Physics-Guided Architecture (PGA) of Neural Networks for Quantifying Uncertainty in Lake Temperature Modeling

Arka Daw * R. Quinn Thomas † Cayelan C. Carey ‡ Jordan S. Read §
Alison P. Appling § Anuj Karpatne *

Abstract

To simultaneously address the rising need of expressing uncertainties in deep learning models along with producing model outputs which are consistent with the known scientific knowledge, we propose a novel physics-guided architecture (PGA) of neural networks in the context of lake temperature modeling where the physical constraints are hard coded in the neural network architecture. This allows us to integrate such models with state of the art uncertainty estimation approaches such as Monte Carlo (MC) Dropout without sacrificing the physical consistency of our results. We demonstrate the effectiveness of our approach in ensuring better generalizability as well as physical consistency in MC estimates over data collected from Lake Mendota in Wisconsin and Falling Creek Reservoir in Virginia, even with limited training data. We further show that our MC estimates correctly match the distribution of ground-truth observations, thus making the PGA paradigm amenable to physically grounded uncertainty quantification.

1 Introduction

Given the success of deep learning methods in commercial domains such as computer vision, speech, and natural language processing, there is a growing interest in the scientific community to unlock the power of deep learning methods for advancing scientific discovery [2, 9, 14, 27]. Out of the many reasons fueling this interest, a primary factor is the rich ecosystem of advanced deep learning frameworks such as Conv Nets [17] and long short term memory (LSTM) models [11] that can handle complex structures in the data common in many scientific applications. Another reason is that with algorithmic innovations such as Dropout [28], we are not only moving toward robustness in deep learning but also toward better approaches for uncertainty quantification in deep learning, e.g., using the Monte Carlo (MC) Dropout method [8]. This is especially important in scientific problems where we need to produce uncertainty bounds in addition to point estimates, e.g., in climate change applications [24].

Despite dramatic advances in many commercial fields, current standards of deep learning has seen limited success in scientific applications (e.g., [4,18,22]), sometimes even leading to spectacular failures (e.g., [18]). This is primarily because of the *black-box* nature of conventional deep learning frameworks, which are trained solely on data and are agnostic to the underlying scientific principles driving real-world phenomena. As a first step in moving beyond black-box applications of deep learning, there is an emerging field of research combining scientific knowledge (or theories) with data science methods, termed *theory-guided data science* [15]. A promising line of research in this field is to guide the learning of neural network models using *physics-based loss functions* [13,16,30], that measure the violations of physical principles in the neural network outputs. We refer to this paradigm as physics-guided learning (PGL) of neural networks.

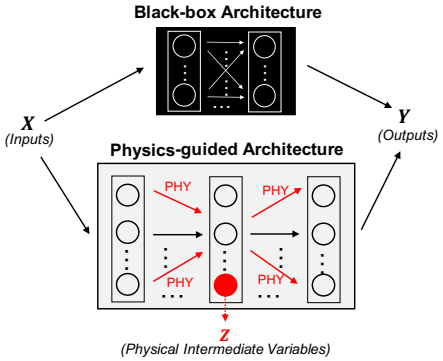


Figure 1: Physics-guided Architecture (PGA) paradigm of neural networks aims to infuse physics in neural network designs through *physics-informed connections* among neurons and through *physical intermediate variables*, shown in red. Note: Figures in this paper are best viewed in color.

While PGL formulations have been shown to improve generalization performance and generate more physically consistent predictions, adding a loss function in the learning objective still does not circumvent the black-box nature of neural network architectures, involving arbitrary design choices (e.g., number of layers and nodes per layer). As a result, black-box architectures are susceptible to producing physically inconsis-

[1]Department of Computer Science, Virginia Tech
[2]Department of Forest Resources and Environmental Conservation, Virginia Tech
[3]Department of Biological Sciences, Virginia Tech
[4]U.S. Geological Survey

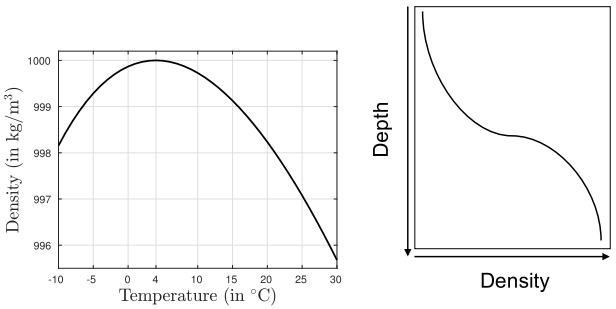
tent solutions with minor perturbations in the network weights, even after being trained with physics-based loss functions. This is a major concern when using uncertainty quantification methods such as MC dropout, where the network edges are randomly dropped with a small probability in the testing stage to produce a distribution of sample predictions for every test instance. Indeed, our results demonstrate that the randomness injected by MC dropout in the network weights easily breaks the ability of the PGL paradigm to preserve physical consistency in the sample predictions, leading to physically non-meaningful uncertainty estimates.

This paper presents innovations in an emerging field of theory-guided data science where, instead of using black-box architectures, we principally embed well-known physical principles in the neural network design. We refer to this paradigm as physics-guided architecture (PGA) of neural networks. Specifically, this paper offers two key innovations in the PGA paradigm for the illustrative problem of lake temperature modeling as illustrated in Figure 1. First, we introduce novel *physics-informed connections* among neurons in the network to capture physics-based relationships of lake temperature. Second, we associate physical meaning to some of the neurons in the network by computing physical intermediate variables Z in the neural pathway from inputs to outputs. By hard-wiring physics in the model architecture, the PGA paradigm ensures physical consistency of results regardless of small perturbations in the network weights, e.g, due to MC dropout. We compare the efficacy of our proposed approach with baseline methods on data collected from two lakes with differing physical characteristics and climatic regimes: Lake Mendota in Wisconsin, U.S.A., and Falling Creek Reservoir in Virginia, U.S.A.

The remainder of the paper is organized as follows. Section 2 provides a brief background of the problem of lake temperature modeling and relevant related work. Section 3 describes our proposed **PGA**-LSTM framework. Section 4 discusses our evaluation procedure while Section 5 presents results. Section 6 provides detailed analysis of our results and Section 7 provides concluding remarks and directions for future research.

2 Background and Related Work

2.1 Lake Temperature Modeling: Modeling the temperature of water in a lake is important from both economic and ecological perspectives. Water temperature is known to be principal driver of the growth, survival, and reproduction of economically viable fish [21,26] (see Appendix for more details). Hence, accurate and timely information about water temperature is necessary to monitor the ecological health of lakes and forecast future populations of fish and other aquatic taxa. Since observations of water temperatures are incomplete at broad spatial scales (or non-existent for most lakes), physics-based models of lake temperature, e.g., the General Lake Model (GLM) [10], are commonly used for studying lake processes. These modeling studies often use temperature of water at the centre of a lake at varying depth values[1] and time points for model validation. We adopt the same formulation to model the temperature of water in a lake, $Y_{d,t}$ at depth d and time t. There are two key physical relationships in lake temperature modeling that serve as the basis of the PGA innovations proposed in this paper. First, the temperature of freshwater can be directly mapped to density water according to the non-linear relationship shown in Figure 2(a) (notice that water is maximally dense at 4°C rather than 0). Second, denser water sinks to the bottom of the lake and thus resides at higher depth. Hence, density of water monotonically increases with depth as shown in the example plot of Figure 2(b). See Appendix for more details on these physical relationships.



(a) Temperature–Density Physics (b) Density–Depth Physics

Figure 2: Plots of two key physical relationships in lake temperature modeling: temperature-density physics and density-depth physics.

2.2 Physics-guided Machine Learning: Physics-guided Learning (PGL) is a recent paradigm in learning neural networks [13,16,30] where along with considering the prediction loss in the target space Y, we also measure the violations of physical principles in the model outputs $\widehat{Y}$, represented as *physics-based loss* in the PGL objective:

$$\arg\min \; Loss(Y, \widehat{Y}) + \lambda_{PHY} \text{ PHY}.Loss(\widehat{Y}). \tag{2.1}$$

where λ_{PHY} is a trade-off hyper-parameter that decides the relative importance of minimising the physical inconsistency compared to the empirical loss and the model complexity. By using physics-based loss, PGL restricts the search space of neural network weights to physically consistent options, thereby aiming to

[1] Depth is measured in the direction from lake surface to lake bottom.

achieve more generalizable and physically relevant predictions. For the problem of lake temperature modeling, Karpatne et al. [16] developed a PGL framework to measure violations of the two physical relationships introduced in Section 2.1. Jia et al. [13] extended this to work with time-based LSTM architectures and implemented an additional physics-based loss term to incorporate energy conservation. Other recent works like Xu et al. [31] integrated probabilistic logic with neural networks using semantic loss for classification tasks. Also, Pathak et al. [25] leveraged a set of linear constraints as loss functions for weakly supervised segmentation. Marquez et al. [23] imposed hard constraints on neural networks using constrained optimization formulations.

A major limitation of the PGL paradigm is that the choice of the neural network architecture is still black-box and not informed by physics. Even though minimizing physics-based loss helps in physically constraining the search space of neural network weights during training, there are no architectural constraints in the neural network design that guarantee the model predictions to be physically consistent on unseen test instances.

Physics-guided architecture of neural networks has recently gained popularity in several domains. Leibo et al. [19] proposed network connections to incorporate Hebbian rule of learning in neuroscience for view-tolerant facial detection. Another line of work has explored ways of embedding various forms of invariance in neural networks for problems in molecular dynamics [1] and turbulence modeling [20]. However, none of these developments are directly applicable to our problem of temperature prediction, where we need to encode physics available in the form of monotonic relationships and presence of intermediate variables.

2.3 Uncertainty Quantification: Uncertainty quantification (UQ) is critical for model evaluation in a number of scientific applications, where rather than producing point estimates of the target variable, it is preferred to have a distribution of the possible values. In our problem of lake temperature modeling, we wish to perform UQ to ascertain the amount of confidence we can place in our temperature predictions and its estimated impact on the population of fish species and other ecological variables.

A standard approach for performing UQ in neural networks is by using dropout [28] on the trained neural network weights in the testing phase, to produce Monte Carlo samples of the target variable for every test instance—a technique called Monte Carlo (MC) dropout [8]. While there are other methods in Bayesian deep learning for UQ that directly estimate posterior probabilities using priors on network weights [7], they are generally slower than MC Dropout. We use MC Dropout in our approach to perform UQ for lake temperature modeling, although our proposed PGA innovations are generic and can be coupled with any other method for UQ in deep learning.

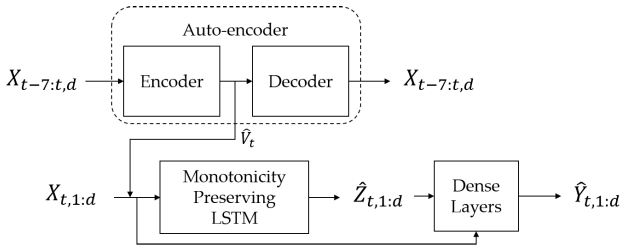


Figure 3: Proposed **PGA**-LSTM framework.

3 Proposed Framework

3.1 Overview of PGA-LSTM: Figure 3 provides an overview of our proposed physics-guided architecture of LSTM (**PGA**-LSTM) for lake temperature modeling. It is comprised of three basic components: (i) An LSTM based auto-encoder framework which extracts temporal features $\widehat{V}_t$ from the data at a given time t, (ii) a *monotonicity-preserving* LSTM which uses $\widehat{V}_t$ along with additional depth-based features $X_{t,1:d}$ to predict an intermediate physical quantity: density $\widehat{Z}_{t,d}$, while ensuring that $\widehat{Z}_{t,d} \geq \widehat{Z}_{t,d-1}$, and (iii) a multi-layer perceptron model which combines the density predictions $\widehat{Z}_{t,d}$ with the input drivers $X_{t,d}$ to finally predict the temperatures $\widehat{Y}_{t,d}$. In the following, we describe these three components in detail and present an end-to-end learning procedure for the complete **PGA**-LSTM framework.

3.2 Temporal Feature Extraction: The problem of lake temperature modeling can be viewed as a spatio-temporal sequential prediction problem. In order to develop a model which addresses both of these aspects simultaneously, we propose a simple yet effective method to incorporate spatio-temporal relationships into our model.

The autoencoder consists of two Recurrent Neural Networks (RNNs), the encoder LSTM and the decoder LSTM, as shown in Figure 4. We construct an input sequence by augmenting the feature vectors of the last 7 days of target date with its feature vector. This input $X_{t-7:t,d}$ is then fed into the encoder which generates some hidden representations $\widehat{V}_t$ for the target date. The decoder is then asked to reconstruct the entire input sequence from just the hidden representation $\widehat{V}_t$ of the target date. In order to do so, the representation must retain information about the sequential nature of the input data corresponding to the last week. Note that the dimensionality of the hidden representations is

intentionally kept smaller than the input dimensionality. This design of auto-encoder is inspired by the earlier work of Srivastava et al. [29].

3.3 Monotonicity-preserving LSTM: We build upon the basic LSTM architecture [11] that is designed to capture long-term and short-term memory effects in predicting a target sequence $Z_{1:d}$ given an input sequence $X_{1:d}$ using a recurrent neural network (RNN) framework. The basic idea of LSTM is to remember information for arbitrary long intervals by maintaining memory cell states, C_d, and hidden states, H_d. The cell state C_d is operated on by two neural network modules (or gates): *input* gate and *forget* gate, which can add or delete information in the cell state, respectively, and are learnable functions of the features, X_d, and the hidden state at the previous index, H_{d-1}. The cell state in turn affects the hidden state H_d using a learnable *output* gate. Finally, H_d is mapped to estimates of the target variable $\widehat{Z}_d$ using a stack of fully-connected dense layers.

While LSTM explicitly captures recurrence relationships between hidden states at consecutive indices, H_d and H_{d-1}, and thus offers some smoothness in its predictions, the choice of the recurrence forms are quite arbitrary and not informed by physics. In our **PGA**-LSTM formulation, $\widehat{Z}_{1:d}$ corresponds to a physically meaningful intermediate quantity: the density of water, which we know from Section 2.1 can only increase or remain constant with depth d. To incorporate this knowledge of density-depth physics, we introduce novel *physics-informed* connections in LSTM to have a monotonic recurrence relationship between Z_d and Z_{d-1}. The proposed monotonicity-preserving LSTM architecture is shown in Figure 5 and described in the following.

The main innovation in our proposed architecture (marked as red in Figure 5) is to not only keep track of the hidden state and cell state, H_d and C_d, respectively, but also the physical intermediate variable, Z_d, which we know can only increase with depth. Hence, we consider the problem of only predicting the positive increment in density, δ_d, as a function of H_d, which when added to Z_{d-1} yields Z_d. In particular, we apply

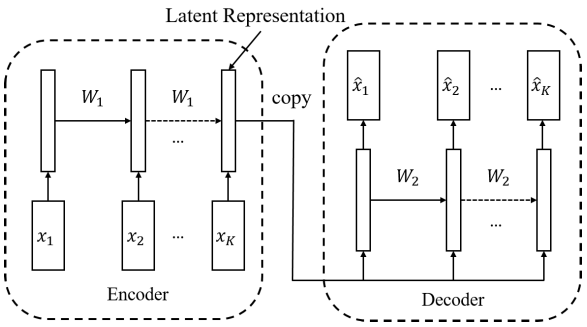


Figure 4: Proposed way for temporal feature extraction

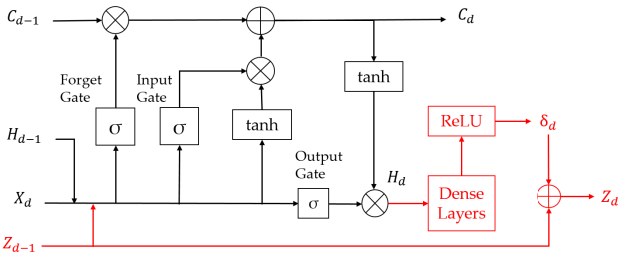


Figure 5: Monotonicity-preserving LSTM Architecture. Components in red represent novel physics-informed innovations in LSTM.

a stack of k dense hidden layers on H_d and pass the outputs through a ReLU activation function to predict positive values of δ_d. The complete set of equations for the forward pass of monotonicity-preserving LSTM is given by:

$$
\begin{aligned}
I_d &= \sigma(W_i[X_d, H_{d-1}, \mathbf{Z_{d-1}}] + b_i) \\
F_d &= \sigma(W_f[X_d, H_{d-1}, \mathbf{Z_{d-1}}] + b_f) \\
C_d &= F_d \circ C_{d-1} + I_d \circ \tanh(W_c[X_d, H_{d-1}, \mathbf{Z_{d-1}}] + b_c) \\
O_d &= \sigma(W_o[X_d, H_{d-1}, \mathbf{Z_{d-1}}] + b_o) \\
H_d &= O_d \circ \tanh(C_d) \\
\mathbf{L_d^1} &= \text{Activation}(\mathbf{W_1 H_d + b_1}) \\
\mathbf{L_d^i} &= \text{Activation}(\mathbf{W_i L_d^{i-1} + b_i}), \text{ where } i = 2,...,k. \\
\delta_d &= \mathbf{ReLU}(W_\delta L_d^k + b_\delta) \\
\mathbf{Z_d} &= \mathbf{Z_{d-1}} + \delta_d
\end{aligned}
$$

where, terms highlighted in red represent novel physics-informed innovations introduced in our proposed architecture compared to conventional LSTMs. Note that $\sigma(.)$ denotes the sigmoid activation, $\circ$ denotes the Hadamard product, $[a, b]$ denotes the concatenation of a and b, and $(W, b)_q$ denotes learnable weight and bias terms for all values of q.

While the physics-informed innovations in our **PGA**-LSTM model were specifically motivated by the density-depth physics in our target application, the idea of preserving monotonicity in LSTM outputs is useful in many other scientific applications. In general, our **PGA**-*monotonicity-preserving* LSTM framework can be used in applications where a target variable obeys monotonic constraints.

3.4 Mapping Density to Temperature: Having computed density as an intermediate variable in our **PGA**-LSTM framework, mapping estimates of density $\widehat{Z}_d$ at depth d to estimates of temperature $\widehat{Y}_d$ at depth d appears quite straightforward. Ideally, one can refer to the temperature-density physics introduced in Section 2.1 to infer density given temperature. However, the physical mapping from density to temperature is one-to-many and thus non-unique (see Figure 2(a)). In

particular, a given value of density Z can be mapped to two possible values of temperature Y, one corresponding to the freezing phase ($Y < 4°C$) and the other corresponding to the warming phase ($Y > 4°C$). To address this, we learn the mapping from $\widehat{Z}_d$ to $\widehat{Y}_d$ directly from the data, by concatenating X_d with $\widehat{Z}_d$ and feeding the concatenated values to a stack of fully-connected dense layers with a single output node predicting the target variable $\widehat{Y}_d$. Since $\widehat{Z}_d$ is already a strong physical predictor of Y_d, we do not need a deep architecture to map density to temperature and thus use a small number of hidden layers.

3.5 End-to-end Learning Procedure: One of the benefits of our **PGA**-LSTM framework is that along with predicting the target variable: temperature, $\widehat{Y}$, it also produces estimates of a physical intermediate variable: density, $\widehat{Z}$, as ancillary outputs. Further, during the training stage, ground-truth observations of temperature, Y, can be converted to ground-truth estimates of density, Z, using the one-to-one physical mapping from temperature to density. Hence, we perform end-to-end training of the complete **PGA**-LSTM model by minimizing the empirical loss over both Y and Z in the following learning objective:

$$\begin{aligned} \underset{(W,b)}{\arg\min} \quad & \text{Loss}(Y, \widehat{Y}) + \lambda_Z \text{ Loss}(Z, \widehat{Z}) + \lambda_R \text{ R}(W) \\ & \text{R}(W) = ||W||_2. \end{aligned} \tag{3.2}$$

Here, $Y_{d,t}$ and $Z_{d,t}$ are the observed temperature and density values, respectively, at depth d and time t, N is the total number of observations, (W, b) is the combined set of weights and bias terms, respectively, across all components of **PGA**-LSTM, and λ_Z and λ_R are the trade-off parameters for the density prediction loss and regularization loss, respectively.

4 Evaluation Setup

4.1 Data and Experiment Design: Our proposed **PGA**-LSTM model was trained and tested on two lakes that differed in depth, size, and climatic conditions. The first lake, Lake Mendota in Wisconsin, USA, is approximately 40 km^2 in surface area with a maximum depth of approximately 25 m. Lake Mendota is a dimictic lake with seasonal variation in water temperatures from 0° in the winter to nearly 30° in the summer. The overall data for lake Mendota consisted of 35,213 observations. Falling Creek Reservoir (FCR) in Virginia, USA, is approximately 0.119 km^2 in surface area with a maximum depth of 9.3 m [5]. Similar to Mendota, FCR is also dimictic and has seasonal variation in water temperatures from 0° in the winter to nearly 30° in the summer. The overall data for FCR consisted of 7588 observations [6].

We partitioned the data into two contiguous time windows to be used for training and testing, such that there is no temporal auto-correlation between the training and test sets. The first 4 years was used for training for both the lakes and the remaining for testing. Both the input and density outputs were normalised to zero mean unity standard deviation. The main temperature output Y was not normalised. Please see Appendix for detailed description of the data and model specifications [2].

4.2 Evaluation Metrics: We consider the root mean square error (RMSE) of a model on the test set as a metric of generalizability of the model. We also consider Physical Inconsistency as another evaluation metric, which is defined as the fraction of times the MC sample predictions at consecutive depths are physically inconsistent, i.e., they violate the density-depth relationship. We have used a tolerance value of 10^{-5} kg/m^3 to decide if a difference in density across consecutive depths is physically inconsistent or not.

4.3 Baselines: We have chosen the following baselines for comparison. (1) An LSTM with similar architecture as a **PGA**-LSTM was chosen for comparison. The black-box LSTM has 8 memory units followed by four dense layers with 5 hidden units each, finally followed by a dense layer with one unit. (2) The PGL-LSTM was used as another baseline with a similar architecture to the LSTM but using physics-guided learning in the form of loss functions. The PGL-LSTM tries to minimise the physics based loss that can evaluated by computing the physical inconsistency of two consecutive depth outputs. Note that we did not consider the PGRNN approach [12] for lake temperature modeling using physics-guided learning in RNNs as a baseline in this paper due to two main reasons. First, this paper only considers density-depth relationships while PGRNN also considers energy conservation. Second, PGRNN builds RNNs in time dimension whereas we are building LSTM in the depth dimension, to make use of the density-depth relations directly in the creation of physics-guided architecture.

5 Results

5.1 Comparing PGA-LSTM with Baselines: Tables 1 and 2 compare the performance of **PGA**-LSTM with baseline methods on Lake Mendota and FCR, respectively, using 40% data for training on both lakes. We are interested in two evaluation metrics: Test RMSE and Physical Inconsistency of the predicted temperature profiles. Both metrics can be evaluated either

[2] The codes for **PGA**-LSTM are available on Github: https://github.com/arkadaw9/PGA_LSTM

	Test RMSE (in $°C$)		Physical Inconsistency	
	Per Sample	*Mean*	*Per Sample*	*Mean*
LSTM	2.25 ± 0.14	1.63 ± 0.08	0.32 ± 0.03	0.10 ± 0.02
PGL	2.30 ± 0.12	$\mathbf{1.62 \pm 0.10}$	0.34 ± 0.02	0.12 ± 0.02
PGA	$\mathbf{2.09 \pm 0.18}$	1.63 ± 0.04	$\mathbf{0.01 \pm 0.01}$	$\mathbf{0.00 \pm 0.00}$

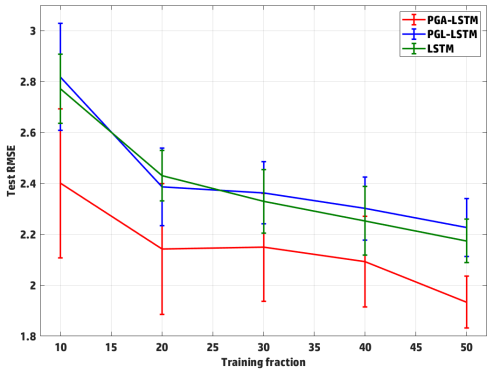
Table 1: Results on Lake Mendota using 40% training data.

	Test RMSE (in $°C$)		Physical Inconsistency	
	Per Sample	*Mean*	*Per Sample*	*Mean*
LSTM	2.96 ± 0.22	2.27 ± 0.17	0.28 ± 0.02	0.07 ± 0.03
PGL	2.84 ± 0.16	2.12 ± 0.13	0.27 ± 0.02	0.08 ± 0.03
PGA	$\mathbf{2.19 \pm 0.21}$	$\mathbf{1.88 \pm 0.12}$	$\mathbf{0.00 \pm 0.01}$	$\mathbf{0.00 \pm 0.00}$

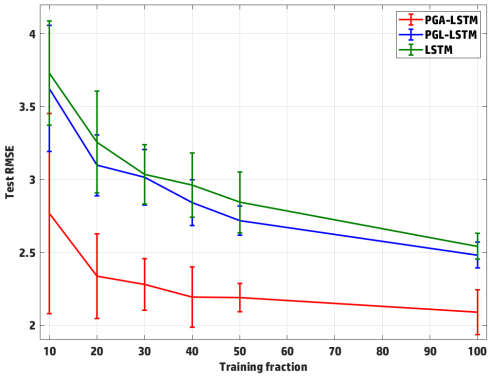
Table 2: Results on FCR using 40% training data.

on the individual MC samples (referred to as *Per Sample*) or on the mean of the MC samples (referred to as *Mean*). (Note that we are interested in *Per Sample* evaluation as we want every MC sample to be accurate and physically consistent for the *Mean* results to make sense in scientific applications.) We can see from Table 1 that on Lake Mendota, LSTM has fairly high *Per Sample* Test RMSE, illustrating the limitations of black-box models in achieving good generalizability. Further, the *Per Sample* Physical inconsistency of LSTM is 0.32, which indicates that the MC samples generated from LSTM are physically inconsistent 32% of the time. If we consider the mean of MC samples generated from LSTM, we can get lower *Mean* Test RMSE, due to the cancellation of noise through aggregation. However, the *Mean* Physical Inconsistency of LSTM is still fairly high.

If we employ the PGL paradigm, we can see that PGL-LSTM shows little to no improvement in RMSE or Physical Inconsistency in comparison to LSTM. Note that every dropout network represents a slightly perturbed version of the trained neural network model. Ideally, we want every dropout network to produce physically consistent simulations of the target variable, so that the UQ analysis is physically meaningful. However, if we use black-box architectures, it is highly likely to obtain dropout networks that produce physically inconsistent solutions even after using the PGL paradigm. This is because the dropout procedure effectively injects a small amount of randomness in the neural network weights, which may be sufficient to *unlearn* the physical consistency introduced during training by the PGL paradigm. In contrast to the baseline methods, we can observe that our proposed **PGA**-LSTM model shows the smallest *Per Sample* Test RMSE while always preserving physical consistency, even after performing MC dropout. Table 2 shows similar trends in the results of



(a) Lake Mendota

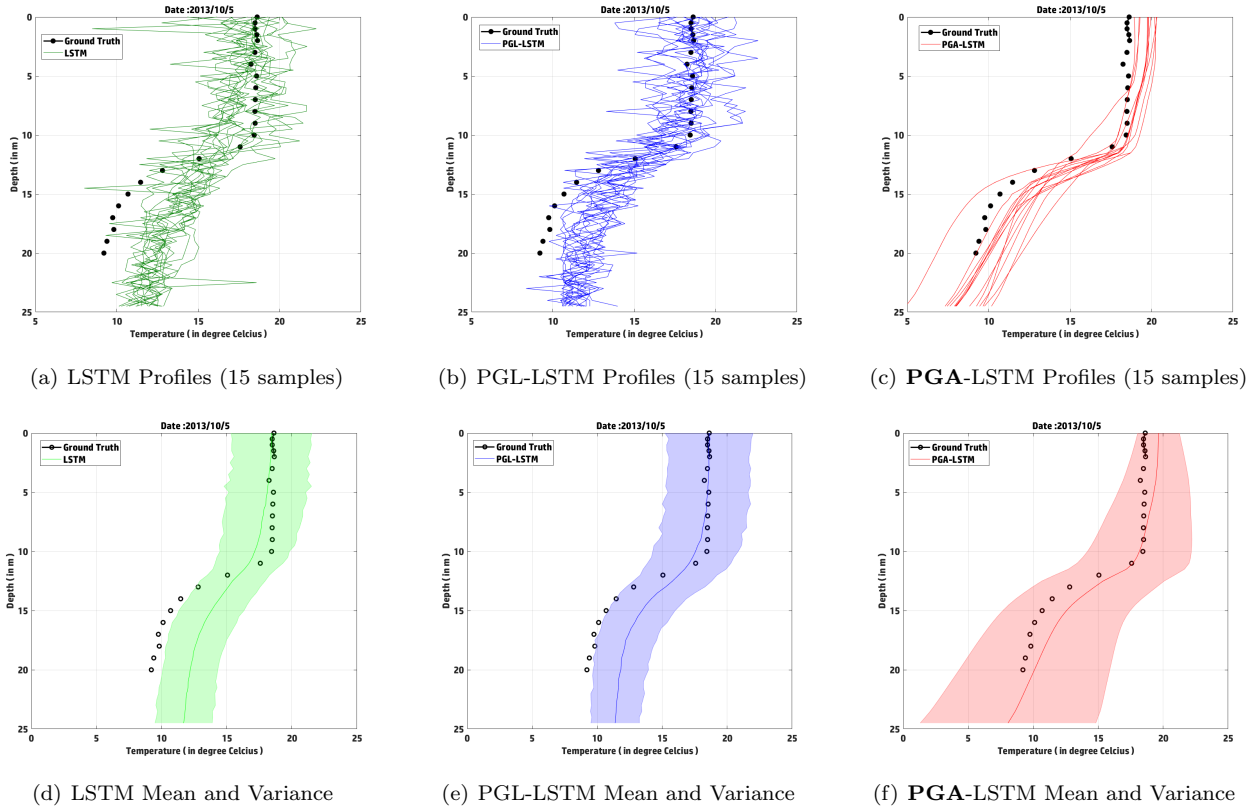


(b) FCR

Figure 6: Test RMSE (*per sample*) on varying training sizes.

PGA-LSTM w.r.t. baseline methods on FCR.

5.2 Effect of Varying Training Size: To demonstrate the effects of reducing training size on the accuracy of comparative models, Figure 6 shows the *Per Sample* Test RMSE of **PGA**-LSTM, PGL-LSTM, and LSTM on varying training fractions. We can see that the test RMSE of all methods increase as we reduce the amount of data available for training in both Lake Mendota and FCR. However, we can see that **PGA**-LSTM shows the lowest Test RMSE for all values of training fractions in both the lakes. While Lake Mendota and FCR represent heavily studied water bodies, a majority of lakes in the USA (and the world) suffer from limited number of observations. Hence, by comparing models in scarcity of training data on these lakes, we intend to simulate real-life scenarios on other unseen lakes where temperature models have to be deployed. Note that in FCR, the rate of increase in RMSE as we reduce training size is lowest for **PGA**-LSTM as compared to all baseline methods. This resonates with the fact that the need for introducing physics to achieve better test RMSE is



(a) LSTM Profiles (15 samples) (b) PGL-LSTM Profiles (15 samples) (c) **PGA**-LSTM Profiles (15 samples)

(d) LSTM Mean and Variance (e) PGL-LSTM Mean and Variance (f) **PGA**-LSTM Mean and Variance

Figure 7: Temperature profiles of comparative models on October 5, 2013 for Lake Mendota.

higher at smaller training sizes, when black-box models have higher risks of over-fitting and learning spurious solutions.

6 Analysis of Results

6.1 Visualizing Temperature Profiles: Figures 7(a), 7(b), and 7(c) show plots of 15 sample temperature profiles generated by comparative models on a representative test date of October 15, 2013 in Lake Mendota, when trained on 40% data. Note that these 15 samples have been selected at random from the pool of dropout MC samples generated on this test date across all 10 random runs of training. We can observe that the sample profiles of LSTM and PGL-LSTM are highly physically inconsistent (i.e., temperature profiles show no monotonic behavior with depth). Hence, despite their RMSE values, a lake scientist will have lower confidence in trusting their results and using them in subsequent scientific analyses. In contrast to baseline methods, **PGA**-LSTM produces sample profiles that are always physically consistent, and thus are useful from a domain perspective.

To analyze the validity of MC profiles in capturing the uncertainty around temperature predictions, Figures 7(d), 7(e), and 7(f) show the mean and variance of comparative models for the complete pool of dropout samples generated on this test date across all 10 random runs of training. The error bars in these plots have been generated using two standard deviations around the mean, thus capturing 99.7% of samples under Gaussian assumption. We can see that the mean profiles of LSTM and PGL-LSTM are close to the ground-truth and the error bars engulf the ground-truth in the shallower portion of the lake (depth < 10m). However, as we move to the deeper portion of the lake (depth > 10m), the ground-truth observations start to depart from the distribution of samples generated by LSTM and PGL-LSTM and escapes outside the error bars. On the other hand, the distribution of samples generated using **PGA**-LSTM accurately envelops the ground-truth observations at every portion of the lake irrespective of depth. [3]

6.2 Assessing Uncertainty Estimates: While Figure 7(f) shows that the ground-truth observations

[3] While Figure 7 provides results over a single test date in Lake Mendota, videos of the results for both lakes for all test dates are available at the following link : **click here**

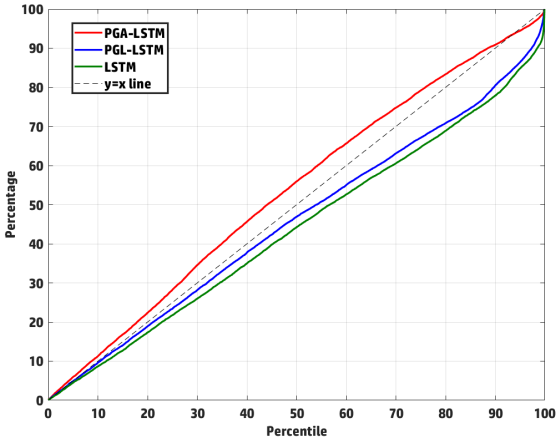


Figure 8: Cumulative percentage of observations within a certain percentile of samples of comparative models on Lake Mendota [3].

are contained within 99.7% samples of **PGA**-LSTM on a given date in Lake Mendota, we attempt to quantitatively assess the validity of our uncertainty estimates as compared to baseline methods across all dates. Ideally, if the uncertainty estimates produced by a model are valid, we should expect the distribution of its samples to accurately match the distribution of ground-truth observations on test points. In other words, if we look at the k^{th} percentile of samples generated by an ideal model, then we should expect to observe $k\%$ of ground-truth test points to fall within it [3]. To capture this idea, we first fit a Gaussian distribution on the complete pool of samples generated by a model on a test point, and then estimate the two-tailed percentile of the ground-truth observed at that point. Figure 8 plots the cumulative percentage of ground-truth observations (Y-axis) that fall within a certain percentile of samples generated by comparative models (X-axis). The ideal model is represented by the diagonal line $y = x$, where the percentage of ground-truth points within a percentile is equal to the percentile value. Models that are over-confident would have fewer ground-truth points within a certain percentile and hence would lie below the diagonal. Conversely, models that are under-confident would reside above the $y = x$ diagonal. We can see from Figure 8 that the baseline models, LSTM and PGL-LSTM, lie just below the diagonal and hence produce slightly over-confident uncertainty estimates, i.e., the distribution of ground-truth points sometimes falls outside the distribution of MC samples. On the other hand, **PGA**-LSTM lies just above the diagonal and hence produces slightly larger uncertainty estimates than what is ideally expected. Note that in the absence of any information about the ground-truth on test dates, it is generally desirable to be slightly under-confident and produce wider uncertainty bounds than to be over-confident with narrower bounds. Further, even though the uncertainty bounds of **PGA**-LSTM are wider, it produces lower *Per Sample* Test RMSE than all other baseline methods. This illustrates that **PGA**-LSTM produces reasonably good uncertainty estimates that captures the distribution of the ground-truth observations.

7 Conclusions and Future Work

This paper explored an emerging direction in theory-guided data science to move beyond black-box neural network architectures and design physics-guided architectures (PGA) of neural networks that are informed by physics. We specifically develop a novel **PGA**-LSTM model for the problem of lake temperature modeling, where we design a *monotonicity-preserving* LSTM module to predict physically consistent densities. We compared our **PGA**-LSTM model with baseline methods to demonstrate its ability to produce generalizable and physically consistent solutions, even after making minor perturbations in the network weights by the Monte Carlo (MC) dropout method for uncertainty quantification.

Future work will explore applications of the proposed **PGA**-LSTM in other scientific problems that show monotonic recurrence relationships. Future work will also explore the effect of other state of the art uncertainty prediction methods on physics-guided architecture models. Extensions of the PGA framework for capturing more complex forms of physical relationships in space and time will be explored as well. Future work can also study the impact of **PGA**-LSTM on physical interpretability of neural networks, since the features extracted at the hidden layers of the network correspond to physically meaningful concepts.

References

[1] Brandon Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. *arXiv preprint arXiv:1906.04015*, 2019.

[2] Tim Appenzeller. The scientists' apprentice. *Science*, 357(6346):16–17, 2017.

[3] Jochen Bröcker and Leonard A Smith. Increasing the reliability of reliability diagrams. *Weather and forecasting*, 22(3):651–661, 2007.

[4] Peter M Caldwell, Christopher S Bretherton, Mark D Zelinka, Stephen A Klein, Benjamin D Santer, and Benjamin M Sanderson. Statistical significance of climate sensitivity predictors obtained by data mining. *Geophysical Research Letters*, 41(5):1803–1808, 2014.

[5] CC Carey, JP Doubek, RP McClure, and PC Hanson. Oxygen dynamics control the burial of organic carbon

in a eutrophic reservoir. *Limnology and Oceanography-Letters*, 3:293–301, 2018.

[6] CC Carey, RP McClure, AB Gerling, JP Doubek, S Chen, ME Lofton, and KD Hamre. Time series of high-frequency profiles of depth, temperature, dissolved oxygen, conductivity, specific conductivity, chlorophyll a, turbidity, ph, and oxidation-reduction potential for beaverdam reservoir, carvins cove reservoir, falling creek reservoir, gatewood reservoir, and spring hollow reservoir in southwestern virginia, usa, 2013 - 2018. *Environmental Data Initiative, link to dataset*, 2019.

[7] Meire Fortunato, Charles Blundell, and Oriol Vinyals. Bayesian recurrent neural networks. *arXiv:1704.02798*, 2017.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, 2016.

[9] D Graham-Rowe, D Goldston, C Doctorow, M Waldrop, C Lynch, F Frankel, R Reid, S Nelson, D Howe, SY Rhee, et al. Big data: science in the petabyte era. *Nature*, 455(7209):8–9, 2008.

[10] Matthew R Hipsey, Louise C Bruce, Casper Boon, Brendan Busch, Cayelan C Carey, David P Hamilton, Paul C Hanson, Jordan S Read, Eduardo de Sousa, Michael Weber, et al. A general lake model (glm 3.0) for linking with high-frequency sensor data from the global lake ecological observatory network (gleon). *Geoscientific Model Development*, 12(1):473–523, 2019.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] Xiaowei Jia, Ankush Khandelwal, Nayak Guru, James Gerber, Kimberly Carlson, Paul West, and Vipin Kumar. Predict land covers with transition modeling and incremental learning. In *SDM)*, 2017.

[13] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *SDM*, pages 558–566. SIAM, 2019.

[14] TO Jonathan, AM Gerald, et al. Special issue: dealing with data. *Science*, 331(6018):639–806, 2011.

[15] Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[16] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[18] David Lazer, Ryan Kennedy, Gary King, and Alessandro Vespignani. The Parable of Google Flu: Traps in Big Data Analysis. *Science (New York, N.Y.)*, 343(6176):1203–5, March 2014.

[19] Joel Z Leibo, Qianli Liao, Fabio Anselmi, Winrich A Freiwald, and Tomaso Poggio. View-tolerant face recognition and hebbian learning imply mirror-symmetric neural tuning to head orientation. *Current Biology*, 27(1):62–67, 2017.

[20] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[21] John J Magnuson, Larry B Crowder, and Patricia A Medvick. Temperature as an ecological resource. *American Zoologist*, 19(1):331–343, 1979.

[22] Gary Marcus and Ernest Davis. Eight (no, nine!) problems with big data. *The New York Times*, 6(04):2014, 2014.

[23] Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Imposing hard constraints on deep networks: Promises and limitations. *arXiv:1706.02025*, 2017.

[24] James M Murphy, David MH Sexton, David N Barnett, Gareth S Jones, Mark J Webb, Matthew Collins, and David A Stainforth. Quantification of modelling uncertainties in a large ensemble of climate change simulations. *Nature*, 430(7001):768, 2004.

[25] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, pages 1796–1804, 2015.

[26] James J Roberts, Kurt D Fausch, Douglas P Peterson, and Mevin B Hooten. Fragmentation and thermal risks from climate change interact to affect persistence of native trout in the colorado river basin. *Global Change Biology*, 19(5):1383–1398, 2013.

[27] Terrence J Sejnowski, Patricia S Churchland, and J Anthony Movshon. Putting big data to good use in neuroscience. *Nature neuroscience*, 17(11):1440–1441, 2014.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958, 2014.

[29] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. In *ICML*, ICML'15, pages 843–852. JMLR.org, 2015.

[30] Russell Stewart and Stefano Ermon. Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, 2017.

[31] Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A semantic loss function for deep learning with symbolic knowledge. *arXiv preprint arXiv:1711.11157*, 2017.