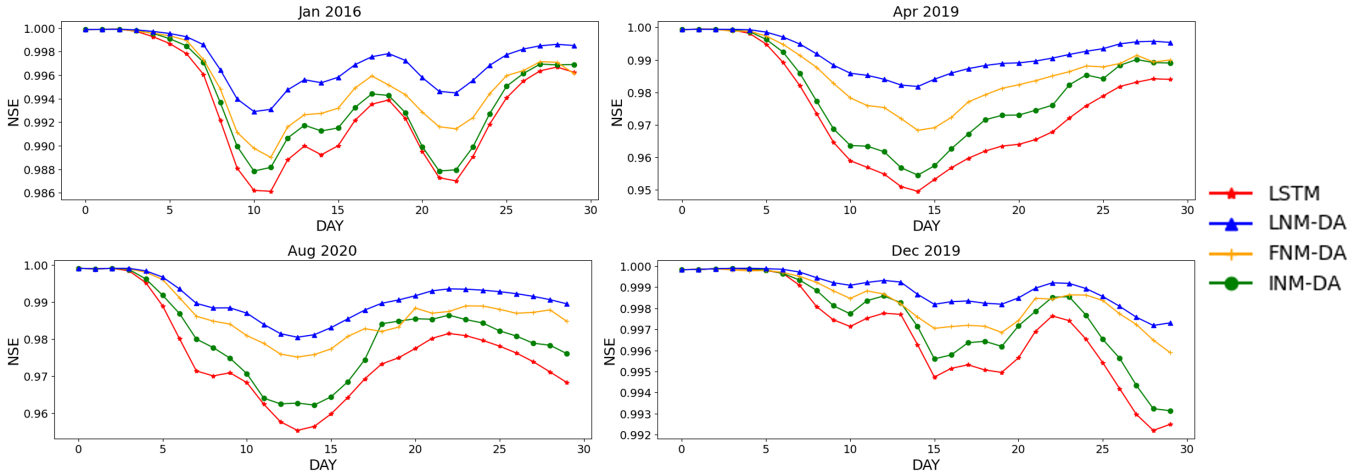

FIGURE 10 NSE over a span of four months is compared for LNM-DA, LSTM, and two alternative methods (FNM-DA and INM-DA). [Colour figure can be viewed at wileyonlinelibrary.com]

between these methods and the observations, as illustrated in Figure 9.

In Figure 9, the black symbols represent the error between the simulation data from the EFAS dataset and the observations from the NRFA dataset. The results indicate that LSTM exhibits the largest discrepancy between predicted values and observations. Similarly, the two alternative methods also show a considerable gap compared with the observations. In contrast, the predictions of our proposed LNM-DA method align closely with those of simulation data from the EFAS dataset, suggesting that LNM-DA has a smaller error compared with LSTM and the two alternative methods. Given that the observations also contain some level of uncertainty, the errors of both our method and EFAS relative to the observations are similar. Therefore, it can be concluded that our method is comparable with EFAS.

Secondly, the NSE (Nash & Sutcliffe, 1970) is a scalar to evaluate the predictive skill of hydrological models, shown in Equation (13):

$$NSE = 1 - \frac{\sum_{i=1}^{n}(Q_o^i - Q_m^i)^2}{\sum_{i=1}^{n}(Q_o^i - \overline{Q})^2}, \tag{13}$$

where Q_o^i represents the observed river discharge at the ith time step, Q_m^i represents the river discharge from the model prediction or assimilation at the ith time step, and $\overline{Q}$ represents the mean of the observed river discharge. Hence, the NSE is utilized as a criterion for comparing the performance of LNM-DA, LSTM (without DA), and two alternative methods, shown in Figure 10. The value of NSE ranges from negative infinity to 1, with values closer to 1 indicating better predictive performance of the model. From the figure, it can be seen that the value of LNM-DA is closer to 1 compared with the two alternative methods, which means that LNM-DA is a model with better predictive performance. Compared with the LSTM prediction, LNM-DA shows an average

improvement of 44.8%, FNM-DA improves by an average of 31.1%, and INM-DA achieves an average improvement of 14.5%. Therefore, the combination of two evaluation indexes, MSE and NSE, proves that LNM-DA has better performance in the prediction of river discharge.

Thirdly, the structural similarity (SSIM) is compared in Figure 11. SSIM is a metric used to assess the structural similarity of an image, ranging from a maximum value of 1 to a minimum value of −1. The larger the value of SSIM, the more structurally similar the two images are deemed to be. SSIM takes the mean, variance, and covariance into account, as shown in Equation (14):

$$
\begin{aligned}
SSIM(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i}) &= [l(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i})]^{\alpha}[c(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i})]^{\beta}[s(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i})]^{\gamma}, \\
l(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i}) &= \frac{2\mu_{\mathbf{X}_{\mathrm{a}}^{i}}\mu_{\mathbf{X}_{\mathrm{gt}}^{i}} + C_1}{\mu_{\mathbf{X}_{\mathrm{a}}^{i}}^{2} + \mu_{\mathbf{X}_{\mathrm{gt}}^{i}}^{2} + C_1}, \\
c(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i}) &= \frac{2\sigma_{\mathbf{X}_{\mathrm{a}}^{i}}\sigma_{\mathbf{X}_{\mathrm{gt}}^{i}} + C_2}{\sigma_{\mathbf{X}_{\mathrm{a}}^{i}}^{2} + \sigma_{\mathbf{X}_{\mathrm{gt}}^{i}}^{2} + C_2}, \\
s(\mathbf{X}_{\mathrm{a}}^{i}, \mathbf{X}_{\mathrm{gt}}^{i}) &= \frac{2\sigma_{\mathbf{X}_{\mathrm{a}}^{i}\mathbf{X}_{\mathrm{gt}}^{i}} + C_3}{\sigma_{\mathbf{X}_{\mathrm{a}}^{i}}^{2} + \sigma_{\mathbf{X}_{\mathrm{gt}}^{i}}^{2} + C_3},
\end{aligned}
\tag{14}
$$

where $\mathbf{X}_{\mathrm{a}}^{i}$ represents the full-model analysis field at the ith time step, $\mathbf{X}_{\mathrm{gt}}^{i}$ represents the ground truth at the ith time step, $l(\mathbf{X}_{\mathbf{a}}^{\mathbf{i}}, \mathbf{X}_{\mathrm{gt}}^{i})$ represents the mean, $c(\mathbf{X}_{\mathbf{a}}^{\mathbf{i}}, \mathbf{X}_{\mathrm{gt}}^{i})$ represents the variance, $s(\mathbf{X}_{\mathbf{a}}^{\mathbf{i}}, \mathbf{X}_{\mathrm{gt}}^{i})$ represents the covariance, α, β, and γ are coefficients, and C_1, C_2, C_3 are variables to stabilize division with a weak denominator. In this experiment, SSIM is used to compare the predicted structural similarity between the full-model analysed state and the ground truth, and the corresponding elements to be considered are the mean and variance of the full-model analysed field. The specific implementation is realized through the scikit-image library (Van der Walt *et al.*, 2014), and the parameters involved in the SSIM formulation are default values ($\alpha = \beta = \gamma = 1$, $C_1 = 0.01$, $C_2 = 0.03$, and $C_3 = 0.015$).

As shown in Figure 11, SSIM of LNM-DA demonstrates an average increase of 3% compared with the LSTM prediction. Additionally, SSIM of FNM-DA displays an average increase of 2.3%, while that of INM-DA shows an average increase of 1.3%. In most instances, the result of LNM surpasses the LSTM prediction and the two alternative methods concerning the SSIM. It can be concluded that the SSIM of the result of LNM-DA is larger than that of LSTM and the two alternative methods. Therefore, the mean and variance of the result of LNM-DA tend to be closer to the ground truth, and the covariance between the result of LNM-DA and the ground truth tends to be close to 1.

Fourthly, the standard deviation (STD) is a statistical indicator that measures the degree of distribution or variation of data, shown in Equation (15):

$$
\begin{aligned}
e_i &= \mathbf{X}_{\mathrm{a}}^{i} - \mathbf{X}_{\mathrm{gt}}^{i}, \\
\bar{e} &= \frac{1}{n}\sum_{i=1}^{n} e_i, \\
STD &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - \bar{e})^2},
\end{aligned}
\tag{15}
$$

where e_i represents the predicted error at the ith time step, and $\bar{e}$ represents the mean value of the predicted error. The results of STD for LNM-DA, LSTM, and two alternative models are depicted in Figure 12.

As shown in Figure 12, on average, the standard deviation (STD) of LNM-DA decreases by 56% compared with

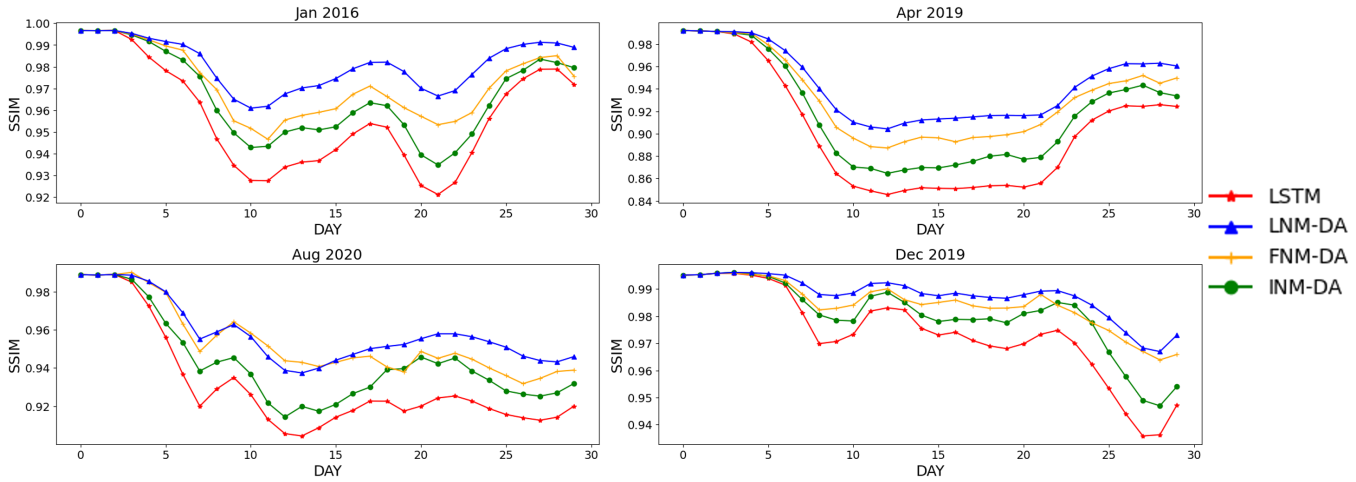

FIGURE 11 SSIM over a span of four months is compared for LNM-DA, LSTM, and two alternative methods (FNM-DA and INM-DA). [Colour figure can be viewed at wileyonlinelibrary.com]

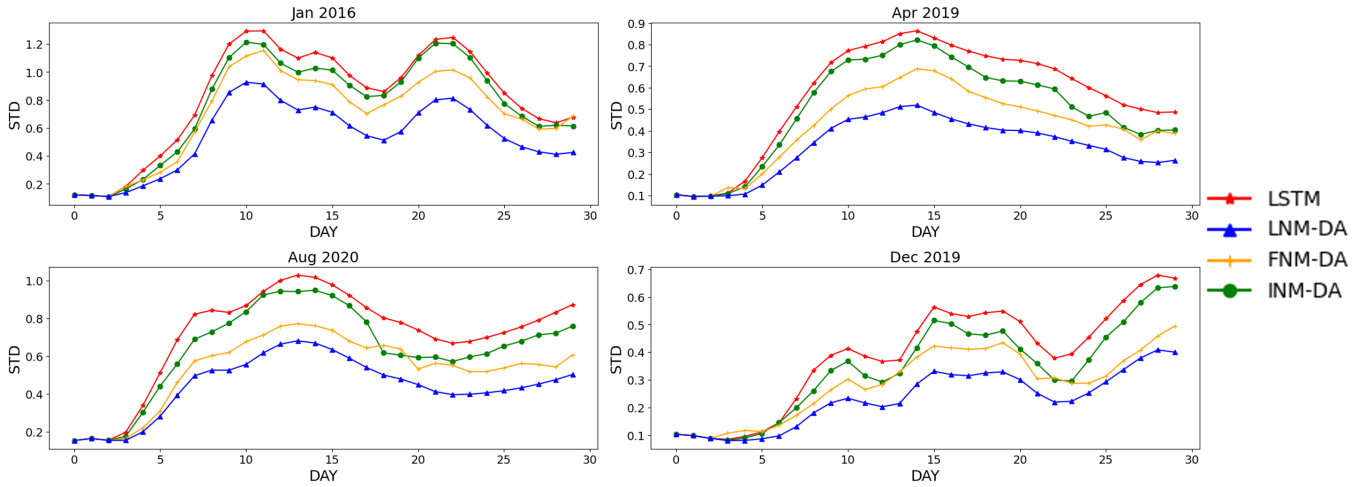

FIGURE 12 STD over a span of four months is compared for LNM-DA, LSTM, and two alternative methods (FNM-DA and INM-DA). [Colour figure can be viewed at wileyonlinelibrary.com]

the LSTM prediction, while FNM-DA achieves a 48.5% reduction and INM-DA shows a 38.4% decrease. When the STD is close to 0, it indicates that the prediction error is tightly clustered around the mean, reflecting more stable and consistent model performance. Conversely, a larger STD suggests greater variation in the prediction error, indicating that the model's performance is less consistent and stable. Therefore, compared with the LSTM prediction and the two alternative methods, LNM-DA exhibits a lower STD, indicating that the results produced by LNM-DA are more reliable and stable. This is particularly crucial for flood prediction, where consistency and stability are of utmost importance.

5.2 | Visualization of the results

To visualize the results, the 2D absolute values of errors between the full-model background field and ground truth standardized by upstream area are plotted for seven continuous chosen time steps, shown in Figure 13. This figure clearly demonstrates that all three methods (LNM-DA, FNM-DA, and INM-DA) effectively reduce the prediction error of the LSTM model to some extent. Moreover, it is evident that the error of LNM-DA is significantly smaller than that of the other two methods, highlighting its superior performance. The 2D absolute error results align closely with the outcomes of our previously calculated evaluation metrics, validating the consistency and reliability further.

Additionally, the 2D errors can be analysed by dividing the data into two regions: the UK and Ireland. The key distinction between the two areas is that the Ireland area lacks observations.

In the UK region, the errors of the three methods are more similar in the England area, as they correct the LSTM prediction error effectively. This improvement is attributed to the higher density of observation points in the England region. The abundance of observational data allows the 3D-Var process to adjust and refine the LSTM predictions better. In contrast, in the Scotland region, the errors of the two alternative models are larger than those of LNM-DA due to the sparser distribution of observation points. This indicates that our method performs significantly better in areas with limited observational data compared with the other two methods, as evidenced by the results in the figure. Similar results are observed in Northern Ireland, where LNM-DA outperforms the other two alternative methods, demonstrating its superior ability to correct prediction errors even in areas with sparse observational data. The reason for this is that FNM-DA relies on a decoder to revert to the full space during data assimilation, leading to an accumulation of model errors in the prediction process. Similarly, INM-DA's field reconstruction network exhibits larger errors in regions with sparse observations, contributing further to the accumulation of errors in these areas.

In Ireland, the results of LNM-DA are clearly superior to those of the other two alternative methods, as it corrects the LSTM predictions significantly. Among the two alternatives, INM-DA performs poorly because its field reconstruction relies on observations, which are absent in the Ireland region. This lack of observational data leads to a larger error in the field reconstruction, preventing effective correction in the data assimilation process and resulting in error accumulation. On the other hand, while FNM-DA performs better than INM-DA due to its approach, it still does not achieve the same level of performance as LNM-DA. This is because, after using the decoder, error accumulation occurs. Additionally, since the observation operator operates in full space, its impact on regions with

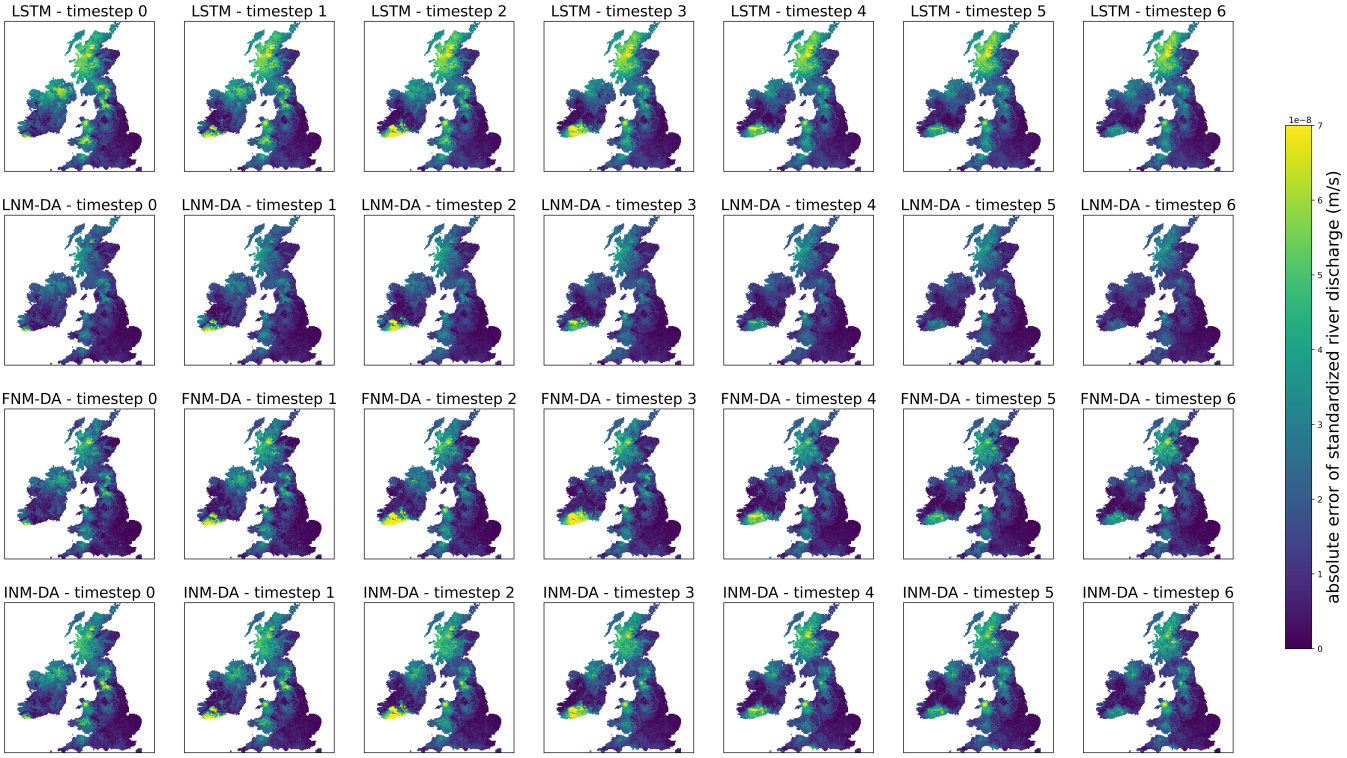

FIGURE 13 2D absolute value of errors of LNM, LSTM, and two alternative methods standardized by upstream area for seven continuous chosen time steps. [Colour figure can be viewed at wileyonlinelibrary.com]

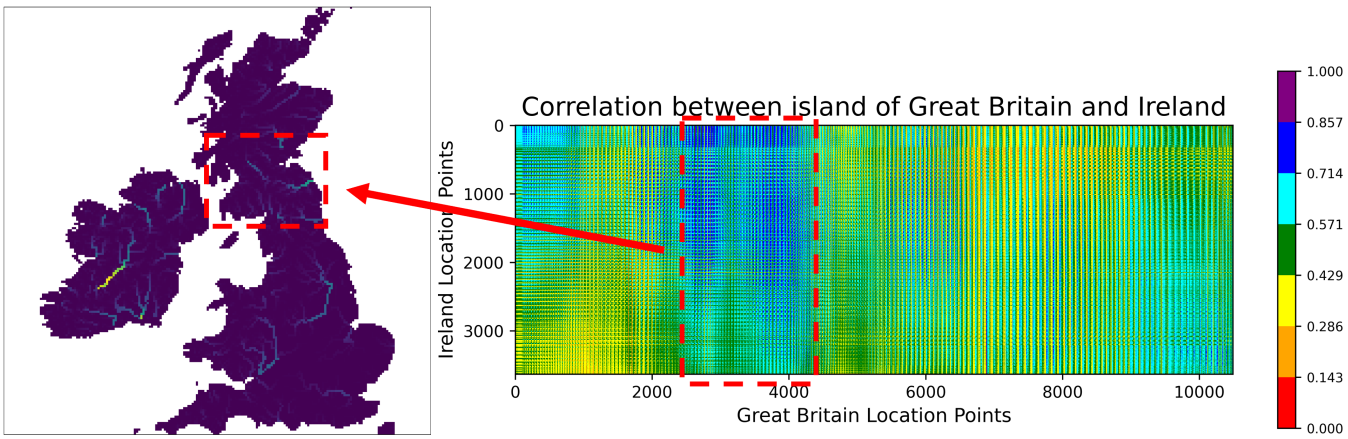

FIGURE 14 Correlation between the islands of Great Britain and Ireland. (The horizontal coordinates of the correlation plot represent location points in Great Britain, while the vertical coordinates correspond to location points in Ireland. The red-boxed regions in the right graph indicate areas where there is relatively high spatial correlation between points in Great Britain and points in Ireland, corresponding to the red-boxed regions in the left graph.) [Colour figure can be viewed at wileyonlinelibrary.com]

no observations is less effective compared with latent space. The encoder–decoder structure in LNM-DA, however, captures spatial correlations and provides additional information, enabling better corrections in areas with sparse or no observations.

To provide detailed information about the spatial correlation in the full-model state field, which helps explain why LNM-DA performs better than the other two alternative methods in areas with sparse (Scotland and North Ireland) or no observations (Ireland), the correlation between the backgrounds in the islands of Great Britain (England, Scotland, and Wales)and Ireland (Northern Ireland and Ireland) is calculated, as illustrated in Figure 14. This figure highlights the spatial relationships within the full-model state field, demonstrating how LNM-DA leverages the CAE to capture these correlations, enabling improved predictions in regions with limited or no observational data.

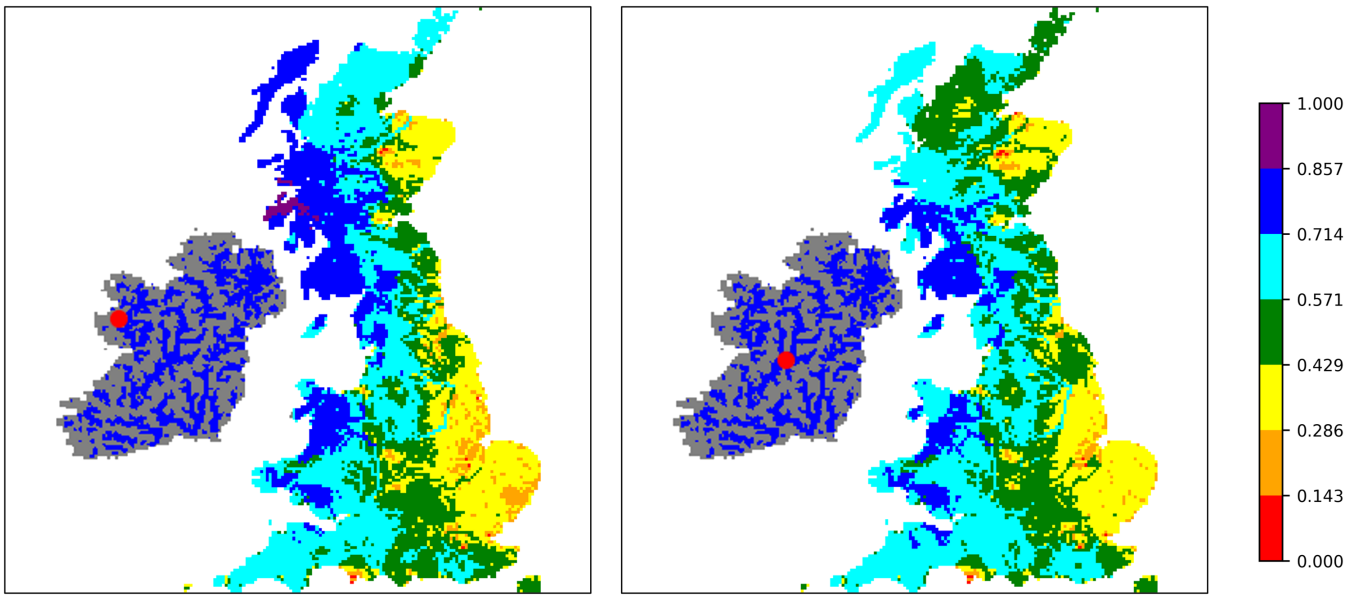

FIGURE 15 The Great Britain part of each panel illustrates the correlation corresponding to the red dot in Ireland. Meanwhile, the blue lines in the Ireland part of each panel indicate rivers with high discharge. [Colour figure can be viewed at wileyonlinelibrary.com]

In Figure 14, the horizontal axis represents the location points in Great Britain, while the vertical axis represents the location points in Ireland. The region outlined in red represents the Great Britain state field, which exhibits high correlation with the Ireland state field. This observation suggests that the patterns of river discharge are similar in the two regions. Additionally, to represent the correlation between the islands of Great Britain and Ireland more accurately, we selected two points on the island of Ireland and plotted the corresponding correlation for the island of Great Britain, shown in Figure 15.

In Figure 15, the Great Britain part of each panel illustrates the correlation corresponding to the red dot in Ireland. Meanwhile, the blue lines in the Ireland part of each panel indicate rivers with high discharge. We observe a strong correlation between the points in the Ireland region and the western coast of Great Britain, as well as a significant correlation with the red-boxed section in Figure 14. Consequently, when using LNM-DA, the CAE can capture this spatial correlation effectively, leading to improved prediction accuracy by leveraging these similarities between the UK and Ireland.

5.3 | Runtime comparison

The runtimes of the LNM-DA and two alternative methods (FNM-DA and INM-DA) are compared, shown in Table 6.

The runtime is calculated by setting the number of iterations to 500 and conducting 3D-Var for a one-day prediction on the GPU. From the table, it is evident that the running times for all three methods are around 1–2 seconds, which largely speeds up 3D-Var and meets the requirements of flood prediction. Since the dimensions of the background-error covariance matrix are the same for LNM-DA, FNM-DA, and INM-DA, and the dimension of the observation-error covariance matrix in INM-DA is smaller than those in FNM-DA and LNM-DA, INM-DA achieves the greatest running speed. Furthermore, since the observation operator in FNM-DA operates in full space, it requires to pass through the decoder during the 3D-Var process, making it the slowest method.

TABLE 6 Runtime of 3D-Var with different methods over 500 iterations for a one-day prediction.

Method	Time (s)
LNM-DA	≈ 1.5
FNM-DA	≈ 2.5
INM-DA	≈ 1

6 | CONCLUSION AND FUTURE WORK

This article proposes a LNM to address the challenges of applying 3D-Var in hydrology for the prediction of river discharge. By conducting 3D-Var in latent space, this method reduces computational cost and enhances the speed of 3D-Var, making it suitable for flood prediction. More importantly, this method resolves the issue of the lack of an observation operator from the latent state

vector to the observations by utilizing NN for this mapping. According to the results, LNM has been demonstrated to be more accurate and efficient compared with the LSTM prediction and two alternative methods.

In hydrology applications, there are several variables related to river discharge, such as soil moisture and rainfall, which do not have an explicit observation function to represent their relationship with river discharge. The LNM presented here has the ability to construct such an observation function using NN. Moreover, it can broaden the range of observation types as long as the observation can be deduced from the state field.

In future research, to achieve more accurate results, we can incorporate a Voronoi operator (Cheng *et al.*, 2024) into the DA, which can represent the relationship between the observations and the full-model state field. This Voronoi operator could contribute to obtaining more accurate results for river discharge, which is crucial for predicting floods. Additionally, we consider that 4D-Var provides more stable and accurate results than 3D-Var by incorporating the time dimension. Therefore, LNM and 4D-Var will be combined to achieve even greater accuracy. Furthermore, the temporal and spatial resolution of the state field is not adequate for predicting floods in small areas. Therefore, we need to devise a method to enhance the resolution by integrating information from observations.

ACKNOWLEDGEMENTS

The authors are grateful to their sponsors at the Resource Geophysics Academy, Imperial College London, for supporting this research. Sarah L. Dance was supported in part by the NERC National Centre for Earth Observation.

DATA AVAILABILITY STATEMENT

The data used in this article are available at NRFA at https://nrfa.ceh.ac.uk/data/search and EFAS at https://ewds.climate.copernicus.eu/datasets/efas-historical?tab=overview.

ENDNOTES

[1]https://github.com/acse-jm122/torchda.
[2]https://nrfa.ceh.ac.uk/.
[3]https://emergency.copernicus.eu/.

ORCID

Kun Wang https://orcid.org/0009-0002-7835-1364
Sarah L. Dance https://orcid.org/0000-0003-1690-3338
Rossella Arcucci https://orcid.org/0000-0002-9471-0585

REFERENCES

Amendola, M., Arcucci, R., Mottet, L., Casas, C.Q., Fan, S., Pain, C. et al. (2020) *Data assimilation in the latent space of a neural network*. arXiv preprint arXiv:201212056.

Arcucci, R., Mottet, L., Pain, C. & Guo, Y.K. (2019) Optimal reduced space for variational data assimilation. *Journal of Computational Physics*, 379, 51–69.

Asch, M., Bocquet, M. & Nodet, M. (2016) *Data assimilation: methods, algorithms, and applications*. Philadelphia, PA: SIAM.

Ashfahani, A., Pratama, M., Lughofer, E. & Ong, Y.S. (2020) DEVDAN: deep evolving denoising autoencoder. *Neurocomputing*, 390, 297–314.

Bannister, R.N. (2008) A review of forecast error covariance statistics in atmospheric variational data assimilation. I: characteristics and measurements of forecast error covariances. *Quarterly Journal of the Royal Meteorological Society*, 134(637), 1951–1970.

Bannister, R.N. (2017) A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703), 607–633.

Barthold, F.K., Wu, J., Vaché, K.B., Schneider, K., Frede, H.G. & Breuer, L. (2010) Identification of geographic runoff sources in a data sparse region: hydrological processes and the limitations of tracer-based approaches. *Hydrological Processes*, 24(16), 2313–2327.

Bengio, Y., Courville, A. & Vincent, P. (2013) Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798–1828.

Brajard, J., Carrassi, A., Bocquet, M. & Bertino, L. (2020) Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: a case study with the Lorenz 96 model. *Journal of Computational Science*, 44, 101171.

Buizza, C., Casas, C.Q., Nadler, P., Mack, J., Marrone, S., Titus, Z. et al. (2022) Data learning: integrating data assimilation and machine learning. *Journal of Computational Science*, 58, 101525.

Carrassi, A., Bocquet, M., Bertino, L. & Evensen, G. (2018) Data assimilation in the geosciences: an overview of methods, issues, and perspectives. *WIREs Climate Change*, 9(5), e535.

Chen, J., Li, Y. & Zhang, S. (2023) Fast prediction of urban flooding water depth based on CNN- LSTM. *Water*, 15(7), 1397.

Chen, X., Lee, R.M., Dwivedi, D., Son, K., Fang, Y., Zhang, X. et al. (2021) Integrating field observations and process-based modeling to predict watershed water quality under environmental perturbations. *Journal of Hydrology*, 602, 125762.

Cheng, S., Argaud, J.P., Iooss, B., Lucor, D. & Ponçot, A. (2021a) Error covariance tuning in variational data assimilation: application to an operating hydrological model. *Stochastic Environmental Research and Risk Assessment*, 35(5), 1019–1038.

Cheng, S., Chen, J., Anastasiou, C., Angeli, P., Matar, O.K., Guo, Y.K. et al. (2023a) Generalised latent assimilation in heterogeneous reduced spaces with machine learning surrogate models. *Journal of Scientific Computing*, 94(1), 11.

Cheng, S., Liu, C., Guo, Y. & Arcucci, R. (2024) Efficient deep data assimilation with sparse observations and time-varying sensors. *Journal of Computational Physics*, 496, 112581.

Cheng, S., Lucor, D. & Argaud, J.P. (2021b) Observation data compression for variational assimilation of dynamical systems. *Journal of Computational Science*, 53, 101405.

Cheng, S., Quilodrán-Casas, C., Ouala, S., Farchi, A., Liu, C., Tandeo, P. et al. (2023b) Machine learning with data assimilation and uncertainty quantification for dynamical systems: a review. *IEEE/CAA Journal of Automatica Sinica*, 10(6), 1361–1387.

Cintra, R., de Campos Velho, H. & Cocke, S. (2016) Tracking the model: data assimilation by artificial neural network. In:

Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN). Vancouver, BC: IEEE, pp. 403–410.

Courtier, P., Andersson, E., Heckley, W., Vasiljevic, D., Hamrud, M., Hollingsworth, A. et al. (1998) The ECMWF implementation of three-dimensional variational assimilation (3D-Var). I: formulation. *Quarterly Journal of the Royal Meteorological Society*, 124(550), 1783–1807.

Dung, N.B., Long, N.Q., Goyal, R., An, D.T. & Minh, D.T. (2022) The role of factors affecting flood hazard zoning using analytical hierarchy process: a review. *Earth Systems and Environment*, 6(3), 697–713.

El Gharamti, M., McCreight, J.L., Noh, S.J., Hoar, T.J., RafieeiNasab, A. & Johnson, B.K. (2021) Ensemble streamflow data assimilation using WRF-hydro and DART: novel localization and inflation techniques applied to hurricane Florence flooding. *Hydrology and Earth System Sciences*, 25(9), 5315–5336.

Evensen, G. (2003) The ensemble Kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 343–367.

Geer, A.J. (2024) Simultaneous inference of sea ice state and surface emissivity model using machine learning and data assimilation. *Journal of Advances in Modeling Earth Systems*, 16(7), e2023MS004080.

Gettelman, A., Geer, A.J., Forbes, R.M., Carmichael, G.R., Feingold, G., Posselt, D.J. et al. (2022) The future of earth system prediction: advances in model-data fusion. *Science Advances*, 8(14), eabn3488.

Ghorbanidehno, H., Kokkinaki, A., Lee, J. & Darve, E. (2020) Recent developments in fast and scalable inverse modeling and data assimilation methods in hydrology. *Journal of Hydrology*, 591, 125266.

Goeury, C., Ponçot, A., Argaud, J., Zaoui, F., Ata, R. & Audouin, Y. (2017) Optimal calibration of TELEMAC-2D models based on a data assimilation algorithm. In: *24th Telemac-Mascaret User Conference, Graz, Austria*, pp. 73–80.

Gomes, G., Thiemig, V., Skøien, J.O., Ziese, M., Rauthe-Schöch, A., Rustemeier, E. et al. (2020) *EMO: a high-resolution multi-variable gridded meteorological data set for Europe*. Dataset. European Commission, Joint Research Centre (JRC) http://data.europa.eu/89h/0bd84be4-cec8-4180-97a6-8b3adaac4d26

Gondara, L. (2016) Medical image denoising using convolutional denoising autoencoders. In: *Proceedings of the 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. Barcelona, Spain: IEEE, pp. 241–246.

Gong, H., Chen, Z., Maday, Y. & Li, Q. (2021) Optimal and fast field reconstruction with reduced basis and limited observations: application to reactor core online monitoring. *Nuclear Engineering and Design*, 377, 111113.

Gong, H., Yu, Y., Li, Q. & Quan, C. (2020) An inverse-distance-based fitting term for 3D-Var data assimilation in nuclear core simulation. *Annals of Nuclear Energy*, 141, 107346.

Hernández, F. & Liang, X. (2018) Hybridizing Bayesian and variational data assimilation for high-resolution hydrologic forecasting. *Hydrology and Earth System Sciences*, 22(11), 5759–5779.

Hernández, F. & Liang, X. (2019) Efficient data assimilation in high-dimensional hydrologic modeling through optimal spatial clustering. In: *World environmental and water resources congress 2019*. Reston, VA: American Society of Civil Engineers, pp. 334–347.

Hinton, G.E. & Salakhutdinov, R.R. (2006) Reducing the dimensionality of data with neural networks. *Science*, 313(5786), 504–507.

Hochreiter, S. & Schmidhuber, J. (1997) Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Janjić, T., Bormann, N., Bocquet, M., Carton, J., Cohn, S.E., Dance, S.L. et al. (2018) On the representation error in data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(713), 1257–1278.

Kratzert, F., Gauch, M., Nearing, G. & Klotz, D. (2022) NeuralHydrology—A python library for deep learning research in hydrology. *Journal of Open Source Software*, 7(71), 4050.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K. & Herrnegger, M. (2018) Rainfall–runoff modelling using long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11), 6005–6022.

Kurtz, W., Lapin, A., Schilling, O.S., Tang, Q., Schiller, E., Braun, T. et al. (2017) Integrating hydrological modelling, data assimilation and cloud computing for real-time management of water resources. *Environmental Modelling & Software*, 93, 418–435.

Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F. et al. (2022) *GraphCast: learning skillful medium-range global weather forecasting*. arXiv preprint arXiv:221212794.

Larnier, K. & Monnier, J. (2023) Hybrid neural network-variational data assimilation algorithm to infer river discharges from SWOT-like data. *Computational Geosciences*, 27(5), 853–877.

LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Lees, T., Reece, S., Kratzert, F., Klotz, D., Gauch, M., De Bruijn, J. et al. (2021) Hydrological concept formation inside long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences Discussions*, 2021, 1–37.

Li, B., Li, R., Sun, T., Gong, A., Tian, F., Khan, M.Y.A. et al. (2023) Improving LSTM hydrological modeling with spatiotemporal deep learning and multi-task learning: a case study of three mountainous areas on the Tibetan plateau. *Journal of Hydrology*, 620, 129401.

Li, P., Zhang, J. & Krebs, P. (2022) Prediction of flow based on a CNN-LSTM combined deep learning approach. *Water*, 14(6), 993.

Li, Z., Kan, G., Yao, C., Liu, Z., Li, Q. & Yu, S. (2014) Improved neural network model and its application in hydrological simulation. *Journal of Hydrologic Engineering*, 19(10), 04014019.

Liang, J., Terasaki, K. & Miyoshi, T. (2023) A machine learning approach to the observation operator for satellite radiance data assimilation. *Journal of the Meteorological Society of Japan Ser II*, 101(1), 79–95.

Liu, D., Jiang, W., Mu, L. & Wang, S. (2020) Streamflow prediction using deep learning neural network: case study of Yangtze River. *IEEE Access*, 8, 90069–90086.

Loshchilov, I. (2017) *Decoupled weight decay regularization*. arXiv preprint arXiv:171105101.

Macdonald, N. & Sangster, H. (2017) High-magnitude flooding across Britain since AD 1750. *Hydrology and Earth System Sciences*, 21(3), 1631–1650.

Makridakis, S. & Hibon, M. (1997) ARMA models and the box–Jenkins methodology. *Journal of Forecasting*, 16(3), 147–163.

Matthews, G., Barnard, C., Cloke, H., Dance, S.L., Jurlina, T., Mazzetti, C. et al. (2022) Evaluating the impact of post-processing medium-range ensemble streamflow forecasts from the European flood awareness system. *Hydrology and Earth System Sciences*, 26(11), 2939–2968.

Matthews, G., Cloke, H.L., Dance, S.L., Hansford, E., Mazzetti, C. & Prudhomme, C. (2023) Co-design and co-production of flood forecast products: summary of a hybrid workshop. *Bulletin of the American Meteorological Society*, 104(5), E1058–E1066.

Mazzetti, C., Decremer, D., Krzeminski, B., Barnard, C., Wetterhall, F. & Prudhomme, C. (2019) *River discharge and related historical data from the European Flood Awareness System, v3.5*. https://cds.climate.copernicus.eu/cdsapp!/dataset/efas-historical [Accessed on 15-Nov-2023 *Copernicus Climate Change Service (C3S) Climate Data Store (CDS)*.

Melinc, B. & Zaplotnik, Ž. (2024) 3D-Var data assimilation using a variational autoencoder. *Quarterly Journal of the Royal Meteorological Society*, 150(761), 2273–2295.

Merz, B., Blöschl, G., Vorogushyn, S., Dottori, F., Aerts, J.C., Bates, P. et al. (2021) Causes, impacts and patterns of disastrous river floods. *Nature Reviews Earth & Environment*, 2(9), 592–609.

Nash, J.E. & Sutcliffe, J.V. (1970) River flow forecasting through conceptual models part I–a discussion of principles. *Journal of Hydrology*, 10(3), 282–290.

Nearing, G.S., Klotz, D., Sampson, A.K., Kratzert, F., Gauch, M., Frame, J.M. et al. (2021) Data assimilation and autoregression for using near-real-time streamflow observations in long short-term memory networks. *Hydrology and Earth System Sciences Discussions*, 2021, 1–25.

Pasetto, D. (2013) *Reduced order models and data assimilation for hydrological applications*. Università degli studi di Padova.

Paszke, A. (2019) Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.

Penny, S.G., Smith, T.A., Chen, T.C., Platt, J.A., Lin, H.Y., Goodliff, M. et al. (2022) Integrating recurrent neural networks with data assimilation for scalable data-driven state estimation. *Journal of Advances in Modeling Earth Systems*, 14(3), e2021MS002843.

Pujol, L., Garambois, P.A. & Monnier, J. (2022) Multi-dimensional hydrological–hydraulic model with variational data assimilation for river networks and floodplains. *Geoscientific Model Development*, 15(15), 6085–6113.

Quilodrán-Casas, C., Arcucci, R., Mottet, L., Guo, Y. & Pain, C. (2021) *Adversarial autoencoders and adversarial LSTM for improved forecasts of urban air pollution simulations*. arXiv preprint arXiv:210406297.

Ridler, M.E., van Velzen, N., Hummel, S., Sandholt, I., Falk, A.K., Heemink, A. et al. (2014) Data assimilation framework: linking an open data assimilation library (OpenDA) to a widely adopted model interface (OpenMI). *Environmental Modelling & Software*, 57, 76–89.

Salah, R., Vincent, P., Muller, X., Glorot, X. & Bengio, Y. (2011) Contractive auto-encoders: explicit invariance during feature extraction. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 833–840.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986) Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.

Saad, O.M. & Chen, Y. (2020) Deep denoising autoencoder for seismic random noise attenuation. *Geophysics*, 85(4), V367–V376.

Sayers, P., Penning-Rowsell, E.C. & Horritt, M. (2018) Flood vulnerability, risk, and social disadvantage: current and future patterns in the UK. *Regional Environmental Change*, 18(2), 339–352.

Schneider, T., Lan, S., Stuart, A. & Teixeira, J. (2017) Earth system modeling 2.0: a blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44(24), 12–396.

Sikorska-Senoner, A.E. & Quilty, J.M. (2021) A novel ensemble-based conceptual-data-driven approach for improved streamflow simulations. *Environmental Modelling & Software*, 143, 105094.

Stich, S.U. (2018) *Local SGD converges fast and communicates little*. arXiv preprint arXiv:180509767.

Storto, A., De Magistris, G., Falchetti, S. & Oddo, P. (2021) A neural network–based observation operator for coupled ocean–acoustic variational data assimilation. *Monthly Weather Review*, 149(6), 1967–1985.

Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N. et al. (2014) Scikit-image: image processing in python. *PeerJ*, 2, e453.

Vandaele, R., Dance, S.L. & Ojha, V. (2021) Deep learning for automated river-level monitoring through river-camera images: an approach based on water segmentation and transfer learning. *Hydrology and Earth System Sciences*, 25(8), 4435–4453.

Vandaele, R., Dance, S.L. & Ojha, V. (2023) Calibrated river-level estimation from river cameras using convolutional neural networks. *Environmental Data Science*, 2, e11.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.A. & Bottou, L. (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12), 3371–3408.

Wang, Y., Chen, J. & Yang, D. (2019) Bayesian assimilation of multiscale precipitation data and sparse ground gauge observations in mountainous areas. *Journal of Hydrometeorology*, 20(8), 1473–1494.

Wang, Y., Shi, X., Lei, L. & Fung, J.C.H. (2022) Deep learning augmented data assimilation: reconstructing missing information with convolutional autoencoders. *Monthly Weather Review*, 150(8), 1977–1991.

Zanchetta, A.D. & Coulibaly, P. (2020) Recent advances in real-time pluvial flash flood forecasting. *Water*, 12(2), 570.

Zhang, L., Qin, H., Mao, J., Cao, X. & Fu, G. (2023) High temporal resolution urban flood prediction using attention-based LSTM models. *Journal of Hydrology*, 620, 129499.

Ziliani, M.G., Ghostine, R., Ait-El-Fquih, B., McCabe, M.F. & Hoteit, I. (2019) Enhanced flood forecasting through ensemble data assimilation and joint state-parameter estimation. *Journal of Hydrology*, 577, 123924.

How to cite this article: Wang, K., Cheng, S., Piggott, M.D., Dance, S.L., Wang, Y. & Arcucci, R. (2025) Latent data assimilation with non-explicit observation operator in hydrology. *Quarterly Journal of the Royal Meteorological Society*, e5009. Available from: https://doi.org/10.1002/qj.5009