PERSPECTIVE

Differentiable modeling unifies machine learning and physical models to advance Geosciences

Chaopeng Shen[1*], Alison P. Appling[2], Pierre Gentine[3], Toshiyuki Bandai[4], Hoshin Gupta[5], Alexandre Tartakovsky[6], Marco Baity-Jesi[7], Fabrizio Fenicia[7], Daniel Kifer[8], Li Li[1], Xiaofeng Liu[1], Wei Ren[9], Yi Zheng[10], Ciaran J. Harman[11], Martyn Clark[12], Matthew Farthing[13], Dapeng Feng[1], Praveen Kumar[6,14], Doaa Aboelyazeed[1], Farshid Rahmani[1], Yalan Song[1], Hylke E. Beck[15], Tadd Bindas[1], Dipankar Dwivedi[16], Kuai Fang[17], Marvin Höge[7], Chris Rackauckas[18], Binayak Mohanty[19], Tirthankar Roy[20], Chonggang Xu[21], Kathryn Lawson[1]

1 Civil and Environmental Engineering, The Pennsylvania State University, University Park, PA, USA.
2 U.S. Geological Survey, Reston, VA, USA
3 National Science Foundation Science and Technology Center for Learning the Earth with Artificial Intelligence and Physics (LEAP), Columbia University, New York, NY USA
4 Life and Environmental Science Department, University of California, Merced, CA, USA
5 Hydrology and Atmospheric Sciences, The University of Arizona, Tucson, AZ, USA.
6 Civil and Environmental Engineering, University of Illinois, Urbana Champaign, IL, USA
7 Eawag: Swiss Federal Institute of Aquatic Science and Technology, Dübendorf, Switzerland
8 Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA
9 Department of Natural Resources and the Environment, University of Connecticut, Storrs, CT, USA
10 Southern University of Science and Technology, Shenzhen, Guangdong Province, China
11 Department of Environmental Health and Engineering, Johns Hopkins University, Baltimore, MD, USA
12 Global Institute for Water Security, University of Saskatchewan, Canmore, Alberta, Canada
13 US Army Engineer Research and Development Center, Vicksburg, MS, USA
14 Prairie Research Institute, University of Illinois, Urbana Champaign, IL, USA
15 Physical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
16 Lawrence Berkeley National Laboratory, Berkeley, CA, USA
17 Department of Earth System Science, Stanford University, Stanford, CA, USA
18 Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, Massachusetts, USA
19 Department of Biological & Agricultural Engineering, Texas A&M University, College Station, TX, USA
20 Civil and Environmental Engineering, University of Nebraska-Lincoln, NE, USA
21 Earth and Environmental Divisions, Los Alamos National Laboratory, NM, USA
* Corresponding author, email cshen@engr.psu.edu

This document is the accepted manuscript version of the following article:
Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., … Lawson, K. (2023). Differentiable modelling to unify machine learning and physical models for geosciences. Nature Reviews Earth, 4, 552-567. https://doi.org/10.1038/s43017-023-00450-9

Abstract (150 words)

In many domains of Geosciences, Process-Based Modeling (PBM) offers benefits in interpretability and physical consistency but struggles to efficiently leverage large datasets. Machine Learning (ML) methods, especially deep networks, have strong predictive skills, yet lack the ability to answer specific scientific questions. In this Perspective, we present differentiable modeling (DM) in geosciences as a pathway that dissolves the perceived barrier between PBM and ML. "Differentiable" refers to accurately and efficiently calculating gradients with respect to model variables or parameters, enabling the discovery of high-dimensional unknown relationships. DM involves connecting (arbitrary amounts of) prior physical knowledge to neural networks, pushing the boundary of physics-informed machine learning. Evidence suggests DM offers improved interpretability, generalizability, and extrapolation capability compared to purely data-driven ML, while approaching its accuracy. DM requires less training data while scaling favorably in performance and efficiency with data. Geoscientists can now ask questions, test hypotheses, and discover unrecognized physical relationships.

Introduction

Geoscientific models encompass a wide range of domains, with evolving scopes and ever-increasing societal importance, especially in the face of rapid climate change. For example, hydrologic models help manage water resources[1,2] and plan for extremes such as floods and droughts[3]. Vegetation models can help predict the fate of the carbon and other key biogeochemical cycles on land[4] or in the ocean[5]. Agricultural models can help estimate crop yields and environmental impacts[6]. Geophysical models aim to predict land-surface changes via processes like landslides[7], land subsidence[8], the impact of future warming on glacial melt[9], and earthquakes. Biogeochemical reactive transport models aim to understand and predict surface and subsurface water chemistry and quality[10–12]. Combining many such components, Earth System Models[13–15] and Integrated Assessment Models[16–18] provide crucial climate projections and guidance for resource managers and policy makers[19,20].

Geoscientific models often share commonalities as they describe the temporally dynamic responses of systems to time-dependent forcings as modulated by static landscape attributes. Many such problems can be described as systems of nonlinear equations, or ordinary and/or partial differential equations (ODE/PDEs). The overall system can contain multiple processes, some of which are well understood while others are only empirically represented. Parameterizations (parametric representations of physical processes) are extensively employed[21,22]. Further, process representations and parameterizations are often subject to considerable uncertainty, some of which is related to the coarse scale of the models, and thus have significant room for improvement.

The recent rapid growth of machine learning (ML) offers new opportunities for learning from big data and filling knowledge gaps in geoscientific models, in particular. While various forms of physics-informed ML have been proposed for several years, there has been a lack of recognition of one core strength of ML --- differentiable programming. Understanding this strength as well as its limitations will enable us to reframe how ML and physical models can best interact and synergize.

In this Perspective, we argue that differentiable implementations of geoscientific models offer a transformative approach to simultaneously improve process representations, parameter estimation, knowledge discovery, and predictive accuracy, by connecting process-based and machine-learning-based model components. It offers immense potential to advance the wide variety of geoscientific domains.

PBMs and ML in the geosciences

While process-based modeling (PBM) and pure machine learning (ML) are both valuable approaches to modeling, they each have their own limitations.

Process-based geoscientific modeling

The traditional process-based modeling (PBM) approach, which uses models derived deductively from physical laws or empirical relationships[23,24], has helped improve our understanding of system functions and behaviors. Due to their deductive nature, they can be leveraged to test hypotheses or to assess the system's response and causation relationships (see the *Physical Laws* row in Table 1). Further, they can simulate a wide range of observed (for example, discharge or leaf area index) and unobserved variables (for example, groundwater recharge or fine-root distribution). Such an ability is critical to both advancing scientific understanding and to providing a narrative when communicating with the public and stakeholders who are engaged in decision making[25]. It is possible to ask specific questions regarding processes within the modeled system, by progressively improving the representations of processes[23,26–28] and evaluating them using controlled experiments.

Despite these benefits, there remain important challenges with PBMs. Process-based models often cannot rapidly evolve with and fully exploit the information in "big data" due to the time needed to develop and test process representations and parameterizations[29,30]. The differences between model predictions and observations are first reconciled by parameter calibration, which can be non-trivial and can add significant uncertainty[31] (more about this later). For model errors beyond parameter tuning, modelers need to hypothesize different causes (for example, missing processes in the governing equation), then implement structural changes and iteratively confront the updated model structure and underlying hypotheses with the validation data[23]. This iterative process is highly expensive (in both labor and time) and complex, and can be biased by a developer's knowledge background[32]. Consequently, it is common that the structural representation of a specific process in a geoscientific model stagnates for years or decades[33–36].

One key reason for this stagnation is that process-based models are limited by knowledge gaps. Extensive physical, biological, and socioeconomic knowledge is required to achieve adequate model structure representations, and any deficiencies can amplify errors and ambiguity. Another major challenge is the interactions of processes across disciplinary boundaries[37]. For instance, vegetation, microbes, human management, and socioeconomic systems all interact with each other and affect the water and carbon and other biogeochemical cycles[38–41]. Interdisciplinary research is highly valuable but challenging, therefore limiting our progress toward accurate model prediction.

Machine-learning-based geoscientific modeling

Irrespective of the domain or application, there has been a rapid increase in the use of purely data-driven machine learning (ML) approaches, especially deep neural networks (NNs). NNs have highly generic model structure and many parameters that are determined from training on data. ML has been applied to a wide range of scientific applications[37,42] (see Discussion A in Supplementary Information S1). In the geosciences, NNs have shown promise in predicting crop production[43,44], precipitation fields[45,46] and clouds[47], water quality variables[48,49] such as water temperature[50–53], dissolved oxygen[54,55], phosphorous[56], and nitrogen[57,58], and the full hydrologic cycle[59] including soil moisture[60–62], streamflow[63–66], evapotranspiration[67–69], groundwater levels[70], and snow[71], etc. Deep networks like long short-term memory (LSTM) networks[72], transformers[73,74], graph neural networks[65], and convolutional neural networks (CNNs)[75,76] have become widely known in geosciences. Many studies reported noticeably better performance than conventional

process-based or statistical models, revealing that earlier work did not fully exploit the information in the data[29] (Table S1 in Supplementary Information S1).

Nevertheless, there remain important challenges with purely data-driven ML:

Machine learning is typically data hungry. The success of deep networks has relied on the availability of "big data", which are, unfortunately, oftentimes not available in many geoscientific applications[58,77], where variables are measured at only tens, hundreds, or thousands of sites. For example, water quality data are sparse and inconsistent in temporal and spatial coverage[10,78]. For rare and extreme events which critically impact human activities, such as floods, droughts, and earthquakes, available data is even scarcer.

ML is not exempt from defaults and can struggle with data errors, incompleteness, out-of-sample or out-of-distribution predictions, or bias in the inputs or training data. The quality of ML models is therefore inherently limited by the quantity, diversity, and quality of the observations[53,79,80]. Since a purely data-driven model can, at best, nearly-perfectly replicate the patterns in the training data, it invariably inherits various issues from the training data including explicit or spurious biases, inadequate spatiotemporal resolutions (such as with satellite-based observations), and the inability to account for non-stationarity (shifting background statistical properties) or unseen extremes in time series due to the short data record.

ML algorithms are based on correlations and not causality, regarding both attributes and temporal changes. There are oftentimes confounding factors in data, so that ML models can produce the "right" results for the wrong (causal) reasons, potentially making predictions less reliable when circumstances are changed or outside of the training domain. Although causal representation learning[81] and explainable AI methods[82–84] have shown promise recently, challenges still remain with learning causality and interpretability. Parallel methods that can flexibly interrogate a model, encode causality and prior information, and identify missing physics anywhere in the model chain can be valuable.

Finally, purely data-driven ML models cannot predict untrained variables (those not provided as training targets). Due to their very nature, ML-based models are designed to only output the training targets. It is difficult for them to reveal how events unfolded. For example, in a study where soil moisture is unobserved, pure ML models cannot state whether "the flood occurred because the soil was saturated". This limits both the formation of hypotheses and communication with stakeholders.

Differentiable programming

Considering the successes and limitations of NNs, we seek to identify the foundational strengths of NNs and overcome its limitations. To this end, we argue that differentiable programming (explained below), is the computing paradigm that supports the efficient training of NNs which, when generalized, can deliver many philosophically and practically transformative outcomes to PBMs. Traditional process-based, statistical or hybrid modeling of Earth systems has been dealing with optimization problems for model parameter tuning (see the *Similarity* block in Table 1). However, only by exploiting the power of gradient-based optimization, which updates weights by explicitly tracking their contributions to the outcome, have researchers been able to learn from big data and efficiently train the large numbers of weights (parameters) necessary to approximate complex unknown functions.

The ability of generic NN architectures such as transformers, CNNs, and recurrent neural networks to approximate unknown functions has produced desirable outcomes (Figure 1 & Table 1). First, researchers from any field can concentrate on a few generic architectures, permitting cross-domain sharing of knowledge and experiences. Second, NNs can help identify previously unrecognized physical relationships. Third, NN training can scale up with the amount of data (in terms of accuracy, generalizability, and efficiency)[79,85],

contrasting traditional modeling where the learning may quickly saturate after some limited calibration of parameters or functions[53].

All of these abilities are possible only because NNs can now be trained with a large number of weights, providing a large learnable function space[86,87]. The number of weights easily exceeds the optimization capabilities of conventional algorithms. The LSTM models widely employed in hydrology can contain ~500,000 weights while recent large language models already have trillions of weights, which can lead to emergence of human-level abilities not observed at smaller scales[88]. In contrast, traditional evolutionary[89–91], or genetic[92] or particle swarm optimization methods[93] can hardly handle more than a few dozen independent parameters (Table 1).

The computing paradigm behind training large amounts of weights is differentiable programming[94,95] (meaning that we design programs in a way that their outputs are differentiable with respect to inputs), as cheaply obtained gradients allow for parameter updates via various first-order gradient-descent methods[96]. In the context of ML, it is largely enabled by automatic differentiation (AD), which decomposes a complex algorithm into a sequence of elementary arithmetic operations, and then applies the chain rule of differentiation to compute the derivatives. Reverse- or forward-mode AD constitutes a powerful functionality provided by ML platforms like PyTorch[97], JAX[98], Julia[99], and Tensorflow[100]. Models written on these platforms can be, often without much effort, programmatically differentiable even with mathematically indifferentiable operations (such as thresholding or *if* statements), as long as they are piecewise differentiable.

This leads us to conclude that it is differentiable programming that distinguishes neural networks from other traditional models, due to its ability to efficiently harvest large amounts of data and tune a very large number of parameters. Recognizing that differentiable programming is not exclusive with process-based modeling, it can serve as the basis for unifying NN and process-based geoscientific modeling**.** As we will discuss next, this unification of PBM and ML requires only minor modifications to our conceptual modeling and implementation strategies, but can open new doors for scientific discovery.

Differentiable modeling (DM)

Here we expand the scope of our discussion beyond differentiable programming and AD, and use the term "differentiable modeling" (Figure 1) to refer to joint physics-ML modeling approaches that use any method to rapidly and accurately produce gradients for large-scale optimization of the combined system. A distinct feature of DM is its predominant programmatical differentiability – that is, the whole model needs to support gradient calculation from the start to the end of the workflow – to ensure that we can trained combined neural networks that can adapt to and evolve from data. Purely data-driven neural networks already use differentiable programming (almost entirely via AD), but here "differentiable modeling" also emphasizes the hybrid nature of the system. A non-AD example is that of adjoint methods, which solve accompanying equations (called adjoint equations)[101–103] for the derivatives, and take advantage of the multiplicative nature of the chain rule to save computational time. AD differentiates through the low-level calculation, while adjoint methods differentiates through higher-level functions or mathematical equations[104]. Some other gradient estimation methods, like finite differences, are intractable for any reasonably-sized NNs (10,000 weights would require 10,001 forward model evaluations) and can be challenged by stiffness. Second-order methods, such as Newton Raphson, have not gained popularity for the training of NNs due to the costs and challenges of computing the Hessian matrix. The vast majority of NNs are implemented on platforms supporting differentiable programming, while most existing PBMs are not. We believe that converging the two (NNs and PBMs) through AD presents a tremendous opportunity for efficient learning from data.

DM pushes the boundary of physics-informed ML and can be considered a branch of scientific ML[105,106] that emphasizes improving process representations and interpretation (Supplementary Information S1, Discussion C). There are two perspectives from which we can view differentiable models (Figure 2a First, they are ML models constrained to a smaller searchable space by the structural priors, and thus can still repeat the benefit of big data when they exist. Second, they are PBMs augmented with learnable and adaptable components (and thus an expanded searchable space) provided by NNs, and can be trained in data-scarce scenarios and provide elucidation of processes.

Table 1. Similarities and differences between purely data-driven NNs and process-based models. [Pro] annotates the comparative strengths, also shown in green text. In the equations, W stands for weights of the neural network g; θ stands for the physical parameters of the process-based model f; x, u and A are dynamic forcings, state variables, and semi-static attributes, respectively; and L represents the loss function which quantifies the difference between simulation outputs and observations y^*.

	Purely data-driven NNs	**Purely process-based models**
	Similarities	
Mathematical form	$y = g^W(u, x, A)$ $W = \text{argmin}(L(y, y^*))$	$y = f^\theta(u, x, A)$ $\theta = \text{argmin}(L(y, y^*))$
Programmatically differentiable	Yes	Traditionally no, but can be reimplemented on ML platforms as shown in the *Applications* section.
	Differences	
Training/ Calibration	[Pro] Trained using data-driven training methods such as gradient descent, with gradient computations supported by differentiable programming	Typically calibrated at limited numbers of sites or for a limited number of parameters, though efficient many-site, multi-objective methods exist.
Architecture	[Pro] Generic structure with many weights that allow the model to flexibly learn a wide range of functions	Physically based equations (structural priors) representing human understanding of physics, with a limited number of parameters.
Data	[Pro] Capable of efficiently gaining accuracy and generalizability as datasets grow, with scaling benefits to big data.	Learning saturates at a small data quantity. [Pro] Can often predict reasonably despite data limitations in accuracy, resolution, and availability.
Unknown processes	[Pro] Can discover patterns and functions from data that might be unknown or uncertain.	Processes must all be explicitly specified by the modeler, even if they are only assumptions.
Domain knowledge	[Pro] Generic model architecture – Easy to develop even without domain expertise and accommodates large knowledge gaps	Specialized domain knowledge required.
Physical laws	Not guaranteed to respect physical laws.	[Pro] Respect physical laws.
Inspection	Outputs trained variables only.	[Pro] Provide access to many intermediate variables that facilitate interpretability. Provides interpretable intermediate variables
Interpretation	Takes much effort to interpret, and internal variables are not guaranteed to have physical meaning.	[Pro] Contain equations representing physical processes, allowing narration of model "reasoning" and formal tests of alternative representations
Education	Taught in computer science or data science curricula.	Taught in engineering or science curricula.

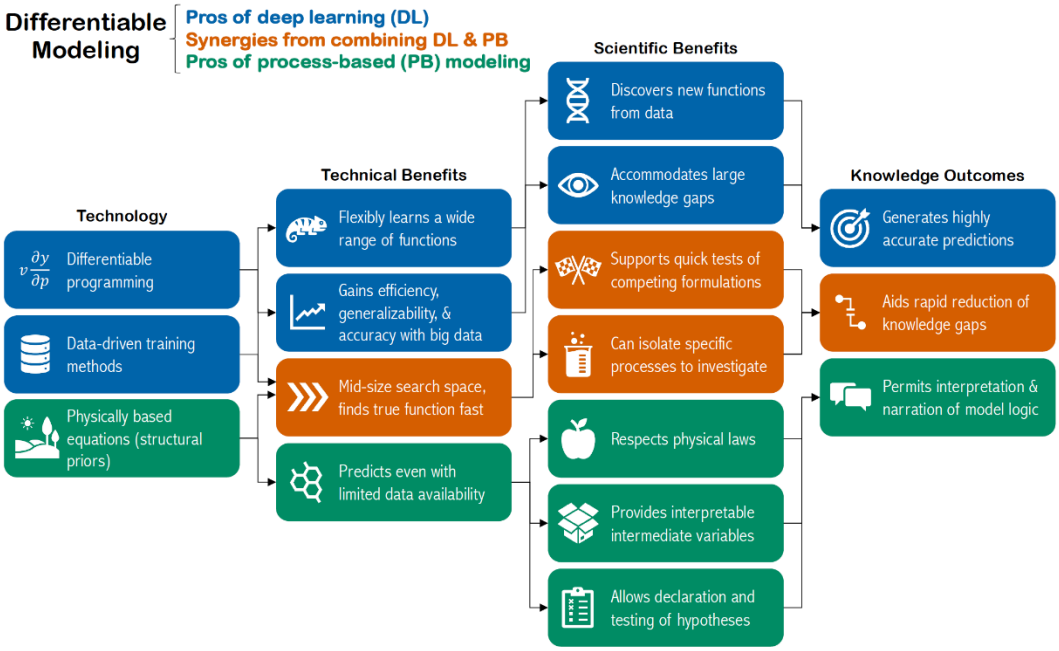

Figure 1. *ML (blue boxes) gives us great results with easy-to-use models, resulting from the complexity of neural networks (many parameters) and the technologies that make it feasible to train such complex models. The most fundamental of these technologies is differentiable programming. In the DG paradigm, which incorporates differentiable non-ML model components (physically based structural priors), we can now obtain additional great features (orange boxes) while retaining and augmenting the old ones (green boxes).*

Approximating functions inside the model

While efficient gradient calculation may appear to be merely a technical change, it is however likely to transform our modeling philosophy. First, the ability to approximate complex, unknown functions using data can greatly broaden the type of questions that can be asked, by enabling us to treat trusted model components as priors and focus on improving the more uncertain components. To explain this idea in concise mathematical terms, let us consider a physics-based model,

$$(1)\quad y = g(u, x, \theta)\ ,$$

where y is the environmental variable to be predicted, and u, x, and θ represent state variables, dynamic forcings, and physical parameters, respectively. This representation encompasses differential equations but is more generic, for example:

$$(2)\quad \partial u/\partial t = g(u, x, \theta).$$

Traditional inversion algorithms only estimate the parameters (essentially asking, "θ =?") while requiring that the functional form g be assumed *a priori* (except for some rigid methods like nonparametric regression, which require complicated derivations and specialized training algorithms, and thus have not gained popularity). However, differentiable models allow us to ask questions about the functional form g itself, by training, for instance, a neural network (*NN*) on observed data to replace g with:

$$(3)\quad y = NN^{W}(u, x, \theta).$$

where W is the high-dimensional weights. With DM, we now can place our question mark precisely in the model. The functions to estimate could be a parameterization scheme, as done in differentiable parameter learning[99], for example:

$$y = g(u, x, \theta = NN^{W}(A)) \quad (4)$$

The function could be a module in a model (Figure 3), where we can replace g_3 in

$$y = g(g_1, g_2, g_3(u, x, \theta)) \quad (5)$$

with a *NN* as done in Feng et al.[107] by optionally replacing the runoff function:

$$y = g(g_1, g_2, NN^{W}(u, x, \theta)) \quad (6)$$

Alternatively, the function could be a part of a governing equation or constitutive laws. For example, we can estimate NN^{W} in the following equation[108,109]:

$$\partial u/\partial t = g(g_1, g_2, NN^{W}(u, x, \theta)) \quad (7)$$

In the above equations, the physical process equations provide a backbone (or inductive bias) for the overall model; in equation 4 the physical backbone is g; in equations 5-7, the physical backbone is g, g_1, g_2 and g_3. The unchanged parts (structural priors) like g, g_1, g_2 critically serve as physical constraints. We may gain insights by simply visualizing the relationships learned by NN^{W} [65,110] or applying knowledge distillation methods[111]. We are also able to evolve better process representations for some model components such as g_3 mentioned above, for example, the relation between soil moisture and effective rainfall in conceptual hydrologic models, without needing a full understanding of all the processes. This precision and latitude of questioning is unprecedented. Moreover, as ML learns an overall mapping from *x* to *y* based on correlations, it can intertwine many processes, making its interpretation difficult. As we break the mapping down into multiple subparts based on inserting prior knowledge, we inherently reduce complexities, reducing the scope of learning, and improve interpretability (Figure 3).

Differentiable modeling provides a framework for combining deductive reasoning and inductive learning. Purely data-driven models are inductive and seek to derive almost all relationships from data, whereas process-based models first posit hypotheses and then test those hypothesis using data. Differentiable modeling posits a user-defined number of structural assumptions, and then identifies other parts of the model from data. This design follows the traditional scientific approach that identifies parsimonious models to reflect the general properties of the phenomenon, along with a quantification of the predictable aspects that are not yet well understood[112]. Moreover, differentiable models can approach state-of-the-art performance that matches data-driven models (Supplementary Information S1, Discussion B).

Differentiable Modeling in Geosciences

Here we advocate for a new modeling genre in modeling all earth and environmental processes: "Differentiable Modeling in Geosciences" (DM). DM in Geosciences intermingles geoscientific physical equations (called structural priors) with NNs to simulate processes, update process representations, learn meaningful parameters, quantify uncertainty and ask a range of questions (Table 2). DG may also exploit gradients for other purposes such as sensitivity analysis or trajectory optimization. DG seeks to marry the core of NN models – their optimizing and learning capabilities – to geoscientific process descriptions.

For geoscientific problems, NNs can be utilized in a wide variety of ways, ranging from learning physical parameters[113] to updating structural assumptions in a component[79], or estimating time-dependent forcing terms of the natural systems. We emphasize that DG is different from previous concepts introduced

in physics-guided machine learning (PGML) or not-fully-differentiable models in terms of methodology (must be fully differentiable), mission (to advance process understanding), and philosophy (whether treating physical law as truth or not). Please see Supplementary Information S1, Discussion C for the comparison.

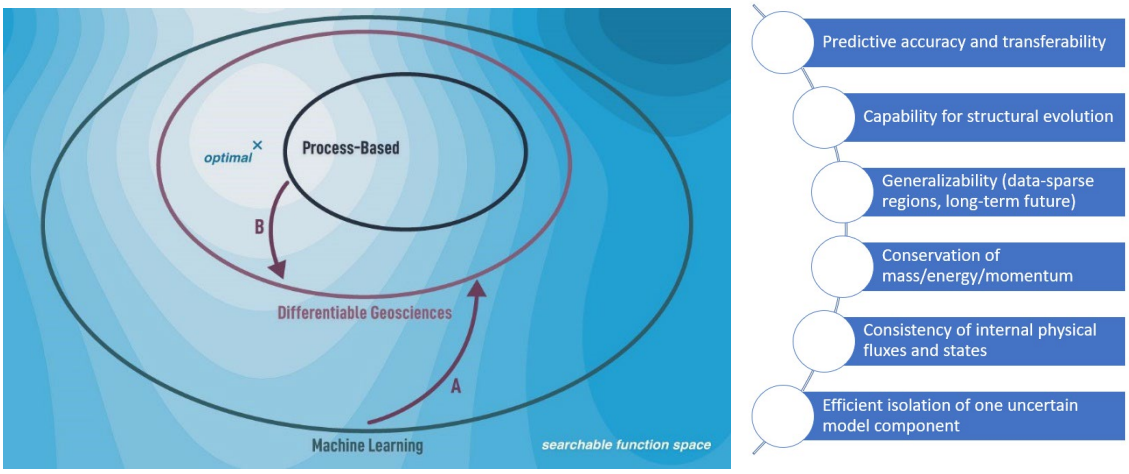

Figure 2. *(Left) Differentiable models can be viewed as (A) machine learning models guided into a smaller searchable space (ovals) by structural priors or (B) process-based models with expanded search space supported by learnable units. The background color gradient indicates model optimality, related to the cost function if we had infinite data, such that "optimal" marks the location of the ideal model solution. (Right) Differentiable models could evolve to gain process knowledge while improving the model predictions. Success can be claimed if DG models can be developed with the following features: Predictive accuracy and transferability equal to or superseding purely data-driven models for extensively measured variables; Capable of structural evolution, enabling improvements to the parameterization and formulation of the processes; Accurate generalizability to data-sparse regions or into the long-term future; Conservation of mass/energy/momentum; Consistency of internal physical fluxes and states that can provide a full narrative of the events and full support to downstream processes; Permits efficient isolation of one uncertain model component at a time to learn physics with less ambiguity.*

Overcoming data limitations

DM is particularly suitable for the geosciences due to the nature of datasets and problems. First of all, geoscientific data are strongly imbalanced in spatial extent, temporal coverage, and the coverage of variables, and also have noise in observational datasets. While satellites can be used to indirectly estimate global leaf area index[114] or coarse-resolution surface soil moisture[115,116], and other variables[117], there are a limited number of sites measuring photosynthesis rates[118], soil respiration or streamflow, especially in Africa and Asia[119]. Globally, there is very limited knowledge of subsurface properties. Purely data-driven ML may be biased by these data limitations, yet these biases could be partially alleviated by the inclusion of physics as an inductive bias. Indeed, preliminary analysis shows that differentiable models with a process-based model as the backbone can outperform LSTM in regional extrapolation[120].

The second major motivation behind the use of differentiable modeling is nonstationarity in the geosciences induced by changes in climate, land-use land-cover, etc., which could drive many systems out of the previously observed range of variability[121]. Earlier tests indicate while ML models presents highly competitive performance[64,120], it still declines substantially in accuracy when faced with nonstationary processes[120,122]. Constrained by physical formulations, DG has a chance to better represent future trends[120].

As DG models can also output any diagnostic variable calculated by the process-based equations within the DG model, we can perform model conditioning and/or data assimilation operations with sparse and scattered data. By conditioning, we mean constraining the model using observations to improve overall model dynamics For example, a hydrologic model can be conditioned by satellite soil moisture or streamflow data so that it can better predict vegetation water use[113], primary productivity, or snow water equivalent[123]. For data assimilation, the model can use recent observations of B to improve the short-term forecast of A, as B can also help to update our model state variables.

DG is primed to greatly improve the quality of physical parameters, which strongly control the behaviors of the models. Quite often, we have no ground truth information for the parameters and they require inversion from observations or high-resolution simulations. Parameter estimation has, for decades, been fraught with uncertainty and ambiguity. Due to different parameters producing very similar output and their sensitivity to spatiotemporal resolutions, calibration at a geographic location can often lead to nonuniqueness (sometimes referred to as "equifinality")[124–126]. Extending parameters to unmonitored locations requires "regionalization", which may improve robustness, but it is difficult for traditional methods to achieve optimal results. Training neural networks as parameter generators has a great potential to improve parameter generalization and performance, while also giving insights about parameter sensitivity. Using all the available data points to constrain the parameters can generate favorable scaling behaviors – more training data leads to improved performance, efficiency, and generalizability[113] (discussed in Supplementary Information Text S1).

Earth science applications

Here we call for more attention to differentiable modeling as a new modeling genre for geosciences. DG holds the potential to tackle a diverse array of novel questions across various geoscientific domains, pursuing ambitious goals ranging from high performance to knowledge discovery (Figure 2b). Some example questions suitable for DG are given in Table 2. This section briefly describes early explorations of DG, categorized by how gradients are computed and employed. This section also gives examples, which are by no means exhaustive, to explain the concepts and to inspire more innovation.

Table 2. Differentiable Geosciences can help almost all geoscientific domains in knowledge discovery and improving simulation quality. We provide some core question types, along with example questions and their domains.

	Question	**Examples**	**Domain**
a	What is the relationship between x and y?	*How do we estimate floodplain hydraulic parameter values efficiently at large scales using new sensing data*	Hydraulics
		How does global groundwater-dominated baseflow respond to climate change	Hydrology
b	What physics is missing from the differential equation?	*Can we find functional forms to express soil hydraulic properties (water retention and hydraulic conductivity function) that describes non-equilibrium flow?*	Soil Science
c	What should be the assumption here?	*What is the main driver of reduced plant production: vapor pressure deficit or deficit in soil moisture?*	Ecosystem
		What is a proper, scale-appropriate way to parameterize groundwater storage and flow at the global scale?	Hydrology

d	How does factor A influence parameter β?	*Can we use space-based observations of geohazards, e.g., landslides[122], to quantify subsurface properties (so we can better predict future events)?*	Geohazards
		How and to what extent do river chemistry and quality vary across gradients of climate, vegetation, land use, and geology conditions? thus how do they change in a warmer climate and intensified human modification (type f)?	Water Quality
e	Is a process causing phenomenon P?	*Is CO_2 fertilizing plants and increasing global photosynthesis?*	Climate
f	What will happen under new environmental conditions?	*How can we predict crop phenology dynamics (e.g., planting, shooting, flowering, harvesting) and assess potential production risk under future climate change, which involves interconnected biotic, abiotic, and human influences?*	Agriculture
		How can we leverage both physics and data to create more accurate models for ice dynamics within the cryosphere and better constrain its fate under climate change?	Cryosphere
g	What is the information content of datasets (inputs, training targets)?	*How can we better leverage emerging sensing platforms while improving our model representations of sediment transport and nonlinear wave-wave interactions in order to infer nearshore bathymetry at large scales*	Coastal

Differentiating through numerical models

Differentiating through numerical models by leveraging modern ML platform is likely the most straightforward and the most similar to traditional models. We leverage both AD and, when necessary, a customized backward function (adjoint) to keep track of gradients at relatively elementary levels of operations. For explicit time stepping, utilizing modern ML platforms like PyTorch, Julia or JAX, one can, in theory, reimplement an existing physical model coded in Fortran or C/C++ to obtain a differentiable model version via AD (and ensure reproducibility). Then the differentiable model is connected to NNs. The physics is clearly enforced, and the user obtains an efficient forward simulator for any initial, boundary and forcing conditions. They can also migrate the learned relationships to existing models to immediately support operations.

For problems that need iterative solvers, e.g., system of nonlinear equations or stiff ODEs requiring implicit time stepping, direct AD may consume too much memory, but adjoint-based backward functions can be employed instead at the iterative solver level (so called "discretize-then-optimize"). Alternatively, adjoint functions can also be written at the differential equation level, in which case we solve an adjoint differential equation backward in time to compute the gradients (so called "optimize-then-discretize")[176]. Care needs to be taken gradients computed in the optimize-then-discretize way as sometimes we obtain lower-accuracy gradients that interfere with training[127]. Adjoint methods have been used to solve optimization problems governed by PDEs, with the adjoint equations derived either manually[128], or, more rarely, by automated programs[129]. They might be more computationally efficient than AD for certain problems by exploiting the structure of the mathematical model. Adjoint solvers have long been successfully employed in numerical weather prediction, like 3D or 4DVar[130] and groundwater modeling[131], for the purpose of efficient data assimilation or calibration. In addition, those methods were traditionally not

connected to the neural network training machinery, perhaps because the role of differentiable programming was not clear at the time.

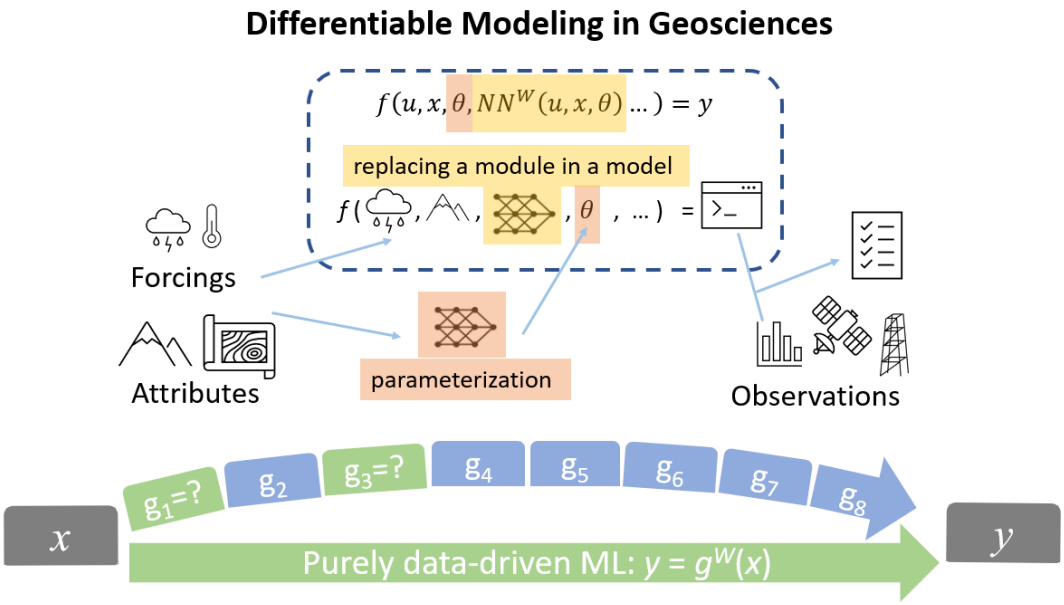

Figure 3. The general idea of a differentiable, learnable process-based model. Differentiable models can help almost all geoscientific domains in knowledge discovery and improving simulation quality. At the bottom, we show purely data-driven ML models that directly learns a mapping relationship from x to y, which intertwines many processes and is thus difficult to understand. Differentiable models allow us to break the model in into portions (g_1, g_2 are to be learned) so that we can narrow the scope of the relationships to be learned (potentially with less data) for better interpretability. At the top, we illustrate that in differentiable modeling, NNs can serve as parameterization schemes or processes inside a model. They can adapt to data and learn from data.

As an example, the conceptual hydrologic model HBV (a system of ODEs) was implemented on PyTorch and coupled NNs provided regionalized parameterization[107] (Figure 4). "Regionalized" means the parameterization is trained by all sites simultaneously, which poses a strong constraint and improves robustness). Strikingly, its streamflow simulation approached the performance level of LSTM, with very similar performance under different forcing datasets. The soil moisture-runoff function can be replaced with NNs, which learns a new moisture-runoff relation (similar to a constitutive relation) where the precipitation amounts heavily influences runoff for threshold-like watershed systems. This implementation also output untrained variables such as evapotranspiration and baseflow, which agreed well with alternative estimates. To improve numerical accuracy and parameter robustness[132,133], we can incorporate adjoint backward functions for implicit time-stepping. Moreover, in spatial extrapolation cases, the differentiable model outperformed ML models (LSTM in this case) with respect to daily metrics and decadal trends[120] (Figure 4) due to the structural constraints, demonstrating its potential for global hydrologic modeling. Similarly, other work[123] encoded the hydrologic model EXP-HYDRO as a recurrent NN architecture and coupling it with fully connected NNs which served as the parameterization pipeline as well as postprocessor to improve runoff. A symbiotic integration between NN and physics led to robust transferability across basins. Recently, a hybrid neural ODE approach, in which NNs were used to substitute the differential equations-

based hydrologic model, thereby improving predictions while keeping full interpretability of a mechanistic model[134]. In the biogeosciences or ecosystem modeling, differentiable models have also been used to improve parameters for photosynthesis[135] at large scales.

Apart from models similar to ODEs, direct differentiation can also be applied to models operating on graphs representing the natural systems, such as river networks. An advective dispersion equation implemented on a river graph to simulate stream water temperature was found to perform better in data-sparse situations[136]. Similarly, when a differentiable river routing model was trained on daily discharge at a gauge downstream of a river network (with pretrained LSTM producing runoff as inputs to the graph) to learn a parameterization scheme for n[137], a power-law-like curve was obtained between Manning's roughness coefficient (n) and catchment area, consistent with the expected behavior.

To give a more adjoint-focused example, a non-linear coefficient in the Poisson equation and a heat equation was recovered when unknown functions or operators in a PDE were replaced by NNs, the PDE was discretized by a finite element method, and the was gradient provided by the adjoint method[128]. To overcome the challenge facing Newton iteration convergence due to the incorporation of NN and the lack of a preconditioner, an operator-splitting approach was used to discretize the PDE into two subproblems. The first subproblem only has differential operators of the PDE, not NNs, while the other subproblem with NNs can be solved by integrating NNs by a Gaussian quadrature rule. The approach can similarly apply to equations in geosciences.

Reimplementing a model into a differentiable version may incur non-trivial development costs. Mathematical changes may be required to adapt previously non-differentiable mathematical operations, for example, by replacing indexing with convolutions, or to improve parallel efficiency. While DG models may not always have to run on Graphical Process Units (GPUs), enabling the use of GPUs would improve the computational efficiency by orders of magnitudes compared to CPUs, notwithstanding some current challenges (described in the *Challenges to address for DG* section). Our position is that in most cases, the cost is well worth the investment due to the potential to interrogate into the model, make changes, and learn physics. The reimplementation may also provide a "reset" opportunity to re-examine many of the commonly-made model assumptions or implementation choices.

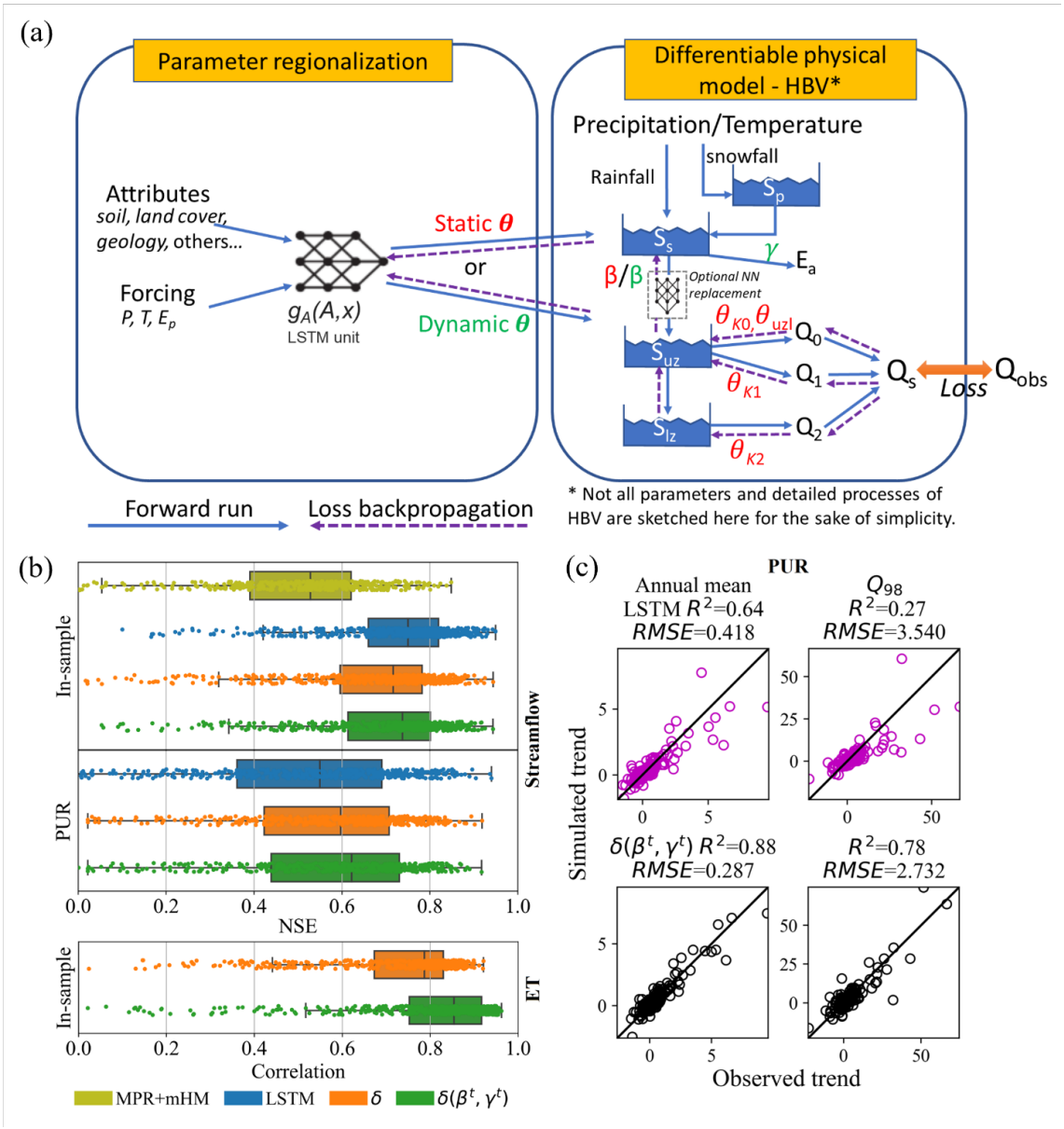

Figure 4. (Adapted from Feng et al.[107,120]). (a) Sketch of a differentiable hydrologic model using process-based hydrologic model HBV as a backbone. The purple dashed lines illustrate an example of the paths of backpropagation to train the two connected neural networks, but backpropagation can in fact update any component including making corrections to precipitation. (b) The differentiable models (with $\delta(\beta^t, \gamma^t)$ indicating model with two time-dependent parameters) can approach the performance of LSTM and greatly outperform traditional approaches (mPR+mHM) on the basins of the CAMELS dataset (higher NSE is better) for the in-sample temporal test and outperforms LSTM for spatial extrapolation test (PUR). δ models can output evapotranspiration (ET) at high accuracy while LSTM cannot (correlation was evaluated against a satellite product); (c) For prediction in ungauged regions (PUR, representing spatial extrapolation: trained in some regions and tested in another large ungauged region), δ models can surpass the performance of LSTM in terms of projecting decadal-scale trends in annual mean streamflow or high-flow (Q_{98}). Please refer to Feng et al.[107,120] for details regarding forcing and benchmarks.

Connecting NNs with PBMs through surrogate models.

If we train a neural network as a surrogate for a PBM and faithfully reproduce its behavior, we can then connect this surrogate model to other NN components in a DG framework because the surrogate is programmatically differentiable. This idea has driven many studies to train surrogate models for hydrologic, hydraulic[138,139], and reactive transport models, and then further using inversion[140] and optimization. For another example, many benefits and a favorable scaling relationships (with more data) were shown with differentiable parameter learning[113], which connected a physics-based model like VIC or its surrogate model to a neural network (g), using some widely available attributes ($\boldsymbol{A}$): $\boldsymbol{\theta} = g(\boldsymbol{A})$, and a network trained on all sites. A more detailed description of training such a neural network on soil moisture data to parameterize VIC is described in Supplementary Information S2. The initial effort of a surrogate model approach is low compared to fully recoding a model, but one may need to continuously retrain the surrogate models as the optimization goes to different regions of the parameter or state space. Furthermore, a surrogate model does not allow direct changes to the model structure. As a result, it is only recommended for highly complex and computationally expensive models that are challenging for reimplementation. Such cases may arise in climate models, hydraulics or subsurface modeling where governing PDEs of fluid dynamics and sediment transport must be solved with high spatial and temporal resolutions and require non-trivial computational codes. Solving PDEs using neural networks has attracted increasing attention, with many studies seeking to use NNs to approximate the numerical solution of those PDEs[141–143], such as for the Richards equation[144]. For another example, for 2D hydraulic simulations, a differentiable surrogate model can be employed for the inversion of bathymetry[145]. While not DG's philosophical theme, surrogate models can certainly accelerate and aid the mission of DG.

PINN method for learning parameters and constitutive relationships

In the physics-informed neural networks (PINN) method[109,146,147], parameterization schemes can be learned by modeling the space-dependent properties of a system (like hydraulic conductivity of porous media) and unknown constitutive relationships (like pressure-dependent permeability of the unsaturated porous media and strain-dependent effective viscosity of non-Newtonian fluids). To make the model fully differentiable, in the PINN method, the states of the system are also modeled with neural networks. Then, all neural networks are jointly trained using the system state measurements and the fundamental conservation law constraints added as penalty terms to the joint loss function. As a result, the PINN method allows for learning systems parameters and constitutive relationships using measurements of the system states that may be easier to collect than the direct measurements of the parameters. The latter would be needed for learning parameters using data only. An example of the application of PINN for learning the constitutive relationship in the unsaturated flow model is given in the Supplementary Information S2. As compared to previously mentioned DM methods, PINN has a unique design that directly learns the problem-dependent space-time solution of states, and thus needs to be trained for each boundary/initial conditions pair. Its focus is on knowledge discovery rather than being an efficient forward simulator.

ML-dominant hybrid models with limited physics.

Another class of models applicable in the data-rich realm employs NNs for the majority of modeling but inserting physical operators for imposing limited physics. For example, previous work used LSTM to estimate physical surface fluxes such as evaporation, runoff, and recharge, with only one constraint of mass balance equations[148]. Since the only supervising task for the fluxes was the observations of discharge, it was uncertain whether the terms maintained their physical meaning. The authors later constrained the system

using more observations[149] which improved the simulations and reduced equifinality. A model learning from two data sources outperformed those learning from only one source for the inference of soil moisture using LSTM trained at 9-km resolution, whose solutions were fed into an averaging operation to obtain outputs at 36-km, and loss functions were computed at both resolutions against in-situ and satellite-based observations[150]. Overall, ML-dominant systems can be strong predictors and a beneficial option in DG, but one needs to carefully assess the interpretability and physical significance of the diagnostic intermediate variables.

Summary and future directions

Throughout our examples, we demonstrated that the DM is a novel framework that allows varying amounts of structural priors to be flexibly employed along with NNs, ranging from having just a few physically-based operators to significant physically-based structures. As a result, with DM the divide between ML and PBM can be dissolved. Understanding the role of differentiable programming allows us to break free from thinking about fixed methods or approaches for their integration – instead focusing on physical priors, uncertainty, unknown relationships and data. DM is particular suited for many geoscientific domains, and the DM system can learn from multiple sources of data, multiscale datasets, and leverage the benefits of big or small data.

Memory usage and vanishing gradients are major issues in NN training, especially where iterative numerical solvers are involved. Keeping track of gradients requiring storing some information (partly alleviated if checkpointing is applied but still a prominent issue) and thus use of memory, which is especially constrained with GPUs. Vanishing gradient means that the parameters in deeper layers have very small gradients, so they become difficult to train[151,152]. Vanishing gradient can happen with recurrent NNs, which are similar to differentiable models. Moreover, differentiable models may have very heterogeneous operations (compared to NNs which are predominantly matrix multiplications) so how to maximizing the utilization of GPUs may pose a challenge. We anticipate new issues to emerge and new solutions to address them.

While numerical solvers for ordinary differentiable equations (ODEs) can be readily accommodated by current differentiable computing platforms, partial differential equations (PDEs) may still be challenging. This is first because solving PDEs requires substantial computation and memory, which makes training by a batch of examples expensive in terms of both compute and memory usage. The architecture suitable for big-data ML training tends to prefer massive parallelism, which reduces the range of suitable numerical algorithms. Differentiable modelers now need to understand both the forward and backward methods, adding to the mathematical learning curve. Nevertheless, some differentiable numerical solvers to PDEs have been proposed and tested in computational fluid mechanics, and appear to be alternative to standard solvers [153].

Since differentiable modeling allows us to learn processes, it is to be expected that we may run into "process non-uniqueness", also called "process equifinality". In traditional hydrologic modeling, "multiple working hypotheses" has been proposed to test different model formulations coupled together[33]. With DM, systematic development approaches can allow solving a part of the problem or determining one process at a time to reduce the interaction of modules. Second, more mature uncertainty quantification techniques are needed, such as going beyond ensemble methods[154–157], to help assess the success and failures of hypotheses. Finally, large and multivariate benchmarks and extrapolation tests are needed that match the intended use cases to verify the validity and realism of physical outputs. For example, models intended for climate change

impact assessment must be tested for long term projection fidelity; models for global-scale applications must pass rigorous spatial extrapolation tests[119].

The progress and emergence of the utilization of AI at large scales (of data and model) have been astonishing[88,158]. We here argue that both prediction accuracy and knowledge discovery in the geosciences can greatly benefit from leveraging advanced AI model architecture combined with physics using a differentiable programming framework. While perceived as a technological breakthrough, differentiable modeling can lead to philosophical changes – we can now ask new questions and test hypotheses on model structure or data usage, and therefore utilize data in new and more optimal ways. The power of DM offers a new pathway toward advances in geosciences.

Acknowledgements

We attribute many ideas of the paper to a discussion in the HydroML symposium, University Park, PA, May 2022, https://bit.ly/3g3DQNX, sponsored by National Science Foundation EAR #2015680 and Penn State Institute for Computational and Data Sciences is a Computational Research. Content related to this paper was also presented in some presentations, including AI4ESP talk online https://bit.ly/3etm5aI in Nov 2021. Shen was supported by National Science Foundation EAR-2221880. Song and Bindas were supported by Office of Science, US Department of Energy under award DE-SC0016605 and Cooperative Institute for Research to Operations in Hydrology (CIROH), award number A22-0307-S003. Gentine acknowledges funding from the National Science Foundational Science and Technology Center, Learning the Earth with Artificial intelligence and Physics (LEAP), award #2019625 and USMILE European Research Council grant. Marty Wernimont at USGS greatly improved the presentation of Figures 1 and 2; Wernimont and Appling were supported by the USGS Water Mission Area, Water Availability and Use Science Program. Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government. Li was supported by the National Science Foundation EAR-2121621 and Department of Energy Office of Science DE-SC0020146. CX acknowledges the support of Next Generation Ecosystem Experiment Project-Tropics sponsored by DOE Office of Science. Ren was supported by the National Science Foundation CBET award # 2045235.

Competing Interests

KL and CS have financial interests in HydroSapient, Inc., a company which could potentially benefit from the results of this research. This interest has been reviewed by the University in accordance with its Individual Conflict of Interest policy, for the purpose of maintaining the objectivity and the integrity of research at The Pennsylvania State University.

Disclaimer

Any use of trade, firm, or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Bibliography

1. Ajami, N. K., Gupta, H., Wagener, T. & Sorooshian, S. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. *Journal of Hydrology* **298**, 112–135 (2004).

2. van Griensven, A. & Meixner, T. A global and efficient multi-objective auto-calibration and uncertainty estimation method for water quality catchment models. *Journal of Hydroinformatics* **9**, 277–291 (2007).
3. Barendrecht, M. H. *et al.* The value of empirical data for estimating the parameters of a sociohydrological flood risk model. *Water Resour. Res.* **55**, 1312–1336 (2019).
4. Post, H., Vrugt, J. A., Fox, A., Vereecken, H. & Franssen, H.-J. H. Estimation of Community Land Model parameters for an improved assessment of net carbon fluxes at European sites. *Journal of Geophysical Research: Biogeosciences* **122**, 661–689 (2017).
5. Aumont, O., Ethé, C., Tagliabue, A., Bopp, L. & Gehlen, M. PISCES-v2: An ocean biogeochemical model for carbon and ecosystem studies. *Geoscientific Model Development* **8**, 2465–2513 (2015).
6. Ahmed, M. *et al.* Calibration and validation of APSIM-Wheat and CERES-Wheat for spring wheat under rainfed conditions: Models evaluation and application. *Computers and Electronics in Agriculture* **123**, 384–401 (2016).
7. Lepore, C., Arnone, E., Noto, L. V., Sivandran, G. & Bras, R. L. Physically based modeling of rainfall-triggered landslides: a case study in the Luquillo forest, Puerto Rico. *Hydrology and Earth System Sciences* **17**, 3371–3387 (2013).
8. Shirzaei, M. *et al.* Measuring, modelling and projecting coastal land subsidence. *Nat Rev Earth Environ* **2**, 40–58 (2021).
9. Biemans, H. *et al.* Importance of snow and glacier meltwater for agriculture on the Indo-Gangetic Plain. *Nat Sustain* **2**, 594–601 (2019).
10. (2021).
79. Fang, K., Kifer, D., Lawson, K., Feng, D. & Shen, C. The data synergy effects of time-series deep learning models in hydrology. *Water Resources Research* **58**, e2021WR029583 (2022).
80. McGovern, A., Ebert-Uphoff, I., Gagne, D. J. & Bostrom, A. Why we need to focus on developing ethical, responsible, and trustworthy artificial intelligence approaches for environmental science. *Environmental Data Science* **1**, e6 (2022).
81. Schölkopf, B. Causality for Machine Learning. in 765–804 (2022). doi:10.1145/3501714.3501755.

82. Bach, S. *et al.* On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE* **10**, e0130140 (2015).
83. Montavon, G., Samek, W. & Müller, K.-R. Methods for Interpreting and Understanding Deep Neural Networks. *Digital Signal Processing* (2017) doi:10/gcvxrb.
84. Toms, B. A., Barnes, E. A. & Ebert-Uphoff, I. Physically interpretable neural networks for the geosciences: Applications to earth system variability. *Journal of Advances in Modeling Earth Systems* **12**, e2019MS002002 (2020).
85. Fleming, S. W., Watson, J. R., Ellenson, A., Cannon, A. J. & Vesselinov, V. C. Machine learning in Earth and environmental science requires education and research policy reforms. *Nat. Geosci.* **14**, 878–880 (2021).
86. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Networks* **4**, 251–257 (1991).
87. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Networks* **2**, 359–366 (1989).
88. Bubeck, S. *et al.* Sparks of Artificial General Intelligence: Early experiments with GPT-4. Preprint at https://doi.org/10.48550/arXiv.2303.12712 (2023).
89. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**, 182–197 (2002).
90. Duan, Q., Sorooshian, S. & Gupta, V. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resources Research* **28**, 1015–1031 (1992).
91. Zitzler, E., Laumanns, M. & Thiele, L. *SPEA2: Improving the strength pareto evolutionary algorithm*. *TIK Report* vol. 103 https://www.research-collection.ethz.ch/handle/20.500.11850/145755 (2001).
92. Liu, S. *et al.* A hybrid approach of support vector regression with genetic algorithm optimization for aquaculture water quality prediction. *Mathematical and Computer Modelling* **58**, 458–465 (2013).
93. Zambrano-Bigiarini, M. & Rojas, R. A model-independent Particle Swarm Optimisation software for model calibration. *Environmental Modelling & Software* **43**, 5–25 (2013).

94. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research* **18**, 1–43 (2018).
95. Innes, M. *et al.* A Differentiable Programming System to Bridge Machine Learning and Scientific Computing. Preprint at http://arxiv.org/abs/1907.07587 (2019).
96. Goodfellow, I., Bengio, Y. & Courville, A. Numerical Computation - Gradient-Based Optimization. in *Deep Learning* 775 (The MIT Press, 2016).
97. Paszke, A. *et al.* Automatic differentiation in PyTorch. in *31st Conference on Neural Information Processing Systems (NIPS 2017)* (2017).
98. Bradbury, J. *et al.* JAX: Autograd and XLA. *Astrophysics Source Code Library* ascl:2111.002 (2021).
99. Bezanson, J., Karpinski, S., Shah, V. B. & Edelman, A. Julia: A fast dynamic language for technical computing. *ArXiv* **abs/1209.5145**, (2012).
100. Abadi, M. *et al.* Tensorflow: A system for large-scale machine learning. in *12th USENIX symposium on operating systems design and implementation (OSDI 16)* 265–283 (USENIX Association, 2016).
101. Errico, R. M. What Is an Adjoint Model? *Bulletin of the American Meteorological Society* **78**, 2577–2592 (1997).
102. Johnson, S. G. Notes on Adjoint Methods for 18.335. 7 (2021).
103. Pal, A., Edelman, A. & Rackauckas, C. Mixing Implicit and Explicit Deep Learning with Skip DEQs and Infinite Time Neural ODEs (Continuous DEQs). Preprint at https://doi.org/10.48550/arXiv.2201.12240 (2022).
104. Ghattas, O. & Willcox, K. Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica* **30**, 445–554 (2021).
105. Baker, N. *et al. Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence*. https://www.osti.gov/biblio/1478744 (2019) doi:10.2172/1478744.
106. Rackauckas, C. *et al.* Universal differential equations for scientific machine learning. Preprint at http://arxiv.org/abs/2001.04385 (2021).

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

108. Huang, D. Z., Xu, K., Farhat, C. & Darve, E. Learning constitutive relations from indirect observations using deep neural networks. *Journal of Computational Physics* **416**, 109491 (2020).

109. Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D. & Barajas-Solano, D. Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research* **56**, e2019WR026731 (2020).

110. Padarian, J., McBratney, A. B. & Minasny, B. Game theory interpretation of digital soil mapping convolutional neural networks. *SOIL* **6**, 389–397 (2020).

111. Udrescu, S.-M. & Tegmark, M. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances* **6**, eaay2631 (2020).

112. Ma, Y., Tsao, D. & Shum, H.-Y. On the principles of parsimony and self-consistency for the emergence of intelligence. Preprint at https://doi.org/10.48550/arXiv.2207.04630 (2022).

113. Tsai, W.-P. *et al.* From calibration to parameter learning: Harnessing the scaling effects of big data in geoscientific modeling. *Nat Commun* **12**, 5988 (2021).

114. Myneni, Ranga, Knyazikhin, Yuri, & Park, Taejin. MCD15A2H MODIS/Terra+Aqua Leaf Area Index/FPAR 8-day L4 Global 500m SIN Grid V006. (2015) doi:10.5067/MODIS/MCD15A2H.006.

115. ESA. About SMOS - Soil Moisture and Ocean Salinity mission. *European Space Agency (ESA)* https://earth.esa.int/eogateway/missions/smos (2022).

116. O'Neill, P. E. *et al.* SMAP Enhanced L3 Radiometer Global and Polar Grid Daily 9 km EASE-Grid Soil Moisture, Version 5 (SPL3SMP_E). (2021) doi:10.5067/4DQ54OUIJ9DL.

117. Mahecha, M. D. *et al.* Earth system data cubes unravel global multivariate dynamics. *Earth System Dynamics* **11**, 201–234 (2020).

118. Lin, Y.-S. *et al.* Optimal stomatal behaviour around the world. *Nature Climate Change* **5**, 459–464 (2015).

119. Feng, D., Lawson, K. & Shen, C. Mitigating prediction error of deep learning streamflow models in large data-sparse regions with ensemble modeling and soft data. *Geophysical Research Letters* **48**, e2021GL092999 (2021).

120. Feng, D., Beck, H., Lawson, K. & Shen, C. The suitability of differentiable, learnable hydrologic models for ungauged regions and climate change impact assessment. *Hydrology and Earth System Sciences Discussions* 1–28 (2022) doi:10.5194/hess-2022-245.

121. Wagener, T. *et al.* The future of hydrology: An evolving science for a changing world. *Water Resources Research* **46**, 1–10 (2010).

122. Liu, J., Hughes, D., Rahmani, F., Lawson, K. & Shen, C. Evaluating a global soil moisture dataset from a multitask model (GSM3 v1.0) with potential applications for crop threats. *Geoscientific Model Development* **16**, 1553–1567 (2023).

123. Jiang, S., Zheng, Y. & Solomatine, D. Improving AI system awareness of geoscience knowledge: Symbiotic integration of physical approaches and deep learning. *Geophysical Research Letters* **47**, e2020GL088229 (2020).

124. Beven, K. A manifesto for the equifinality thesis. *Journal of Hydrology* **320**, 18–36 (2006).

125. Pokhrel, P., Gupta, H. V. & Wagener, T. A spatial regularization approach to parameter estimation for a distributed watershed model. *Water Resour. Res.* **44**, (2008).

126. Wagener, T., McIntyre, N., Lees, M. J., Wheater, H. S. & Gupta, H. V. Towards reduced uncertainty in conceptual rainfall-runoff modelling: Dynamic identifiability analysis. *Hydrol. Process.* **17**, 455–476 (2003).

127. Onken, D. & Ruthotto, L. Discretize-Optimize vs. Optimize-Discretize for Time-Series Regression and Continuous Normalizing Flows. Preprint at https://doi.org/10.48550/arXiv.2005.13420 (2020).

128. Mitusch, S. K., Funke, S. W. & Kuchta, M. Hybrid FEM-NN models: Combining artificial neural networks with the finite element method. *Journal of Computational Physics* **446**, 110651 (2021).

129. Farrell, P. E., Ham, D. A., Funke, S. W. & Rognes, M. E. Automated derivation of the adjoint of high-level transient finite element programs. *SIAM J. Sci. Comput.* **35**, C369–C393 (2013).

130. Fisher, M. & Andersson, E. *Developments in 4D-Var and Kalman Filtering*. https://www.ecmwf.int/sites/default/files/elibrary/2001/9409-developments-4d-var-and-kalman-filtering.pdf (2001).

131. Neupauer, R. M. & Wilson, J. L. Adjoint-derived location and travel time probabilities for a multidimensional groundwater system. *Water Resources Research* **37**, 1657–1668 (2001).

132. Clark, M. P. & Kavetski, D. Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes. *Water Resources Research* **46**, (2010).

133. Kavetski, D. & Clark, M. P. Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction. *Water Resources Research* **46**, (2010).

134. Höge, M., Scheidegger, A., Baity-Jesi, M., Albert, C. & Fenicia, F. Improving hydrologic models for predictions and process understanding using neural ODEs. *Hydrology and Earth System Sciences* **26**, 5085–5102 (2022).

135. Aboelyazeed, D. *et al.* A differentiable ecosystem modeling framework for large-scale inverse problems: demonstration with photosynthesis simulations. *Biogeosciences Discussions* (2022) doi:10.5194/bg-2022-211.

136. Bao, T. *et al.* Partial Differential Equation Driven Dynamic Graph Networks for Predicting Stream Water Temperature. in *2021 IEEE International Conference on Data Mining (ICDM)* 11–20 (2021). doi:10.1109/ICDM51629.2021.00011.

137. Bindas, T. *et al.* Improving large-basin streamflow simulation using a modular, differentiable, learnable graph model for routing. Preprint at https://doi.org/10.1002/essoar.10512512.1 (2023).

138. Forghani, M. *et al.* Application of deep learning to large scale riverine flow velocity estimation. *Stoch Environ Res Risk Assess* **35**, 1069–1088 (2021).

139. Forghani, M. *et al.* Variational encoder geostatistical analysis (VEGAS) with an application to large scale riverine bathymetry. *Advances in Water Resources* **170**, 104323 (2022).

140. Asher, M. J., Croke, B. F. W., Jakeman, A. J. & Peeters, L. J. M. A review of surrogate models and their application to groundwater modeling. *Water Resources Research* **51**, 5957–5973 (2015).

141. Blechschmidt, J. & Ernst, O. G. Three ways to solve partial differential equations with neural networks — A review. *GAMM-Mitteilungen* **44**, e202100006 (2021).

142. Lu, L., Meng, X., Mao, Z. & Karniadakis, G. E. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* **63**, 208–228 (2021).

143. Takamoto, M. *et al.* PDEBENCH: An Extensive Benchmark for Scientific Machine Learning. Preprint at https://doi.org/10.48550/arXiv.2210.07182 (2022).

144. Maxwell, R. M., Condon, L. E. & Melchior, P. A physics-informed, machine learning emulator of a 2D surface water model: What temporal networks and simulation-based inference can help us learn about hydrologic processes. *Water* **13**, 3633 (2021).

145. Liu, X., Song, Y. & Shen, C. Bathymetry inversion using a deep-learning-based surrogate for shallow water equations solvers. Preprint at https://doi.org/10.48550/arXiv.2203.02821 (2022).

146. Raissi, M., Perdikaris, P. & Karniadakis, G. E. 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 (2019).

147. He, Q., Barajas-Solano, D., Tartakovsky, G. & Tartakovsky, A. M. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources* **141**, 103610 (2020).

148. Kraft, B., Jung, M., Körner, M. & Reichstein, M. Hybrid modeling: Fusion of a deep learning approach and a physics-based model for global hydrological modeling. in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* vol. XLIII-B2-2020 1537–1544 (Copernicus GmbH, 2020).

149. Kraft, B., Jung, M., Körner, M., Koirala, S. & Reichstein, M. Towards hybrid modeling of the global hydrological cycle. *Hydrology and Earth System Sciences* **26**, 1579–1614 (2022).

150. Liu, J., Rahmani, F., Lawson, K. & Shen, C. A multiscale deep learning model for soil moisture integrating satellite and in situ data. *Geophysical Research Letters* **49**, e2021GL096847 (2022).

151. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* **06**, 107–116 (1998).

152. Hochreiter, S., Bengio, Y., Frasconi, P., & Jürgen Schmidhuber. Gradient Flow in Recurrent Nets: The Difficulty of Learning Long-Term Dependencies. in *A Field Guide to Dynamical Recurrent Neural Networks* (eds. Kremer, S. C. & Kolen, J. F.) 237–244 (IEEE Press, 2001).

153. Kochkov, D. *et al.* Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences* **118**, e2101784118 (2021).

154. Fang, K., Kifer, D., Lawson, K. & Shen, C. Evaluating the potential and challenges of an uncertainty quantification method for long short-term memory models for soil moisture predictions. *Water Resources Research* **56**, e2020WR028095 (2020).

155. Li, D., Marshall, L., Liang, Z., Sharma, A. & Zhou, Y. Bayesian LSTM with stochastic variational inference for estimating model uncertainty in process-based hydrological models. *Water Resources Research* **57**, e2021WR029772 (2021).

156. Tabas, S. S. & Samadi, S. Variational Bayesian dropout with a Gaussian prior for recurrent neural networks application in rainfall–runoff modeling. *Environ. Res. Lett.* **17**, 065012 (2022).

157. Krapu, C. & Borsuk, M. A differentiable hydrology approach for modeling with time-varying parameters. *Water Resources Research* **58**, e2021WR031377 (2022).

158. Brown, T. B. *et al.* Language Models are Few-Shot Learners. Preprint at https://doi.org/10.48550/arXiv.2005.14165 (2020).

159. Fleming, S. W., Garen, D. C., Goodbody, A. G., McCarthy, C. S. & Landers, L. C. Assessing the new Natural Resources Conservation Service water supply forecast model for the American West: A challenging test of explainable, automated, ensemble artificial intelligence. *Journal of Hydrology* **602**, 126782 (2021).

160. Li, L. *et al.* Developing machine learning models with multi-source environmental data to predict wheat yield in China. *Comput. Electron. Agric.* **194**, (2022).

161. Paudel, D. *et al.* Machine learning for regional crop yield forecasting in Europe. *Field Crops Research* **276**, 108377 (2022).

162. Shahhosseini, M., Hu, G., Huber, I. & Archontoulis, S. V. Coupling machine learning and crop modeling improves crop yield prediction in the US Corn Belt. *Sci Rep* **11**, 1606 (2021).

163. Chen, S., Zwart, J. A. & Jia, X. Physics-guided graph meta learning for predicting water temperature and streamflow in stream networks. in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* 2752–2761 (Association for Computing Machinery, 2022). doi:10.1145/3534678.3539115.

164. Rahmani, F. *et al.* Data Release: Deep learning approaches for improving prediction of daily stream temperature in data-scarce, unmonitored, and dammed basins: U.S. Geological Survey data release. *U.S. Geological Survey* https://doi.org/10.5066/P9VHMO56 (2021).

165. Daraio, J. A., Bales, J. D. & Pandolfo, T. J. Effects of land use and climate change on stream temperature II: Threshold exceedance duration projections for freshwater mussels. *JAWRA Journal of the American Water Resources Association* **50**, 1177–1190 (2014).

166. van Vliet, M. T. H. *et al.* Coupled daily streamflow and water temperature modelling in large river basins. *Hydrol. Earth Syst. Sci.* **16**, 4303–4321 (2012).

167. He, X. *et al.* Improving predictions of evapotranspiration by integrating multi-source observations and land surface model. *Agricultural Water Management* **272**, 107827 (2022).

168. Talib, A. *et al.* Evaluation of prediction and forecasting models for evapotranspiration of agricultural lands in the Midwest U.S. *Journal of Hydrology* **600**, 126579 (2021).

169. Seibert, J., Vis, M. J. P., Lewis, E. & Meerveld, H. J. van. Upper and lower benchmarks in hydrological modelling. *Hydrological Processes* **32**, 1120–1125 (2018).

170. Mohamoud, Y. M. & Parmar, R. S. Estimating streamflow and associated hydraulic geometry, the Mid-Atlantic Region, USA. *JAWRA Journal of the American Water Resources Association* **42**, 755–768 (2006).

171. Merritt, A. M., Lane, B. & Hawkins, C. P. Classification and prediction of natural streamflow regimes in arid regions of the USA. *Water* **13**, (2021).

172. Stefan, H. G. & Fang, X. Dissolved oxygen model for regional lake analysis. *Ecological Modelling* **71**, 37–68 (1994).

173. Heddam, S. Simultaneous modelling and forecasting of hourly dissolved oxygen concentration (DO) using radial basis function neural network (RBFNN) based approach: a case study from the Klamath River, Oregon, USA. *Modeling Earth Systems and Environment* **2**, 135 (2016).

174. Keshtegar, B. & Heddam, S. Modeling daily dissolved oxygen concentration using modified response surface method and artificial neural network: a comparative study. *Neural Computing and Applications* **30**, 2995–3006 (2018).

175. Haber, E. & Ruthotto, L. Stable architectures for deep neural networks. *Inverse Problems* **34**, 014004 (2018).

176. Chen, R. T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. Neural ordinary differential equations. in *Proceedings of the 32nd International Conference on Neural Information Processing Systems* 6572–6583 (Curran Associates Inc., 2018).

177. Shen, C. Deep learning: A next-generation big-data approach for hydrology. *Eos* vol. 99 (2018).

178. Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nat Rev Phys* **3**, 422–440 (2021).

179. Karpatne, A. *et al.* Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering* **29**, 2318–2331 (2017).

180. Khandelwal, A. *et al.* Physics guided machine learning methods for hydrology. Preprint at https://doi.org/10.48550/arXiv.2012.02854 (2020).

181. Pawar, S., San, O., Aksoylu, B., Rasheed, A. & Kvamsdal, T. Physics guided machine learning using simplified theories. *Physics of Fluids* **33**, 011701 (2021).

182. Bennett, A. & Nijssen, B. Deep learned process parameterizations provide better representations of turbulent heat fluxes in hydrologic models. *Water Resources Research* **57**, e2020WR029328 (2021).

183. Schaap, M. G., Leij, F. J. & van Genuchten, M. Th. Rosetta: a Computer Program for Estimating Soil Hydraulic Parameters With Hierarchical Pedotransfer Functions. *Journal of Hydrology* **251**, 163–176 (2001).

184. Rasp, S., Pritchard, M. S. & Gentine, P. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences of the United States of America* **115**, 9684–9689 (2018).

185. Zhu, Y. *et al.* Physics-informed deep-learning parameterization of ocean vertical mixing improves climate simulations. *National Science Review* **9**, nwac044 (2022).

186. Koppa, A., Rains, D., Hulsman, P., Poyatos, R. & Miralles, D. G. A deep learning-based hybrid model of global terrestrial evaporation. *Nat Commun* **13**, 1912 (2022).

187. Liu, B. *et al.* Physics-guided long short-term memory network for streamflow and flood simulations in the Lancang–Mekong river basin. *Water* **14**, 1429 (2022).

188. Frame, J. M. *et al.* Post-Processing the National Water Model with Long Short-Term Memory Networks for Streamflow Predictions and Model Diagnostics. *JAWRA Journal of the American Water Resources Association* **57**, 885–905 (2021).

189. Sun, A. Y., Jiang, P., Yang, Z.-L., Xie, Y. & Chen, X. A graph neural network approach to basin-scale river network learning: The role of physics-based connectivity and data fusion. *Hydrology and Earth System Sciences Discussions* (2022) doi:10.5194/hess-2022-111.

190. Reichstein, M. *et al.* Deep learning and process understanding for data-driven Earth system science. *Nature* **566**, 195–204 (2019).

191. Wang, N., Zhang, D., Chang, H. & Li, H. Deep learning of subsurface flow via theory-guided neural network. *Journal of Hydrology* **584**, 124700 (2020).

Supplementary Information

S1. Supplementary Discussion

A. Recent progress in geoscientific domains from purely data-driven machine learning.

ML has gradually but pervasively permeated the vast majority of scientific disciplines, and is transforming those sciences at an unprecedented pace. In hydrology, deep networks such as long short-term memory (LSTM) networks[72], and convolutional neural networks (CNNs)[75,76] have shown strong ability with regard to prediction of soil moisture[60–62], water supply[159], streamflow[63–66], evapotranspiration[67–69], groundwater levels[70], snow[71], and other aspects of the water cycle[59]. In water quality studies, LSTMs and CNNs have shown promise in simulating water temperature[50–53], dissolved oxygen[54], phosphorous[56], and nitrogen[57,58], among others[48,49]. In agriculture, ML approaches have been widely applied for crop production prediction[160–162]. In regional climate studies, CNN-based schemes or generative algorithms have been found to improve the forecasting of precipitation fields[45,46] and prediction of clouds (deep clouds)[47]. Often the studies have reported state-of-the-art performance when compared with conventional approaches. Typically, such high-quality predictions can be made even when a good understanding of the underlying processes is not available. We made an effort to collect a list of somewhat comparable studies with metrics for both traditional and ML models (Figure S3 and Table S1). Previous models have been highly useful in advancing science, but these results imply that they were not fully exploiting the information available in the data[29], and they can benefit from leveraging the strength of ML.

Table S1. *ML vs. traditional model performances for a number of scientific applications with data from many sites. The metrics were computed based on simulations and observations. The lower the values, the better for RMSE, while higher is better for Pearson's correlation (COR), R^2, and Nash-Sutcliffe model efficiency coefficient (NSE). This is presented with many caveats, such as the ML model is optimized to match observations while traditional models have many other constraints; a selection bias – where ML did not outperform did not get published (nevertheless, one could also argue studies where PBM outperformed were not easily found). The point of this table was not to show that ML was always better, but to support the argument that ML tends to have advantages in accuracy. Also note the limitations of ML discussed in the Introduction.*

Variable	**Metric**	**Deep networks**	**Traditional**	**Reference**
Stream Temperature	RMSE (°C)	1.91	4.01	Chen et al.[163]
	RMSE (°C)	0.89	1.80	Rahmani et al.[164] and Daraio et al.[165]
	Pearson COR	0.99	0.91	Rahmani et al.[164] and van Vliet et al.[166]
	R_2	0.942	0.93	Rahmani et al.[164]
	NSE	0.98	0.93	Rahmani et al.[164]
Evapotranspiration	R^2	0.67	0.21	He et al.[167]
	RMSE (mm/day)	1.21	2.56	
	NSE	0.65	0.57	Talib et al.[168]
Soil Moisture	RMSE	0.027	0.085	Fang et al.[60]
	Pearson COR	0.87	0.72	
	RMSE	0.027	0.035	
	Pearson COR	0.87	0.82	
	Pearson COR	0.91	0.77	Liu et al.[150]
	RMSE	0.034	0.08	
Streamflow	NSE	0.76	0.68	Seibert et al.[169] and Kratzert et al.[64]
	NSE	0.9 /0.68	-	Mohamoud and Parmar[170]
	Mean R^2	0.71	-	Merritt et al.[171]
	NSE	0.78	-	Zhi et al.[171]
Dissolved oxygen	Median R^2	-	0.64	Stefan and Fang[172]
	CC (correlation Coefficient)	0.972	-	Heddam[173]
	Median NSE	0.760	-	Keshtegar and Heddam[174]

B. *Why can differentiable models (DMs) achieve state-of-the-art predictive performance?*

Purely data-driven ML architectures have set a high bar for accuracy in multiple geoscience domains, such that one would be tempted to predict a substantial loss in accuracy when adding in less-flexible process-based components. However, it is still uncertain whether generic ML architectures are necessarily needed to achieve good model accuracy. As long as some model components are adaptable and learnable, we can learn from data. If we view the model as a more strongly constrained ML model (perspective "A" in Figure 2a), it is easy to see that there is a potential to achieve ML-level performance if the searchable space of PBM is enlarged to include a good approximation of the true function, directed by gradient-based training. The paths taken to upgrade the models will be expert-dependent (prior-dependent), so one should not expect a unified approach at present.

Many dynamical systems in Geosciences can be written as ordinary differential equations (ODEs), such as rainfall runoff in a basin, crop growth, or nutrient release. While solving these equations, the numerical model is run for many steps. This is mathematically similar to recurrent neural networks, and the time integration operation is similar to the functionality achieved by some neural networks like the Residual Networks[175,176]. It should not be surprising that learnable process-based models with some ML components can perform as well as deep networks.

As discussed in Section S1, multiple studies have already shown that differentiable, learnable models can approach the performance of purely data-driven models, or exhibit advantages in some cases where extrapolation is key. Differentiable model formulations can maintain at least two of the three desirable features: approximating complex, previously unknown functions, and the ability to assimilate information from big data. Compared to purely data-driven ML, DM trades genericity for interpretability and the ability to ask specific questions. Deep networks like CNNs, LSTMs, and attention layers will be an ingrained part of differentiable modeling in geosciences. Eventually, deep learning will become part of the repertoire of geoscientists, just like with numerical methods[177].

C. *How is DM related to physics-guided machine learning (PGML) and how are they different?*

Many ML-physics integration strategies with a wide variety of complexity have been proposed in the past in a seemingly scattered manner, such that a clear classification is difficult[178]. It has not been sufficiently recognized that some of these algorithms work fundamentally because they leveraged the differentiable programming tools. The scattered nature of those publications makes the landscape of ML-physics integration daunting and confusing, while hindering us from making innovations based on first principles. However, the concept of "differentiability" can serve as a compass to guide us in understanding newly proposed methods. We can ask if a method is fully (end-to-end) differentiable, how it uses gradients, how much prior information is inserted, what questions are asked, and how it scales with data. Here we outline some similarities and differences between DG and some existing methods.

DM and physics-guided (or physics-informed, theory-guided, or knowledge-guided) machine learning (PGML)[179–181] both seek to combine physics with ML, but they differ in their approaches, purposes, and philosophies. Many PGML studies seek to introduce physical constraints, for example, as regularization or pretraining, to ML methods to gain better generalizability with less training data. PGML does not in theory need differentiable programming and partial physics could be enforced. In contrast, DM is more thorough in that it uses the numerical physical model as the backbone and demands that the entire workflow be differentiable. In terms of purposes, PGML is tasked to make the ML model more robust, while differentiable modeling seeks to update our assumptions or discover new knowledge. Relatedly, in terms of philosophies, when a physical law was introduced in PGML, it was treated as truth (albeit sometimes with some tolerance level[53]). Often, this includes all the calculations and assumptions to

support the law. In DM, we do not presume the physical laws to be correct, and, rather, are constantly looking for opportunities to update existing knowledge.

There are many not-fully-differentiable methods that could be valuable for various applications but are outside of the scope of DM for this paper. For one, it is possible to incorporate ML algorithms trained offline on datasets as part of a physical model, like training a neural network on turbulent heat fluxes and inserting into a hydrologic model[182]; training pedotransfer functions to infer soil parameters from soil hydraulic data[183]; training an atmospheric parameterization network on short-term cloud-resolving simulations[184]; or training ocean-mixing parameterizations on data and physical constraints[185]. While this approach has the advantage that the physical meaning of the NN is clear and stands alone, direct training data are needed for the variable of interest (thus having issues with pure ML as discussed in the main text) and the network can no longer evolve and adapt in an interactive fashion, for instance to further update the model when exposed to observations. In the future these NNs could be further incorporated into DG models. Some other offline coupling methods include providing outputs of process-based models as inputs to neural networks (this helps to integrate over spatiotemporal heterogeneity)[186,187], or training ML models to predict the PBM residuals[155,188,189]. Readers are referred to Reichstein et al.[190] which promoted a number of ways to connect physics and ML for geosciences, with a brief mention of differentiable programming.

S2. Details for some examples.

Example 1. Part of the effort in Tsai et al.[113], which proposed differentiable parameter learning (dPL), connected the Variable Infiltration Capacity (VIC) process-based hydrologic model to a neural network (g) that estimates physical parameters of VIC ($\boldsymbol{\theta}$) using some widely available attributes ($\boldsymbol{A}$): $\boldsymbol{\theta} = g(\boldsymbol{A})$. In an "end-to-end" workflow, $\boldsymbol{\theta}$ is then sent to VIC, whose outputs are compared with observations, effectively turning the parameter calibration problem into a machine learning problem, trained on all sites simultaneously using backpropagation and gradient descent (Figure S1a). As a result of this global loss function, dPL exhibits advantages over traditional calibration on multiple fronts, for three different datasets (soil moisture, CAMELS streamflow, and global headwater runoff). The parameter sets are spatially coherent (Figure S1b-c) and extrapolate better in space (Figure S1d-e). dPL is hyper efficient: a job that normally takes a 100-CPU cluster 2-3 days now takes a single Graphical Processing Unit (GPU) one hour. dPL allows the combined model to output unobserved variables while alleviating the notorious problem of parameter equifinality[124]. As summarized earlier, these are the great advantages we expect to harness with differentiable modeling.

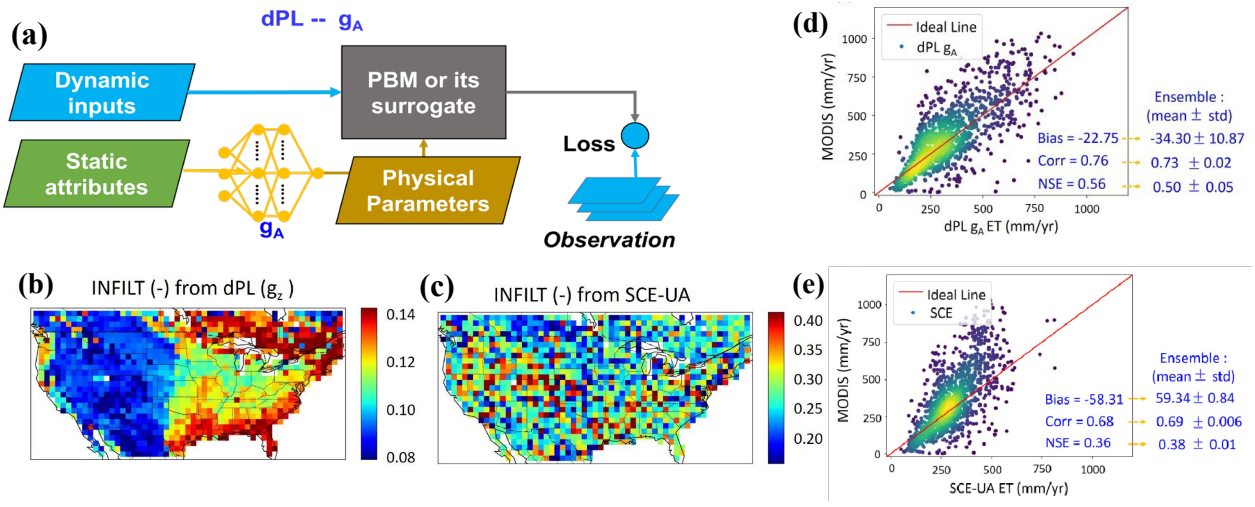

Figure S1. (From Tsai et al.[113], reprint allowed via Creative Commons Attribution 4.0 International License, http://creativecommons.org/licenses/by/4.0/) (a) Structural diagram of one of the dPL frameworks called g_A; (b & c) The estimated infiltrating curve parameter (INFILT) from dPL vs. the siteby-site calibrated shuffled complex evolutionary algorithm (SCE-UA);(d & e) dPL better matches the MODIS satellite product for uncalibrated variable ET than does SCE-UA.

Example 2. Physics-informed neural networks (PINNs)[146,147], while first published in 2017, could be perceived as a genre of DG as the gradient information is critically employed. PINNs pose problems in a unique way, seeking to train a neural network with space-time coordinates as inputs, *h(t,x)* where *x* represents spatial coordinates and *t* is time such that (i) *h(t,x)* agrees with known data points at *(t,x)*, and (ii) the derivatives *dh/dx*, *dh/dt,* etc. agree with the governing partial differential equations. Physical parameters could also be part of the inputs to the *h* network[191]. PINNs are a highly innovative approach tested on a large variety of applications in many domains, and there have been a number of good reviews of this work[142,178]. PINNs have made enormous strides, with novel inversion uses such data assimilation[147] and learning governing equations, but, as with other methods, there are also some limitations. Obviously, the function *h(t,x)* is tied to the initial and boundary conditions so it needs to be trained separately for each initial/boundary condition pair, and the form of the inputs limits the neural network to certain types (multilayer perceptron network) that are not the easiest to train. However, the learned parameters and constitutive relationships can describe the system under a wide range of boundary and initial conditions. Furthermore, the fidelity of the trained network to physical equations must be carefully examined.

In geosciences, a PINN method for learning unknown parameter fields and constitutive relationships was proposed[109] (Figure S2). As an example, steady-state groundwater flow in an aquifer with an unknown conductivity field and unsaturated flow in the vadose zone with an unknown pressuredependent conductivity were considered. In the unsaturated flow application, it was assumed that only sparse measurements of pressure head were available. The quantities of interest were the unsaturated conductivity as a function of the pressure head, and the pressure head field. Notably, it was assumed that no measurements of the unknown parameters were available. In the proposed PINN method, both quantities of interest were represented with neural networks (NNs) (with unknown parameters). This step created a differentiable model of the unsaturated flow in the vadose zone. It was also assumed that the pressure head measurements could be described by the steady-state Richards equation. Substituting the NN approximations into this equation formed the axillary residual NN, which shared the (unknown) parameters with the primary NNs. For the primary NNs to satisfy the governing equation, the residual NN should be zero everywhere in the domain – in other words, the exact measurements of the residuals are available everywhere in the domain. The NNs were trained jointly using the pressure head measurements. Since the conductivity and residual NNs share the same parameters, estimating parameters in the residual NN also provides the parameterization of the conductivity NN. Figure S2a shows the reference pressure head field and the locations of the measurements. Figure S2b shows the point errors in the estimated pressure head field. The reference and estimated unsaturated conductivity functions are shown in Figure S2c. These figures demonstrate that the PINN method can learn both the state variable and the constitutive relationship very accurately.

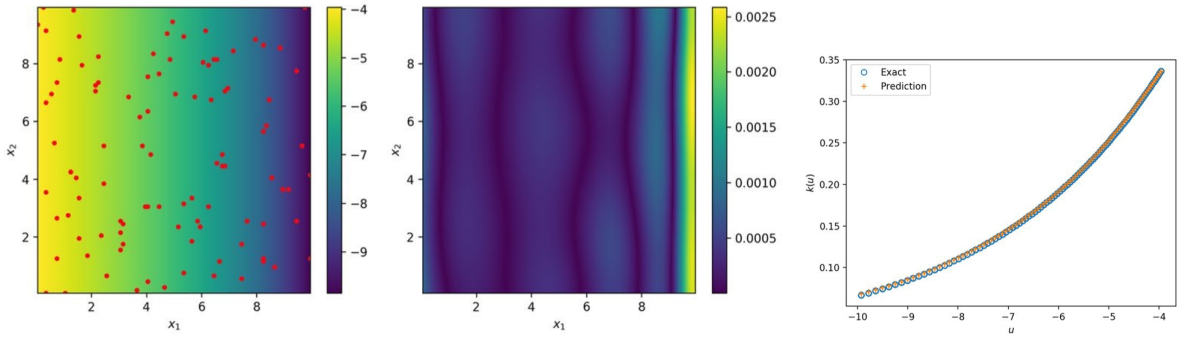

Figure S2. (from Tartakovsky et al.[109], reprint permission obtained) (a) The reference pressure head field and the locations of the measurements. (b) The point errors in the estimated head field. (c) The reference and estimated conductivities as functions of the pressure head.