Scientifically-Interpretable Reasoning Network (ScIReN): Discovering Hidden Relationships in the Carbon Cycle and Beyond

Joshua Fan*[1], Haodi Xu*[2], Feng Tao*[3]†, Md Nasim[1], Marc Grimson[1], Yiqi Luo[2], Carla P. Gomes[1]

[1]Cornell University, Department of Computer Science
[2]Cornell University, School of Integrative Plant Science, Soil and Crop Sciences Section
[3]Cornell University, Department of Ecology and Evolutionary Biology
{jyf6,hx293,feng.tao,mn637,mg2422,yl2735}@cornell.edu, gomes@cs.cornell.edu

Abstract

Soils have potential to mitigate climate change by sequestering carbon from the atmosphere, but the soil carbon cycle remains poorly understood. Scientists have developed *process-based models* of the soil carbon cycle based on existing knowledge, but they contain numerous unknown parameters and often fit observations poorly. On the other hand, neural networks can learn patterns from data, but do not respect known scientific laws, and are too opaque to reveal novel scientific relationships. We thus propose *Scientifically-Interpretable Reasoning Network (ScIReN)*, a fully-transparent framework that combines interpretable neural and process-based reasoning. An interpretable encoder predicts *scientifically-meaningful latent parameters*, which are then passed through a differentiable process-based decoder to predict labeled output variables. While the process-based decoder enforces existing scientific knowledge, the encoder leverages Kolmogorov-Arnold networks (KANs) to reveal interpretable relationships between input features and latent parameters, using novel smoothness penalties to balance expressivity and simplicity. ScIReN also introduces a novel hard-sigmoid constraint layer to restrict latent parameters into prior ranges while maintaining interpretability. We apply ScIReN on two tasks: simulating the flow of organic carbon through soils, and modeling ecosystem respiration from plants. On both tasks, ScIReN outperforms or matches black-box models in predictive accuracy, while greatly improving scientific interpretability – it can infer latent scientific mechanisms and their relationships with input features.

Code — https://github.com/gomes-lab/ScIReN
Extended version — https://arxiv.org/pdf/2506.14054

1 Introduction

Climate change poses major challenges to humanity, such as sea level rise, damages to agricultural production, and increased natural disasters (Lee et al. 2023). Soils have the potential to mitigate climate change by sequestering carbon from the atmosphere (Lal, Negassa, and Lorenz 2015), as they store more carbon than plants and the atmosphere combined (Jobbágy and Jackson 2000). Yet it is difficult to understand how carbon flows through the soil, leading to high uncertainties in future climate projections (Luo et al. 2016).

Neural networks can predict the amount of organic carbon stored in the soil at each location (Tan et al. 2024). However, neural networks can produce predictions that violate existing scientific knowledge, such as mass conservation (Karpatne, Kannan, and Kumar 2022). Also, scientists want to *understand* the biogeochemical processes governing the soil carbon cycle – which control how much carbon can be stored in the soil and how long it will remain sequestered there. However, the black-box nature of neural networks makes it challenging to translate their predictive ability into novel scientific insights (Marcinkevičs and Vogt 2023).

To gain insight into these biogeochemical processes, ecologists use domain expertise to develop process-based models that simulate the soil carbon cycle (Luo and Smith 2022). These models are highly sophisticated, and may contain hundreds of pools representing different types of carbon, and matrices specifying flux rates between each pair of pools (Luo et al. 2022; Huang et al. 2018; Lu et al. 2020). However, these models contain many unknown parameters, which traditionally must be tuned by human experts through an inefficient trial-and-error process (Luo and Schuur 2020). These models often cannot fit observed data well, especially in cross-scale predictions, due to a poor understanding of the relationships between environmental conditions and parameter values. This is a key bottleneck in using process-based models for verifiable predictions on the global soil carbon cycle and its response to climate change.

A few prior works in hydrology (Tsai et al. 2021; Feng et al. 2022), phenology (van Bree, Marcos, and Athanasiadis 2025), ecosystem modeling (Reichstein et al. 2022), and soil biogeochemistry (Xu et al. 2025) have proposed implementing process-based models in a differentiable way so that poorly-understood processes can be optimized by backpropagation or replaced with neural networks. While these approaches integrate scientific knowledge with deep learning,

*These authors contributed equally.

†Now at Department of Informatics and Intelligent Systems, Institute of Energy and the Environment, The Pennsylvania State University.

the neural network components remain too opaque for scientists to understand, which leaves the relationships between the latent parameters and input features unclear.

To address these limitations, we propose *Scientifically-Interpretable Reasoning Network (ScIReN)*, an end-to-end differentiable framework that embeds scientific process-based models into a *fully-transparent* neural model, creating a system that respects scientific knowledge, learns from data in an end-to-end manner, and can discover new scientific relationships. ScIReN contains three main components. First, a learnable encoder takes in environmental features for a given location, and predicts scientifically-meaningful latent parameters. We use an *interpretable model* such as a sparse Kolmogorov-Arnold network (Liu et al. 2024c) to make this portion fully transparent, allowing scientists to understand the relationships between latent parameters and input features. Second, a hard-sigmoid constraint layer projects these parameters into a physically-plausible range set by prior knowledge. Finally, a process-based decoder uses these predicted parameters to simulate the flow of carbon through the soil based on scientific knowledge, and finally predicts the output variables (e.g. amount of organic carbon at each soil depth). We compare our predictions with ground-truth labels, and backpropagate to train the entire system.

Our main contributions are: (1) We propose ScIReN, a *fully-interpretable* framework for combining scientific reasoning (in the form of process-based models) with data-driven learning. Unlike prior work, our model can infer *unobserved physical processes* and *their relationships to input variables* in a fully-transparent way. **(2)** We balance smoothness and expressivity in the learned functional relationships by using B-splines with novel smoothness penalties. **(3)** We propose a novel hard-sigmoid constraint layer to constrain the scientific parameters to fall within physically-plausible ranges. **(4)** We validate ScIReN on two scientific domains. First, in ecosystem respiration, ScIReN discovers the correct relationships between latent parameters and input features, while other methods do not. It also improves out-of-distribution extrapolation compared to existing methods. Second, we test ScIReN on a more complex task of modeling the soil carbon cycle through 140 soil pools at each location. On a synthetic dataset, ScIReN is able to predict *unlabeled* biogeochemical parameters and accurately retrieve their relationships with environmental features. With real data, ScIReN simulates observed soil carbon amounts with accuracy comparable to black-box methods while being completely interpretable. We hope our work inspires other researchers to apply and extend ScIReN across diverse scientific tasks, advancing AI's capacity for interpretable scientific discovery.

2 Related Work

Knowledge-Guided Machine Learning. There is a rich history of incorporating prior knowledge into neural networks by modifying the loss function, pretraining procedure, or model architecture (Karpatne, Jia, and Kumar 2024; Willard et al. 2022). A common approach is to add a loss term that penalizes when physical laws are violated (Daw et al. 2022; Beucler et al. 2019; Jia et al. 2019). For example, the density of water is known to increase monotonically with depth; thus, Daw et al. (2022) and Jia et al. (2019) use a "monotonicity loss" to penalize violations of this. Physics-informed neural networks assume that the governing equation of a system is known, and penalize when the model predictions or gradients violate this equation (Raissi, Perdikaris, and Karniadakis 2019), but are hard to train (Krishnapriyan et al. 2021). Process-based models can generate synthetic data to pretrain the network (Jia et al. 2021; Liu et al. 2024a). While these approaches use prior knowledge to guide the model, they cannot guarantee that physical constraints will be satisfied (Willard et al. 2022), and cannot easily provide new insights into physical processes.

One can also design model architectures to encode prior knowledge. Convolutional neural networks encode inductive biases such as translation equivariance and locality into the model architecture (LeCun, Bengio et al. 1995). In lake temperature modeling, Daw et al. (2020) design an LSTM variant that produces monotonically-increasing intermediate variables by design. In agriculture, Liu et al. (2024a) design a hierarchical neural network that incorporates causal relations between different variables. However, it is difficult to design a new architecture for every problem.

Combining reasoning and learning. A few works reason about constraints and prior knowledge within the network itself. Deep Reasoning Networks use entropy-based losses to encourage the latent space to be interpretable and satisfy constraints (e.g. in Sudoku, each row must contain exactly one of each number) (Chen et al. 2020). This approach was used to solve the phase-mapping problem in materials discovery, where the constraints are thermodynamic rules (Chen et al. 2021). CLR-DRNets enhanced the reasoning process using a modified LSTM, and used curriculum learning to improve trainability (Bai, Chen, and Gomes 2021). Physically-informed Graph-based DRNets add a physical decoder that reconstructs X-ray diffraction patterns based on Bragg's law (Min et al. 2023). CS-SUNet encourages pixels with similar input features to have similar predictions, enabling weakly-supervised prediction (Fan et al. 2022). While these models have an interpretable latent space, the bulk of the network remains uninterpretable.

Process-based models. Scientists develop *process-based models* to simulate physical processes based on domain knowledge (Cuddington et al. 2013). These models consist of mathematical equations that describe relationships between various variables. In soil science, *pool-and-flux* models are common, where a matrix equation tracks the amount of carbon at each soil depth and matter type (Luo and Smith 2022). Transition matrices encode the rate at which carbon is transferred between pools, which are functions of soil and climate properties. Many scientific models can be unified under this matrix form (Huang et al. 2018; Luo et al. 2022).

Unfortunately, despite their sophistication, these models have difficulty matching real observations, and have numerous unknown parameters that are traditionally set in an ad-hoc way. These unknown parameters need to vary across space and (sometimes) time, but are difficult to estimate

(Luo and Schuur 2020). A state-of-the-art approach for setting these parameters is PRODA (Tao et al. 2020). PRODA first runs Bayesian data assimilation at each location separately to find optimal biogeochemical parameters for each location. Then, a neural network is trained to predict these optimal parameters given environmental covariates. While this approach is effective, it is computationally expensive, and is not always robust since each location's parameters are estimated with only a few observations.

Differentiable Process-Based Models. A few works integrate process-based models and neural networks in an end-to-end differentiable framework; this has been called *differentiable parameter learning* (Tsai et al. 2021), *hybrid modeling* (Reichstein et al. 2022), or *differentiable process-based modeling* (Shen et al. 2023). For example, (van Bree, Marcos, and Athanasiadis 2025; Reichstein et al. 2022; Tsai et al. 2021) used a process-based model as the main backbone, but replaced some poorly-understood components with neural networks. By implementing the process-based model in a differentiable way, the model could be trained end-to-end, and unknown components could be fit using data. Xu et al. (2025) scaled this approach up to a more complex soil carbon model with 21 unknown parameters and 140 carbon pools. However, the neural network component is still opaque, making it difficult for scientists to discover new relationships and insights.

Interpretable ML. A subfield of machine learning aims to interpret how neural networks make predictions (Molnar 2025). *Post-hoc feature attribution methods* such as SHAP (Lundberg and Lee 2017) or Integrated Gradients (Sundararajan, Taly, and Yan 2017) estimate the impact of each feature on the model's prediction for a given example. *Local surrogate models* such as LIME (Ribeiro, Singh, and Guestrin 2016) fit an interpretable model that approximates the black-box model in a small region. *Marginal effect plots* such as Partial Dependence Plots (Friedman 2001) or Accumulated Local Effects plots (Apley and Zhu 2020) plot how each feature affects the output on average. These methods are mere approximations of a black-box model, and can produce misleading explanations (Rudin 2019).

On the other hand, *inherently-interpretable models* make the entire model transparent by design (Rudin 2019). In linear regression, coefficients precisely reveal how each input affects the prediction. Neural additive models (Agarwal et al. 2021) and Kolmogorov Arnold Networks (Liu et al. 2024c) provide greater expressivity while maintaining interpretability; we discuss these in the next section. These methods are typically applied in supervised settings; to the best of our knowledge, our work is the first to combine them with scientific knowledge to predict *unlabeled* variables.

3 Methods

To combine scientific knowledge and data-driven learning into a fully-transparent model, ScIReN contains three main components. First, a learned encoder f_{NN} (with learnable weights θ) takes in input features $\mathbf{x} \in \mathbb{R}^D$ (e.g. soil/climate variables at a given location), and outputs unconstrained latent parameters $\tilde{\mathbf{p}} \in \mathbb{R}^P$: $\tilde{\mathbf{p}} = f_{NN}(\mathbf{x}; \theta)$. In ScIReN, f_{NN} should be a fully-transparent model, such as a neural additive model or sparse Kolmogorov-Arnold network. The latent parameters are scientifically-meaningful variables that govern underlying physical processes, yet cannot be observed directly. Secondly, a constraint layer projects the unconstrained parameters into scientifically-plausible ranges given by prior knowledge: $\mathbf{p} = \text{Proj}(\tilde{\mathbf{p}})$. Finally, the constrained parameters are passed through a *fixed, deterministic* process-based decoder g_{PBM}, which simulates physical processes and predicts output variables: $\hat{y} = g_{PBM}(\mathbf{p})$. We compare this with the true label and backpropagate. The framework is summarized in Figure 1, and we elaborate on each component below.

3.1 Encoder: Learned Interpretable Relationships

We want to learn a function f_{NN} mapping observed input features $\mathbf{x}$ to latent scientific parameters $\mathbf{p}$. In prior work, f_{NN} is typically a fully-connected neural network (Xu et al. 2025; van Bree, Marcos, and Athanasiadis 2025; Reichstein et al. 2022; Tsai et al. 2021). However, scientists want to understand how biogeochemical parameters (e.g. transfer rates between pools) depend on input features (e.g. temperature); neural networks do not provide these insights.

Recently, a line of work has aimed to produce neural networks that are inherently interpretable while being expressive. Neural additive models (NAM) (Agarwal et al. 2021) model the output as the sum of single-variable functions of each input feature. Specifically, they learn a neural network $\phi_j : \mathbb{R} \to \mathbb{R}$ (with one input and one output) for each feature x_j, and sum contributions from each feature into the output:

$$NAM(\mathbf{x}) = b + \sum_{j=1}^{D} \phi_j(x_j; \theta_j) \tag{1}$$

where $(b, \{\theta_j\}_{j=1}^D)$ are learnable parameters trained via backpropagation. While NAM is quite expressive, it cannot model non-additive feature interactions, which are important in soil science (Dieleman et al. 2012).

To increase the expressivity of neural additive models, we can generate intermediate variables using neural additive model of the inputs, then further use a neural additive model on the intermediate variables to generate the output. Specifically, generate intermediate variables $\mathbf{z} = \{z_1, \ldots, z_H\}$ as

$$z_h = NAM_h(\mathbf{x}) = b_h + \sum_{j=1}^{D} \phi_{h,j}(x_j), \quad \forall h \in [1, H] \tag{2}$$

Now define each output variable $\tilde{p}_i$ as a neural additive model over the intermediate variables

$$\tilde{p}_i = NAM_i(\mathbf{z}) = b_i + \sum_{h=1}^{H} \Phi_{i,h}(z_h) \tag{3}$$

$$= b_i' + \sum_{h=1}^{H} \Phi_{i,h}\left(\sum_{j=1}^{D} \phi_{h,j}(x_j)\right) \tag{4}$$

where all the bias terms are collected into b_i'. This is now the same form as a two-layer Kolmogorov-Arnold network

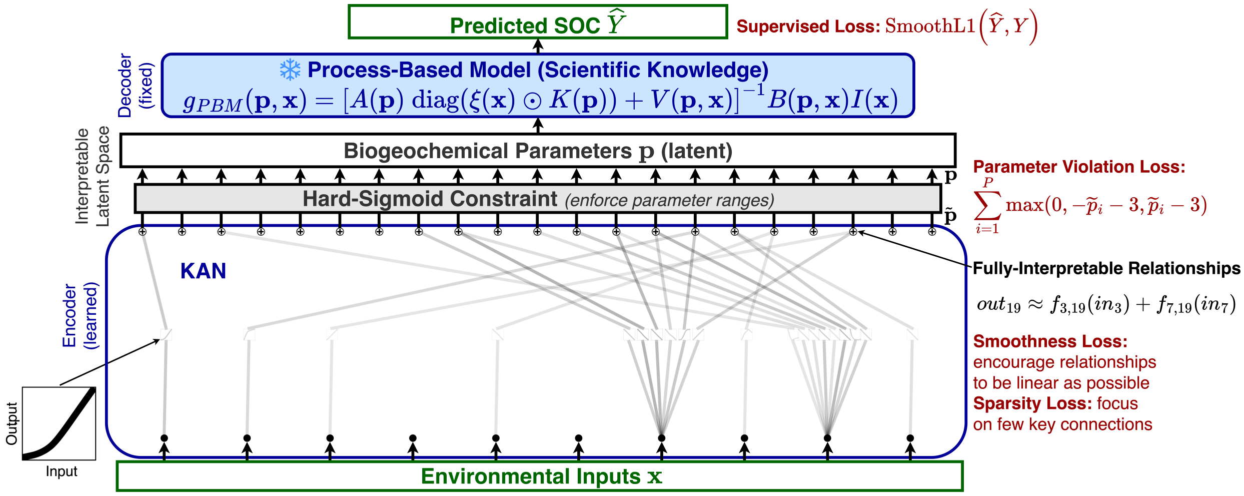
Predicted SOC $\widehat{Y}$
Supervised Loss: $\mathrm{SmoothL1}\left(\widehat{Y}, Y\right)$
Decoder (fixed)
Process-Based Model (Scientific Knowledge)
$g_{PBM}(\mathbf{p},\mathbf{x}) = [A(\mathbf{p})\ \mathrm{diag}(\xi(\mathbf{x}) \odot K(\mathbf{p})) + V(\mathbf{p},\mathbf{x})]^{-1} B(\mathbf{p},\mathbf{x}) I(\mathbf{x})$
Interpretable Latent Space
Biogeochemical Parameters $\mathbf{p}$ (latent)
$\mathbf{p}$
Hard-Sigmoid Constraint (enforce parameter ranges)
$\tilde{\mathbf{p}}$
Parameter Violation Loss:
$\sum_{i=1}^{P} \max(0, -\tilde{p}_i - 3, \tilde{p}_i - 3)$
KAN
Fully-Interpretable Relationships
$out_{19} \approx f_{3,19}(in_3) + f_{7,19}(in_7)$
Encoder (learned)
Smoothness Loss: encourage relationships to be linear as possible
Sparsity Loss: focus on few key connections
Output
Input
Environmental Inputs $\mathbf{x}$

Figure 1: Overview of ScIReN. The encoder reveals interpretable functional relationships between environmental inputs (e.g. temperature) and latent scientific parameters (e.g. transfer rates between soil pools). A constraint layer forces latent parameters into a prior range, and the process-based decoder simulates the physical process with the given latent parameters.

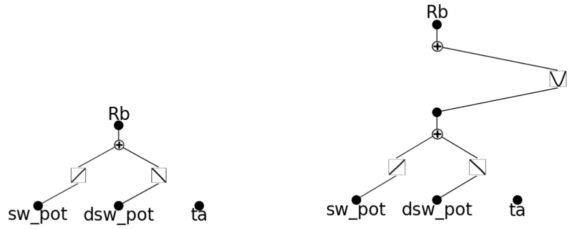

Figure 2: Learned encoder examples: 1-layer KAN (left) and 2-layer KAN (right)

(KAN, see equation 2.1 in (Liu et al. 2024c)). By the Kolmogorov-Arnold theorem, this multi-layer stack of neural additive models can approximate any multivariate continuous function (Liu et al. 2024c).

Two examples of learned encoders are shown in Figure 2. On the left, a 1-layer KAN models the latent parameter $Rb \approx w_1 \cdot sw\_pot + (-w_2) \cdot dsw\_pot$. One can immediately see how each feature influences Rb. On the right, a 2-layer KAN is needed to capture this nonlinear relationship, where an intermediate variable $Rb' \approx w_1 \cdot sw\_pot + (-w_2) \cdot dsw\_pot$, and then $Rb \approx |Rb'|$.

3.2 Sparsity and Smoothness Regularization

Kolmogorov-Arnold networks are still hard to interpret if each output depends on many inputs. Liu et al. (2024b) propose entropy regularization to sparsify the network. Specifically, we compute an importance score for each edge, as the mean absolute deviation of the output activations from the edge, weighted by their eventual contribution to the final output variables (Liu et al. 2024b); see Appendix for details. We denote this score for edge $(i \to j)$ in layer l as $B^l_{i,j}$. We encourage the entropy of the edge importance distribution to be low (making the network choose a few important edges and push others towards zero). We also use a L1 penalty to further promote sparsity by shrinking the magnitudes of each edge's outputs towards zero:

$$e^l_{i,j} = \frac{B^l_{i,j}}{\sum_{i',j'} B^l_{i',j'}} \quad \text{(make edge importances sum to 1)} \tag{5}$$

$$\mathcal{L}_{entropy} = -\sum_l \sum_{i,j} e^l_{i,j} \log e^l_{i,j};\ \mathcal{L}_{L1} = -\sum_l \sum_{i,j} |B^l_{i,j}| \tag{6}$$

The model can be viewed as using gradient descent to reason over possible scientific relationships, ultimately choosing a sparse set of relationships that fits the data.

Note that KANs parameterize the learnable edge activation functions Φ, ϕ using B-splines instead of neural networks. B-splines represent a curve as a weighted sum of basis functions, each of which peaks at a different point on the x-axis; see (Eilers and Marx 1996) for details. We can apply a second-order difference penalty on the spline coefficients – this encourages the coefficients to change in a linear way, making the function more linear (Eilers and Marx 2010). To our knowledge this has not been proposed with KANs before; this allows us to increase the number of basis functions (knots) and the function's expressivity while maintaining smoothness and preventing overfitting. If $c_1 \dots c_G$ are B-spline coefficients, the penalty is

$$\mathcal{L}_{smooth} = \sum_{i=1}^{G-2} ((c_{i+2} - c_{i+1}) - (c_{i+1} - c_i))^2 \tag{7}$$

3.3 Linear Parameter Constraint Layer

For some latent scientific parameters p_i, there is a known prior range $[p_i^{min}, p_i^{max}]$ based on prior knowledge and physical plausibility. Xu et al. (2025) applied a sigmoid function to the encoder output $\tilde{p}_i$ to force the predicted parameter into the prior range: $p_i = \sigma(\tilde{p}_i)$. However, this adds

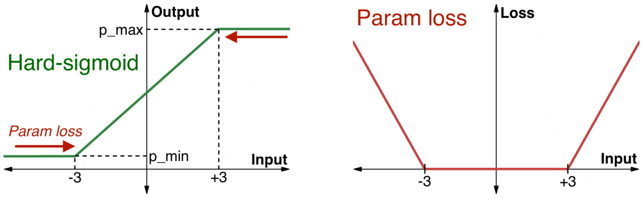

Figure 3: **Left:** The hard-sigmoid function constrains parameters to $[p_{min}, p_{max}]$, without adding nonlinearity. **Right:** parameter violation loss pushes the hard-sigmoid input away from flat regions.

nonlinearity and harms interpretability. For example, suppose a parameter p_i is actually a linear function of input variable x_j, $p_i = wx_j$. If the parameter was constrained using a sigmoid, the unconstrained encoder would have to learn $\tilde{p}_i = \sigma^{-1}(wx_j)$, so that after the sigmoid function the parameter becomes $p_i = \sigma(\sigma^{-1}(wx_j)) = wx_j$. The additional inverse sigmoid makes the function less interpretable.

Instead, we use a piecewise linear *hard-sigmoid* function (Figure 3 left) to constrain the parameters:

$$p_i = \begin{cases} p_i^{min} & \text{if } \tilde{p}_i \le -3, \\ p_i^{max} & \text{if } \tilde{p}_i \ge +3, \\ \dfrac{p_i^{max} - p_i^{min}}{6} \cdot \tilde{p}_i + \dfrac{p_i^{max} + p_i^{min}}{2} & \text{otherwise} \end{cases} \tag{8}$$

While the gradient $\frac{\partial p_i}{\partial \tilde{p}_i}$ is zero when $\tilde{p}_i$ is outside the range $[-3, 3]$, we can add "parameter violation loss" that places a penalty when $\tilde{p}_i$ is in the flat area (figure 3 right).

$$\mathcal{L}_{param} = \sum_{i=1}^{P} \max(0, -\tilde{p}_i - 3, \tilde{p}_i - 3) \tag{9}$$

This provides a gradient that pushes $\tilde{p}_i$ towards the linear range $[-3, 3]$ when it is in the flat range.

For the ecosystem respiration model, we only know that the latent parameter is nonnegative, so we use ReLU to impose the constraint, and a flipped ReLU loss to push inputs out of the flat region.

$$p_i = \max(\tilde{p}_i, 0); \quad \mathcal{L}_{param} = \max(-\tilde{p}_i, 0) \tag{10}$$

3.4 Differentiable Process-Based Decoder

ScIReN uses a process-based model that expresses output variables as a *fixed, differentiable function* of scientific parameters and input variables: $\hat{Y} = g_{PBM}(\mathbf{p}, \mathbf{x})$. Two examples are described below.

Ecosystem respiration. Consider the model of ecosystem respiration in (Reichstein et al. 2022). Based on scientific knowledge, we write the output variable R_{eco} (ecosystem respiration) as a differentiable function of two latent parameters, base respiration R_b and temperature sensitivity Q_{10}:

$$R_{eco} = g_{PBM}(\mathbf{p}, \mathbf{x}) = R_b(\mathbf{x}) \cdot Q_{10}^{\frac{t_a - T_{ref}}{10}} \tag{11}$$

where the latent parameters are $\mathbf{p} = \{R_b, Q_{10}\}$, the input features are $\mathbf{x} = \{sw\_pot, dsw\_pot, t_a\}$, and $T_{ref} = 15$. R_b has an unknown relationship with the input features (learned in the encoder), while Q_{10} is a learnable constant.

Soil carbon modeling. As a more complex process, we use the soil organic carbon module from Community Land Model 5 (CLM5) (Lu et al. 2020), which tracks the amount of soil organic carbon (SOC) in 140 pools in the soil (20 depths and 7 material types per depth). Denote the amount of carbon in pool i as Y_i. The core of the model is a mass conservation equation for each pool, where the change in carbon equals inflow (from plants and other pools) minus outflow (to other pools or the atmosphere):

$$\frac{dY_i(t)}{dt} = \text{inflow to pool } i - \text{outflow from pool } i \tag{12}$$

The equations for each pool can be combined into a single matrix equation. If we assume steady state ($\frac{dY_i(t)}{dt} = 0$), we can write Y as a function of biogeochemical parameters $\mathbf{p}$ and input features $\mathbf{x}$:

$$\hat{Y}(\mathbf{p}, \mathbf{x}) = \left[A(\mathbf{p})\,(\xi(\mathbf{p}, \mathbf{x}) \odot K(\mathbf{p})) + V(\mathbf{p}, \mathbf{x})\right]^{-1} I(\mathbf{p}, \mathbf{x}) \tag{13}$$

The details of this equation are explained in the Appendix. For now, it is sufficient to note that the process-based model takes in 21 latent biogeochemical parameters $\mathbf{p}$, uses the parameters to construct matrices describing carbon fluxes and decomposition, and finally predicts the amount of carbon in 140 pools (20 layers and 7 pools), $\hat{Y}$. Each operation (including matrix inversion) is differentiable and can be implemented in PyTorch. Note that our labeled data only contains aggregate SOC amounts at specific depths (which may not match the 20 fixed layers). Thus, we sum up the SOC pools at each layer, and linearly interpolate to predict SOC at the observed depths.

3.5 Final Loss

The final loss contains a smooth L1 loss between the predicted and ground-truth output variables, as well as the parameter violation and KAN regularization losses:

$$\mathcal{L} = \sum_{i=1}^{N} \left[SmoothL1(\hat{Y}_i, Y_i)\right] + \lambda_{param}\mathcal{L}_{param} + \lambda_{L1}\mathcal{L}_{L1} + \lambda_{entropy}\mathcal{L}_{entropy} + \lambda_{smooth}\mathcal{L}_{smooth}$$

Since the entire network is differentiable, we can backpropagate the loss through the process-based model to optimize the latent parameters and the learnable weights of the neural network. The loss weights λ are tuned on a validation set for each domain; they are intuitive to tune since we can visualize whether the KAN is too sparse/dense and whether the functions are too jagged/smooth. The Appendix examines how sensitive ScIReN is to hyperparameter choices.

4 Experiments

We test our approach on two domains: the model of ecosystem respiration in (Reichstein et al. 2022), and the CLM5 soil carbon cycle model (Xu et al. 2025).

4.1 Evaluation Metrics and Baselines

For each dataset, we first verify that methods are able to predict observed variables, by measuring R^2 on the test set.

However, the main novelty of ScIReN is its ability to infer hidden functional relationships between input features and latent parameters. We thus conduct experiments on synthetic data to verify that ScIReN infers the correct latent parameters and functional relationships. To evaluate functional relationship quality, for both ground-truth (synthetic) relationships and our learned models, we first compute the fraction of variance in the output that is explained by each input feature. For 1-layer KAN, we can simply compute the variance of each edge's post-activation outputs, and divide by the total variance in the output. For other models, this is non-trivial; we use Partial Dependence Variance (Greenwell, Boehmke, and McCarthy 2018) to estimate this. We then use KL divergence to measure how far the model's learned feature importance distributions are from the ground-truth.

For baselines, we compare against a pure neural network that only predicts observed variables (and cannot infer latent variables), and a blackbox-hybrid model (Reichstein et al. 2022; Xu et al. 2025) where a neural network (or linear model) predicts latent parameters, which are passed through the process-based model. We run ScIReN and Blackbox-Hybrid using a nonlinear constraint (sigmoid or softplus) and our proposed linear constraint (hard-sigmoid or ReLU).

4.2 Ecosystem Respiration

For ecosystem respiration, we used the same dataset and splits as (Reichstein et al. 2022), except we removed the 20% highest-temperature examples from the train set, forcing the model to extrapolate to higher temperatures than seen during training. We created two sets of latent R_b (base respiration) values. First, we model R_b as a linear function of 2 features sw_pot, dsw_pot (as in Reichstein et al. (2022)):

$$R_b = 0.0075 \cdot sw\_pot - 0.00375 \cdot dsw\_pot + 1.03506858$$

Second, to create a setting where a 2-layer KAN is needed, we add an absolute value.

$$R_b' = 0.0075 \cdot sw\_pot - 0.00375 \cdot dsw\_pot \tag{14}$$

$$R_b = \left| \frac{R_b' - mean(R_b')}{stdev(R_b')} \right| + 0.1 \tag{15}$$

We then generate the observed variable R_{eco} using the process-based model (Eq. 11) with multiplicative noise:

$$R_{eco} = R_b \cdot Q_{10}^{\frac{t_a - T_{ref}}{10}} \cdot (1 + \epsilon), \quad \epsilon \sim N(0, 0.1), \tag{16}$$

Table 1 shows results for the first setting (linear R_b). For predicting the observed variable R_{eco}, Blackbox-Hybrid and ScIReN outperform Pure-NN as the process-based model provides prior knowledge that helps the model extrapolate out-of-distribution. For inferring the latent variable R_b and functional relationships, ScIReN with linear constraint does best; it correctly learns that R_b only depends on sw_pot and dsw_pot, not t_a (see Figure 2 left). This is difficult because the irrelevant feature t_a (air temperature) is highly correlated with feature sw_pot. Also, if the model learns the wrong Q_{10} value, it can make R_b depend on t_a to compensate. ScIReN's entropy loss pushes it to eliminate as many variables as possible, and the smoothness loss (with the linear constraint) makes the relationship as linear as possible. Other methods learn complex relationships that perform worse.

Table 2 shows results for nonlinear R_b, where a 2-layer KAN is needed to model the complex relationship (1 layer is insufficient). ScIReN predicts the observed variable, latent variable, and functional relationships almost perfectly, greatly outperforming Pure-NN and Blackbox-Hybrid. Figure 2 (right) shows that ScIReN learned the true relationship.

4.3 Soil Carbon Cycle

For soil carbon, we split the US into 2×2 degree blocks and randomly assign the blocks to five folds, as in (Wang et al. 2020). We average across five data splits – each split uses one fold for testing, one for validation, and the other folds for training. Each split also uses its own initial seed.

First, we synthetically generate functional relationships between the 10 input features and 4 most sensitive biogeochemical parameters from (Xu et al. 2025). Of the $10 \times 4 = 40$ possible relationships, we select 20% and randomly assign each to (linear, quadratic, log, exp, abs) with random affine shifts. We set the other parameters to default values as they are poorly constrained by data (Xu et al. 2025); this *equifinality* is an inherent limitation of process-based models, which we mitigate by only predicting the 4 most sensitive parameters. We then use the CLM5 process-based model to generate synthetic SOC labels from these parameters. Table 3 shows how various methods perform in recovering these functional relationships. ScIReN (1-layer KAN) recovers the ground-truth relationships (see Figure 4) and observed/latent variables almost perfectly, while Blackbox-Hybrid is not incentivized to produce sparse relationships and mixes correlated features in.

Finally, we train all methods on real carbon labels in Table 4. ScIReN's accuracy in predicting SOC amounts is comparable to Blackbox-Hybrid and outperforms pure neural networks (in terms of R^2). This indicates that we can obtain full interpretability without significantly sacrificing predictive accuracy. Note that a 1-layer KAN is enough to achieve good accuracy, making the encoder even easier to interpret. Qualitatively, the predicted relationships are consistent with ecological knowledge; see Appendix for details.

5 Conclusion

We have proposed ScIReN, an end-to-end framework that combines data-driven learning with established scientific knowledge (in the form of process-based reasoning) to discover interpretable relationships between latent scientific parameters and inputs. The entropy loss helps the model discover robust relationships and disregard irrelevant variables, while the smoothness loss enables expressivity without overfitting. A limitation of ScIReN is that the KAN encoder is somewhat sensitive to initialization and hyperparameters; future work could pursue making KANs easier to train. Nevertheless, ScIReN demonstrates excellent performance at retrieving true functional relationships, can extrapolate out-of-distribution, and contains useful biases towards simplicity. Most importantly, ScIReN is fully transparent – it respects existing scientific knowledge while generating novel insights, thus facilitating data-driven scientific discovery.

Method	R^2 (observed, ↑)	R^2 (latent, ↑)	KL, functional relationships (↓)
Pure-NN	0.968 ± 0.004	N/A	N/A
Blackbox-Hybrid, nonlinear constraint	0.972 ± 0.003	0.818 ± 0.052	0.284 ± 0.041
Blackbox-Hybrid, linear constraint	**0.976** ± 0.001	0.941 ± 0.031	0.161 ± 0.068
ScIReN, nonlinear constraint (1-layer KAN)	0.975 ± 0.000	0.995 ± 0.000	0.002 ± 0.001
ScIReN, linear constraint (1-layer KAN)	**0.976** ± 0.000	**1.000** ± 0.000	**0.001** ± 0.001

Table 1: Ecosystem respiration, linear R_b. Mean and standard deviation across 5 seeds.

Method	R^2 (observed, ↑)	R^2 (latent, ↑)	KL, functional relationships (↓)
Pure-NN	0.946 ± 0.004	N/A	N/A
Blackbox-Hybrid, nonlinear constraint	0.948 ± 0.008	-0.265 ± 0.355	0.937 ± 0.168
Blackbox-Hybrid, linear constraint	**0.961** ± 0.001	0.949 ± 0.059	0.199 ± 0.094
Linear-Hybrid, linear constraint	0.400 ± 0.002	-0.204 ± 0.080	1.078 ± 0.115
ScIReN, linear constraint (1-layer KAN)	0.651 ± 0.014	-1.225 ± 0.668	1.182 ± 0.085
ScIReN, linear constraint (2-layer KAN)	**0.961** ± 0.001	**0.993** ± 0.005	**0.078** ± 0.045

Table 2: Ecosystem respiration, nonlinear R_b. Mean and standard deviation across 5 seeds.

Method	R^2 (observed, ↑)	R^2 (latent, ↑)	KL, functional relationships (↓)
Pure-NN	0.933 ± 0.015	N/A	N/A
Blackbox-Hybrid, nonlinear constraint	0.996 ± 0.003	0.226 ± 0.800	1.312 ± 0.170
Blackbox-Hybrid, linear constraint	0.995 ± 0.003	0.721 ± 0.226	1.082 ± 0.258
Linear-Hybrid, linear constraint	0.979 ± 0.013	0.087 ± 1.014	1.727 ± 0.322
ScIReN, linear constraint (1-layer KAN)	**0.999** ± 0.002	**0.989** ± 0.019	**0.080** ± 0.042

Table 3: Soil carbon cycle (synthetic parameters). Mean and standard deviation across 5 splits/seeds.

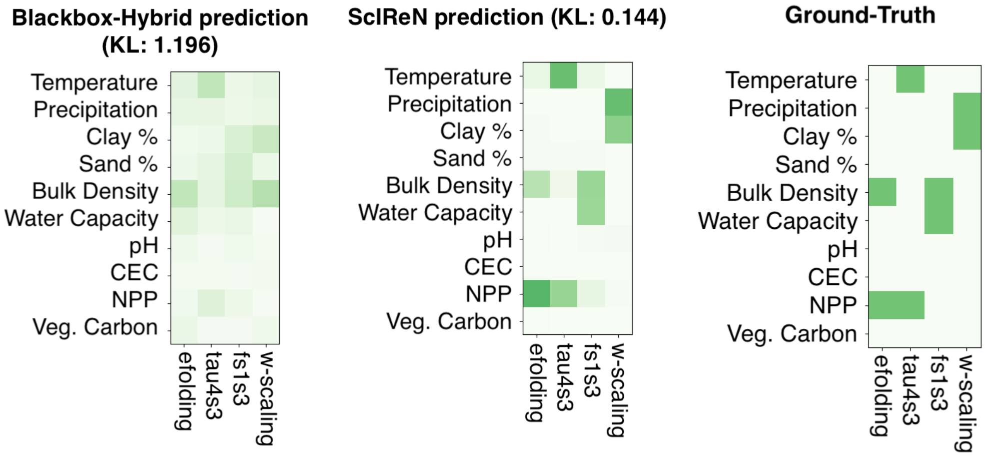

Figure 4: Functional relationships learned by Blackbox-Hybrid (left) and ScIReN (center) vs. truth (right), on synthetic labels. ScIReN recovers the true relationships much more accurately.

Method	R^2(↑)	MAE (↓)	Pearson correlation (↑)
Pure-NN	0.552 ± 0.173	**4609.3** ± 356.8	**0.780** ± 0.053
Blackbox-Hybrid, nonlinear constraint	0.584 ± 0.082	4726.2 ± 727.3	0.776 ± 0.048
Blackbox-Hybrid, linear constraint	**0.589 ± 0.070**	4849.7 ± 650.3	0.774 ± 0.040
Linear-Hybrid, linear constraint	0.552 ± 0.082	4984.8 ± 771.6	0.761 ± 0.046
ScIReN, linear constraint (1-layer KAN)	0.582 ± 0.080	4708.2 ± 673.1	0.769 ± 0.049
ScIReN, linear constraint (2-layer KAN)	0.571 ± 0.094	4707.3 ± 826.3	0.765 ± 0.052

Table 4: Soil carbon cycle (real labels). Mean and standard deviation across 5 splits/seeds.

Acknowledgements

This research is supported by AI-LEAF: "AI Institute for Land, Economy, Agriculture, and Forestry", funded by the USDA National Institute of Food and Agriculture (NIFA) and the NSF National AI Research Institutes Competitive Award. This research is also partially supported by Schmidt Sciences programs, an AI2050 Senior Fellowship and an Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship; the National Science Foundation; the US Department of Energy; the CALS Moonshot Seed Grant program; the "NYS Connects: Climate Smart Farms & Forestry" project, funded by the USDA, the New York State Department of Environmental Conservation, and the New York State Department of Agriculture and Markets; the National Science Foundation (NSF) through an NSF Research Traineeship (NRT) project in AI for Sustainability under Grant No. 2345579; and the Air Force Office of Scientific Research.

References

Agarwal, R.; Melnick, L.; Frosst, N.; Zhang, X.; Lengerich, B.; Caruana, R.; and Hinton, G. E. 2021. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34: 4699–4711.

Apley, D. W.; and Zhu, J. 2020. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 82(4): 1059–1086.

Bai, Y.; Chen, D.; and Gomes, C. P. 2021. CLR-DRNets: Curriculum learning with restarts to solve visual combinatorial games. In *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*, 17–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Beucler, T.; Rasp, S.; Pritchard, M.; and Gentine, P. 2019. Achieving conservation of energy in neural network emulators for climate modeling. *arXiv preprint arXiv:1906.06622*.

Chen, D.; Bai, Y.; Ament, S.; Zhao, W.; Guevarra, D.; Zhou, L.; Selman, B.; van Dover, R. B.; Gregoire, J. M.; and Gomes, C. P. 2021. Automating crystal-structure phase mapping by combining deep learning with constraint reasoning. *Nature Machine Intelligence*, 3(9): 812–822.

Chen, D.; Bai, Y.; Zhao, W.; Ament, S.; Gregoire, J.; and Gomes, C. 2020. Deep reasoning networks for unsupervised pattern de-mixing with constraint reasoning. In *International Conference on Machine Learning*, 1500–1509. PMLR.

Cuddington, K.; Fortin, M.-J.; Gerber, L.; Hastings, A.; Liebhold, A.; O'connor, M.; and Ray, C. 2013. Process-based models are required to manage ecological systems in a changing world. *Ecosphere*, 4(2): 1–12.

Daw, A.; Karpatne, A.; Watkins, W. D.; Read, J. S.; and Kumar, V. 2022. Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge guided machine learning*, 353–372. Chapman and Hall/CRC.

Daw, A.; Thomas, R. Q.; Carey, C. C.; Read, J. S.; Appling, A. P.; and Karpatne, A. 2020. Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling. In *Proceedings of the 2020 siam international conference on data mining*, 532–540. SIAM.

Dieleman, W. I.; Vicca, S.; Dijkstra, F. A.; Hagedorn, F.; Hovenden, M. J.; Larsen, K. S.; Morgan, J. A.; Volder, A.; Beier, C.; Dukes, J. S.; et al. 2012. Simple additive effects are rare: a quantitative review of plant biomass and soil process responses to combined manipulations of CO2 and temperature. *Global change biology*, 18(9): 2681–2693.

Eilers, P. H.; and Marx, B. D. 1996. Flexible smoothing with B-splines and penalties. *Statistical science*, 11(2): 89–121.

Eilers, P. H.; and Marx, B. D. 2010. Splines, knots, and penalties. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(6): 637–653.

Fan, J.; Chen, D.; Wen, J.; Sun, Y.; and Gomes, C. 2022. Monitoring Vegetation From Space at Extremely Fine Resolutions via Coarsely-Supervised Smooth U-Net. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5066–5072.

Feng, D.; Liu, J.; Lawson, K.; and Shen, C. 2022. Differentiable, learnable, regionalized process-based models with multiphysical outputs can approach state-of-the-art hydrologic prediction accuracy. *Water Resources Research*, 58(10): e2022WR032404.

Friedman, J. H. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Greenwell, B. M.; Boehmke, B. C.; and McCarthy, A. J. 2018. A simple and effective model-based variable importance measure. *arXiv preprint arXiv:1805.04755*.

Huang, Y.; Lu, X.; Shi, Z.; Lawrence, D.; Koven, C. D.; Xia, J.; Du, Z.; Kluzek, E.; and Luo, Y. 2018. Matrix approach to land carbon cycle modeling: A case study with the Community Land Model. *Global Change Biology*, 24(3): 1394–1404.

Jia, X.; Willard, J.; Karpatne, A.; Read, J.; Zwart, J.; Steinbach, M.; and Kumar, V. 2019. Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM international conference on data mining*, 558–566. SIAM.

Jia, X.; Willard, J.; Karpatne, A.; Read, J. S.; Zwart, J. A.; Steinbach, M.; and Kumar, V. 2021. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science*, 2(3): 1–26.

Jobbágy, E. G.; and Jackson, R. B. 2000. The vertical distribution of soil organic carbon and its relation to climate and vegetation. *Ecological applications*, 10(2): 423–436.

Karpatne, A.; Jia, X.; and Kumar, V. 2024. Knowledge-guided machine learning: Current trends and future prospects. *arXiv preprint arXiv:2403.15989*.

Karpatne, A.; Kannan, R.; and Kumar, V. 2022. *Knowledge guided machine learning: Accelerating discovery using scientific knowledge and data*. CRC Press.

Krishnapriyan, A.; Gholami, A.; Zhe, S.; Kirby, R.; and Mahoney, M. W. 2021. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34: 26548–26560.

Lal, R.; Negassa, W.; and Lorenz, K. 2015. Carbon sequestration in soil. *Current Opinion in Environmental Sustainability*, 15: 79–86.

LeCun, Y.; Bengio, Y.; et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10): 1995.

Lee, H.; Calvin, K.; Dasgupta, D.; Krinner, G.; Mukherji, A.; Thorne, P.; Trisos, C.; Romero, J.; Aldunce, P.; Barret, K.; et al. 2023. IPCC, 2023: Climate change 2023: Synthesis report, summary for policymakers. Contribution of working groups i, II and III to the sixth assessment report of the intergovernmental panel on climate change [core writing team, h. Lee and j. Romero (eds.)]. IPCC, geneva, Switzerland.

Liu, L.; Zhou, W.; Guan, K.; Peng, B.; Xu, S.; Tang, J.; Zhu, Q.; Till, J.; Jia, X.; Jiang, C.; et al. 2024a. Knowledge-guided machine learning can improve carbon cycle quantification in agroecosystems. *Nature communications*, 15(1): 357.

Liu, Z.; Ma, P.; Wang, Y.; Matusik, W.; and Tegmark, M. 2024b. Kan 2.0: Kolmogorov-arnold networks meet science. *arXiv preprint arXiv:2408.10205*.

Liu, Z.; Wang, Y.; Vaidya, S.; Ruehle, F.; Halverson, J.; Soljačić, M.; Hou, T. Y.; and Tegmark, M. 2024c. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*.

Lu, X.; Du, Z.; Huang, Y.; Lawrence, D.; Kluzek, E.; Collier, N.; Lombardozzi, D.; Sobhani, N.; Schuur, E. A.; and Luo, Y. 2020. Full implementation of matrix approach to biogeochemistry module of CLM5. *Journal of Advances in Modeling Earth Systems*, 12(11): e2020MS002105.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Luo, Y.; Ahlström, A.; Allison, S. D.; Batjes, N. H.; Brovkin, V.; Carvalhais, N.; Chappell, A.; Ciais, P.; Davidson, E. A.; Finzi, A.; et al. 2016. Toward more realistic projections of soil carbon dynamics by Earth system models. *Global Biogeochemical Cycles*, 30(1): 40–56.

Luo, Y.; Huang, Y.; Sierra, C. A.; Xia, J.; Ahlström, A.; Chen, Y.; Hararuk, O.; Hou, E.; Jiang, L.; Liao, C.; et al. 2022. Matrix approach to land carbon cycle modeling. *Journal of Advances in Modeling Earth Systems*, 14(7): e2022MS003008.

Luo, Y.; and Schuur, E. A. 2020. Model parameterization to represent processes at unresolved scales and changing properties of evolving systems. *Global Change Biology*, 26(3): 1109–1117.

Luo, Y.; and Smith, B. 2022. *Land carbon cycle modeling: Matrix approach, data assimilation, & ecological forecasting*. CRC Press.

Marcinkevičs, R.; and Vogt, J. E. 2023. Interpretable and explainable machine learning: A methods-centric overview with concrete examples. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(3): e1493.

Min, Y.; Chang, M.-C.; Kong, S.; Gregoire, J. M.; van Dover, R. B.; Thompson, M. O.; and Gomes, C. P. 2023. Physically Informed Graph-Based Deep Reasoning Net for Efficient Combinatorial Phase Mapping. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, 392–399. IEEE.

Molnar, C. 2025. *Interpretable Machine Learning*. 3 edition. ISBN 978-3-911578-03-5.

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378: 686–707.

Reichstein, M.; Ahrens, B.; Kraft, B.; Camps-Valls, G.; Carvalhais, N.; Gans, F.; Gentine, P.; and Winkler, A. J. 2022. Combining system modeling and machine learning into hybrid ecosystem modeling. In *Knowledge guided machine learning*, 327–352. Chapman and Hall/CRC.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5): 206–215.

Shen, C.; Appling, A. P.; Gentine, P.; Bandai, T.; Gupta, H.; Tartakovsky, A.; Baity-Jesi, M.; Fenicia, F.; Kifer, D.; Li, L.; et al. 2023. Differentiable modelling to unify machine learning and physical models for geosciences. *Nature Reviews Earth & Environment*, 4(8): 552–567.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, 3319–3328. PMLR.

Tan, T.; Genova, G.; Heuvelink, G. B.; Lehmann, J.; Poggio, L.; Woolf, D.; and You, F. 2024. Importance of Terrain and Climate for Predicting Soil Organic Carbon Is Highly Variable across Local to Continental Scales. *Environmental science & technology*, 58(26): 11492–11503.

Tao, F.; Zhou, Z.; Huang, Y.; Li, Q.; Lu, X.; Ma, S.; Huang, X.; Liang, Y.; Hugelius, G.; Jiang, L.; et al. 2020. Deep learning optimizes data-driven representation of soil organic carbon in earth system model over the conterminous United States. *Frontiers in Big Data*, 3: 17.

Tsai, W.-P.; Feng, D.; Pan, M.; Beck, H.; Lawson, K.; Yang, Y.; Liu, J.; and Shen, C. 2021. From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nature communications*, 12(1): 5988.

van Bree, R.; Marcos, D.; and Athanasiadis, I. 2025. Hybrid Phenology Modeling for Predicting Temperature Effects on Tree Dormancy. *arXiv preprint arXiv:2501.16848*.

Wang, S.; Chen, W.; Xie, S. M.; Azzari, G.; and Lobell, D. B. 2020. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 12(2): 207.

Willard, J.; Jia, X.; Xu, S.; Steinbach, M.; and Kumar, V. 2022. Integrating scientific knowledge with machine learning for engineering and environmental systems. *ACM Computing Surveys*, 55(4): 1–37.

Xu, H.; Fan, J.; Tao, F.; Jiang, L.; You, F.; Houlton, B. Z.; Sun, Y.; Gomes, C. P.; and Luo, Y. 2025. Biogeochemistry-Informed Neural Network (BINN) for Improving Accuracy of Model Prediction and Scientific Understanding of Soil Organic Carbon. *arXiv preprint arXiv:2502.00672*.