Towards Entity-Aware Conditional Variational Inference for Heterogeneous Time-Series Prediction: An application to Hydrology

Rahul Ghosh* Arvind Renganathan* Wallace McAliley† Michael Steinbach*
Christopher Duffy‡ Vipin Kumar*

Abstract

Many environmental systems (e.g., hydrology basins) can be modeled as entity whose response (e.g., streamflow) depends on drivers (e.g., weather) conditioned on their characteristics (e.g., soil properties). We introduce Entity-aware Conditional Variational Inference (EA-CVI), a novel probabilistic inverse modeling approach, to deduce entity characteristics from observed driver-response data. EA-CVI infers probabilistic latent representations that can accurately predict response for diverse entities, particularly in out-of-sample few-shot settings. EA-CVI's latent embeddings encapsulate diverse entity characteristics within compact, low-dimensional representations. EA-CVI proficiently identifies dominant modes of variation in responses and offers the opportunity to infer a physical interpretation of the underlying attributes that shape these responses. EA-CVI can also generate new data samples by sampling from the learned distribution, making it useful in zero-shot scenarios. EA-CVI addresses the need for uncertainty estimation, particularly during extreme events, rendering it essential for data-driven decision-making in real-world applications. Extensive evaluations on a renowned hydrology benchmark dataset, CAMELS-GB, validate EA-CVI's abilities.

Keywords: representation learning, meta-learning, few-shot learning, zero-shot learning, environmental applications

1 Introduction

Across numerous scientific and environmental disciplines, researchers study how engineered and natural systems/entities respond to external factors [12]. In hydrology, e.g., predicting the streamflow (response) of a river basin/catchment (entity) due to external drivers (meteorological data, e.g., air temperature and precipitation) is crucial to understanding hydrology cycles, water management, flood mapping, and making operational decisions. An entity's response to external drivers is influenced by its inherent properties, referred to as entity characteristics. For instance, the streamflow of two river basins can vary significantly in response to the same amount of precipitation due to differences in their land-cover types [25]. Despite increasing data availability, in many of these applications, data seldom exist at appropriate spatiotemporal resolution or coverage for scientific studies or management decisions. Developing models that can transfer information from highly observed systems to sparsely observed or unmonitored systems is of interest in many environmental applications [32]. Traditionally, this transfer of information has relied on the regionalization of process-based models (PBMs), particularly in hydrology [30, 15]. Regionalization techniques require significant amounts of site-specific data collection and computational power to relate the parameters of a PBM (that may already be calibrated to the data of a monitored system) to the inherent characteristics of a sparsely observed or unmonitored system.

Machine learning (ML) models are increasingly being considered as an alternative to PBMs due to their ability to benefit from training data from diverse entities [32], enabling them to transfer knowledge between them. There are two primary methods of transferring this knowledge. The first approach involves incorporating ancillary characteristics of the entities as features (e.g., CTLSTM [20]) to account for their diversity and effectively transfer information to both less-observed (few-shot setting) and unobserved (zero-shot setting) entities. However, these characteristics can be difficult to measure accurately, leading to uncertainty or incomplete data. They may also be unknown, poorly understood, or absent in available entity characteristics. The second approach, termed inverse modeling, has been used to infer time-invariant entity characteristics from its driver-response data [11] in a deterministic fashion. A prominent example of these methods, Knowledge-Guided Self-supervised Learning (KGSSL) [11], offers a solution for performing entity-specific modulation for less-observed entities by conditioning the entity's response to external drivers on attributes inferred from the available few-shot responses, without requiring entity characteristics. They are shown to outperform the

*University of Minnesota. {ghosh128, renga016, stei0062, kumar001}@umn.edu
†U.S. Geological Survey. wmcaliley@usgs.gov
‡Penn State University. cxd111@psu.edu

state-of-the-art forward model that uses the actual incomplete characteristics in a few-shot setting [11].

This paper introduces a new approach called Entity-aware Conditional Variational Inference (EA-CVI) for probabilistic inverse modeling. EA-CVI infers entity-specific attributes as a distribution over a latent space from driver-response data. Compared to deterministic models like KGSSL, EA-CVI has several advantages. First, the latent representations are built probabilistically, which aligns well with Bayesian reasoning and allows for principled approaches to tasks such as Bayesian inference and posterior estimation. Second, EA-CVI captures the inherent uncertainty associated with limited data, enabling more flexible generalization to new entities. Third, EA-CVI can generate new data samples by sampling from the learned distribution, thus rendering it useful even in a zero-shot setting. Fourth, the variational latent space of EA-CVI is parsimonious, with most of the variability captured in a few latent dimensions. Lastly, this latent space has a semantic meaning that produces a coherent effect on the predicted response, making the response generation mechanism controllable with physical interpretability.

Next, we provide a brief overview of the key features of our proposed model and discuss how these features enable the advantages mentioned above. Specifically, EA-CVI consists of an entity *Encoder* that uses the driver and response of an entity to infer the data-driven posterior distribution over a latent space, followed by a response *Decoder* to perform the inference and generative steps (see Figure 1b). The latent space holds information about entity characteristics, and embeddings sampled from this posterior distribution are used to predict responses. We derive the evidence lower bound (ELBO) [18] of the loss function used to train EA-CVI. The ELBO comprises two key components: prediction error, which penalizes deviations between predicted responses and ground truth, and KL Divergence, which regularizes the latent space.

The KL-divergence loss of the variational approach shapes the latent space representation by aligning the approximate posterior distribution (obtained from the *Encoder*) with a predefined prior (e.g., a multivariate Gaussian in our case). It offers a crucial advantage over the deterministic KGSSL approach in its inherent embrace of variability and uncertainty within this space, especially when dealing with limited data. By training on extensive driver-response data from diverse entities, EA-CVI's adaptable yet structured latent space allows the discovery of dominant modes within the entity distribution. Sampling the entity characteristics from this latent space allows the model to generate responses for unobserved entities in a zero-shot setting. In environmental science, the search for a deeper understanding of how various entity physio-graphic factors influence response generation mechanisms has long been a fundamental endeavor. Notably, EA-CVI introduces a novel perspective in identifying the physical attributes associated with different response variation modes, opening the door to exploring entity responses under diverse bio-geo-physical conditions. Associating physical attributes with each response mode significantly enhances the interpretability of variations in the output model.

Operational decisions (e.g., probabilistic characterization of design-relevant extremes) [2] often need to consider relatively rare but high-impact events. A principled method of managing this uncertainty during regionally unprecedented events can improve trust in data-driven decision-making from these methods. Deep learning approaches to inverse problems [11, 26] can return high-quality point estimates but usually do not provide uncertainty estimates, which are essential to aid decision-makers. Although techniques such as Bayesian Neural Networks [4, 31] focus on modeling uncertainty in the predictions by treating the parameters of the neural network as random variables. EA-CVI implicitly characterizes uncertainty within the latent space, making it well-suited for representation learning in inverse problems [5]. Our method provides better estimates of uncertainty, particularly during periods of high streamflow response, rendering it essential for real-world applications.

We evaluate our proposed framework for predicting streamflow using CAMELS-GB (Catchment Attributes and MEteorology for Large-sample Studies) [7], a widely used hydrology benchmark dataset, for understanding the Earth's interconnected ecosystems and how they are impacted by humans and changing environment. CAMELS input data are freely available on the website of UK Centre for Ecology & Hydrology, and the code is available at GitHub[1].

2 Related Works

2.1 Few-Shot Learning Meta-learning [14] is a widely used approach when few observation samples are available for the out-of-sample entities. Meta-learning methods leverage the shared structure between different training tasks. This leads to better generalization and adaptation for new entities when only a small number of labels are available. Model Agnostic Meta-Learning (MAML) [8] is a popular approach that learns a global meta-model, which can then be easily adapted to create personalized models for each entity using limited

[1] https://github.com/2021rahul/Towards-Entity-Aware-Conditional-Variational-Inference-for-Heterogeneous-Time-Series-Prediction

data. Along these lines several advancements [27, 33] of MAML have been proposed that essentially tackle the same problem. Another line of research is to encode tasks into low-dimensional latent embeddings, which will modulate the prediction function's behavior for diverse entities [10]. The prediction function is conditioned on entity observations using an inferred embedding that is encoded from an entity's input and output pairs. Recently, Ghosh et al. [11] introduced KGSSL which infers time-invariant entity characteristics from its driver-response data. Similarly, Botterill et al. [6] used an encoder-decoder structure to obtain learned encodings similar to hydrological signatures. Further advancements include using bootstrapping [22] to have multiple latent embeddings or attention-based versions of NP [17]. It is important to note that the encoder in these approaches can be viewed as an inverse network, where the objective is to infer task or context characteristics from input and output pairs. Our work proposes a variational approach to such encoder-based inverse models and thus can be easily incorporated in those approaches.

2.2 Uncertainty Quantification Many methods in Bayesian deep learning have been developed to accurately predict outcomes and provide estimates of uncertainty. Monte Carlo Dropout [9] and weight perturbation schemes [24] are examples of approximate Bayesian inference when making predictions during the testing. Variational inference is another technique used to improve learning in Bayesian networks [4, 31]. Stochastic variational inference is used to estimate predictive uncertainty in Bayesian LSTM models. Mixture density networks [3] are used for multi-modal data where each modality can be captured using mixing components. [19] investigate using mixture density networks and Monte Carlo Dropout to estimate the uncertainty in streamflow predictions. Ensemble modeling, which includes techniques like variational mode decomposition or data assimilation [1], is a popular approach to predicting uncertainty. However, to the best of our knowledge, a variational inference framework has yet to be explored for entity response prediction.

3 Methods

3.1 Problem Formulation This work focuses on learning ML models for a set of entities. For each entity i, we have access to multiple driver/response pairs of time series sequences, as $\{(\boldsymbol{x_i^1}, y_i^1), (\boldsymbol{x_i^2}, y_i^2), \ldots, (\boldsymbol{x_i^{T_i}}, y_i^{T_i})\}$, where elements of $\boldsymbol{x_i^t}$ are drivers, y_i^t is a response, $\boldsymbol{x_i^t} \in \mathbb{R}^{D_x}$, $y_i^t \in \mathbb{R}$, and superscripts indicate time step indices. The objective is to learn the mapping function from input time-varying drivers $\boldsymbol{X_i^t} = [\boldsymbol{x_i^1}, \boldsymbol{x_i^2}, \ldots, \boldsymbol{x_i^t}]$ to a target variable y_i^t. In conventional supervised ML, we train a predictive model $p_{\theta_i}(y_i^t|\boldsymbol{X_i^t})$ by finding the parameters θ that maximize the likelihood of the observed data,

$$\theta_i^* = \arg\max_{\theta_i} \log p_{\theta_i}(y_i^t|\boldsymbol{X_i^t}). \tag{3.1}$$

Given sufficient training data for each entity, we can train individual ML models that capture these inherent biases in each entity within the learned parameters θ_i^*. However, this is not feasible as many entities lack sufficient training data. Hence, we consider learning a global model combining data from all the entities. The major challenge in building this mapping is to handle the heterogeneity across different sites $i \in \{1, ..., N\}$ to achieve good performance over all the entities. These entities' behavior is often governed by their inherent characteristics $\boldsymbol{z_i}$, i.e., the conditional distribution is of the form $p_\theta(y_i^t|z_i, \boldsymbol{X_i^t})$, where θ denotes the function class shared by the target systems and $\boldsymbol{z_i}$ denotes entity-specific inherent characteristics. In many scenarios, measurement of the entity characteristics may be entirely unavailable. Without these entity attributes, the global model cannot accurately predict each entity's response; thus, we present a variational inference method to address this challenge in this paper.

3.2 Architecture Our proposed method infers latent entity characteristics ($\boldsymbol{z_i} \in \mathbb{R}^{D_z}$) given the time-varying drivers $\boldsymbol{X_i} = [\boldsymbol{x_i^1}, \boldsymbol{x_i^2}, \ldots, \boldsymbol{x_i^T}]$) and response ($\boldsymbol{Y_i} = [y_i^1, y_i^2, \ldots, y_i^T]$) data and uses these latent characteristics to predict an entity's response from the drivers. We use a temporal deep latent variable model (DLVM) that comprises a sequence encoder (inference network) and a decoder (generator network), as shown in Figure 1a. The inference network ($q_\phi : \mathbb{R}^{T\times(D_x+1)} \rightarrow \mathbb{R}^{D_z}$) is trained to encode entities into the latent space. The generator network ($p_\theta : \mathbb{R}^{T\times(D_x+D_z)} \rightarrow \mathbb{R}^{T\times 1}$) is trained to decode latent vectors and driver data into the response space. During training, latent vectors are encouraged to contain the minimum amount of information needed to reconstruct the entity response from latent vectors and drivers. In the following sections, we describe the choice of neural network architectures. Subsequently, we will describe the training process and the novel loss function, focusing on how they facilitate variational modeling.

3.2.1 Inference Network (Encoder) Because the exact posterior inference is intractable, an inference model, $q_\phi(\boldsymbol{z}|[\boldsymbol{x^t}; y^t]_{1:T})$, that approximates the true posterior, $p_\theta(\boldsymbol{z}|[\boldsymbol{x^t}; y^t]_{1:T})$, for variational inference [16] is introduced. This can also be viewed as encoding the driver and response data interaction for an entity to learn an approximate posterior over latent variables in these sequences. This distribution $q_\phi(\boldsymbol{z}|[x^t; y^t]_{1:T})$

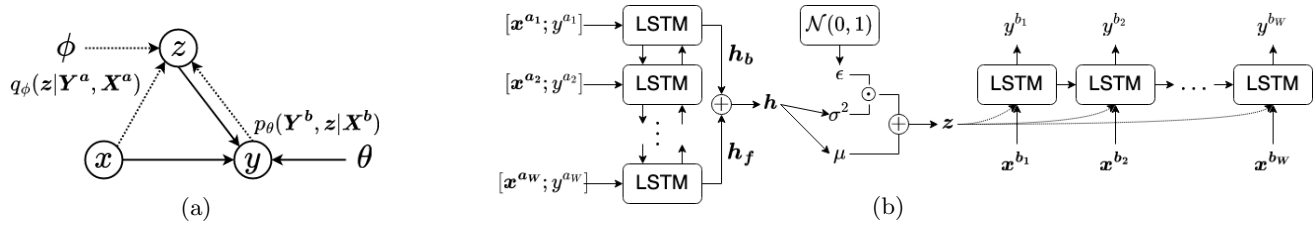

Figure 1: (a) *Graphical model of the proposed deep latent variable model (DLVM)*, (b) *Architectural diagram of our proposed method. We use a bidirectional-LSTM as encoder network and an LSTM as the conditional decoder.*

is modeled using a neural network, where ϕ are the related weight parameters (Figure 1a). We implement this using a bidirectional RNN-based sequence encoder. LSTM [13] is particularly suited for our task where long-range temporal dependencies between driver and response exist as they are designed to avoid exploding and vanishing gradient problems. The final hidden states for the forward ($\boldsymbol{h_f}$) and backward LSTM ($\boldsymbol{h_b}$) are added to get the final embeddings $\boldsymbol{h}$ as shown in Figure 1b. We define the posterior distribution as a function of $\boldsymbol{h}$, using multi-layer perceptrons (MLPs) to infer the parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ of a multivariate normal distribution with a diagonal covariance matrix, as

$$\begin{aligned} \boldsymbol{h} &= \text{BiLSTM}([\boldsymbol{x^t}; y^t]_{1:T}; \phi_{\boldsymbol{h}}) \\ \boldsymbol{\mu} &= \text{MLP}(h; \phi_{\boldsymbol{\mu}}) \\ \boldsymbol{\sigma}^2 &= \text{diag}(\exp(\text{MLP}(h; \phi_{\boldsymbol{\sigma}^2}))) \\ q_\phi(\boldsymbol{z}|[\boldsymbol{x^t}; y^t]_{1:T}) &= \mathcal{N}(\boldsymbol{z}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2), \end{aligned} \tag{3.2}$$

where the parameter set ϕ is divided into the LSTM parameters ($\phi_{\boldsymbol{h}}$) and the two MLP parameters ($\phi_{\boldsymbol{\mu}}$ and $\phi_{\boldsymbol{\sigma}^2}$). To draw a sample of $\boldsymbol{z}$, we use the reparameterization trick [18], given as $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma}^2\boldsymbol{\epsilon}$, where $\epsilon_i \sim \mathcal{N}(0,1) \; \forall \; \epsilon_i \in \boldsymbol{\epsilon}$.

3.2.2 Generator Network (Decoder) The generator network allows for the conditional generation of response data given the latent variable ($\boldsymbol{z}$) from the inference network and the driver data. The conditional generative process of the model is given in Figure 1a as follows: for a given sequence of driver and response data ($\boldsymbol{X^a}$ and $\boldsymbol{Y^a}$), $\boldsymbol{z}$ is drawn from the posterior distribution $q_\phi(\boldsymbol{z}|[\boldsymbol{X^a}; \boldsymbol{Y^a}])$, and the sequence of response data for another time-period is generated from the distribution $p_\theta(\boldsymbol{Y^b}|\boldsymbol{z}, \boldsymbol{X^b})$. Specifically, we construct an LSTM-based conditional sequence generator $y^t = LSTM(\boldsymbol{z}, [\boldsymbol{x^{1:t}}; \theta)$, where $y^t \in \boldsymbol{Y^b}$ and $\boldsymbol{x^t} \in \boldsymbol{X^b}$.

3.3 Learning Consider two time periods: a period of known sequences $\boldsymbol{X^a}$ and $\boldsymbol{Y^a}$, and a period where the drivers $\boldsymbol{X^b}$ are known, but the responses $\boldsymbol{Y^b}$ are to be predicted. We assume that the entity attributes do not change over time, which implies that the data from each period contain similar information about the latent variables $\boldsymbol{z}$. We train the framework to extract these latent attributes by maximizing an ELBO on the conditional log-likelihood $p_\theta(\boldsymbol{Y^b}|\boldsymbol{X^b})$, given by

$$\begin{aligned} \text{ELBO} =& \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a}, \boldsymbol{X^a})}\left[\log p_\theta(\boldsymbol{Y^b}|\boldsymbol{z}, \boldsymbol{X^b})\right] \\ & - \mathcal{D}_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{Y^a}, \boldsymbol{X^a})||p_\theta(\boldsymbol{z}|\boldsymbol{X^b})\right). \end{aligned} \tag{3.3}$$

Equation (3.4) results in a training approach wherein data from different periods are provided as inputs to the encoder and decoder. As suggested by [29], we let the latent variables be independent of the drivers so that the prior distribution becomes an unconditional prior, i.e., $p_\theta(\boldsymbol{z}|\boldsymbol{X^b}) = p_\theta(\boldsymbol{z})$. We choose a multivariate standard normal distribution prior, i.e. $\boldsymbol{z} \sim \mathcal{N}(0,1)$. Note that 3.3 differs from the standard CVAE [29] objective. Rather than maximizing the log-likelihood of the observations given their corresponding drivers (as proposed by CVAE), we maximize the conditional log-likelihood of one set of observations, b, given other observations, a, and drivers for all observations.

Maximizing 3.3 increases the probability of training data under the generative model and encourages the inference model to be similar to the unknown exact posterior distribution. When the inference process is ambiguous, the inference model is incentivized to produce a wide latent distribution such that all the latent encodings are needed for the generative model to produce all possible responses. Thus, a wide range of possible responses can be produced, and uncertainty in the responses can be expressed. Note that the approximate posterior is for one period, and the true posterior is for another. If the data from each period provided the same information about the latent variables, then this term would be zero for a perfect approximate distribution $q_\phi(\boldsymbol{z}_i|\boldsymbol{Y^b}_i, \boldsymbol{X^b}_i)$. Here we justify 3.3 by showing that it still forms a valid ELBO.

THEOREM 3.1. *With our parameterization of $\boldsymbol{z}$, 3.3 is a valid lower bound of the conditional log-likelihood $p_\theta(\boldsymbol{Y^b}|\boldsymbol{X^b})$.*

Proof. We seek to maximize the conditional likelihood $p_\theta(\boldsymbol{Y^b}|\boldsymbol{X^b})$ of all response sequences in the training

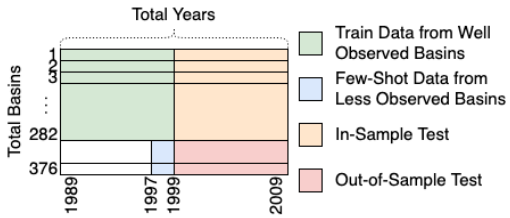

Figure 2: *Experimental setting followed in the paper for training and testing of the ML models.*

data, conditioned on their corresponding driver sequences. We can write the conditional log-likelihood as

$$
\begin{aligned}
&\log p_\theta(\boldsymbol{Y^b}|\boldsymbol{X^b}) \\
&\quad = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log p_\theta(\boldsymbol{Y^b}|\boldsymbol{X^b})\right] \\
&\quad = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log \frac{p_\theta(\boldsymbol{Y^b},\boldsymbol{z}|\boldsymbol{X^b})}{p_\theta(\boldsymbol{z}|\boldsymbol{Y^b},\boldsymbol{X^b})}\right] \\
&\quad = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log \frac{p_\theta(\boldsymbol{Y^b},\boldsymbol{z}|\boldsymbol{X^b})q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})p_\theta(\boldsymbol{z}|\boldsymbol{Y^b},\boldsymbol{X^b})}\right] \\
&\quad = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log \frac{p_\theta(\boldsymbol{Y^b},\boldsymbol{z}|\boldsymbol{X^b})}{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\right] \\
&\qquad + \mathcal{D}_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})||p_\theta(\boldsymbol{z}|\boldsymbol{Y^b},\boldsymbol{X^b})\right).
\end{aligned} \tag{3.4}
$$

The second term is the KL divergence of the approximation to the posterior distribution from the true posterior. Because the KL divergence is always non-negative, it is valid to maximize the first term of 3.4 during training because it is a lower bound on the conditional log-likelihood, resulting in the ELBO expression shown in 3.3.

$$
\begin{aligned}
\text{ELBO} &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log \frac{p_\theta(\boldsymbol{Y^b},\boldsymbol{z}|\boldsymbol{X^b})}{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log p_\theta(\boldsymbol{Y^b}|\boldsymbol{z},\boldsymbol{X^b})\right] \\
&\quad + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log \frac{p_\theta(\boldsymbol{z}|\boldsymbol{X^b})}{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})}\left[\log p_\theta(\boldsymbol{Y^b}|\boldsymbol{z},\boldsymbol{X^b})\right] \\
&\quad - \mathcal{D}_{KL}\left(q_\phi(\boldsymbol{z}|\boldsymbol{Y^a},\boldsymbol{X^a})||p_\theta(\boldsymbol{z}|\boldsymbol{X^b})\right).
\end{aligned} \tag{3.5}
$$

□

4 Dataset and Baselines

CAMELS-GB (Catchment Attributes and MEteorology for Large-sample Studies) [7] is part of a family of continental scale datasets that are used extensively by the hydrology community to assess the quality of PBMs and ML models [23]. CAMELS-GB provides daily meteorological forcing data (precipitation, evapotranspiration and air temperature), daily streamflow observation, and basin characteristics (refer to the code repository for the complete set) for 671 basins in the UK. Our study uses data for 376 basins (entities) from CAMELS from Oct 01, 1989, to Sep 30, 2009. Data from 1989-1999 is used for model training, and 1999-2009 is used for testing, as shown in Figure 2. The basins are divided into two subsets: in-sample basins, which are used to build and train ML models, and out-of-sample basins, which are not encountered during training.

Table 1: *Mean R^2 values for streamflow modeling on CAMELS-GB for EA-CVI and the baselines in a few-shot setting. The amount of data (in years) used as few-shot are denoted as column names. We exclude CTLSTM and MAML_{CTLSTM} when determining the best performing model because they require entity characteristics which may not be available in this problem.*

MODELS	Few-Shot in years			
	0.5	1	2	3
MAML_{LSTM}	-2.313	-0.823	-0.625	-0.288
KGSSL	-1.352	0.523	0.540	0.599
$\text{KGSSL}_{Bayesian}$	0.330	0.504	0.531	0.604
EA-CVI	**0.443**	**0.580**	**0.607**	**0.628**
CTLSTM	0.339	0.339	0.339	0.339
MAML_{CTLSTM}	0.451	0.532	0.554	0.578

We compare the performance of EA-CVI to state-of-the-art methods in few-shot learning and inverse modeling. $\textbf{MAML}_{\boldsymbol{LSTM}}$ trains a meta-LSTM base model using model agnostic meta-learning (MAML) [8] approach for fast adaptation of the base LSTM model. We use the streamflow from the out-of-sample basins in a few-shot setting and five inner optimization steps to finetune the meta-model. **KGSSL** is the state-of-the-art purely deterministic inverse framework [11] for few-shot settings to infer the entity attributes in the form of embeddings and further use them to predict the streamflow. $\textbf{KGSSL}_{\boldsymbol{Bayesian}}$ [28] further extends the KGSSL framework by defining probability distributions for model weights using a Bayesian approach [4]. Lastly, for comparison only, we also present results using **CTLSTM** [20] and $\textbf{MAML}_{\boldsymbol{CTLSTM}}$, both of which have access to the actual basin characteristics not used in our proposed method.

We create input sequences of length 365 using a stride of half the sequence length, i.e., 183. All LSTMs used in the response predictor for EA-CVI (decoder) and the baselines have one hidden layer with 128 units, whereas the LSTMs used in the encoder of EA-CVI and KGSSL have a hidden layer with 32 units. The feed-forward network used to get the mean and standard deviation also has one hidden layer with 32 units. In our experiments, we perform extensive hyperparameter search with the list provided in the code repository. To reduce the randomness typically expected with network initialization, we report the result of ensemble prediction obtained by averaging predictions from five models with different weight initializations.

5 Experiment and Results

5.1 Predictive Performance In Table 1 we evaluate the performance in terms of mean coefficient of determination (R^2) for each streamflow prediction method. Here we report the performance on the out-of-sample basins (i.e., the training and testing data are from different basins and different years) in a few-shot setting by varying the amount of data available as few-shots. We observe that MAML_{LSTM} has relatively lower R^2 values, indicating poorer predictive performance. This is because all model parameters are adapted for each entity using a few shots during finetuning, resulting in a suboptimal model. KGSSL performs better than MAML_{LSTM} (with positive mean R^2 values across all the few-shot settings) due to efficient use of the few-shot settings. Instead of adapting the whole model parameter set, KGSSL infers the entity attributes using the few-shot samples and uses them to modulate the predictor model. $\text{KGSSL}_{Bayesian}$ performs similarly to KGSSL, showing positive mean R^2 values, and the performance tends to improve with more few-shot samples. EA-CVI outperforms the previous models, consistently exhibiting the highest mean R^2 values across all few-shot settings. This indicates that EA-CVI is more sample-efficient and effective at predicting streamflow in a few-shot setting than KGSSL. Interestingly, when more years of observation are available for the out-of-sample entities, CTLSTM and MAML_{CTLSTM} are outperformed by the inverse modeling methods (KGSSL and EA-CVI) that infer the characteristics from the driver-response data. This shows that the known characteristics present may be incomplete, and both inverse modeling methods infer the entire latent variable space as the embeddings represent known and unknown static attributes. The EA-CVI approach has the added benefits of creating a semantically meaningful latent space, zero-shot prediction, and uncertainty quantification, which we discuss in the following sections.

5.2 Semantic Meaning of Latent Space This section provides a semantic analysis of EA-CVI's latent embeddings. First, we demonstrate its effectiveness in encapsulating diverse entity characteristics within compact, low-dimensional representations. Second, we show how different latent components affect the streamflow generation, highlighting its ability to generate possible scenarios of entity response under different bio-geophysical conditions. Lastly, we provide a physical interpretation of these latent dimensions with known physical characteristics of entities.

5.2.1 Efficient Latent Space To explore the latent spaces learned by KGSSL and EA-CVI, we measure the activity of each latent vector dimension, where a latent dimension's activity is defined as its variance over all the entities in the in-sample set. For both methods, we analyze the information in each dimension by gradually incorporating an increasing number of these latent variables ordered by their activity. Table 2 shows the predictive performance on test years of the in-sample entities for both the methods where we use top-k most active latent dimensions with the remaining dimensions replaced by zero values. The performance drop in EA-CVI is significantly less than KGSSL with fewer latent dimensions. This demonstrates EA-CVI's efficiency in encoding more information in the most active latent dimensions. The latent dimensions are thus analogous to principal component analysis (PCA), as the most active latent variables can be viewed as the dominant modes of variation in the response.

Table 2: *Mean R^2 values for EA-CVI and KGSSL streamflow modeling with decreasing number of latent dimensions for in-sample basins in test years.*

Method	All	Top 10	Top 5	Top 2	Top 1
EA-CVI	0.7271	0.7182	0.7169	0.7010	0.6869
KGSSL	0.6957	0.2713	0.1957	0.0741	-0.4625

5.2.2 Coherent Streamflow Generation We observe independent modes of response variation under the posterior distribution by changing one latent variable at a time. In Figure 3, we tested the effect of the top two most and least active latent variables on the output to show how the EA-CVI inverse framework exposes modes of variation in the output variable. We sweep these latent variables' values from -3σ to 3σ with a step of 0.25σ. Using these sweeps, we create the sets of predictions shown in Figure 3. For example, the most active latent component of EA-CVI (the top row of Figure 3) captures high variation around peak streamflows. By contrast, the second most active latent component consistently affects the entire streamflow time series. We also observe that less active dimensions have more complex modes of variations, which supports our analogy to PCA. For KGSSL, however, the most and least active latent dimensions have similar effects on streamflow prediction. In addition, changing the latent variables in EA-CVI has a coherent effect on streamflow, i.e., the streamflow either increases or decreases consistently throughout the time series. We do not observe such a coherent effect in the plots of KGSSL. This further shows that the information is encoded uniformly across the dimensions of the latent vector in KGSSL and thus lacks interpretability in inferring the dominant modes of variation. In the following section we provide a physical interpretation of the effects of the latent variables on streamflow by calculating the correlation of latent vectors and basin characteristics over all the entities.

5.2.3 Correlation with Entity Characteristics Figure 4 shows the correlations between each latent di-

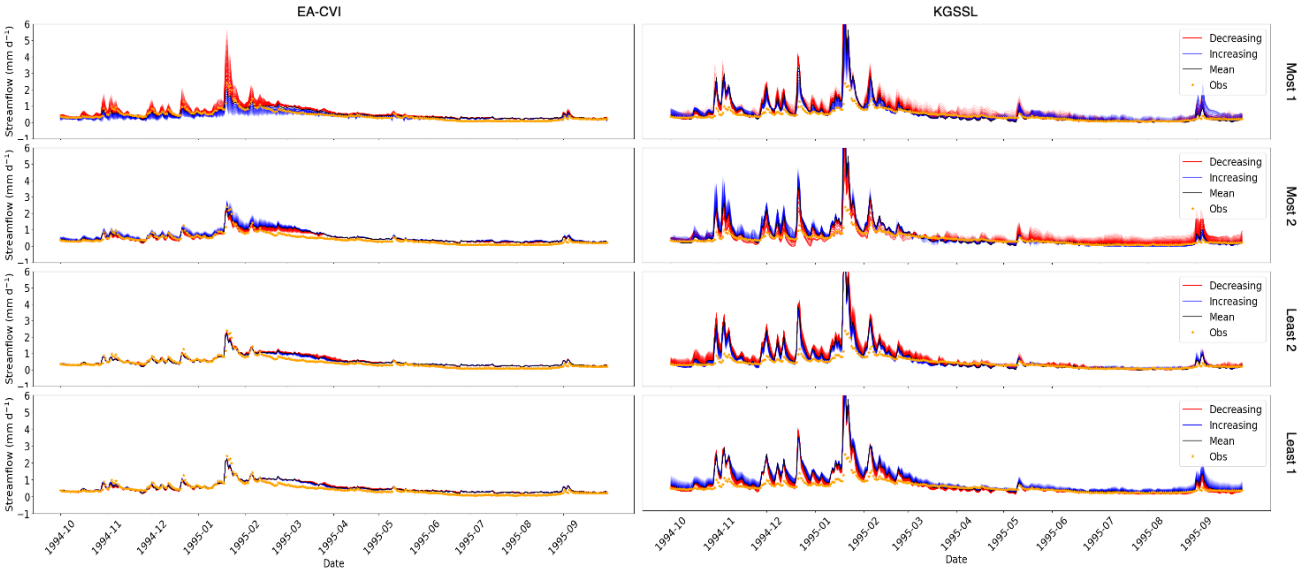

Figure 3: *Streamflow profiles of a basin generated by increasing (blue plots) and decreasing (red plots) the value of the top two and bottom two most active components of the latent vector for both EA-CVI and KGSSL.*

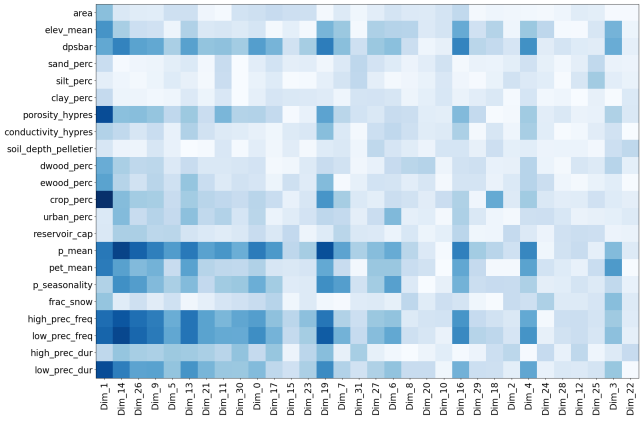

Figure 4: *Correlation of each dimension of EA-CVI's learned embeddings with each physical characteristic.*

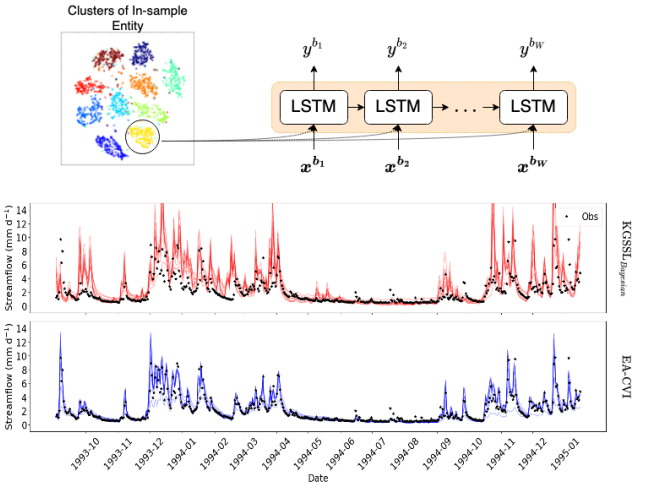

Figure 5: *Performance of models in zero-shot setting. Top figure shows the architectural setup. For a randomly chosen out-of-sample basin, KGSSL (red) predictions using cluster centroids do not match its observations. EA-CVI (blue) produces conditional predictions using cluster centroids that contain the observations.*

mension and basin characteristics. The vertical axis represents the actual entity characteristics, and the horizontal axis shows the latent variables ranked from most to least activity across entities. The most active latent dimensions, which encode most of the entity-specific information, correlate most strongly to physical characteristics. In particular, the most active latent variable correlates with attributes like soil porosity (degree of porosity of soil) and crop percentage (amount of vegetation), which are known to hydrologists to reduce the streamflow for similar weather drivers. Similar conclusions can be drawn for the other latent variables, thus leading to a knowledge-guided exploration of the latent space and providing explainability to the predictions. Additionally, the correlations tend to decrease as activity decreases. This result illustrates that the most active, information-dense latent dimensions correlate best to physical properties.

5.3 Zero-Shot Streamflow Generation In many situations, building a reliable model for response generation in out-of-sample entities in zero-shot settings is necessary. The CTLSTM model cannot be used in this scenario without entity characteristics, and KGSSL cannot infer these characteristics without few-shot data. EA-CVI allows us to generate conditional streamflow based on the dominant modes of latent characteristics inferred from the entities observed during training. Specifically, we cluster the latent vectors of in-sample entities to create categories of different types of entities based on their inferred attributes. Given an out-of-sample en-

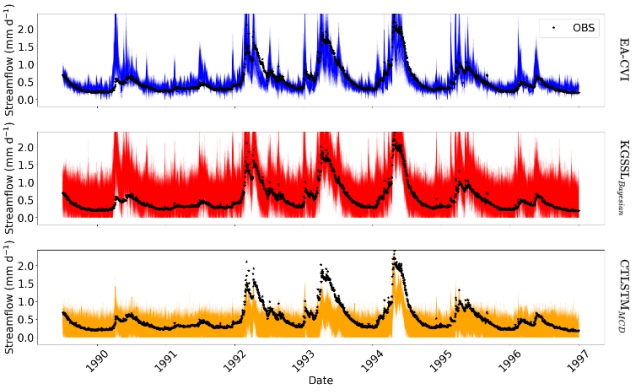

Figure 6: *Observed streamflow and 100 predicted streamflow realizations for EA-CVI (blue), KGSSL$_{Bayesian}$ (red) and CTLSTM$_{MCD}$ (orange) for a randomly chosen Out-of-Sample basin.*

tity, we obtain the centroid from each cluster of entities and use it in the decoder to provide conditional streamflow prediction, as shown in Figure 5. In the figure, many generated streamflows from EA-CVI (blue lines) overlap with the observed streamflow, showing that the cluster centroids can be used for conditional streamflow prediction as the embedding space of EA-CVI is regularized. B, none of the generated streamflow from KGSSL (red lines) lies on the observations. This is because KGSSL, like other deterministic autoencoders, has not been trained to use the latent space continuously. Therefore, the decoder's output for the cluster centroids is not valid as these points in the latent space have not been encountered during training.

5.4 Uncertainty Quantification In this section, we aim to evaluate the uncertainty estimations of EA-CVI and compare with the Monte Carlo Dropout (MCD) [9] version of CTLSTM (CTLSTM$_{MCD}$) and KGSSL$_{Bayesian}$. First, we visually compare the estimated uncertainty from the two approaches. Second, we quantitatively evaluate the estimated uncertainty distributions using commonly used metrics. We generate multiple inferences by running the model 100 times.

5.4.1 Visualizing Predicted Uncertainty In Figure 6 we visually compare the predictions from EA-CVI, KGSSL$_{Bayesian}$ and CTLSTM$_{MCD}$ on a randomly selected test basin during the test years. We observe that EA-CVI produces high-resolution prediction with uncertainty increasing during times with high streamflow response, whereas the predictions predicted by KGSSL$_{Bayesian}$ and CTLSTM$_{MCD}$ are of lower resolution with wide uncertainty bands at all times. In addition, the uncertainty bands from EA-CVI better capture the observations (especially during peak/major events) denoting that EA-CVI better estimates uncertainty while producing accurate predictions.

Table 3: *Resolution of the predictions from the two methods.*

Metric	CTLSTM$_{MCD}$	KGSSL$_{Bayesian}$	EA-CVI	ALL Basins
Mean Absolute Deviation	0.2260	0.5207	0.1366	1.3408
Standard Deviation	0.2793	0.6422	0.1721	2.0294
Variance	0.1972	0.5907	0.0784	8.3943
Quantile Distance 0.75-0.25	0.3878	0.9008	0.2262	1.6446
Quantile Distance 0.9-0.1	0.7107	1.6548	0.4302	3.5870

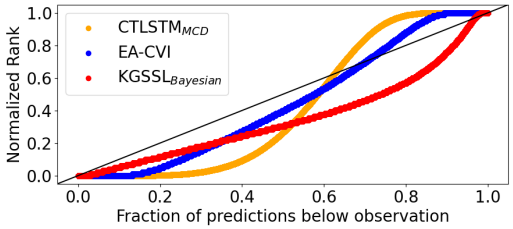

Figure 7: *Probability plot to show the reliability of the predictions. An optimal model's plot lies near the 1:1 line, shown in black.*

5.4.2 Estimating Predicted Uncertainty We also evaluate the predicted distributions from two perspectives: a) measures of dispersion and b) reliability of the distributional predictions. Table 3 reports measures of dispersion for the methods and the empirical distribution from the observations aggregated over all the basins as a reference ("ALL Basins"). The "ALL Basins" statistics should be used as a reference to contextualize the statistics from the modeled distributions. The table shows that EA-CVI predicts higher resolution distributions. Next, we use a probability plot [21] to evaluate how well the distributions of predictions match the true distributions of their corresponding observations. We compute the fraction of corresponding predictions that are less than the observation for each observation. Those fractions will be distributed uniformly between (0,1) if the prediction distribution matches the distribution of the observation. We evaluate whether the fractions are uniformly distributed by ranking the fractions from lowest to highest and plotting the normalized ranks against the fractions. The plotted points fall close to the 1:1 line if the fractions are distributed uniformly. From Figure 7, we can observe that the line corresponding to EA-CVI lies closer to the 1:1 line than CTLSTM$_{MCD}$ and KGSSL$_{Bayesian}$. EA-CVI's predicted probabilities match the distribution of the observations better than the baselines. The KGSSL$_{Bayesian}$ line lies below the 1:1 line, indicating a bias toward low values.

6 Conclusion

In this work, we presented a novel inverse model using the variational framework to infer the entity characteristics in a latent space and leverage them for the conditional prediction of responses given driver data. Extensive experiments on a hydrological benchmark dataset showed that in a few-shot setting, EA-CVI is more sample-efficient and outperforms baseline models for less-observed entities (even models with access

to the actual entity characteristics). EA-CVI's ability to identify the physical attributes associated with different response variation modes has the potential to offer deeper insights. The proposed method can add value in other applications in environmental sciences where global models are to be learned for a diverse set of entities. Our framework can further be extended to handling missing observations in the driver or response data. In its current form, EA-CVI does not use known entity characteristics. Thus, incorporating partially known or noisy basin characteristics as prior knowledge to modulate the latent dimension is a direction of further research. Additionally, the methods presented here can be applied to other methods for task-aware modulation in machine-learning and may be considered in future work.

7 Acknowledgement

This work is supported by the NSF award 2313174. Access to computing facilities was provided by the Minnesota Supercomputing Institute. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

References

[1] Peyman Abbaszadeh et al. The quest for model uncertainty quantification: A hybrid ensemble and variational data assimilation framework. *WRR*, 2019.

[2] Keith Beven. Facets of uncertainty: epistemic uncertainty, non-stationarity, likelihood, hypothesis testing, and communication. *Hydrological Sciences Journal*, 2016.

[3] Christopher M. Bishop. Mixture density networks. 1994.

[4] Charles Blundell et al. Weight uncertainty in neural network. *ICML*, 2015.

[5] Vanessa Böhm et al. Uncertainty quantification with generative models. *arXiv:1910.10046*, 2019.

[6] Tom E Botterill et al. Using machine learning to identify hydrologic signatures with an encoder-decoder framework. *Authorea Preprints*, 2022.

[7] Gemma Coxon et al. CAMELS-GB: hydrometeorological time series and landscape attributes for 671 catchments in Great Britain. *Earth System Science Data*, 2020.

[8] Chelsea Finn et al. Model-agnostic meta-learning for fast adaptation of deep networks. *ICML*, 2017.

[9] Yarin Gal et al. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *ICML*, 2016.

[10] Marta Garnelo et al. Neural processes. *CoRR*, 2018.

[11] Rahul Ghosh et al. Robust inverse framework using knowledge-guided self-supervised learning: An application to hydrology. *KDD*, 2022.

[12] Rahul Ghosh et al. Entity aware modelling: A survey. *arXiv:2302.08406*, 2023.

[13] Alex Graves et al. Framewise phoneme classification with bidirectional lstm networks. *IJCNN*, 2005.

[14] Timothy Hospedales et al. Meta-learning in neural networks: A survey. *IEEE TPAMI*, 2021.

[15] Xiaowei Jia et al. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS TDS*, 2021.

[16] Michael I Jordan et al. An introduction to variational methods for graphical models. *Machine Learning*, 1999.

[17] Hyunjik Kim et al. Attentive neural processes. *ICLR*, 2019.

[18] Diederik P Kingma et al. Auto-encoding variational bayes. *ICLR*, 2014.

[19] Daniel Klotz et al. Uncertainty estimation with deep learning for rainfall–runoff modeling. *HESS*, 2022.

[20] Frederik Kratzert et al. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *HESS*, 2019.

[21] Francesco Laio et al. Verification tools for probabilistic forecasts of continuous hydrological variables. *HESS*, 2007.

[22] Juho Lee et al. Bootstrapping neural processes. *NeurIPS*, 2020.

[23] Thomas Lees et al. Benchmarking data-driven rainfall–runoff models in Great Britain: a comparison of long short-term memory (LSTM)-based models with four lumped conceptual models. *HESS*, 2021.

[24] Aryan Mobiny et al. Dropconnect is effective in modeling uncertainty of bayesian deep networks. *Scientific Reports*, 2021.

[25] Andrew J. Newman et al. Gridded ensemble precipitation and temperature estimates for the contiguous united states. *Journal of Hydrometeorology*, 2015.

[26] Patrick Putzky and Max Welling. Recurrent inference machines for solving inverse problems. *arXiv:1706.04008*, 2017.

[27] Aniruddh Raghu et al. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv:1909.09157*, 2019.

[28] Somya Sharma et al. Probabilistic inverse modeling: An application in hydrology. *SDM*, 2023.

[29] Kihyuk Sohn et al. Learning structured output representation using deep conditional generative models. *NeurIPS*, 2015.

[30] Jung-Hun Song et al. Regionalization of a rainfall-runoff model: Limitations and potentials. *Water*, 2019.

[31] Yeming Wen et al. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *ICLR*, 2018.

[32] Jared D Willard et al. Daily surface temperatures for 185,549 lakes in the conterminous united states estimated using deep learning (1980–2020). *Limnology and Oceanography Letters*, 2022.

[33] Han Zhao et al. On learning invariant representations for domain adaptation. *ICML*, 2019.