Uncertainty Quantification in Inverse Models in Hydrology

SOMYA SHARMA, University of Minnesota - Twin Cities, USA
RAHUL GHOSH, University of Minnesota - Twin Cities, USA
ARVIND RENGANATHAN, University of Minnesota - Twin Cities, USA
XIANG LI, University of Minnesota - Twin Cities, USA
SNIGDHANSU CHATTERJEE, University of Minnesota - Twin Cities, USA
JOHN NIEBER, University of Minnesota - Twin Cities, USA
CHRISTOPHER DUFFY, Pennsylvania State University, USA
VIPIN KUMAR, University of Minnesota - Twin Cities, USA

In hydrology, modeling streamflow remains a challenging task due to the limited availability of basin characteristics information such as soil geology and geomorphology. These characteristics may be noisy due to measurement errors or may be missing altogether. To overcome this challenge, we propose a knowledge-guided, probabilistic inverse modeling method for recovering physical characteristics from streamflow and weather data, which are more readily available. We compare our framework with state-of-the-art inverse models for estimating river basin characteristics. We also show that these estimates offer improvement in streamflow modeling as opposed to using the original basin characteristic values. Our inverse model offers 3% improvement in R^2 for the inverse model (basin characteristic estimation) and 6% for the forward model (streamflow prediction). Our framework also offers improved explainability since it can quantify uncertainty in both the inverse and the forward model. Uncertainty quantification plays a pivotal role in improving the explainability of machine learning models by providing additional insights into the reliability and limitations of model predictions. In our analysis, we assess the quality of the uncertainty estimates. Compared to baseline uncertainty quantification methods, our framework offers 10% improvement in the dispersion of epistemic uncertainty and 13% improvement in coverage rate. This information can help stakeholders understand the level of uncertainty associated with the predictions and provide a more comprehensive view of the potential outcomes.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Modeling and simulation**; • **Applied computing** → **Physical sciences and engineering**.

Additional Key Words and Phrases: hydrology, neural networks, probabilistic models, uncertainty quantification

ACM Reference Format:
Somya Sharma, Rahul Ghosh, Arvind Renganathan, Xiang Li, Snigdhansu Chatterjee, John Nieber, Christopher Duffy, and Vipin Kumar. 2018. Uncertainty Quantification in Inverse Models in Hydrology. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Researchers in scientific communities study engineered or natural systems and their responses to external drivers. In hydrology, streamflow prediction [11, 12] is one crucial research problem for understanding hydrology cycles, flood mapping, water supply management, and other operational decisions. For a given entity (river-basin/catchment), the response (streamflow) is governed by external drivers (meteorological data) and complex physical processes specific to each entity (basin/entity characteristics). Machine learning (ML) paradigms in inverse modeling enable us to infer entity characteristics from streamflow response. In our study, for the same amount of precipitation (external driver), two river basins (entities) can have very different streamflow (response) values depending on their soil geology (entity characteristic) [26] - this presents the issue of navigating a large search space to learn one of the many right model structures. Recently, Knowledge-guided self-supervised learning (KGSSL) [12], the state-of-the-art inverse model for extracting these entity characteristics, was proposed. The framework uses a self-supervised learning paradigm, where ML models are trained using labels that can be generated without any external annotation process. However, it is not capable of quantifying uncertainty. This may present challenges in its adoption for real-life decision making. This is because in hydrology, observed data are not only impacted by measurement uncertainty in static characteristics, arising from measurement errors and use of estimation methods, but may further be affected by uncertainties arising from hydrological approximations, weather forecasting based distributional shifts, and dam regulations.

Developing inverse models that can quantify uncertainty requires addressing several challenges. Often, the measured characteristics are only surrogate variables for the actual entity characteristics, leading to inconsistencies and high uncertainty. Moreover, in real-world applications these characteristics may be essential in modeling the driver-response relation. However, they may be completely unknown, not well understood, or not present in the available set of entity characteristics. A principled method of managing this uncertainty due to imperfect data can contribute in improving trust of data-driven decision making from these methods.

In this paper, we introduce uncertainty quantification in learning representations of static characteristics. Such a framework complements explainability efforts by providing additional context and insights into the reliability, limitations, and decision-making process of machine learning models. For instance, Equifinality of hydrological modeling (different model representation result in same model results) is a widely known phenomenon affecting the adoption of hydrology models in practice [16]. Uncertainty in model structure

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

and input data are also widespread. Studying the effect of such challenges can help improve trust of water managers, improve process understanding, reduce costs and make predictions explainable [24].

To achieve this, we propose a Bayesian inverse model for simultaneously learning representations of static characteristics and quantifying uncertainty in these predictions. As a consequence, we analyze the framework's reconstruction capabilities. We modify the KGSSL autoencoder architecture such that the parameters in the encoder are estimated using the *Bayes by Backprop* weight perturbation method. This enables learning of the posterior weight distribution and uncertainty quantification in static characteristic reconstructions. We also propose an uncertainty based learning (UBL) method to reduce epistemic uncertainty (uncertainty in predictions due to imperfect model and imperfect data) in our reconstructions. This method utilizes a spectral regularization based objective formulation wherein reconstructions with higher uncertainty are penalized in the loss. We also demonstrate the improvement in streamflow prediction (in the forward model) using these robust reconstructed static characteristics (6% increase in test R^2). We provide model performance for inverse and forward models and compare it against the baselines, KGSSL [12] and CT-LSTM [20], both state-of-the-art frameworks for streamflow modeling. We also compute the coverage rate of how often the observed values lie within the bounds of the inferred static characteristics' posterior prediction distribution. In practice, this can help water managers and the public to understand if we can reliably obtain a close enough prediction, even if we are not always accurate - analysis that can not be done with the deterministic inverse model. UBL offers a 4% increase in coverage rate.

2 RELATED WORK

In hydrology, river flow modeling is a well-studied problem, with several recent advances focusing on using ML methods to build streamflow prediction models [20]. However, estimating the inverse mapping from streamflow to river basin characteristics remains less explored. Due to the problem of equifinality, the estimation of river basin characteristics still remains a challenging task. In physical sciences [6, 27, 36], several recent advances have focused on solving inverse problems. Unlike standard inversion methods in mathematics, which rely on non-linear optimization for calculating the inverse of a forward model, recent machine learning methods allow us to learn the inverse mapping from datasets. This makes it imperative to mitigate any representation error and data biases before solving the inverse problem [2]. Further, within this vast array of methodologies, the selection of the right method is crucial - since searching for an inverse mapping may be difficult due to the large search space. Bayesian optimization and iterative gradient descent-based methods may only provide a locally optimal inverse map [21]. Therefore, a principled MAP formulation and generative modeling may be viable for addressing these data-related issues [32, 35]. Moreover, inverse modeling approaches that rely on a single neural network may not accurately capture the spatio-temporal heterogeneity in basin characteristics. Also, entity characteristics can be unknown or noisy (due to measurement or estimation bias). Therefore, a robust framework for learning entity characteristics can be useful in hydrology. A recent study proved the efficacy of

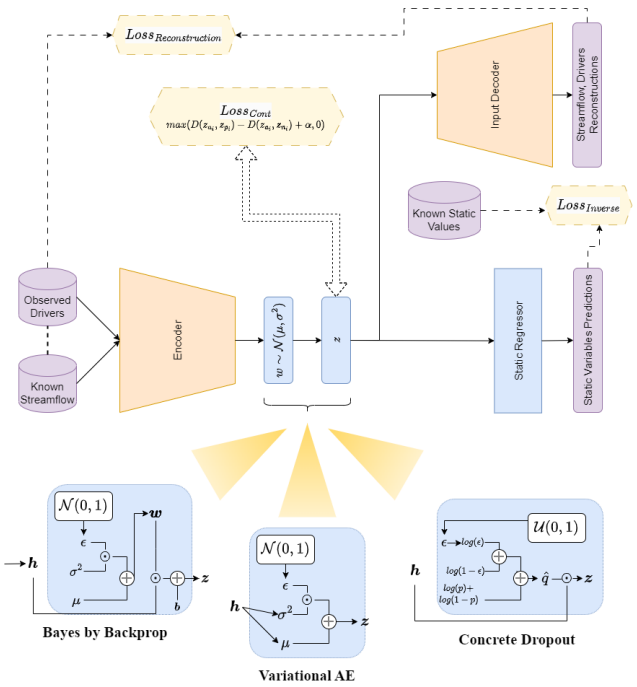


Fig. 1. Bayesian Inverse Model (BIM)

self-supervised autoencoder-based machinery for recovering static characteristics from streamflow values [12]. However, due to the measurement uncertainty and hydrological uncertainty in the input data, it is difficult to evaluate if the predictions from such models are trustworthy. In such a case, using generative models for uncertainty quantification not only allows for a complete recovery of the entity characteristics distribution but also allows us to evaluate uncertainty arising from different sources within the framework [2, 7, 35].

We develop a Bayesian inverse model for robust recovery of the complete distribution of the entity characteristics. Our framework achieves this by obtaining estimates of static variables from time series driver-response data. This, however, introduces temporal bias in our static characteristics. We also propose an uncertainty-based learning scheme to reduce the uncertainty associated with this temporal bias in inverse model estimates of static characteristics.

3 METHOD

3.1 Inverse Model

Streamflow dynamics can vary widely depending on the inherent basin-level static characteristics. Motivated by the recent success of autoencoders in estimating the static characteristics information from streamflow [12], we also incorporate an autoencoder based inverse model for learning basin characteristics. In our problem setting, each river basin , i's ($i = 1, ..., N$), weather drivers ,$x_i^j \in \mathbb{R}^{\mathcal{D}_x}$, and streamflow data, $y_i^j \in \mathbb{R}^{\mathcal{D}_y}$ can be leveraged to learn an inverse mapping to the static characteristics, $z_i^j \in \mathbb{R}^{\mathcal{D}_z}$ at the j^{th} time step [12]. The Sequence Encoder, comprised of a bidirectional LSTM, encodes the driver-response time-series. Each (forward and backward) LSTM use $[\boldsymbol{x^t}; y^t]$ input to generate the carry state and the hidden state $h = [h_{\text{forward}}; h_{\text{backward}}]$. Using a ReLU tranformation

, a linear layer is used in the encoder to get a transformation of the hidden embedding. These transformed embeddings are used as input to the LSTM decoder $\mathcal{D}$. The observed sequence $\mathcal{S}_{e_i}$ are compared with the reconstructed sequence $\hat{\mathcal{S}}_{e_i}$ from the decoder in the reconstruction loss, $\mathcal{L}_{Rec} = \frac{1}{2N} \sum_{e \in \{a,p\}} \sum_{i=1}^{N} MSE(\hat{S}_{e_i}, S_{e_i})$.

$$\begin{aligned} \boldsymbol{i_t} &= \sigma(\boldsymbol{W_i}\left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_i}) \\ \boldsymbol{f_t} &= \sigma(\boldsymbol{W_f}\left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_f}) \\ \boldsymbol{g_t} &= \sigma(\boldsymbol{W_g}\left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_g}) \\ \boldsymbol{o_t} &= \sigma(\boldsymbol{W_o}\left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_o}) \\ \boldsymbol{c_t} &= \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i} \odot \boldsymbol{g_t} \\ \boldsymbol{h_t} &= \boldsymbol{o_t} \odot \tanh(\boldsymbol{c_t}) \end{aligned} \tag{1}$$

The spatial heterogeneity among different river basins can further be leveraged to learn the differences in basin characteristics. Knowledge-guided Contrastive Loss ensures that the inherent hydrological and physical association among similar entities can allow for more efficient representation learning. The implicit physical properties (in embeddings h_{a_i} and h_{b_i}) of "positive pairs" of sequences (S_{a_i} and S_{p_i}, respectively) are compared to other entity sequences. Here, positive pairs (of sequences) refers to learning from temporal associations in the basin while negative pairs (of basins) refers to samples that enable learning from spatial correlation among basins.

$$\begin{aligned} l(a_i, p_i) = &\frac{\exp(sim(\boldsymbol{h_{a_i}}, \boldsymbol{h_{p_i}})/\tau)}{\sum_{e \in \{a,p\}} \sum_{j=1}^{N} \exp(sim(\boldsymbol{h_{a_i}}, \boldsymbol{h_{e_j}})/\tau)} \\ &+ \frac{\exp(sim(\boldsymbol{h_{p_i}}, \boldsymbol{h_{a_i}})/\tau)}{\sum_{e \in \{a,p\}} \sum_{j=1}^{N} \exp(sim(\boldsymbol{h_{p_i}}, \boldsymbol{h_{e_j}})/\tau)} \end{aligned} \tag{2}$$

where, $sim(\boldsymbol{h_{a_i}}, \boldsymbol{h_{p_i}}) = \frac{\boldsymbol{h_{a_i}}^T \boldsymbol{h_{p_i}}}{\|\boldsymbol{h_{a_i}}\| \|\boldsymbol{h_{p_i}}\|}$. Thus, the total contrastive loss for 2N such positive pairs is given as, $\mathcal{L}_{Cont} = \frac{1}{2N} \sum_{i=1}^{N} l(a_i, p_i)$.

$\mathcal{L}_{Cont}$ and $\mathcal{L}_{Rec}$ do not require any supervised information. This enables us to evaluate these losses on a large number of samples. Pseudo-Inverse Loss allows for a source of supervision to be based on the available static feature data. A feed-forward layer I on sequence encoder output is used to estimate $\hat{\mathbf{z}} = I(\mathbf{h})$.

$$\mathcal{L}_{Inv} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{z} \sum_{j=1}^{z} (z_i^j - \hat{z}_i^j)^2 \tag{3}$$

Temporal heterogeneity in driver-response time series is a source of uncertainty in static feature reconstructions. For T time steps and W window size, unc_i provides us with this standard deviation in static feature reconstruction over time,

$$\boldsymbol{unc}_i = \sqrt{\frac{W}{T} \sum_{j=1}^{T/W} (\hat{\boldsymbol{z}}_i^j - \hat{\boldsymbol{z}}_i)^2} \tag{4}$$

The objective function is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{Rec} + \lambda_2 \mathcal{L}_{Cont} + \lambda_3 \mathcal{L}_{Inv} \tag{5}$$

where, reconstruction loss $\mathcal{L}_{Rec}$ enables accurate representation learning of $[x, y]$; contrastive loss $\mathcal{L}_{Cont}$ utilizes the implicit relationships among driver-response time series data, enabling invariant approximation of static features; pseudo-inverse loss (or static loss) $\mathcal{L}_{Inv}$ utilizes available static variable information to enable accurate representation learning. The loss coefficients are learned using hyper-parameter tuning.

3.2 Uncertainty Estimation

The uncertainty in the estimation of static characteristics is obtained using a perturbation-based weight uncertainty method called *Bayes by Backprop* [3, 34]. As a method that relies on learning the posterior distribution of weight parameters, *Bayes by Backprop* can make different layers of the architecture non-deterministic. This allows us to measure and mitigate the uncertainty from different components incorporated in the framework. More recent studies also look at Bayesian deep learning models for their robustness properties [4, 5, 29].

Introducing perturbations in weights while training has historically been used as a regularization method [13, 15, 18, 23, 31]. Some recent advances utilize perturbations to induce non-deterministic behavior in supervised learning models [14, 33]. Several variations of Bayesian neural networks implement the reparameterization trick [19] to learn affine transformation of perturbation using variational inference. All these methods rely on drawing a Gaussian perturbation term $\epsilon \sim \mathcal{N}(0, 1)$. The scale and shift parameters Σ and μ can be learned by optimizing for variational free energy [14]. Therefore, the weight parameters, w, are learned as, $w = \mu + log(1 + exp(\Sigma)) \odot \epsilon$. Here, $log(1 + exp(\Sigma))$ is non-negative and differentiable. The variational parameters $\theta = \{\mu, \Sigma\}$ are minimized by variational free energy [8, 14, 17, 25, 37] that ensures a trade-off between learning a complex representation of the data (the likelihood cost) and learning a parsimonious representation similar to the prior (complexity cost). The variational free energy cost [3] can be written as,

$$\mathcal{F} = KL[q(w|\theta)||Pr(w)] - \mathbb{E}_{q(w|\theta)}[log\ Pr(\mathcal{D}|w)] \tag{6}$$

The complexity cost is the KL divergence between the learned posterior distribution of weight parameters $q(w|\theta)$ and the prior probability $Pr(w)$. The likelihood cost includes the negative log likelihood indicating the probability that the weight parameters capture the complexity of the dataset $\mathcal{D}$. Through this cost we are able to ensure that the weight distribution learns a rich representation and also does not overfit. The Gaussian perturbations in each mini-batch allows the gradient estimates of the cost to be unbiased. In our sequence encoder, we obtain ReLU transformation of the final embeddings h in a final linear layer. The weight distribution in the linear layer are learned using *Bayes by Backprop*. We also tried other model layers for learning parameter distribution (given in [30]).

3.3 Uncertainty Based Learning (UBL)

It is also imperative to manage uncertainty in complex deep learning architectures that may arise due to imperfect data. This can be achieved by penalizing static characteristics estimates with higher uncertainty. Uncertainty estimates from probabilistic models can therefore enable formulation of a regularization scheme to obtain lower uncertainty estimates. We can penalize the pseudo-inverse loss (Equation 3) such that the characteristics with higher uncertainty in the estimates will have higher loss due to bigger penalty coefficients w, such that $w \propto ||\hat{Z} - \bar{Z}||$. In our work, we prove that the optimal coefficients that penalize the pseudoinverse loss the

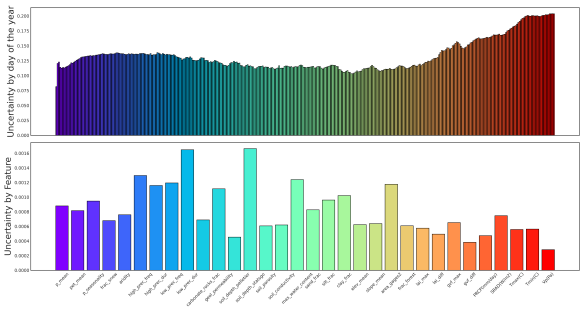


Fig. 2. Model Uncertainty by day of the year and basin characteristic variables. We see higher uncertainty for periods with greater hydrological variability (precipitation and snow events). We see higher uncertainty for soil geomorphological features (turquoise/green/ orange colored middle bars) since those are the most difficult to recover from streamflow.

most is the eigenvector that corresponds with the largest eigenvalue of σ, the epistemic uncertainty matrix [30].

4 RESULTS

Dataset: We use the CAMELS dataset, which is a publicly available hydrology dataset for multiple hydrology entities (including the 531 entities that were included in our study). The input variables for the forward model are 5 time-varying weather drivers and 27 static characteristics about the entities (climate features, soil-based, and geo-morphology-based features are included in the study. These affect the streamflow). The response variable is streamflow values. In practical setting, static characteristics information for all the entities may not be known. This makes it imperative to explore models that can provide inferred characteristics for predicting streamflow for all entities. In our inverse model, the streamflow - weather time series are used for learning static characteristics.

Experimental Setup: Daily data from years 1980 - 2000 are used for training, years 2000 - 2005 are used for validation, and year 2005 - 2015 are used for testing. We stride over half a year and use a year as lookback period for making predictions. We divide the river basins and put 400 of the river basins into train and the rest 131 into a test set. We report NSE (Nash-Sutcliff Efficiency is a measure similar to R^2 and is used to measure prediction performance in time-series hydrological models). To evaluate the uncertainty estimates, we report dispersion, coverage rate, and prediction interval width. In streamflow modeling, ensemble learning has been proven to outperform individual model prediction performance [12, 20, 22, 30]. We use an ensemble of 5 such Bayesian inverse models (BIM) to learn basin characteristic estimates and compare them with individual model predictions. More details on the experimental setup are given in the Appendix.

4.1 Static Characteristic Estimation

The Bayesian inverse model can be compared with the state-of-the-art static characteristic estimation model, KGSSL [12], in terms of prediction NSE. Table 1 shows these results along with UBL variants, wherein, we also learn penalty coefficients for different static variables penalizing those predictions that have higher epistemic uncertainty. Since KGSSL is a deterministic model, it is unable to provide coverage rate. BIM achieves higher NSE and the calibration of uncertainty using UBL mitigates the problem of under-coverage for the 95% confidence interval coverage rate. In our work, we have also shown that the UBL based calibration of uncertainty results in reducing the temporal artifacts in static characteristic predictions by 17% and also reduces the epistemic uncertainty by 36% [30].

Model	NSE	63% C.I. Coverage Rate	95% C.I. Coverage Rate
KGSSL	0.6556	-	-
BIM	**0.6858**	0.8169	0.9386
KGSSL (UBL)	0.6587	-	-
BIM (UBL)	0.6669	**0.8220**	**0.9783**

Table 1. Static reconstruction NSE and coverage rate. We can compare the static characteristic reconstruction NSE values among the deterministic and probabilistic models. Probabilistic models also ensure that our predictions will lie within the (mean $\pm z_\alpha$ s.d) interval.

Forward Model Input	Average NSE	Ensemble NSE
Baseline LSTM (original static characteristics)	0.7031	0.7238
KGSSL Estimates	0.7501	0.7574
IM + CD Estimates	0.7308	0.7502
IM + VAE Estimates	0.7333	0.7565
BIM Estimates	0.7561	0.7597
KGSSL (UBL) Estimates	0.7611	0.7582
BIM (UBL) Estimates	**0.7636**	**0.7659**

Table 2. NSE in forward model streamflow prediction using reconstructed static characteristics as input. Over 5 runs, we build 5 inverse and forward models. Average NSE is average of test NSEs obtained from each forward model. Ensemble NSE is computed from average of predictions from the 5 runs. IM stands for Inverse Model and BIM stands for Bayesian Inverse Model.

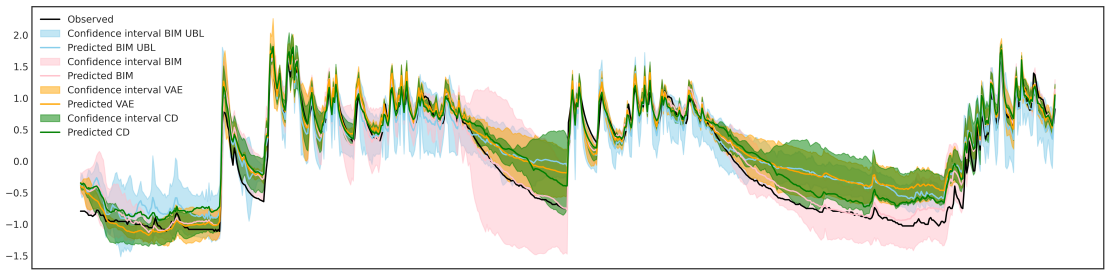
4.2 Streamflow estimation

We can also evaluate the utility of static characteristics predictions based on how they impact streamflow estimation in the forward model setup. We use LSTM model as the forward model since it has been proven to be state-of-the-art in streamflow prediction [20]. We evaluate how the forward model performance changes when we use estimates of static characteristics as input instead of the observed values. Table 2 showcases the forward model performance results, comparing using original static characteristics as input (first row in the table) to the LSTM as opposed to using predictions from the inverse model as input to forward model for forecasting streamflow in test set years. The BIM provides the best streamflow prediction model performance for individual model results (average NSE column) and ensemble results (Ensemble NSE column).

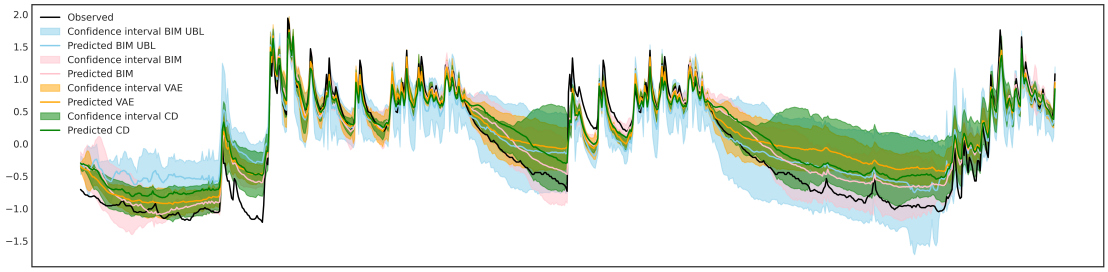
4.3 Uncertainty Quantification

To evaluate the quality of uncertainty estimates, we compare the Bayesian inverse modeling framework against two other popular uncertainty quantification methods (outlined in Figure 1). The first variant (IM + VAE) uses the reparameterization technique [19] to estimate posterior distribution for the hidden encodings as part of a variational autoencoder framework. The second variant (IM + CD) uses concrete dropout method to learn the dropout rate in a linear layer that enables estimation of the posterior for the hidden encodings [9].

Inverse model performance for IM+VAE and IM+CD models are significantly lower than other methods. The basin characteristic estimates are still useful in their ability to leverage spatial heterogeneity



(a) Test Set Prediction. Streamflow in log scale.



(b) Test Set Prediction. Streamflow in log scale.

Fig. 3. Random Test Sample Predictions. Pink line is proposed method and black line is ground truth streamflow.

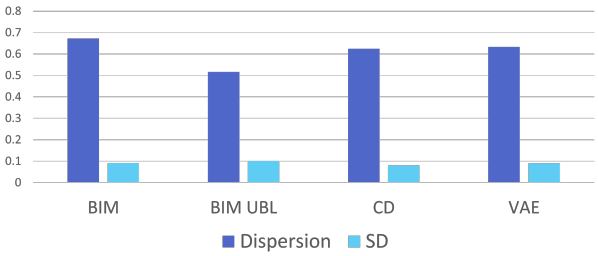


Fig. 4. Uncertainty Statistics - Dispersion and standard deviation (SD) in epistemic uncertainty in inverse model

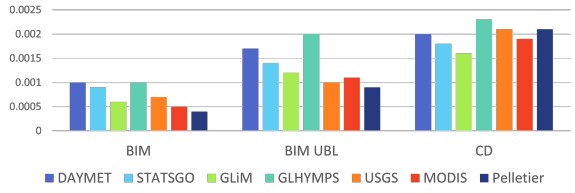
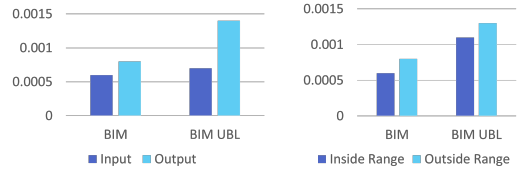


Fig. 5. Uncertainty in variables by data sources used to derive the CAMELS data variables



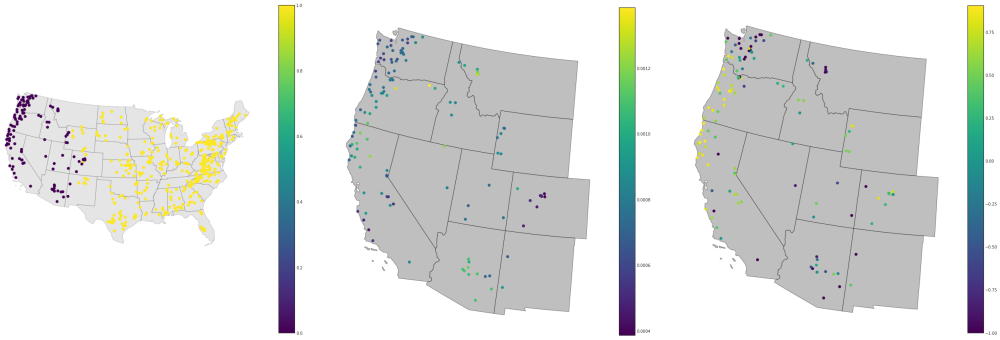
(a) Uncertainty in input reconstructions and output of the inverse model

(b) Uncertainty in predictions within and outside the observed range

Fig. 6. Uncertainty Analysis

in multi-basin streamflow modeling. This is in agreement with previous literature suggesting that randomly generated vectors or noisy characteristics can still enable LSTM to learn hydrological behavior and sustain benchmark streamflow prediction performance [22]. While, Table 2 compares the streamflow prediction capacity of these frameworks, in Figure 3, we compare the uncertainty estimates in individual predictions for randomly selected samples in the test set. We can see the BIM predictions (pink line) are relatively closer to the ground truth streamflow (black line). For other baselines, when the predictions are far from ground truth, even the 95% confidence intervals are unable to capture the ground truth streamflow values. The higher streamflow values relate to precipitation events while the slow decline after that relates to recession baseflow. The larger confidence bounds during the recession baseflow period suggests that predictions are more uncertain in these periods where additional water beyond direct precipitation is impacting streamflow. This may be because of soil-based heterogeneity in baseflow, which is difficult to estimate.

We can also compare the statistical consistency of uncertainty estimates as part of Figure 4. While a similar root mean squared calibration error of around 7.04 for all methods suggests similarity



(a) Train Test Split by Location (b) BIM Uncertainty in Streamflow (c) BIM NSE

Fig. 7. BIM Results by Test Locations

in statistical consistency, we can further compare the dispersion in the uncertainty distribution to evaluate the quality of uncertainty estimates. Higher dispersion for BIM uncertainty estimates suggests that it is a more disperse model that can potentially be more robust to distributional shifts during model inference [28]. Lower standard deviation values for uncertainty estimates also suggest better prediction interval width and improved sharpness. However, concrete dropout achieves a sharper uncertainty distribution compared to BIM.

Since the ground truth static variables are obtained from different data sources in the CAMELS dataset, in Figure 5, we evaluate the uncertainty associated with any noise in these different data sources. Within the methods used for uncertainty quantification, VAE has the highest uncertainty. DAYMET, STATSGO and GLHYMPS has relatively higher uncertainty estimates, while GLim, MODIS and Pelletier result in slightly lower uncertainties. These results are in agreement with the evidence of uncertainty in these datasets suggested in previous literature [1].

The better statistical consistency of BIM methods suggests that these uncertainty estimates are relatively more trustworthy. These uncertainty estimates can potentially be used to derive further insights. For instance, uncertainty can be compared between outputs that have different levels of supervision. Input reconstructions obtained in the inverse model decoder have a better source of supervision, while the static regressor outputs have a lower level of supervision. In Sub-figure 6a, we can see that the uncertainties associated with the input reconstructions are lower than the static regressor output uncertainties. This is also reflected in Figure 2. Similarly, predictions that lie outside the observed range should have higher uncertainty due to their implausibility. This is reflected in Sub-figure 6b, where we see a difference in uncertainty in predictions that lie within and outside the observed value ranges in the inverse model.

5 DISCUSSION

In ML applications, where prediction models may be used by stakeholders for operational decision-making, integrating uncertainty quantification methods improve the model utility and explainability. In hydrology, a probabilistic inverse model offers us the ability to infer basin characteristics that are more trust-worthy. This eliminates the need for thorough curating of large datasets that might be very expensive and time-consuming [10]. In our framework, we quantify uncertainty arising from different sources. Once such classification can be by different data sources that were used to create the CAMELS data. For instance, uncertainty is higher in static variable predictions from STATSGO data. This may be because of bias in estimates for soil depth and soil related features that were used to create the CAMELS dataset. This is also suggested in the CAMELS data paper [1]. Similarly, uncertainty estimates may also shed light on how the model behaves for different time periods (Figure 2) and under different dominant hydrological processes (Figure 3).

Uncertainty estimates can also offer insights about spatial variability in hydrological processes over different river basins. For instance, Figure 7 presents the BIM uncertainty estimates and NSE scores for the test set locations. We can notice higher uncertainties over the pacific northwest region that may arise from a higher frequency of precipitation events. Moreover, we see also see higher NSE for river basins in Northern California and Oregon as compared to other basins. There are differences in the dominant hydrological processes in the river basins that may have resulted in differences in model performance. For instance, Washington gets more high flow and high precipitation days as compared to Oregon and California. Washington also has a higher baseflow index and also higher soil depth.

In hydrology, probabilistic inverse modeling can offer many insights. Better reconstructions for variables like soil porosity and conductivity imply their impact on the streamflow generation process is easily predictable as they govern soil water storage and permeability behavior more closely. In contrast, variables like carbonate rock fraction are poorer because the fraction by itself is not directly related to flow characteristics; a more predictable alternate would be the fraction of solution channels. This effect is also showcased in the lower prediction skills of the inverse model and higher uncertainty. Therefore, model users can be more cautious about inferred basin characteristics that have higher uncertainty.

Quantifying uncertainty in hydrology can aid in assessing the reliability of the models, establishing decision thresholds for acceptable levels of uncertainty, and identification of high-risk scenarios - all of which can enable improved explainability of ML models and can provide decision-makers with a clear understanding of when to trust the model's outputs.

REFERENCES

[1] Nans Addor et al. 2017. The CAMELS data set: Catchment attributes and meteorology for large-sample studies. *Hydrology and Earth System Sciences* 21, 10 (2017), 5293–5313. https://doi.org/10.5194/hess-21-5293-2017

[2] Muhammad Asim et al. 2020. Invertible generative models for inverse problems: mitigating representation error and dataset bias. In *International Conference on Machine Learning*. PMLR, 399–409.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International Conference on Machine Learning*. PMLR, 1613–1622.

[4] Ginevra Carbone, Matthew Wicker, Luca Laurenti, Andrea Patane, Luca Bortolussi, and Guido Sanguinetti. 2020. Robustness of bayesian neural networks to gradient-based attacks. *Advances in Neural Information Processing Systems* 33 (2020), 15602–15613.

[5] Luca Cardelli, Marta Kwiatkowska, Luca Laurenti, Nicola Paoletti, Andrea Patane, and Matthew Wicker. 2019. Statistical guarantees for the robustness of Bayesian neural networks. *arXiv preprint arXiv:1903.01980* (2019).

[6] Phuong D Dao et al. 2021. Improving hyperspectral image segmentation by applying inverse noise weighting and outlier removal for optimal scale selection. *ISPRS Journal of Photogrammetry and Remote Sensing* 171 (2021), 348–366.

[7] Arka Daw, M Maruf, and Anuj Karpatne. 2021. PID-GAN: A GAN Framework based on a Physics-informed Discriminator for Uncertainty Quantification with Physics. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 237–247.

[8] Karl Friston, Jérémie Mattout, Nelson Trujillo-Barreto, John Ashburner, and Will Penny. 2007. Variational free energy and the Laplace approximation. *Neuroimage* 34, 1 (2007), 220–234.

[9] Yarin Gal, Jiri Hron, and Alex Kendall. 2017. Concrete dropout. *arXiv preprint arXiv:1705.07832* (2017).

[10] Timnit Gebru, Judy Hoffman, and Li Fei-Fei. 2017. Fine-grained recognition in the wild: A multi-task domain adaptation approach. In *Proceedings of the IEEE international conference on computer vision*. 1349–1358.

[11] Sujan Ghimire, Zaher Mundher Yaseen, Aitazaz A Farooque, Ravinesh C Deo, Ji Zhang, and Xiaohui Tao. 2021. Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks. *Scientific Reports* 11, 1 (2021), 1–26.

[12] Rahul Ghosh, Arvind Renganathan, Kshitij Tayal, Xiang Li, Ankush Khandelwal, Xiaowei Jia, Christopher Duffy, John Nieber, and Vipin Kumar. 2022. Robust Inverse Framework using Knowledge-guided Self-Supervised Learning: An application to Hydrology. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 465–474.

[13] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. In *International conference on machine learning*. PMLR, 1319–1327.

[14] Alex Graves. 2011. Practical variational inference for neural networks. *Advances in neural information processing systems* 24 (2011), 2348–2356.

[15] Stephen Hanson and Lorien Pratt. 1988. Comparing biases for minimal network construction with back-propagation. *Advances in neural information processing systems* 1 (1988), 177–185.

[16] Younggu Her, Seung-Hwan Yoo, Jaepil Cho, Syewoon Hwang, Jaehak Jeong, and Chounghyun Seong. 2019. Uncertainty in hydrological analysis of climate change: multi-parameter vs. multi-GCM ensemble predictions. *Scientific reports* 9, 1 (2019), 1–22.

[17] Tommi S Jaakkola and Michael I Jordan. 2000. Bayesian parameter estimation via variational methods. *Statistics and Computing* 10, 1 (2000), 25–37.

[18] Guoliang Kang, Jun Li, and Dacheng Tao. 2016. Shakeout: A new regularized deep neural network training scheme. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[19] Durk P Kingma, Tim Salimans, and Max Welling. 2015. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems* 28 (2015), 2575–2583.

[20] Frederik Kratzert et al. 2019. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences* 23, 12 (2019), 5089–5110.

[21] Alexander Lavin, Hector Zenil, Brooks Paige, David Krakauer, Justin Gottschlich, Tim Mattson, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, et al. 2021. Simulation Intelligence: Towards a New Generation of Scientific Methods. *arXiv preprint arXiv:2112.03235* (2021).

[22] Xiang Li, Ankush Khandelwal, Xiaowei Jia, Kelly Cutler, Rahul Ghosh, Arvind Renganathan, Shaoming Xu, JL Nieber, Christopher J Duffy, Michael Steinbach, et al. 2022. Regionalization in a global hydrologic deep learning model: from physical descriptors to random vectors. (2022).

[23] Yinan Li and Fang Liu. 2016. Whiteout: Gaussian adaptive noise regularization in deep neural networks. *arXiv preprint arXiv:1612.01490* (2016).

[24] Hilary K McMillan, Ida K Westerberg, and Tobias Krueger. 2018. Hydrological data uncertainty and its implications. *Wiley Interdisciplinary Reviews: Water* 5, 6 (2018), e1319.

[25] Radford M Neal and Geoffrey E Hinton. 1998. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*. Springer, 355–368.

[26] Andrew J Newman et al. 2015. Gridded ensemble precipitation and temperature estimates for the contiguous United States. *Journal of Hydrometeorology* 16, 6 (2015), 2481–2500.

[27] Petr Pecha et al. 2021. Determination of radiological background fields designated for inverse modelling during atypical low wind speed meteorological episode. *Atmospheric Environment* 246 (2021), 118105.

[28] Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. 2023. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *J. Comput. Phys.* (2023), 111902.

[29] Somya Sharma and Snigdhansu Chatterjee. 2021. Winsorization for Robust Bayesian Neural Networks. *Entropy* 23, 11 (2021), 1546.

[30] Somya Sharma, Rahul Ghosh, Arvind Renganathan, Xiang Li, Snigdhansu Chatterjee, John Nieber, Christopher Duffy, and Vipin Kumar. 2023. Probabilistic Inverse Modeling: An Application in Hydrology. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 847–855.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[32] He Sun and Katherine L Bouman. 2020. Deep probabilistic imaging: Uncertainty quantification and multi-modal solution characterization for computational imaging. *arXiv preprint arXiv:2010.14462* 9 (2020).

[33] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International conference on machine learning*. PMLR, 1058–1066.

[34] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. 2018. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386* (2018).

[35] Jay Whang, Qi Lei, and Alex Dimakis. 2021. Solving inverse problems with a flow-based noise model. In *International Conference on Machine Learning*. PMLR, 11146–11157.

[36] R Iestyn Woolway et al. 2021. Winter inverse lake stratification under historic and future climate change. *Limnology and Oceanography Letters* (2021).

[37] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. 2000. Generalized belief propagation. In *NIPS*, Vol. 13. 689–695.

A REPRODUCIBILITY

The code and data is shared here. The daily-level CAMELS dataset used in this study is available here. The framework is built in PyTorch, with the Bayes by Backprop module being a modification from Blitz package.

B EVALUATING MODEL PERFORMANCE:

To ensure there is no data leakage when using basin characteristics predictions from the inverse model to make predictions in the test set using the forward model, we estimate basin characteristics using the validation data. The inverse model, $g_{\mathcal{S}}$, is trained on training set $\mathcal{S}$ such that, $g_S : [x_t^i, y_t^i] \rightarrow z_i$. In our case, training, validation and test set are divided to test temporal generalizability. Let validation set be $\mathcal{S}_{val}$ and test set by $\mathcal{S}_{test}$. We obtain static characteristic reconstructions on validation set. Now for forward modeling, we average the validation set reconstructions over time to obtain $\hat{z}_{val}$. Since the reconstructions are supposed to remain static over time, these can be used as input to the forward model when evaluating model performance on the test set. Hence, for the forward model, say $\mathcal{F}$, $\mathcal{F} : [x_{test}, \hat{z}_{val}] \rightarrow y_{test}$ mapping can be used to evaluate model performance. This allows us to eliminate overfitting that would have happened had $\hat{z}_{test}$ been used since it would have been computed from $[x_{test}, y_{test}]$.

C EVALUATION METRIC DEFINITIONS

NSE

NSE (Nash-Sutcliff Efficiency) is a measure similar to R^2 and is used to measure prediction performance in time-series hydrological models. Q refers to streamflow at time step i. In our study, we also evaluate the static variable estimates using the same formula - which is equivalent to R^2 score.

$$\text{NSE} = 1 - \frac{\sum_i^N (Q_i - \hat{Q}_i)}{Q_i - \bar{Q}_i}$$

We also evaluate the quality of uncertainty estimates. A detailed definition of these metrics can be found in this paper [28].

Calibration Error: We estimate the calibration error in the prediction distribution using root mean squared calibration error given as,

$$\text{Calibration Error} = \sqrt{\frac{1}{N_p} \sum_j^{N_p} [p_j - \frac{1}{N} \sum_i^N \mathbb{1}\{u_i \leq \hat{u}(x_i)_{p_j}\}]}$$

Here, p_j refers to different percentiles for which the u observations are compared against the percentiles from the predicted distribution $\hat{u}$.

Dispersion

Dispersion has been proposed as a measure for evaluating statistical consistency of uncertainty estimates. A more disperse model is more robust to distributional shifts [28].

$$\text{Dispersion} = \frac{SD_\sigma}{\mu_\sigma}$$

In the dispersion formula, SD and μ refer to the standard deviation and mean of the epistemic uncertainty, σ.

Coverage Rate

Here, the coverage rate evaluates the proportion of times the observed value is being captured by the predicted confidence interval bounds.

$$\text{coverage rate} = \frac{\sum_i^N \mathbb{1}(z_i \in [\mu_{z_i} - \sigma_{z_i}, \mu_{z_i} + \sigma_{z_i}])}{N}$$

Here, we evaluate the number of times the observed value z_i lies within the confidence bounds defined as $(\mu_{z_i} - \sigma_{z_i}, \mu_{z_i} + \sigma_{z_i})$.