Probabilistic Inverse Modeling: An Application in Hydrology

Somya Sharma* Rahul Ghosh* Arvind Renganathan* Xiang Li*
Snigdhansu Chatterjee* John Nieber* Christopher Duffy+ Vipin Kumar*

Abstract

Rapid advancement in inverse modeling methods have brought into light their susceptibility to imperfect data. This has made it imperative to obtain more explainable and trustworthy estimates from these models. In hydrology, basin characteristics can be noisy or missing, impacting streamflow prediction. We propose a probabilistic inverse model framework that can reconstruct robust hydrology basin characteristics from dynamic input weather driver and streamflow response data. We address two aspects of building more explainable inverse models, uncertainty estimation (uncertainty due to imperfect data and imperfect model) and robustness. This can help improve the trust of water managers, handling of noisy data and reduce costs. We also propose an uncertainty based loss regularization that offers removal of 17% of temporal artifacts in reconstructions, 36% reduction in uncertainty and 4% higher coverage rate for basin characteristics. The forward model performance (streamflow estimation) is also improved by 6% using these uncertainty learning based reconstructions.

1 Introduction

Researchers in scientific communities study engineered or natural systems and their responses to external drivers. In hydrology, streamflow prediction [16, 17] 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). Process-based models are commonly used to study the relation between basin characteristics and streamflow. However, these hydrological models are constrained by assumptions, contain many parameters that need calibration and incur enormous computation cost. Machine learning (ML) paradigms in inverse modeling offer a promising alternate to infer entity characteristics from streamflow response. In our study, an entity's response to external drivers depends on its inherent properties (called entity characteristics). For example, for the same amount of precipitation (external driver), two river basins (entities) can have very different streamflow (response) values depending on their land-cover type (entity characteristic) [38] - this presents the issue of navigating a large search space to learn one of the many right model structures. Knowledge-guided self-supervised learning (KGSSL) [17] is an inverse model that can extract these entity characteristics using the input drivers and output-response data. The framework uses a self-supervised paradigm, where ML models are trained using labels that can be generated without any external annotation process. Through contrastive learning, the framework leverages the spatio-temporal correlation among the basins for more effective learning.

Developing such inverse models requires addressing several challenges. Often, the measured characteristics are only surrogate variables for the actual entity characteristics, leading to inconsistencies and high uncertainty. Uncertainty can arise due to several reasons, such as measurement error, missing data, and temporal changes in characteristics. 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 can help quantify the effect of multiple sources of uncertainty that introduce bias and error in decision-making. 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 [21]. Uncertainty in model structure and input data are also widespread. In real world applications, studying these can help improve trust of water managers, improve process understanding, reduce costs and make predictions more explainable and robust [36].

To achieve this, we propose a probabilistic inverse model for simultaneously learning representations

*University of Minnesota - Twin Cities. {sharm636, ghosh128, renga016, lixx5000, chatt019, nieber, kumar001}@umn.edu, + Pennsylvania State University {cxd11}@psu.edu

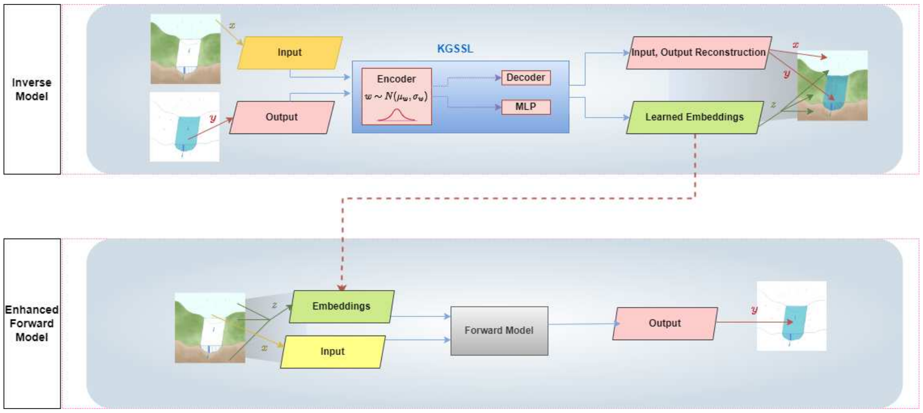

Figure 1: KGSSL for representation learning. The forward model learns streamflow (y) as functional approximation of weather drivers (x) and river basin static attributes (z). KGSSL leverages an inverse modeling framework for learning robust static attribute estimates ($\hat{z}$). The robust estimates aid in improving prediction performance of the forward model.

of static characteristics and quantifying uncertainty in these predictions. As a consequence, we analyze the framework's reconstruction capabilities and its susceptibility to adversarial perturbations - our model is able to maintain the same level of robustness as KGSSL. We use the same loss formulation as KGSSL and the same autoencoder based inverse model architecture. We modify the architecture such that the parameters in the encoder are estimated using the *Bayes by Backprop* method, enabling learning of the posterior weight distribution and uncertainty quantification in static characteristic reconstructions. To study robustness, we investigate the impact of different levels of Gaussian noise in training data on test set 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 show that it results in reducing the temporal artifacts in static characteristic predictions by 17% and also reduces the epistemic uncertainty by 36%. Bayesian neural networks, in an overparameterized regime, are known to provide robust results. In practice, prediction skill, robustness and uncertainty go hand-in-hand. Therefore, 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 reconstruction and forward modeling and compare it against the baselines, KGSSL [17] and CT-LSTM [30], both state-of-the-art frameworks for streamflow prediction. Since, we use a probabilistic model for estimating static characteristics, we obtain a posterior distribution instead of point estimates. This enables us to compute 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

Robustness: In several large-scale applications in areas like computer vision and natural language processing, the presence of even small, imperceptible perturbation can exacerbate model performance [10, 48, 51]. While, several robustness studies focus on the effect of different noises [32, 56], many other studies focus on methods to mitigate the adverse effects of these perturbations [35, 42, 50]. Through objective modification [11, 25, 27] or propagation of input-output relationship constraints [9, 22], deep learning architectures have been modified to improve robustness. In real-world data, natural variations (like blurring) have been studied [41, 47]. Issues like adversarial transferability [23] and data brittleness (robustness issues due to overfitting [40]) bring to light the limitations of modern machine learning methods. More recent studies also look at Bayesian deep learning models for their robustness properties [4, 5, 43].

Inverse Problems: In physical sciences [7, 39, 54], several recent advances have focused on solving inverse problems. Unlike standard inversion methods in mathematics, that 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, 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 [31]. Therefore, a principled MAP formulation or generative modeling may be viable for addressing these data related issues [46, 53]. Inverse modeling approaches that rely on a single neural network may not accurately capture the spatio-temporal heterogeneity in basin characteristics. Moreover, entity characteristics can be unknown or noisy (due to measurement or estimation bias). Therefore, a robust framework for learning entity characteristic can be useful in hydrology applications. Uncertainty quantification in inverse models also provides a viable pathway for learning entity characteristic. We discuss this more in the next section.

Probabilistic Modeling: Probabilistic inverse models offer richer learning than a deterministic inverse model as they offer distributional recovery for the input embeddings. For a given input x_i, the standard deviation of the prediction distribution is called the epistemic uncertainty estimate (σ_i). Any deficiencies in the model framework (e.g., capturing only linear effects) and input data (e.g., missing / noisy data, low sample size) would increase this uncertainty. Smaller uncertainty estimates mean more "confident" predictions. Several recent efforts use generative models for inverse problem solving [2, 8, 53].

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 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 Autoencoder - based Inverse Model We use the CAMELS [1] dataset with streamflow, weather drivers and entity characteristics information (lake, river-basin or streams in river network). Each entity i ($i = 1, ..., N$) has daily information, where $x_i^j \in \mathbb{R}^{\mathcal{D}_x}$ represents dynamic characteristics (weather drivers), $y_i^j \in \mathbb{R}^{\mathcal{D}_y}$ represents streamflow, $z_i^j \in \mathbb{R}^{\mathcal{D}_z}$ represents static characteristics for the i^{th} entity at the j^{th} time step. Similar to KGSSL [17], our framework estimates static characteristics (z) from time-series information ($[x, y]$). Contrastive learning allows us to utilize the spatio-temporal correlation among river basins [6]. The objective function for training KGSSL is

$$\mathcal{L} = \lambda_1 \mathcal{L}_{Rec} + \lambda_2 \mathcal{L}_{Cont} + \lambda_3 \mathcal{L}_{Inv} \tag{3.1}$$

where, reconstruction loss $\mathcal{L}_{Rec}$ enables accurate reconstruction 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 examples to enable accurate representation learning. The loss weights are learned using hyper-parameter tuning. The Sequence Encoder, comprised from 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, $\mathcal{L}_{Rec} = \frac{1}{2N} \sum_{e \in \{a,p\}} \sum_{i=1}^{N} MSE(\hat{S}_{e_i}, S_{e_i})$.

$$\begin{aligned}
\boldsymbol{i_t} &= \sigma(\boldsymbol{W_i} \left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_i}) \\
\boldsymbol{f_t} &= \sigma(\boldsymbol{W_f} \left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_f}) \\
\boldsymbol{g_t} &= \sigma(\boldsymbol{W_g} \left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_g}) \\
\boldsymbol{o_t} &= \sigma(\boldsymbol{W_o} \left[[\boldsymbol{x^t}; y^t]; \boldsymbol{h^{t-1}}\right] + \boldsymbol{b_o}) \\
\boldsymbol{c_t} &= \boldsymbol{f_t} \odot \boldsymbol{c_{t-1}} + \boldsymbol{i} \odot \boldsymbol{g_t} \\
\boldsymbol{h_t} &= \boldsymbol{o_t} \odot \tanh{(\boldsymbol{c_t})}
\end{aligned} \tag{3.2}$$

Knowledge-guided Contrastive Loss ensures that the 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 basin while negative pairs (of basins) refers to samples that enable learning from spatial correlation among basins.

$$\begin{aligned}
l(a_i, p_i) =& \frac{\exp{(sim(\boldsymbol{h_{a_i}}, \boldsymbol{h_{p_i}})/\tau)}}{\sum_{e \in \{a,p\}} \sum_{j=1}^{N} \exp{(sim(\boldsymbol{h_{a_i}}, \boldsymbol{h_{e_j}})/\tau)}} \\
&+ \frac{\exp{(sim(\boldsymbol{h_{p_i}}, \boldsymbol{h_{a_i}})/\tau)}}{\sum_{e \in \{a,p\}} \sum_{j=1}^{N} \exp{(sim(\boldsymbol{h_{p_i}}, \boldsymbol{h_{e_j}})/\tau)}}
\end{aligned} \tag{3.3}$$

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 $\hat{\mathbf{z}} = I(\mathbf{h})$.

$$\mathcal{L}_{Inv} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{z}\sum_{j=1}^{z}(z_i^j - \hat{z}_i^j)^2 \tag{3.4}$$

Temporal heterogeneity in driver-response time-series is a source of uncertainty in the 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,

$$\boldsymbol{unc}_i = \sqrt{\frac{W}{T}\sum_{j=1}^{T/W}(\hat{\boldsymbol{z}}_i^j - \hat{\boldsymbol{z}}_i)^2} \tag{3.5}$$

3.2 Uncertainty Estimation We modify the KGSSL framework to aid in uncertainty quantification. The uncertainty in estimation of static characteristics is obtained using a perturbation-based weight uncertainty method called *Bayes by Backprop* [3, 52]. As a method that relies on learning posterior distribution of weight parameters, *Bayes by Backprop* makes different layers of the architecture non-deterministic. This allows us to measure and mitigate the uncertainty from different components incorporated in the framework.

Introducing perturbations in weights while training has historically been used as a regularization method [18, 20, 26, 34, 44]. Some recent advances utilize perturbations to induce non-deterministic behavior in supervised learning models [19, 49]. Several variations of Bayesian neural networks implement the reparameterization trick [28] 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 [19]. 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 [12,19,24,37,55] 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,

$$\mathcal{F} = KL[q(w|\theta)||Pr(w)] - \mathbb{E}_{q(w|\theta)}[log\ Pr(\mathcal{D}|w)] \tag{3.6}$$

The complexity cost is the KL divergence between the learned posterior distribution of weight parameters $q(w|\theta)$ and the prior probabilitiy $Pr(w)$. 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 layers for learning parameter distribution (Table 5).

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 3.4) such that the characteristics with higher uncertainty in the estimates will have higher loss due to bigger penalty coefficients. In order to do that we look at the following theorem.

Theorem 3.1. *Let g be our inverse model receiving training set S as input such that, $g_S : [x_t^i, y_t^i] \rightarrow z_i$. A loss $\mathcal{L}_g$ for prediction function g_S defined as $\frac{1}{t}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|z|}\sum_{j=1}^{z} w^j(z_i^j - \hat{z}_i^j)^2$ minimizes uncertainty σ_S where w^j corresponds with E_{λ_1}, the eigenvector corresponding to the largest eigenvalue of σ.*

The proof for this and more discussion are given in Appendix B. If our inverse model $g \in \mathcal{G}$, where $\mathcal{G}$ is a hypothesis class, the empirical risk of the hypothesis class, $ERM_{\mathcal{G}}$, can be decomposed into an approximation and an estimation error. While, our uncertainty guided methodology provides prior knowledge on the hypothesis class with lower uncertainty estimation for certain characteristics, the estimation error for other characteristics with lower penalty coefficients might be higher. As we will see in our experimental results, a subset of features (geo-morphology based features) that were reconstructed successfully using KGSSL, saw an increase in bias in estimates from the UBL method. Also, the soil based features that were experiencing higher model approximation error in KGSSL estimates, saw an improvement in uncertainty estimates.

Average Metrics	Climate	Soil Geology	Geo-morphology
Deterministic KGSSL RMSE	0.292	0.575	0.430
Probabilistic KGSSL RMSE	0.294	0.580	0.438
Deterministic KGSSL CORR	0.958	0.792	0.878
Probabilistic KGSSL CORR	0.958	0.788	0.873
Deterministic KGSSL UNC	0.182	0.290	0.218
Probabilistic KGSSL UNC	0.175	0.258	0.205

Table 1: Model performance of deterministic and probabilistic KGGSL in terms of test RMSE, test correlation between predicted and observed static feature and test uncertainty over time, *unc*. The metrics are averaged over 9 features in each category. Probabilistic KGSSL is able to achieve same level of model performance and allows for uncertainty quantification as well.

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 (weather descriptive statistics, soil based features, and geo-morphology based features are included in the study. These affect the streamflow, water runoff processes). 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 representation learning frameworks 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 year 1980 - 2000 are used for training, year 2000 - 2005 are used for validation and year 2005 - 2015 are used for testing. We report mean squared error (MSE), NSE (Nash-Sutcliff Efficiency is a measure similar to R^2. It is used to measure prediction performance in time-series hydrological models), and uncertainty estimates (standard error in prediction estimates). We predict static characteristics for all 531 river basins in the test period (Appendix E, Figure 5). These predictions are made from KGSSL and probabilistic encoder based KGSSL.

4.1 Epistemic Uncertainty The probabilistic inverse model can be obtained by making different components of the framework non-deterministic as suggested in Sub-section 3.2. One variant is probabilistic encoder based inverse model that not only provides us with best validation inverse loss (Appendix C, Table 5), but also allows simultaneous uncertainty quantification in basin characteristics and driver-response reconstructions. More details on exploring different sources of uncertainty in Appendix C. Moreover, uncertainty over time, unc_i, and epistemic uncertainty, σ_i, have been shown to be correlated (Appendix D, Figure 4). This association enables a loss formulation where we can penalize reconstructions with higher epistemic uncertainty to obtain a learned model where uncertainty over time is also reduced. More discussion provided in Appendix D.

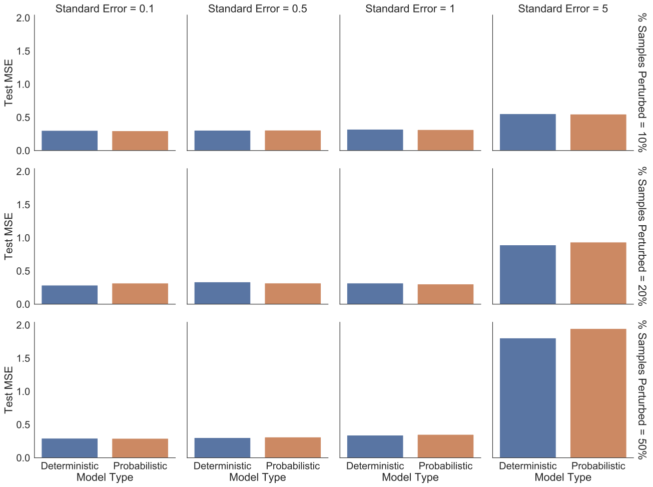

Figure 2: Robustness to noise with 10% samples perturbed, 20% samples perturbed and 50% samples perturbed at 4 levels of standard error. The probabilistic model is able to maintain same level of bias as the deterministic model.

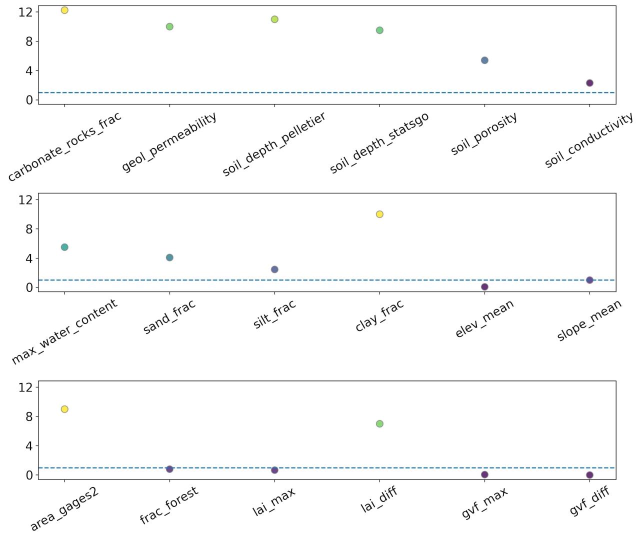

Figure 3: Trade-off ratio is computed as ratio of fraction decrease in uncertainty estimates and fraction increase in MSE values using the Uncertainty Based Learning.

4.2 Robustness Due to the imperfection in driver-response data, it is important to address robustness of our framework to noise. We evaluate the static characteristics reconstruction by perturbing different fractions of training data with different size of perturbations. Table 2 shows the reconstruction performance for weather drivers and streamflow as *reconstruction MSE loss* and reconstruction performance for static characteristics as the *static loss*. Since accurate reconstruction of static variables is of interest, the epistemic uncertainty in the static estimates are mentioned as *static uncertainty*. The percent of training examples that are perturbed with noise is varied from 1% to 50% while the perturbation size is determined by the standard error of the Gaussian noise that is added to the training data. Training examples are perturbed and model performance is evaluated on the clean test set. The static reconstruction loss increases marginally with the addition of noise when the standard error remains below 5. In Table 2 and Figure 2, for the deterministic and probabilistic models, we notice similar levels of bias in static feature reconstruction, where the probabilistic model can also measure the variability in the prediction estimates.

The bias in prediction estimates are robust to corruption in training data, but the variance in estimates increases. This may also impact the association that we observe between the uncertainty over years and epistemic uncertainty. Figure 4 shows us the correlation between the two uncertainties. To observe the effect of higher levels of perturbation, we look at the scenario when 50% of the training data is perturbed.In sub-figure 4b, we notice increased correlation between the weather and geo-morphology based static features. The increased epistemic uncertainty due to perturbations (as shown in Table 2), indicate a greater variability in static features, over time and otherwise. Therefore, with increased variance in static feature reconstruction, the correlation between the two types of uncertainty (measuring the deficiency in our input data), also increases with higher perturbation levels.

4.3 Uncertainty Based Learning (UBL) Learning static characteristics using the UBL methodology, we penalize reconstructed representations with higher uncertainty estimates in preliminary modeling performance (on validation dataset). Allowing for regularization that manages uncertainty results in a model that has lower uncertainty estimates and lower variance in static features over time. This however, also leads to an increase in bias for all static features. We can look at the percent decrease in uncertainty estimates and percent increase in mean squared error values on the test set to understand the trade off between variance and bias. Figure 3 shows us the trade off ratio computed as the ratio of percent decrease in uncertainty and percent increase in test set MSE values. The blue dashed line represents a trade-off ratio of 1. A trade-off ratio above 1 represents scenarios where percent improvement in uncertainty is more than the percent increment in MSE values for a given variable. For instance, for the "carbonate rocks fraction" variable, the percent decrease in uncertainty was 25% while the increase in MSE was 2% - therefore, the trade-off ratio is 12.5. We notice, for the soil based static features, UBL resulted in improvement in uncertainty. Since, geo-morphology characteristics like "elevation mean" and "slope mean" are already well approximated with lower temporal variance in the estimates, the trade-off ratios reflect the ineffectiveness of UBL for those cases.

We can also compare the UBL model performance in terms of test set NSE values with the deterministic and probabilistic KGSSL model (Table 3). In addition to improvement in static feature reconstruction, the probabilistic model also enables us to measure if our mean prediction estimates lie within the confidence intervals created using the standard error in the predictions. We look at the one standard deviation and two standard deviation confidence interval coverage rates.

$$\text{coverage rate} = \frac{\mathbb{I}(z_i \in [\mu_{z_i} - \sigma_{z_i}, \mu_{z_i} + \sigma_{z_i}])}{N \times |z|} \tag{4.7}$$

While we obtain a lower test NSE using the UBL method, we also obtain a higher coverage rate than the probabilistic KGSSL method. This may indicate that the regularization of uncertain predictions results in more confident estimation with wider coverage. In Table 2, we also compare the results in the presence of noise in training examples. We notice similar levels of robustness as compared to the previous models.

4.4 Enhanced Forward Model In our analysis, we investigate methods to reconstruct static characteristic values that are not only more robust to noise but also enable estimation in scenarios where samples might be missing in the observed data. KGSSL enables representation learning that solves the data missing-ness and robustness challenges in the real world static characteristic data [17]. We compare forward modeling results in streamflow prediction using original and reconstructed static characteristic values. As can be seen in Table 4, KGSSL offers improvement in forward model performance using reconstructed static features. We are also able to see improvement using the probabilistic KGSSL model and the UBL variants.

5 Conclusion

Rapid advancements in deep learning methods have made it imperative to obtain more explainable and

Method	Noise	Reconstruction Loss	Static Loss	Static Uncertainty
Deterministic	-	**1.940**	0.3137	0.000
Probabilistic Encoder	-	2.022	**0.2829**	0.0011
Deterministic	1% corruption, 10 s.d.	1.937	0.3001	0.000
Probabilistic Encoder	1% corruption, 10 s.d.	1.935	0.3119	0.0017
Deterministic	5% corruption, 10 s.d.	1.947	0.3095	0.000
Probabilistic Encoder	5% corruption, 10 s.d.	2.004	0.3054	0.0021
Deterministic (UBL)	1% corruption, 10 s.d.	1.955	0.3021	0.000
Probabilistic Encoder (UBL)	1% corruption, 10 s.d.	1.938	0.3135	0.0008
Deterministic (UBL)	5% corruption, 10 s.d.	1.944	0.3128	0.000
Probabilistic Encoder (UBL)	5% corruption, 10 s.d.	1.946	0.3234	**0.0007**

Table 2: Streamflow, weather driver reconstruction MSE test loss, static characteristic MSE test loss and static characteristic estimate epistemic uncertainty. The model performance in terms of test MSE for higher levels of noise in training data are given in Figure 2.

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

Table 3: 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.

Model	Average NSE	Ensemble NSE
Baselines EALSTM (original static characteristics)	0.7031	0.7238
KGSSL	0.7501	0.7570
Probabilistic KGSSL	0.7561	**0.7597**
KGSSL (UBL)	0.7611	0.7582
Probabilistic KGSSL (UBL)	**0.7636**	0.7594

Table 4: 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.

trust-worthy decisions from these models. For solving inverse problem, ensuring explainability is even more critical. 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 [15]. This method offers us improvement in streamflow prediction skill, offers the same level of robustness as previous methods and provides a wider coverage rate as well. We also incorporate inductive bias through our prior assumption on behavior of static characteristics. While this method utilizes penalty coefficients that were estimated in "one-shot" on the validation dataset, in future work a more adaptive approach can be explored to learn the penalty coefficients during training. Similar to a learnable dropout rate [13, 14], such a method may utilize a variational energy optimization scheme to learn these coefficients.

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

Beyond the scientific applications of streamflow in river basins, the need to build robust and personalized prediction models exists in several real-world applications. For example, while predicting mood (response), the personality (characteristics) of the patient (entity) will impact how the patient responds to outside weather (driver), or while assessing the aesthetics of an image (driver), the rating (response) by the assessor (entity) depends on the personality (characteristics). Thus, we hope that framework can be generalized to setting where taking into account these entity characteristics will be essential.

6 Acknowledgments

This work was funded by the NSF awards 1934721. Access to computing facilities was provided by the Minnesota Supercomputing Institute.

References

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

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

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

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

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

[6] Ting Chen et al. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[8] Arka Daw, M Maruf, and Anuj Karpatne. 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*, pages 237–247, 2021.

[9] Krishnamurthy Dvijotham, Sven Gowal, Robert Stanforth, Relja Arandjelovic, Brendan O'Donoghue, Jonathan Uesato, and Pushmeet Kohli. Training verified learners with learned verifiers. *arXiv preprint arXiv:1805.10265*, 2018.

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[11] Mahyar Fazlyab, Manfred Morari, and George J Pappas. Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming. *IEEE Transactions on Automatic Control*, 2020.

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

[13] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

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

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

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

[17] Rahul Ghosh, Arvind Renganathan, Kshitij Tayal, Xiang Li, Ankush Khandelwal, Xiaowei Jia, Chris Duffy, John Neiber, and Vipin Kumar. Robust inverse framework using knowledge-guided self-supervised learning: An application to hydrology. 2021.

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

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

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

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

[22] Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*, 2019.

[23] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

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

[25] Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*, 2019.

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

[27] Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *International Conference on Computer Aided Verification*, pages 443–452. Springer, 2019.

[28] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization

trick. *Advances in neural information processing systems*, 28:2575–2583, 2015.

[29] Z Kolter. Linear algebra review and reference. course notes cs 229, 2008.

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

[31] Alexander Lavin, Hector Zenil, Brooks Paige, David Krakauer, Justin Gottschlich, Tim Mattson, Anima Anandkumar, Sanjay Choudry, Kamil Rocki, Atılım Güneş Baydin, et al. Simulation intelligence: Towards a new generation of scientific methods. *arXiv preprint arXiv:2112.03235*, 2021.

[32] Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*, 2018.

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

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

[35] Hui Liu, Yongzheng Zhang, Yipeng Wang, Zheng Lin, and Yige Chen. Joint character-level word embedding and adversarial stability training to defend adversarial text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8384–8391, 2020.

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

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

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

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

[40] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[41] Alexander Robey, Hamed Hassani, and George J Pappas. Model-based robust deep learning: Generalizing to natural, out-of-distribution data. *arXiv preprint arXiv:2005.10247*, 2020.

[42] Ali Shafahi, Mahyar Najibi, Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *arXiv preprint arXiv:1904.12843*, 2019.

[43] Somya Sharma and Snigdhansu Chatterjee. Winsorization for robust bayesian neural networks. *Entropy*, 23(11):1546, 2021.

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

[45] Lina Stein, Martyn P Clark, Wouter JM Knoben, Francesca Pianosi, and Ross A Woods. How do climate and catchment attributes influence flood generating processes? a large-sample study for 671 catchments across the contiguous usa. *Water Resources Research*, 57(4):e2020WR028300, 2021.

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

[47] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.

[48] Eric Wallace, Mitchell Stern, and Dawn Song. Imitation attacks and defenses for black-box machine translation systems. *arXiv preprint arXiv:2004.15015*, 2020.

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

[50] Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attacks and defenses in word level. *arXiv preprint arXiv:1909.06723*, 2019.

[51] Xingxing Wei, Jun Zhu, Sha Yuan, and Hang Su. Sparse adversarial perturbations for videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8973–8980, 2019.

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

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

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

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

[56] Yichao Zhou, Jyun-Yu Jiang, Kai-Wei Chang, and Wei Wang. Learning to discriminate perturbations for blocking adversarial attacks in text classification. *arXiv preprint arXiv:1909.03084*, 2019.