Knowledge Guided Machine Learning for Extracting, Preserving, and Adapting Physics-aware Features

Erhu He[1], Yiqun Xie[2], Licheng Liu[3], Zhenong Jin[3], Dajun Zhang[4], Xiaowei Jia[1]

[1] University of Pittsburgh, [2] University of Maryland, [3] University of Minnesota, [4] Carleton University

[1]{erh108,xiaowei}@pitt.edu,[2]xie@umd.edu,[3]{lichengl,jinzn}@umn.edu,,[4]dajunzhang@cmail.carleton.ca

Abstract

Training machine learning (ML) models for scientific problems is often challenging due to limited observation data. To overcome this challenge, prior works commonly pre-train ML models using simulated data before having them fine-tuned with small real data. Despite the promise shown in initial research across different domains, these methods cannot ensure improved performance after fine-tuning because (i) they are not designed for extracting generalizable physics-aware features during pre-training, (ii) the features learned from pre-training can be distorted by the fine-tuning process. In this paper, we propose a new learning method for extracting, preserving, and adapting physics-aware features. We build a knowledge-guided neural network (KGNN) model based on known dependencies amongst physical variables, which facilitate extracting physics-aware feature representation from simulated data. Then we fine-tune this model by alternately updating the encoder and decoder of the KGNN model to enhance the prediction while preserving the physics-aware features learned through pre-training. We further propose to adapt the model to new testing scenarios via a teacher-student learning framework based on the model uncertainty. The results demonstrate that the proposed method outperforms many baselines by a good margin, even using sparse training data or under out-of-sample testing scenarios.

Keywords: physics-aware features, knowledge-guided neural networks, data mining

1 Introduction

Environmental processes involve complex interactions amongst physical variables, such as weather, water, soil conditions, plants, and microbes, at different spatial and temporal scales. These processes, which jointly form the cycling of energy, water, and carbon, are simulated by existing physics-based models developed across many scientific domains, such as climate science, hydrology, agriculture, and meteorology [1, 2, 3, 4]. However, these models often use approximations or parameterizations due to incomplete knowledge or excessive complexity in modeling certain processes [5, 6, 7].

Due to the importance of this problem, there is an increasing interest in building data-driven machine learning (ML) models for modeling environmental processes in many societally important applications, such as monitoring agriculture production [8], predicting water temperature and streamflow [9], and forecasting weather and climate [10]. As standard ML models need large training data for capturing complex patterns amongst all the physical variables, one promising direction is to transfer knowledge from physics-based models for training ML models [11]. In particular, prior work has shown that ML models can achieve better accuracy after being pre-trained using simulated data generated by physics-based models, especially when observation data are scarce [12, 13, 14, 15]. Despite the promise of initial research on this topic, these methods remain limited in model generalization due to two reasons. First, they often use standard model architectures and training procedures in pre-training, which cannot ensure learning generalizable features related to physical processes. Second, the standard fine-tuning process (over the entire model) adopted in these works may distort useful features learned from large simulated data [16, 17, 18], leading to the overfitting issue.

To address these challenges, we propose a new method, **K**nowledge-**G**uided **P**retraning, **F**inetuning, and **A**daptation (KGPFA), in the context of predicting crop yield, which is critical for ensuring food security for the growing population nowadays. The proposed KGPFA method uses simulated data and true observations to train a customized ML model for modeling complex physical processes through three learning stages. First, we build a knowledge-guided model architecture to embed physical information through the pre-training process using simulated data generated by physics-based models. Second, we develop a new fine-tuning method, which alternately updates the decoder and encoder of the ML model. The decoder is updated while we freeze the encoder to maintain the extracted feature representation from the pre-training process. Then the encoder is updated in a conservative manner by reducing the model uncertainty for only the training samples with significantly higher prediction accuracy. Third, we conduct model adaptation in the testing phase using a teacher-student approach. The teacher model is updated using only confident samples while preserving physical constraints, and then used for guiding the training of the student model.

We evaluate the proposed method using real corn yield data over a 21-year period in Iowa and Illinois, two leading states of corn production in the United States. The results demonstrate that our proposed method can achieve good prediction accuracy in data-sparse and out-of-sample scenarios. We also verify the effectiveness of each step in the proposed KGPFA method.

2 Related Work

Recent research has shown immense success in integrating physics knowledge into ML models to improve predictive performance and solve general scientific problems. The most common ways include applying additional loss functions [15, 19] and other hybrid approaches [20]. Our recent survey [11] summarized existing methods for integrating scientific knowledge into ML models. For example, Hanson et al. [21] added ecological principles as physical constraints into the loss function of ML models to improve the lake phosphorus prediction. In a case to simulate the lake temperature, Karpatne et al. [19] introduced new training loss to enforce the physical relationship that the density of water at a lower depth is always greater than the density of water at any depth above. Then, our previous works [15, 22, 23] further proposed new methods to reduce search space and improve prediction accuracy by penalizing violations of energy and mass conservation.

Advanced ML models often require a large amount of representative training samples, which can be expensive to obtain in scientific applications. To address this issue, one solution is to augment the model training with simulated data generated by physics-based models under varying yet realistic physical parameters. Prior work has shown that simulated data generated by physics-based models can be used to improve the prediction through residual modeling [24] and augmentation of model input [19]. Both of these methods aim to reduce the complexity of the prediction task, but their performance can still degrade for complex problems given limited training samples. Prior work also investigated another approach that pre-trains ML models using simulated data for either final output variables or intermediate physical variables, and found they can perform much better under data-scarce scenarios in a range of scientific applications [14, 15, 25, 26, 27, 28, 29, 30, 31].

However, the fine-tuning process is found to distort feature representation learned from the pre-training phase [16]. This is likely to degrade the model generalizability, especially when real observations are sparse and testing data are in a different distribution. An alternative approach is to freeze the pre-trained feature representation and tune only a few remaining layers [32]. Prior work also employed this approach before fine-tuning the entire model, and achieved better accuracy and generalization performance [16]. Some other works further investigated improving the fine-tuning process by modifying the loss function [17] and creating additional synthetic samples [33].

Unsupervised model adaptation techniques have been widely studied to enhance the prediction on the target data (i.e., unlabeled testing data) with distribution shifts. Existing methods can be classified into two categories, the methods with access to the source data (i.e., labeled training data), e.g., domain adaptation [34] , and the methods without access to the source data, e.g., test-time adaptation [35, 36, 37]. These approaches refine the model during the testing phase either by reducing the distributional gap between the source and target data [34], or by optimizing additional objectives, e.g., entropy [36] and rotation prediction [35].

3 Problem definition

The objective of this work is to predict the county-level yield for corn in target years. For each county, we are provided with daily input features within each year. Specifically, we use the index i to represent a specific combination of a county and a year, and the input features for the sample i are represented as $\boldsymbol{x}_i = \{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, ..., \boldsymbol{x}_i^T\}$, where $T = 365$ in a non-leap year. The daily features $\boldsymbol{x}_i^t$ include weather drivers (e.g., precipitation, solar radiation), and soil and crop properties. The feature values are obtained as the average of the variable values from a set of randomly sampled farm locations in each county. More details can be found in Section 5.2. Additionally, we have the access to the yearly crop yield labels $\boldsymbol{Y} = \{y_i\}$ from agricultural surveys in the training set $\mathcal{R}$. In the testing set $\mathcal{T}$, we only have the input features but do not have the crop yield labels. We create the training and testing sets by splitting the available data based on different years while keeping the same set of counties across training and testing data.

In addition to the real crop yield dataset, we also run the physics-based Ecosys model [1] to simulate crop yield. We use $\mathcal{S}$ to represent the simulated dataset on a set of combinations of (counties, years). Another benefit of the physics-based model is that it can simulate intermediate physical variables in the crop growing process, such as variables involved in carbon cycling (e.g., autotrophic respiration (Ra), heterotrophic respiration (Rh), and net ecosystem exchange (NEE)). It is noteworthy that physics-based models are often biased as they are necessarily approximations of reality. Hence, the simulations can only be used for weak supervision.

4 Method

In this section, we will describe the proposed KGPFA method, which is outlined in Fig. 1. The proposed method aims to extract physics-aware features through pre-training, preserves the learned features in fine-tuning, and adapts the features to the new environment during the test phase. In particular, we first introduce a knowledge-guided neural network (KGNN) model that integrates known physical knowledge to enhance the pre-training from simulated data. Next, we decompose the KGNN model into the encode and decoder components and fine-tune them alternately using real data. In the fine-tuning phase, the decoder is updated to transform learned physics-aware feature representation to better fit observed data samples while the encoder is updated moderately to refine physics-aware features to mitigate bias learned from the simulated data. Finally, we propose an adaptation method to update the KGNN model in the testing phase. A separate teacher model is constructed to guide the adaptation of the KGNN model, and the teacher model is trained in a conservative manner using only confident samples.

4.1 Knowledge-guided networks for pre-training. Environmental processes in agricultural systems involve the interactions amongst different physical variables (e.g., weather, soil conditions, plant, and respiration). These variables can be either observed or unobserved. These processes, which jointly form the cycling of energy, water, and carbon, are simulated by existing physics-based [1, 2, 3, 4] through a series of mathematical equations. However, these models often use approximations or parameterizations, and also require high computational cost [5, 6, 7].

Standard ML models need large training data for capturing complex patterns among all the physical variables. Recent research in integrating physical knowledge into ML has shown promise in a variety of scientific applications [11]. Despite the promise of initial research on this topic, existing KGNN models remain limited in capturing inter-dependencies amongst multiple processes in complex systems.

We aim to build a new KGNN with a customized network structure that is consistent with existing physics-based models. This entails an interpretable and differentiable model architecture by ascribing physical meaning to intermediate network outputs. A major advantage of this method is the ability to output many intermediate variables in addition to the final target variable (e.g., yield), which enables interpreting and tracking the states of the target systems and applying relevant physical constraints over different physical variables (e.g., the conservation of mass and energy).

In the context of predicting crop yield, we take weather and soil variables as input drivers to simulate the carbon cycle in the crop growing process. The KGNN architecture used in this work is illustrated in Fig. 1. Starting from the sequence of daily input features $\boldsymbol{x}_i^t$, we first use long-short term memory (LSTM) layers to embed the temporal patterns in the input data, as $\{\boldsymbol{h}_i^t\}_{t=1}^T = \text{LSTM}(\{\boldsymbol{x}_i^t\}_{t=1}^T)$. Then we transform the LSTM embeddings through two separate fully-connected network branches, to embed plant-related and soil-related information, respectively. We use $\boldsymbol{hp}_i^t$ and $\boldsymbol{hs}_i^t$ to represent the obtained plant-related embedding and soil-related embedding, respectively.

Next, we simulate three key variables in the carbon cycle: (i) the ecosystem autotrophic respiration (Ra) is generated from the plant-related embedding, (ii) the ecosystem heterotrophic respiration (Rh) is generated from the soil-related embedding, and (iii) the net ecosystem exchange (NEE) is generated from the concatenation of plant-related and soil-related embeddings. The entire carbon cycle can be captured by a mass conservation relation, as $-\text{NEE} = \text{GPP} - \text{Ra} - \text{Rh}$, where GPP represents the gross primary production, and can be estimated from remote sensing. The estimated GPP values are available over large regions and thus are used as input to the KGNN model.

During the pre-training phase, we are provided with the simulated values for Ra, Rh, NEE generated by the physics-based Ecosys model. Then we define a mean square error (MSE)-based loss function for measuring the difference between KGNN-predicted values (denoted by $\widehat{\text{Ra}}$, $\widehat{\text{Rh}}$, $\widehat{\text{NEE}}$) and the simulated values, as follows:

$$\begin{aligned}\mathcal{L}_{\text{sim}} = \sum_i \sum_t (||\text{Ra}_i^t - \widehat{\text{Ra}}_i^t||^2 + ||\text{Rh}_i^t - \widehat{\text{Rh}}_i^t||^2 \\ + ||\text{NEE}_i^t - \widehat{\text{NEE}}_i^t||^2)/(|\mathcal{S}|T)\end{aligned} \tag{4.1}$$

We also define a physical loss that measures the violation of the mass conservation law, as follows:

$$\mathcal{L}_{\text{phy}} = \sum_i \sum_t (\text{GPP}_i^t - \widehat{\text{Ra}}_i^t - \widehat{\text{Rh}}_i^t + \widehat{\text{NEE}}_i^t)^2/(|\mathcal{S}|T) \tag{4.2}$$

Besides, we generate the target variable, i.e., crop yield $\hat{y}_i$, from the plant-related embedding $\{\boldsymbol{hp}_i^t\}_{t=1}^T$. As the crop yield is available yearly while the input is available daily, we use an attention layer to aggregate the plant-related embeddings over time and generate predictions, as follows:

$$\begin{aligned}\hat{y}_i = f(\sum_t \alpha_i^t \, \boldsymbol{hp}_i^t), \\ \{\alpha_i^t\}_{t=1}^T = \text{softmax}(\{g(\boldsymbol{x}_i^t)\}_{t=1}^T),\end{aligned} \tag{4.3}$$

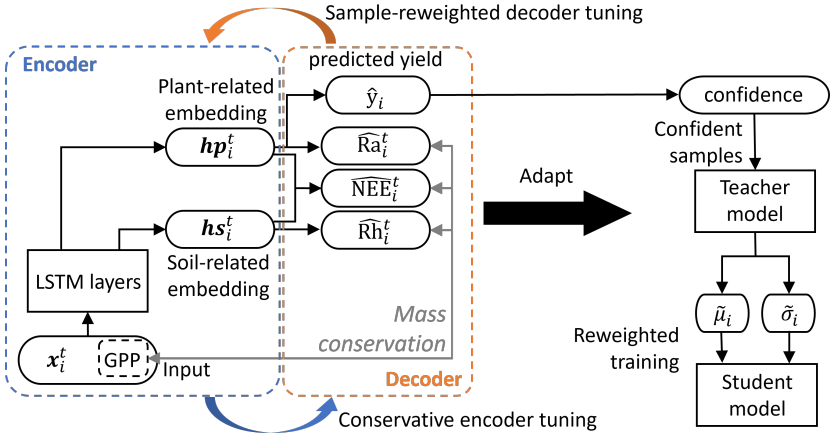


Figure 1: The overall flow of the proposed KGPFA method. The left side shows the KGNN model architecture. The encoder and decoder are trained alternately in fine-tuning. Given new testing data, a teacher-student learning framework (on the right side) is proposed to adapt the learned KGNN model to the unlabeled testing data based on the confidence estimates.

where α_i^t denotes the attention weight at time step t, and is normalized over all the time steps through a softmax function. The transformation functions $f(\cdot)$ and $g(\cdot)$ are implemented using fully connected layers. Then a supervised MSE loss is defined to measure the difference between predicted crop yield and simulated crop yield $\tilde{y}$, as $\mathcal{L}_{sup} = \sum_i (\hat{y}_i - \tilde{y}_i)^2 / |\mathcal{S}|$.

The complete pre-training loss combines the aforementioned loss functions, as follows

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{sim}} + \lambda \mathcal{L}_{\text{phy}}, \tag{4.4}$$

where λ is a hyper-parameter to control the weight of physical loss. In our implementation, we use normalized values for Ra, Rh, NEE, and yield in computing $\mathcal{L}_{\text{sup}}$ and $\mathcal{L}_{\text{sim}}$, and do not include additional hyper-parameters to balance their weights.

4.2 Alternate fine-tuning using real data.

Given limited observation data, We need to fine-tune the pre-trained model to fit the real environment and mitigate the bias in simulated data. However, prior studies found that directly fine-tuning the entire neural network model is likely to distort informative features learned through pre-training [16], and thus undermine the model generalizability. To overcome this issue, we decompose the KGNN model into the encoder and decoder (as shown in Fig. 1) and fine-tune them alternatively in different ways.

Encoder tuning aims to refine the physics-aware feature representation to mitigate the bias learned from simulated data. Since we do not have access to intermediate variables (e.g., Ra, Rh, NEE) in real observations, we will refine the encoder in a conservative way so that the learned feature representation is not distorted.

In particular, we propose to tune the encoder by reducing the model uncertainty on samples with low prediction errors. To quantify the uncertainty, we assume the observations follow a Gaussian distribution with the mean of $\hat{y}_i^t$ (i.e., the original predicted value) and the standard deviation of $\hat{\sigma}$. We create an additional network branch from the plant-related embedding $\boldsymbol{hp}$ to predict the standard deviation from the plant-related embedding. Before alternative training starts, This additional branch is first trained separately by minimizing the negative log-likelihood (NLL) on real training data. The NLL for each data sample i is defined as follows:

$$\mathcal{L}_{\text{nll}} = \frac{\log 2\pi(\hat{\sigma}_i^t)^2}{2} + \frac{(y_i^t - \hat{y}_i^t)^2}{2(\hat{\sigma}_i^t)^2}. \tag{4.5}$$

Here we omit the constants in the NLL. It is also noteworthy that we add the uncertainty modeling (i.e., the prediction of $\hat{\sigma}$) in fine-tuning with real data, but not in pre-training. This is because the real observations can be affected by minor environmental factors that are not considered in generating simulated data.

When tuning the encoder, we will optimize the encoder to reduce the uncertainty on a selected set of 'easy' samples $\mathcal{E} \subseteq \mathcal{R}$, which will be discussed later. Instead of directly minimizing their uncertainty (i.e., variance or standard deviation), we use the same NLL objective on these selected training samples in our tests because it also ensures that the predicted values remain close to the true observations (the last term in Eq. 4.5).

To conduct conservative tuning, we select only 'easy' samples, i.e., the samples that can be well predicted by the current model, i.e., $\mathcal{E} = \{(\boldsymbol{x}_i, y_i)\} \subseteq \mathcal{R}$ that satisfy $|\text{KGNN}(\boldsymbol{x}_i) - y_i| < \tau$, and τ is a performance threshold. To determine the threshold, we adopt

a statistical test, where the null hypothesis H_0 states that the prediction error $e_i = |\text{KGNN}(\boldsymbol{x}_i) - y_i|$ for all the training samples follow a single normal distribution while the alternative hypothesis H_1 states that there exists a subset of samples $U \subseteq \mathcal{R}$, and their prediction errors $\text{err}(\mathcal{U})$ follow a different normal distribution from the errors of the remaining samples U'. Here U can be either the 'easy' or 'hard' samples. The optimal set U can be obtained by solving $\text{argmax}_U \log \frac{\text{Likelihood } (H_1|U)}{\text{Likelihood } (H_0)}$.

According to our prior work [38], this can be solved by minimizing the sum of the variance of $\text{err}(U)$ and $\text{err}(U')$. Hence we can select the threshold γ that leads to the smallest value of $\text{Var}(\text{err}(U)) + \text{Var}(\text{err}(U'))$, where Var denotes variance.

Decoder tuning aims to modify how the physics-aware feature representations are transformed into the output variables while fixing the physics-aware feature representations. Before we start tuning the decoder, we reweight all the samples based on prediction errors, as $w_i = (e_i - \min\{e_i\})/(\max\{e_i\} - \min\{e_i\}) + 1$. The sample weights range in [1, 2], and the training samples with larger errors have higher weights. Then we tune the decoder using a weighted MSE loss, as follows:

$$\mathcal{L}_{\text{wmse}} = \frac{\sum_{i \in \mathcal{R}} w_i (y_i - \hat{y}_i)^2}{|\mathcal{R}|}. \tag{4.6}$$

Traditional unweighted training often compromises the performance on a subset of samples in exchange for better overall prediction. In the proposed approach, the sample weights are re-estimated every few epochs, which ensures that the decoder training improves the prediction over diverse training samples with worse performance. This approach also helps improve the model's generalizability.

4.3 Adaptation to the testing environment

We now introduce the adaptation method to transfer the learned model to the testing data. In contrast to the fine-tuning process, the adaptation only has access to the input features of testing samples but not their labels (i.e., crop yield y). We will leverage the uncertainty measures learned from the fine-tuning phase as proxies for prediction performance, and use them for selecting samples and guiding the adaptation process.

In particular, we propose to conduct model adaptation via a teacher-student learning framework. The idea is to construct a teacher model using only confident samples in the new environment and then use the teacher model to guide the original KGNN model, i.e., the student. Specifically, we select confidence samples based on an uncertainty threshold, which is determined based on the easy-to-hard sample partitioning in the previous fine-tuning phase. In particular, we consider the range of standard deviation σ for easy samples in the training set $\mathcal{R}$ as $[a,b]$ and the range of for the hard samples in $\mathcal{R}$ as $[c,d]$. For reasonable uncertainty measures, they are often related to prediction errors, and thus we have the relation $a < c < b < d$. We set the uncertainty (standard deviation) threshold as $(b + c)/2$, and then use it to identify confident samples, i.e., $\hat{\sigma}_i < (b + c)/2$.

We will use the selected confident samples to train the teacher model by minimizing the NLL loss (Eq. 4.5) by setting the labels y as the current model predictions. The intuition is to further reduce the uncertainty on the confident samples while maintaining the prediction values. After training the teacher model, we will apply the obtained teacher model to the testing data, which produces the estimated prediction mean and standard deviation for each test sample, which are represented by $\tilde{\mu}_i$ and $\tilde{\sigma}_i$, respectively.

Next, we will use the output of the teacher model (i.e., predicted mean values $\{\tilde{\mu}_i^t\}$) as labels to train the student model. As the teacher model is trained using confident samples in the testing data, it has a better chance at identifying samples that cannot be handled by the current model by increasing the modeled uncertainty. Hence, we will reweight different testing samples based on the standard deviation $\tilde{\sigma}_i$ estimated by the teacher model. For each sample i, its weight is equal to $\tilde{w}_i = \eta(1/\tilde{\sigma}_i)$, where $\eta(\cdot)$ is a normalization function over all the test samples. It can be seen that the testing samples with lower uncertainty have higher weights. The final loss for the student model is

$$\mathcal{L}_{\text{stu}} = \frac{\sum_{i \in \mathcal{T}} \tilde{w}_i (\hat{y}_i - \tilde{\mu}_i)^2}{|\mathcal{T}|} \tag{4.7}$$

In our tests, we also added the physical loss (Eq. 4.2) when training both teacher and student models. We also tested different tuning strategies, either tuning the entire model or only the decoder in the adaptation phase, and observed similar performance.

5 Experimental Results

5.1 Dataset We use the corn yield data in Illinois and Iowa from the years 2000-2020 provided by USDA National Agricultural Statistics Service (NASS) [1]. In particular, there are in total 199 counties in our study region (100 counties in Illinois and 99 counties in Iowa). The corn yield data (in gCm^{-2}) are available for each county each year. The input features have 19 dimensions, including NLDAS-2 climate data [39], 0-30cm gSSURGO soil properties [2], crop type information, the

[1] https://quickstats.nass.usda.gov/
[2] https://gdg.sc.egov.usda.gov/

250m Soil Adjusted Near-Infrared Reflectance of vegetation (SANIRv) based daily GPP product [40], and calendar year. Moreover, we use the physics-based Ecosys model [1] to simulate Ra, Rh, NEE, and crop yield for 10,335 synthetic sample locations in the United States from the years 2001-2018.

5.2 Evaluation details We conduct experiments to answer the following questions:

- *Q1.* Whether the proposed method can improve the predictive performance, especially when the training data are sparse?
- *Q2.* What is the effect of the alternate fine-tuning in enhancing the prediction?
- *Q3.* Can the proposed method improve the prediction under out-of-sample testing scenarios?
- *Q4.* Whether the knowledge-guided network model can preserve the mass conservation?

We implement the proposed model using GTX 3090 24GB GPU and AMD Ryzen 9 5950X 16-Core 3.40 GHz Processor with 64GB RAM. The training uses the ADAM optimizer [41] with an initial learning rate of 0.002. The LSTM outputs, $\boldsymbol{hp}$, and $\boldsymbol{hs}$ have 64 dimensions, the embeddings $\boldsymbol{hp}$ and $\boldsymbol{hs}$ are then transformed through a two-layer fully connected network (the first layer with 32 dimensions) to produce the output variables. The weight hyper-parameter for the physical loss is set to be 0.1.

We implement a diverse set of methods:

- LSTM-ATT uses only the LSTM to transform input features and then uses the attention layer to predict the yield [23].
- KGNN_{AG} is based on the proposed KGNN architecture, and uses simulated data to augment the training, i.e., having both simulated and real training data in the loss function with equal weights.
- KGNN_{FT} directly fine-tunes all the parameters in the KGNN model after being pre-trained on the simulated data.
- KGNN_{LP} fine-tunes only the decoder layers in KGNN, as inspired by previous work [16].
- KGPF fine-tunes KGNN using the proposed fine-tuning method (Section 4.2).
- KGPF_{DA} adapts the model to the target data using the adversarial domain adaptation approach [34].
- KGPF_{TT} adapts the model by test-time training on an additional task, as inspired by [35]. The additional task aims to predict GPP from other input features (by masking out GPP from input).
- KGPFA is the proposed three-stage method.

5.3 Predictive performance Table 1 summarizes the testing performance of different methods on the years 2018-2020. We also test each method using different numbers of years of real data for training.

Table 1: The prediction root mean squared errors (RMSE) by different approaches using the training data from the last 2, 5, or 18 years before 2018. The performance is measured on the next three years 2018-2020 as testing years.

Method	2 years	5 years	18 years
LSTM-ATT	64.129	50.030	44.719
KGNN_{AG}	58.309	46.553	39.053
KGNN_{FT}	58.308	46.551	38.905
KGNN_{LP}	55.130	44.862	38.528
KGPF	48.202	42.557	36.591
KGPF_{DA}	56.615	44.414	38.260
KGPF_{TT}	48.013	42.137	36.136
KGPFA	47.838	41.714	34.818

It can be seen that the proposed KGPFA outperforms other methods by a considerable margin. The improvement can also be seen from the spatial distribution of prediction errors, as shown in Fig. 2 (a)-(c). As we reduce the training data, all the methods have degraded performance. Nevertheless, KGPFA has led to substantial performance improvement compared to most baselines even with limited training data.

Besides, we have the following observations. First, KGNN_{AG} performs better than LSTM-ATT, which demonstrates the benefit of using the physics-aware model structure and the simulated data in the learning process. Second, KGNN_{LP} and KGPF perform better than KGNN_{AG}. This is because KGNN_{AG} directly uses the combination of simulated and real data in tuning the entire model but can be affected by the bias in simulated data. KGNN_{LP} does not perform as well as KGPF because it only updates the decoder while the extracted physical features may also need to be updated. Third, the adaptation approaches KGPF_{TT} and KGPFA improve the performance compared to KGPF. This shows the need for model adaptation. The domain adaptation method KGPF_{DA} does not perform well compared to other adaptation methods. This is because it focuses on reducing the distributional gap between training and testing input data but may not preserve the discriminative information about yield.

5.4 Analysis on model tuning We study the change of prediction performance in the fine-tuning process over training epochs, as shown in Fig. 3. It can be seen that the standard linear probing method over the KGNN model (KGNN_{LP}) also improves the performance when the fine-tuning starts, but after more epochs it cannot further improve the performance because it only modifies the transformation on the fixed feature representation. In contrast, the proposed

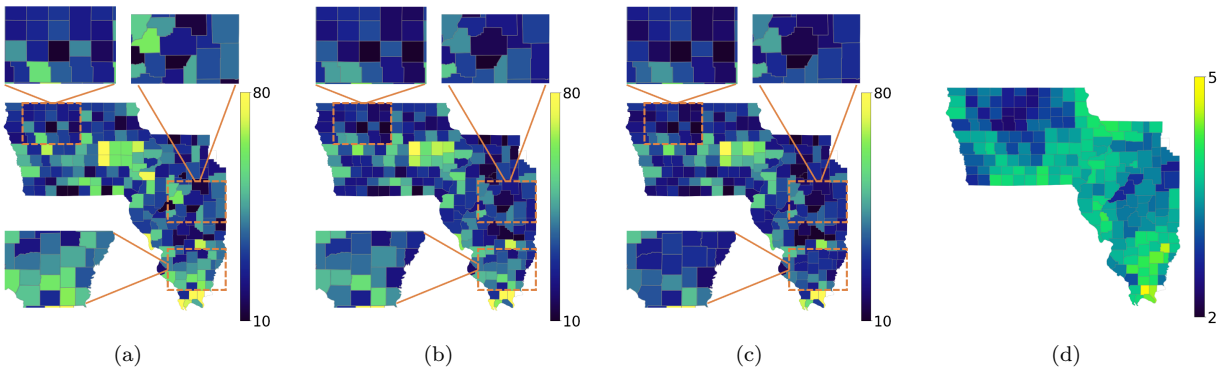


Figure 2: (a)-(c) The spatial distribution of prediction errors (RMSE) over different counties by (a) KGNN_{FT}, (b) KGPF, and (c) KGPFA. (d) The uncertainty value, i.e., the standard deviation, measured by the KGPF method. The results (a)-(d) in each county are averaged over three testing years.

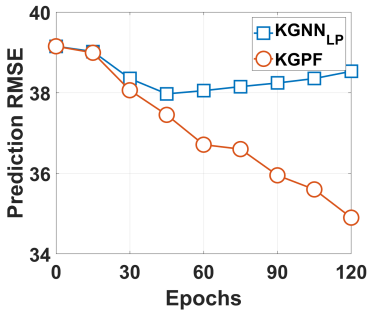


Figure 3: The change of performance over epochs for KGNN_{LP} and the proposed KGPF methods.

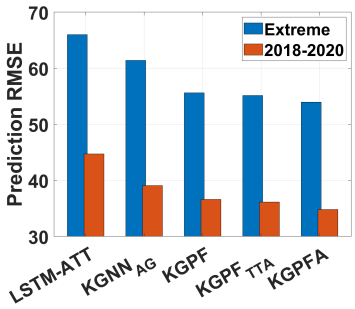


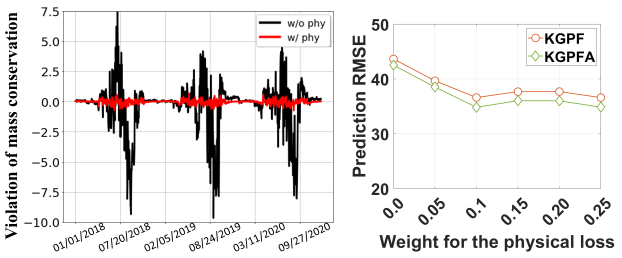
Figure 4: The testing performance of different approaches on years with extreme weathers (2002, 2003, 2020) vs. the years 2018, 2019, and 2020.

method alternately updates the encoder and decoder and leads to consistent improvement.

According to the method described in Section 4.3, the quantification of model uncertainty by the KGPF model (before the adaptation) is the key to the success of the adaptation process. We illustrate the estimated uncertainty (standard deviation) of the KGPF model in Fig. 2 (d). It can be observed that the uncertainty is in general consistent with the distribution of prediction errors presented in Fig. 2 (b). Therefore, the uncertainty measures can be used as proxies to pick testing samples that can be well predicted by the current model.

5.5 Out-of-sample performance To test the capacity of the proposed method under out-of-sample scenarios, we conduct another experiment by testing the model on the three years with extreme weather conditions (2002, 2003, and 2012) and training on the remaining 18 years. In Fig. 4, we show the results of different methods when tested in the years with extreme weather and the last three years (2018-2020). We can observe that all the methods perform much worse when tested in extreme weather conditions as the testing data are in a different distribution. The results show that KGPF substantially improves the performance compared to LSTM-ATT and KGNN_{AG} in extreme weather conditions as the proposed fine-tuning approach to some extent preserves the physically meaningful features learned through pre-training. The proposed adaptation approach also slightly enhances the testing performance, i.e., RMSE 53.95 for KGPFA vs. 55.63 for KGPF.

5.6 Validation of mass conservation Next, we verify that the results produced by the proposed method are indeed consistent with known physical laws. Specifically, we validate the conservation of mass between GPP, NEE, Ra, and Rh in the testing years, as shown in Fig. 5 (a). We implement a model that predicts Ra, Rh, and NEE (with supervision in synthetic data) but does not consider their relationship. When applied to the testing period 2018-2020, this model significantly violates the mass conservation. In contrast, the proposed method leads to a much smaller degree of violation, i.e., $\text{GPP}_i^t - \widehat{\text{Ra}}_i^t - \widehat{\text{Rh}}_i^t + \widehat{\text{NEE}}_i^t$, over the entire testing period. Fig. 5 (b) also shows that the performance is stable when the weight of physical loss is above 0.1.



(a) (b)

Figure 5: (a) The violation of mass conservation law, i.e., $\mathrm{GPP}_i^t - \hat{\mathrm{Ra}}_i^t - \hat{\mathrm{Rh}}_i^t + \hat{\mathrm{NEE}}_i^t$, over time in the testing period. The value is averaged over all the counties in our study region. The two curves represent the complete version of KGPFA and the KGPFA without using the physical loss (Eq. 4.2). (b) The change of prediction performance by KGPF and KGPFA when using different weights for the physical conservation loss.

6 Conclusion

This paper proposes a new multi-stage learning method for extracting, preserving, and adapting physics-aware features. We first build a KGNN model for extracting physics-aware feature representation through pre-training from simulated data generated by physics-based models. Then we propose a fine-tuning approach that alternately updates the encoder and decoder in the KGNN model to enhance the prediction while preserving the learned physics-aware features. Next, we conduct unsupervised adaptation to transfer the model to the testing data using proxy labels from a conservative model as a teacher. The results demonstrate that the proposed method can substantially improve the prediction accuracy, even using sparse training data or under out-of-sample scenarios. Besides, we analyze the alternate training procedure in the proposed fine-tuning approach, and show that the uncertainty estimates obtained after fine-tuning can reveal the general patterns of model prediction errors. This ensures that truly confident samples can be selected in the adaptation process for training the teacher model. Additionally, we show that the proposed method indeed preserves the mass conservation.

The proposed method is generally applicable to many disciplines, (e.g., freshwater science, hydrology, climate science, and material science) in which physics-based models are being used for modeling interacting and evolving processes. Future directions include (i) exploring more advanced KGNN structures in representing complex physical systems (e.g., [29, 42]), and (ii) extending the uncertainty quantification approach to consider the uncertainty from different sources.

Acknowledgements

This work was supported by the NSF awards 2147195, 2239175, 2316305, 2105133, and 2126474, the NASA award 80NSSC22K1164, the Momentum award at the University of Pittsburgh, the DRI award at the University of Maryland, and the University of Pittsburgh Center for Research Computing.

References

[1] Wang Zhou et al. Quantifying carbon budget, crop yields and their responses to environmental variability using the ecosys model for us midwestern agroecosystems. *Agricultural and Forest Meteorology*, 2021.

[2] Jeffrey G Arnold et al. Swat: Model use, calibration, and validation. *Transactions of the ASABE*, 2012.

[3] Steven L Markstrom. *P2S–coupled Simulation with the Precipitation-Runoff Modeling System (PRMS) and the Stream Temperature Network (SNTemp) Models*. US Department of the Interior, USGS, 2012.

[4] Matthew R Hipsey 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*, 2019.

[5] Hoshin V Gupta et al. Debates—the future of hydrological sciences: A (common) path forward? using models and data to learn: A systems theoretic perspective on the future of hydrological science. *WRR*, 2014.

[6] Upmanu Lall. Debates—the future of hydrological sciences: A (common) path forward? one water. one world. many climes. many souls. *WRR*, 2014.

[7] Jeffrey J McDonnell and Keith Beven. Debates—the future of hydrological sciences: A (common) path forward? a call to action aimed at understanding velocities, celerities and residence time distributions of the headwater hydrograph. *WRR*, 2014.

[8] Joshua Fan, Junwen Bai, Zhiyun Li, Ariel Ortiz-Bobea, and Carla P Gomes. A gnn-rnn approach for harnessing geospatial and temporal information: application to crop yield prediction. In *AAAI*, 2022.

[9] Dapeng Feng, Kuai Fang, and Chaopeng Shen. Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *WRR*, 2020.

[10] Sijie He et al. Sub-seasonal climate forecasting via machine learning: Challenges, analysis, and advances. In *AAAI*, 2021.

[11] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 2022.

[12] Mohammad M Sultan et al. Transferable neural networks for enhanced sampling of protein dynamics. *Journal of Chemical Theory and Computation*, 2018.

[13] David Menéndez Hurtado, Karolis Uziela, and Arne Elofsson. Deep transfer learning in the assessment of the quality of protein models. *arXiv preprint arXiv:1804.06281*, 2018.

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

[15] Jordan S Read et al. Process-guided deep learning predictions of lake water temperature. *WRR*, 2019.

[16] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.

[17] Sachin Goyal et al. Finetune like you pretrain: Improved finetuning of zero-shot vision models. In *CVPR*, 2023.

[18] Junjiao Tian, Zecheng He, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, and Zsolt Kira. Trainable projected gradient method for robust fine-tuning. In *CVPR*, 2023.

[19] Anuj Karpatne et al. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[20] Dehao Liu and Yan Wang. Multi-fidelity physics-constrained neural network and its application in materials modeling. *Journal of Mechanical Design*, 2019.

[21] Paul C Hanson, Aviah B Stillman, Xiaowei Jia, Anuj Karpatne, Hilary A Dugan, Cayelan C Carey, Jemma Stachelek, Nicole K Ward, Yu Zhang, Jordan S Read, et al. Predicting lake surface water phosphorus dynamics using process-guided machine learning. *Ecological Modelling*, 430:109136, 2020.

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

[23] Erhu He, Yiqun Xie, Licheng Liu, Weiye Chen, Zhenong Jin, and Xiaowei Jia. Physics guided neural networks for time-aware fairness: an application in crop yield prediction. In *AAAI*, 2023.

[24] Tianfang Xu and Albert J Valocchi. Data-driven methods to improve baseflow prediction of a regional groundwater model. *Computers & geosciences*, 2015.

[25] Stephan Rasp and Nils Thuerey. Data-driven medium-range weather prediction with a resnet pretrained on climate simulations: A new model for weatherbench. *JAMES*, 2021.

[26] Herim Han et al. Transfer learning from simulation to experimental data: Nmr chemical shift predictions. *The Journal of Physical Chemistry Letters*, 2021.

[27] Felix Finkeldey et al. Learning quality characteristics for plastic injection molding processes using a combination of simulated and measured data. *Journal of Manufacturing Processes*, 2020.

[28] Xiaowei Jia et al. Physics guided rnns for modeling dynamical systems: A case study in simulating lake temperature profiles. In *SDM*, 2019.

[29] Licheng Liu, Shaoming Xu, Jinyun Tang, Kaiyu Guan, Timothy J Griffis, Matthew D Erickson, Alexander L Frie, Xiaowei Jia, Taegon Kim, et al. Kgml-ag: a modeling framework of knowledge-guided machine learning to simulate agroecosystems: a case study of estimating n2o emission using data from mesocosm experiments. *Geoscientific Model Development*, 2022.

[30] Shengyu Chen, Jacob A Zwart, and Xiaowei Jia. Physics-guided graph meta learning for predicting water temperature and streamflow in stream networks. In *SIGKDD*, 2022.

[31] Xiaowei Jia, Shengyu Chen, Can Zheng, Yiqun Xie, Zhe Jiang, and Nasrin Kalanat. Physics-guided graph diffusion network for combining heterogeneous simulated data: An application in predicting stream water temperature. In *SDM*. SIAM, 2023.

[32] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022.

[33] Yao Xiao, Ziyi Tang, Pengxu Wei, Cong Liu, and Liang Lin. Masked images are counterfactual samples for robust fine-tuning. In *CVPR*, 2023.

[34] Yaroslav Ganin et al. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016.

[35] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*. PMLR, 2020.

[36] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[37] Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *CVPR*, 2023.

[38] Yiqun Xie, Erhu He, Xiaowei Jia, Han Bao, Xun Zhou, Rahul Ghosh, and Praveen Ravirathinam. A statistically-guided deep network transformation and moderation framework for data with spatial heterogeneity. In *ICDM*. IEEE, 2021.

[39] Youlong Xia et al. Continental-scale water and energy flux analysis and validation for north american land data assimilation system project phase 2 (nldas-2): 2. validation of model-simulated streamflow. *Journal of Geophysical Research: Atmospheres*, 2012.

[40] Chongya Jiang et al. A daily, 250 m and real-time gross primary productivity product (2000–present) covering the contiguous united states. *Earth System Science Data*, 2021.

[41] Diederik P Kingma et al. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

[42] Dapeng Feng et al. Differentiable, learnable, regionalized process-based models with multiphysical outputs can approach state-of-the-art hydrologic prediction accuracy. *WRR*, 2022.