nature reviews earth & environment

https://doi.org/10.1038/s43017-023-00450-9

Perspective

Check for updates

Differentiable modelling to unify machine learning and physical models for geosciences

A list of authors and their affiliations appears at the end of the paper

Abstract

Process-based modelling offers interpretability and physical consistency in many domains of geosciences but struggles to leverage large datasets efficiently. Machine-learning methods, especially deep networks, have strong predictive skills yet are unable to answer specific scientific questions. In this Perspective, we explore differentiable modelling as a pathway to dissolve the perceived barrier between process-based modelling and machine learning in the geosciences and demonstrate its potential with examples from hydrological modelling. 'Differentiable' refers to accurately and efficiently calculating gradients with respect to model variables or parameters, enabling the discovery of high-dimensional unknown relationships. Differentiable modelling involves connecting (flexible amounts of) prior physical knowledge to neural networks, pushing the boundary of physics-informed machine learning. It offers better interpretability, generalizability, and extrapolation capabilities than purely data-driven machine learning, achieving a similar level of accuracy while requiring less training data. Additionally, the performance and efficiency of differentiable models scale well with increasing data volumes. Under data-scarce scenarios, differentiable models have outperformed machine-learning models in producing short-term dynamics and decadal-scale trends owing to the imposed physical constraints. Differentiable modelling approaches are primed to enable geoscientists to ask questions, test hypotheses, and discover unrecognized physical relationships. Future work should address computational challenges, reduce uncertainty, and verify the physical significance of outputs.

01 mingxi zhang

02 mingxi zhang

Sections

e-mail: cshen@engr.psu.edu

Introduction

Geoscientific models encompass a wide range of domains, with evolving scopes and ever-increasing societal importance, especially in the face of climate change. For example, hydrological models can be used to help manage water resources[1,2] and plan for extremes such as floods and droughts[3]. Vegetation models can predict the impacts of future climate changes on the carbon cycle and other key biogeochemical cycles on land[4] or in the ocean[5]. Agricultural models can estimate crop yields and environmental impacts[6]. Geophysical models aim to predict land-surface changes caused by processes such as landslides[7] and subsidence[8]; the impact of future warming on glacial melt[9]; and the occurrence of earthquakes. Biogeochemical reactive transport models are used to understand and predict changes in surface and subsurface water chemistry and quality[10–12]. Earth system models[13–15] and integrated assessment models[16–18] combine many of these model types to provide crucial climate projections and guidance for resource managers and policy makers[19,20].

Geoscientific models describe the temporally dynamic responses of systems to time-dependent forcings, which are modulated by static landscape attributes; as such, the different model types often have features in common with one another. Many geoscientific models can describe multiple processes and are formulated as systems of nonlinear equations, ordinary differential equations (ODEs), or partial differential equations (PDEs). Some geoscientific processes are well understood whereas others are only assumed or empirically represented. Many such models extensively use parameterizations, where parameters either represent the processes too small for the computational grid, as in climate modelling, or modulate model behaviour based on landscape or vegetation characteristics, as in land-surface and hydrological modelling[21,22]. However, process representations and parameterizations are often subject to considerable uncertainty, some of which is due to the coarse scale of the models or data noise.

The rapid growth of machine learning (ML) since the 2010s offers new opportunities to learn from big data and fill knowledge gaps in geoscientific models. Although various forms of physics-informed ML have been proposed, there has been a lack of recognition of one core strength of ML — differentiable programming. 'Differentiable' refers to the ability to accurately and efficiently calculate gradients with respect to model variables or parameters, enabling the discovery of high-dimensional relationships. Understanding the potential of differentiable programming and its limitations will show a clear path towards combining the strengths of ML and physical models.

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 components from process-based models (PBMs) and ML-based models. We discuss the benefits and problems with traditional PBMs and ML. By contrasting them, we identify core strengths of ML and discuss the possibility of combining the strengths of both while mitigating their limitations. Then we formally introduce differentiable modelling (DM) as a new genre of modelling. We describe various classes of DM and give examples in geosciences to demonstrate its promise.

PBMs and ML in the geosciences

PBMs and purely data-driven ML are two valuable approaches for modelling geoscientific systems; however, each has limitations. There are various similarities and differences between the two models, which shows that their advantages are not mutually exclusive (Box 1).

Process-based models

Traditional PBMs use mathematical equations to describe physical processes and are deductively derived from established physical laws or empirical relationships[23,24]. They are used to understand system functions and behaviours, test hypotheses, and assess the response of a system to changes in the driving forces or properties. Further, they can simulate a wide range of observed variables (such as volumetric streamflow or leaf area index) and unobserved variables (for example, groundwater recharge or fine-root distribution). Such abilities are critical to advancing scientific understanding and providing a narrative when communicating with the public and stakeholders who are engaged in decision making[25]. With PBMs, it is possible to ask specific questions regarding processes within the modelled system such as how land-cover change affects water and carbon cycles, by progressively improving the representations of processes[23,26–28] and evaluating the results using controlled experiments.

However, PBMs have some important limitations. Notably, often PBMs cannot rapidly evolve with and fully exploit information from big data owing 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 add substantial uncertainty[31]. For model errors beyond parameter tuning, potential causes of the differences (for example, missing processes in the governing equation) must be hypothesized and structural changes implemented; then the updated model structure and underlying hypotheses are confronted with validation data[23]. This iterative process is very expensive (in both labour and time) and complex, and can be biased by the knowledge background of the modeller[32]. Consequently, the structural representation of a specific process in a geoscientific model can often stagnate for years or decades[33–36], meaning no new knowledge is gained and prediction performance is not improved.

Knowledge gaps further compound PBM stagnation. Extensive physical, biological, and socioeconomic knowledge is required to adequately define model structures, and any deficiencies can amplify errors and ambiguity. Another major challenge is accounting for process interactions that occur across disciplinary boundaries[37]. For instance, vegetation, microbes, human management, and socioeconomic systems all interact with each other and affect water, carbon, and other biogeochemical cycles[38–41]. Interdisciplinary research is highly valuable but challenging; therefore, there is a lack of data on these cross-disciplinary processes, which limits progress towards obtaining accurate model predictions.

Machine-learning-based models

Data-driven ML approaches, especially deep neural networks (NNs), have rapidly permeated the vast majority of scientific disciplines and are transforming those disciplines at an unprecedented pace[37,42]. NNs have highly generic model structures and many parameters that are determined from training on data. ML has been applied to a wide range of scientific applications, and deep networks like long short-term memory (LSTM) networks[43], transformers[44,45], graph neural networks[46], and convolutional neural networks (CNNs)[47,48] have become widely known. In the geosciences, NNs have shown promise in predicting crop production[49,50], precipitation fields[51,52] and clouds[53], water quality variables[54,55] such as water temperature[56–59], dissolved oxygen[60,61], phosphorus[62], and nitrogen[63,64], and the full hydrologic cycle[65] including soil moisture[66–68], streamflow[46,69–71], evapotranspiration[72–74], groundwater levels[75], and snow[76]. Often state-of-the-art performance was

Box 1

Comparing purely data-driven neural networks and process-based models

Similarities

Mathematical form: Purely data-driven neural networks (NNs) and purely process-based models (PBMs) have similar mathematical forms. NNs are described by

$$y = g^{W}(u, x, A) \quad (8)$$

where y is the simulation output, x is the dynamic forcings, u is the state variables, A is the semi-static attributes and W=argmin($L(y,y^*)$), which describes the weights of the neural network, g, where L is the loss function, which quantifies the difference between simulation outputs y and observations y^*.

PBMs have the form

$$y = f^{\theta}(u, x, A) \quad (9)$$

where θ=argmin($L(y,y^*)$), which describes the physical parameters of the PBM, f.

Programmatically differentiable: Purely data-driven NNs are programmatically differentiable, and although conventional PBMs are not programmatically differentiable they can be reimplemented in machine-learning (ML) platforms.

Differences

Training and calibration: NNs can be trained using data-driven training methods such as gradient descent, with gradient computations supported by differentiable programming, whereas PBMs are typically calibrated at limited numbers of sites or for a limited number of parameters, although efficient many-site, multi-objective methods exist.

Architecture: NNs have generic structures with many weights that allow the model to flexibly learn a wide range of functions. PBMs use physically based equations (structural priors) representing human understanding of physics, with a limited number of parameters.

Data: NNs are capable of efficiently gaining accuracy and generalizability as data grow, with beneficial scaling for big data. By contrast, PBM learning saturates at small quantities of data, although they can often make reasonable predictions despite limitations in data accuracy, resolution, and availability.

Unknown processes: NNs can discover patterns and functions from data that might be unknown or uncertain, whereas PBMs all processes must be explicitly specified by the modeller, even if they are only assumptions.

Domain knowledge: The generic model architecture of NNs makes them easy to develop even without domain expertise, and they can accommodate large knowledge gaps. PBMs require specialized domain knowledge.

Physical laws: NNs are not guaranteed to respect physical laws, unlike PBMs, which always respect physical laws.

Extraction: NNs only output trained variables, whereas PBMs provide access to many intermediate variables that aid interpretability.

Interpretation: NNs require much effort to interpret, and internal variables are not guaranteed to have physical meaning. PBMs contain equations representing physical processes, allowing narration of model reasoning and formal tests of alternative representations.

Education: NNs are taught in computer science or data science curricula, whereas PBMs are taught in engineering or science curricula.

reported when compared with conventional approaches, and such high-quality predictions can be made even when a good understanding of the underlying processes is not available. These results imply that previous models, despite their usefulness, were not fully exploiting the information available in the data[29], and they can benefit from leveraging the strength of ML (Supplementary Table 1).

Nevertheless, purely data-driven ML approaches have important limitations. First, NNs typically requires large volumes of data, which unfortunately are not often available in many geoscientific applications[64,77], where variables are only measured at 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 that critically affect human activities, such as floods, droughts, and earthquakes, available data are even scarcer.

Second, NNs are not exempt from deficiencies and can struggle with data errors, incompleteness, out-of-sample or out-of-distribution predictions, and 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[59,79,80]. Purely data-driven ML models can, at best, nearly perfectly replicate patterns in the training data; therefore, they invariably inherit 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 owing to the short data record.

Third, ML algorithms are based on correlations and not causal-regarding both attributes and temporal changes. There are often confounding factors in data, meaning that ML models can produce the

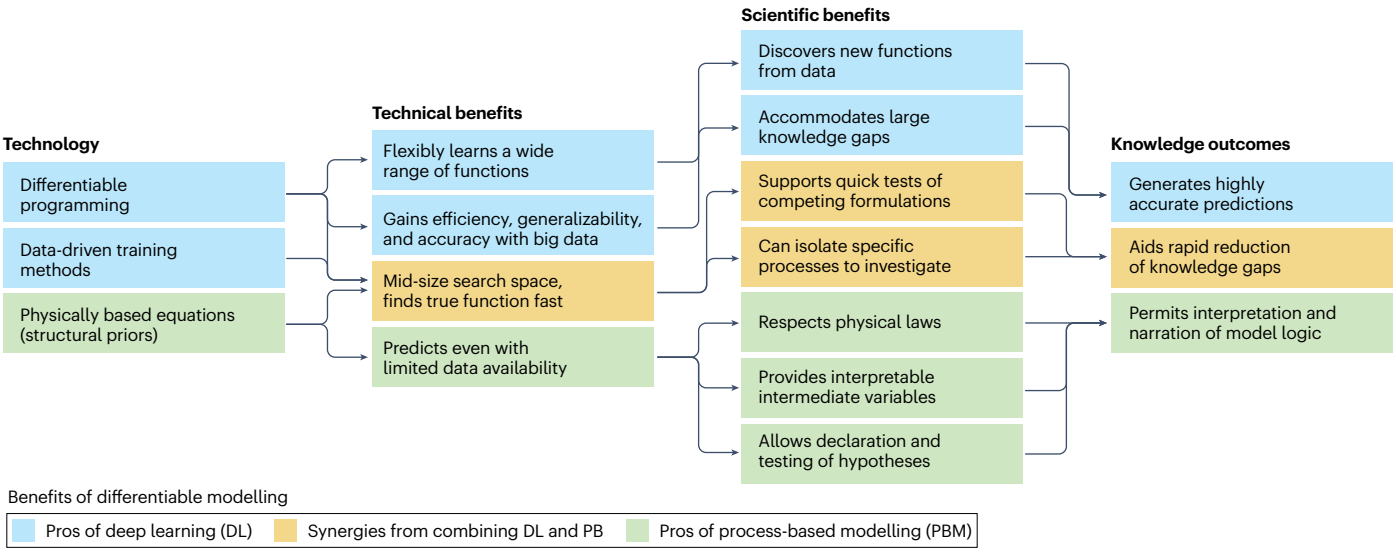


Fig. 1 | Synergies from combining machine learning and process-based modelling. Machine learning (ML, blue boxes) produces accurate results with easy-to-use models, resulting from the complexity of neural networks and the technologies that make it feasible to train such complex models. The most fundamental of these technologies is differentiable programming. Process-based models (PBMs, green boxes) permit human definition and interpretation of model logic. With differentiable modelling (DM), which incorporates differentiable non-ML model components from PBMs such as physically based structural priors, additional features can be obtained (orange boxes) while retaining and augmenting the advantages of both ML models and PBMs.

14-15 2 notes: results for the wrong (causal) reasons, potentially making predictions under different circumstances or outside the training domain less reliable. Although causal representation learning[81] and explainable artificial intelligence (AI) methods[82–84] are promising approaches for overcoming this limitation, challenges still remain with learning causality and interpretability.

Finally, purely data-driven ML models cannot predict untrained variables (those not provided as training targets) because ML-based models are inherently designed to only output the training targets. 16 mingxi zhang Therefore, it is difficult for ML approaches to elucidate how events unfolded. For example, if soil moisture is unobserved, pure ML models cannot state whether a flood occurred because the soil was saturated. 17-18 2 notes: Therefore, it is difficult to use ML for hypotheses formation and communicating with stakeholders.

In summary, ML alone is unlikely to satisfy geoscience modelling needs or answer specific scientific questions. Methods that can flexibly interrogate an ML model, encode causality and prior information, and identify missing physics anywhere in the model chain could be valuable. 19 mingxi zhang

Differentiable programming

Having considered the successes and limitations of NNs, it is important to identify their foundational strengths and work to overcome their limitations. In this section, we explain how differentiable programming is a computing framework that supports the efficient training of NNs and how, when generalized, it could deliver many philosophically and practically transformative outcomes to geoscientific modelling.

Explaining the success of machine learning

Traditional process-based, statistical, or hybrid modelling approaches for Earth systems have long used optimization, such as for parameter calibration (see 'Similarities' in Box 1), but high-dimensional optimization is always challenging because of computational expenses. Only 20 mingxi zhang gradient-based optimization, which updates the network weights by explicitly tracking their contributions to the outcome, makes it computationally tractable to learn from big data and efficiently train the large numbers of parameters necessary to approximate complex unknown functions.

The ability of generic NN architectures such as transformers, CNNs, and recurrent NNs to approximate unknown functions has produced desirable outcomes (Fig. 1). First, researchers from any field can concentrate on a few generic architectures, permitting cross-domain sharing of knowledge and experience. Second, NNs can help to identify previously unrecognized physical relationships. Third, NN training can scale up with the data (in terms of accuracy, generalizability, and efficiency)[79,85], unlike PBMs in which learning can quickly saturate after some limited calibration of parameters or functions[59]. All of these abilities are only possible because NNs can be trained with a large number of network weights, providing a large learnable function space[86,87]. The number of weights easily exceeds the capabilities of conventional optimization algorithms for PBMs. The LSTM models widely used in hydrology can contain ~500,000 weights whereas large language models developed since about 2018 already have trillions of weights, which can lead to the emergence of intelligence not observed at smaller scales[88]. In contrast, traditional evolutionary[89–91], genetic[92], or particle swarm optimization methods[93] can hardly handle more than a few dozen independent parameters (Box 1).

The computing framework that trains NNs with large amounts of weights is known as differentiable programming[94,95]. This approach involves designing programmes in such a way that their outputs are differentiable with respect to inputs, using cheaply obtained gradients to update parameters via various first-order gradient-descent methods[96]. Differentiable programming in NNs is largely enabled by automatic differentiation (AD), which decomposes a complex algorithm into a sequence of elementary arithmetic operations and then applies the

21 mingxi zhang rule of differentiation to compute the derivatives. Reverse-mode or forward-mode AD is provided by ML platforms such as PyTorch[97], JAX[98], Julia[99], and Tensorflow[100]. Models written on these platforms can often be easily made programmatically differentiable even with mathematically indifferentiable operations (such as thresholding or IF statements), as long as they are piecewise differentiable.

Therefore, we suggest that differentiable programming is the feature that distinguishes NNs from other types of models, owing to its ability to efficiently learn from large amounts of data and tune a very large number of parameters. Recognizing that differentiable programming is not exclusive to ML reveals a pathway to unify NNs and geoscientific PBMs. This unification requires only minor modifications to conceptual modelling and implementation strategies but could open new doors for scientific discovery.

Differentiable modelling

22-23 2 notes: In this section, the scope of the discussion is expanded beyond differentiable programming and AD, and the term differentiable modelling (DM) (Fig. 1) is used to refer to joint physics–NN modelling approaches that use any method for rapidly and accurately producing gradients to achieve the large-scale optimization of the combined system. A distinct

24 mingxi zhang feature of DM is the requirement for predominant programmatical differentiability – that is, the whole model must support gradient calculation from the start to the end of the workflow – to ensure that the trained NNs can adapt and evolve based on the data. Purely data-driven NNs already use differentiable programming (almost entirely through AD), but 'differentiable modelling' is used here to also emphasize the hybrid nature of the overall approach.

25-26 2 notes: An alternative to AD is 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 low-level calculations, whereas adjoint methods differentiate using higher-level functions or mathematical equations such as nonlinear equations or differential equations[104]. Other gradient estimation methods, such as finite

27 mingxi zhang difference approaches, are intractable for any reasonably sized NNs (10,000 weights would require 10,001 forward model evaluations). Second-order methods, such as the Newton–Raphson method, have not gained popularity for the training of NNs owing to the costs and

28 mingxi zhang challenges of computing the Hessian matrix. Many NNs are implemented on platforms that support differentiable programming, whereas most existing PBMs are not.

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 interpretations. There are two perspectives from which differentiable models can be viewed (Fig. 2b). First, they can be viewed as ML models that are constrained to a smaller searchable space by the structural priors (model structures and equations representing scientific understanding or hypotheses and kept unchanged during model training). Thus, DM can still reap the benefits of big data when available. Second, they can be viewed as PBMs that are augmented with learnable and adaptable components (and thus an expanded searchable space) provided by NNs, can be trained in data-scarce scenarios, and provide elucidation of processes.

Approximating functions inside the model

Although efficient gradient calculation might seem to be merely a technical change, it could also lead to a transformation of modelling philosophies. First, the ability to approximate complex, unknown functions using data can broaden the type of questions that can be asked, by treating trusted model components as priors and focusing on improving representations of the uncertain components. This idea can be explained in concise mathematical terms using a physics-based model g,

$$y = g(u, x, \theta), \tag{1}$$

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 of a physics-based model is generic and encompasses differential equations, for example:

$$\partial u/\partial t = g(u, x, \theta). \tag{2}$$

Traditional inversion algorithms estimate values of parameters in question (essentially asking, "θ = ?") and require that the functional form of the model g is assumed a priori (except for some rigid methods such as non-parametric regression, which require complicated derivations and specialized training algorithms, and thus have not gained popularity). However, differentiable models make it possible to interrogate the functional form of g, by training, for instance, a neural network (NN) on observed data to replace g:

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

where W represents the high-dimensional weights. The function that is estimated with this approach could also be a parameterization scheme, as in differentiable parameter learning[99], for example:

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

where A is some raw information relevant to the physical parameters θ. DM makes it possible to place questions precisely in the model, to extract fine-grained relationships from data (Supplementary Fig. 1). For example, for a model written simply as

$$y = g(g_1, g_2, g_3(u, x, \theta)) \tag{5}$$

where g_1, g_2, g_3 are process equations as subcomponents of the model, g_3 can be replaced with an NN:

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

As mentioned earlier, equation (6) can also encompass differential equations:

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

The differential equation terms including the NN^{W} will be integrated in time using numerical approaches. NN^{W} could represent rainfall–runoff relationships[107], or a constitutive relationship producing effective hydraulic conductivities in a subsurface reactive transport model[108,109].

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, and g_2. The unchanged parts (structural priors) such as g, g_1, g_2 serve as physical constraints. Insights can be gained by simply visualizing the

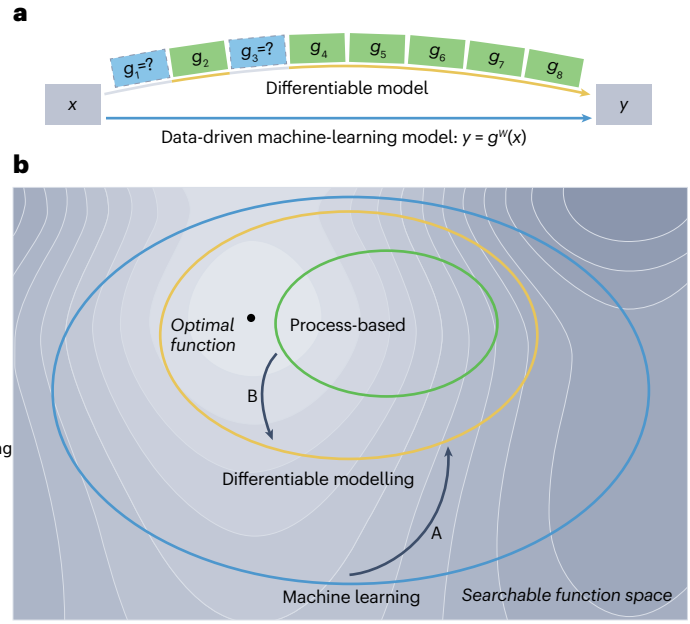


Fig. 2 | Why differentiable models are interpretable and performant. **a**, Purely data-driven machine-learning (ML) models learn direct mapping relationships from x to y, which intertwines many processes and is thus difficult to understand. Differentiable models make it possible to break the model into portions g_n (for example, here g_1 and g_3 are unknown) to narrow the scope of the relationships to be learned (potentially with fewer data than training a pure ML model) for better interpretability. **b**, Differentiable models can be viewed as either: ML models that are guided into smaller searchable spaces (ovals) by structural priors; or process-based models with expanded search space supported by learnable units. The background colour gradient indicates model optimality, related to the cost function if there were an infinite amount of data available. The location marked as optimal (with a dot) indicates the ideal model solution. Differentiable models are more likely to contain and discover the ideal model solution, thus enhancing both process understanding and predictive accuracy.

29
mingxi zhang

30
mingxi zhang

relationships learned from NN^w (refs. 46,110) or by applying knowledge distillation methods[111]. Better process representations can also be obtained for some model components such as g_3, meaning that questions can be posed with high precision and flexibility. Moreover, as a purely data-driven NN learns an overall mapping from x to y, it can intertwine many processes, making it and its results difficult to interpret. By breaking the mapping down into multiple subparts and inserting prior knowledge, the scope of learning and the complexities of the learned relationships are inherently reduced, improving the interpretability and robustness of the conclusions (Fig. 2a; Supplementary Fig. 1).

31-32
2 notes:

DM 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 PBMs first posit hypotheses and then test them using data. DM posits a user-defined number of structural assumptions, and then identifies other parts of the model from the 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].

State-of-the-art predictive performance of DM

Purely data-driven ML architectures have set a high bar for accuracy in multiple geoscience domains; therefore, it could seem plausible that there would be a substantial loss in accuracy when less-flexible process-based components are added. However, it is uncertain whether generic ML architectures are necessary to achieve good model accuracy when we can use NNs as components of a model to learn from and adapt to data. It is easy to see that ML-level performance could be achieved if the searchable space of the PBM is enlarged to include a good approximation of the true function (Fig. 2b), directed by gradient-based training. The paths taken to upgrade the models will be dependent on experts' intuition; therefore, it could take some time for unified approaches to er[illegible].

[illegible] dynamical systems in the geosciences, such as rainfall–runoff in a basin, crop growth, or nutrient release, can be described by ODEs. To solve these equations, the numerical model is run for many steps. This approach is mathematically similar to recurrent NNs, and the time integration operation is similar to the functionality achieved by some NNs such as residual networks[113,114]. Therefore, it should not be surprising that learnable PBMs with some ML components can perform as well as deep networks.

As we will discuss with some geoscience examples, it has already been demonstrated that the performance of differentiable, learnable models can approach that of purely data-driven models, and in some cases, when extrapolation is key, even exhibit advantages. Compared with purely data-driven ML, DM trades generality for interpretability and the ability to ask specific questions, and yet may also not sacrifice accuracy.

Differentiable modelling in geosciences

Here we advocate for a new modelling genre for Earth and environmental processes, which we call DM in geosciences. DM in geosciences combines 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 (Box 2). DM could also exploit gradients for other purposes such as sensitivity analysis or trajectory optimization where human influences on Earth systems are simulated, such as with reservoir management. DM seeks to integrate the optimizing and learning capabilities of NN models with geoscientific process descriptions. Differentiable models could evolve to gain process knowledge while improving the model predictions.

The features of a successful DM model could include predictive accuracy and transferability equal to or greater than that of purely data-driven models for extensively measured variables; structural evolution capabilities, which could improve the parameterization and formulation of the processes; accurate generalizability to data-sparse regions or the long-term future; conservation of mass, energy, and momentum; consistency of internal physical fluxes and states that can provide a full narrative of the events and full support to downstream processes; and efficient isolation of one uncertain model component at a time to learn physics with reduced ambiguity.

Suitability of DM for geosciences

DM is well suited to geoscience applications owing to the nature of the datasets and problems. First, geoscientific data are strongly imbalanced in spatial extent, in temporal coverage, and in the number of variables observed, and there is noise in the observational datasets owing to instrument limitations and observability. Although satellites

can be used to indirectly estimate many variables[115] including global leaf area index[116] and coarse-resolution surface soil moisture (SMOS, soil moisture and ocean salinity)[117,118], other variables such as photosynthesis rates[119], soil respiration, and streamflow are only measured at a limited number of sites, especially in Africa and Asia[120]. Additionally, there is very limited knowledge of subsurface properties, such as the thicknesses, depths, and conductivities of aquifers. Purely data-driven ML can be biased by these data limitations; however, these biases could be partially alleviated by including physics as an inductive bias. Indeed, preliminary analysis shows that differentiable models with a PBM as the backbone can outperform LSTM in regional extrapolation[121].

The second major motivation for using DM in the geosciences is the non-stationarity of processes such as climate and land use and cover, which could drive many systems out of the previously observed range of variability[122]. Although the performance of ML models is highly competitive[70,121], the accuracy declines substantially when faced with non-stationary processes[121,123]. Therefore, DM could potentially represent future trends better than purely data-driven models because it is constrained by physical formulations[121].

Third, DM can output any diagnostic variable calculated using the process-based equations within a model; therefore, model conditioning and/or data assimilation operations can be performed with sparse and scattered data. Model conditioning is where the model is constrained using observations to improve overall model dynamics. For example, a hydrological model can be conditioned by satellite soil moisture or streamflow data so that it can obtain more accurate predictions of vegetation water use[124], primary productivity, or snow water equivalent[125]. For data assimilation, the model uses observations of one variable to improve the short-term forecast of another variable. Additionally, observations of the first variable can be used to update the state variables of the model.

Finally, DM could improve the quality of physical parameters, which strongly control the behaviours of the models. There is often no ground-truth information for the parameters, and they require inversion from observations or high-resolution simulations. Since about the 1980s, parameter estimation has been fraught with uncertainty and ambiguity. Because different parameters can produce similar outputs and are sensitive to spatiotemporal resolutions, calibration at a geographic location can often lead to equifinality[126–128]. Extending parameters to unmonitored locations requires regionalization, which can improve robustness, but it is difficult for traditional regionalization methods to achieve optimal results. Training NNs as parameter generators could improve parameter generalization and performance, while also providing insights about parameter sensitivity. Using all the available data points to constrain the parameters can generate favourable scaling behaviours with more training data leading to improved performance, efficiency, and generalizability[124].

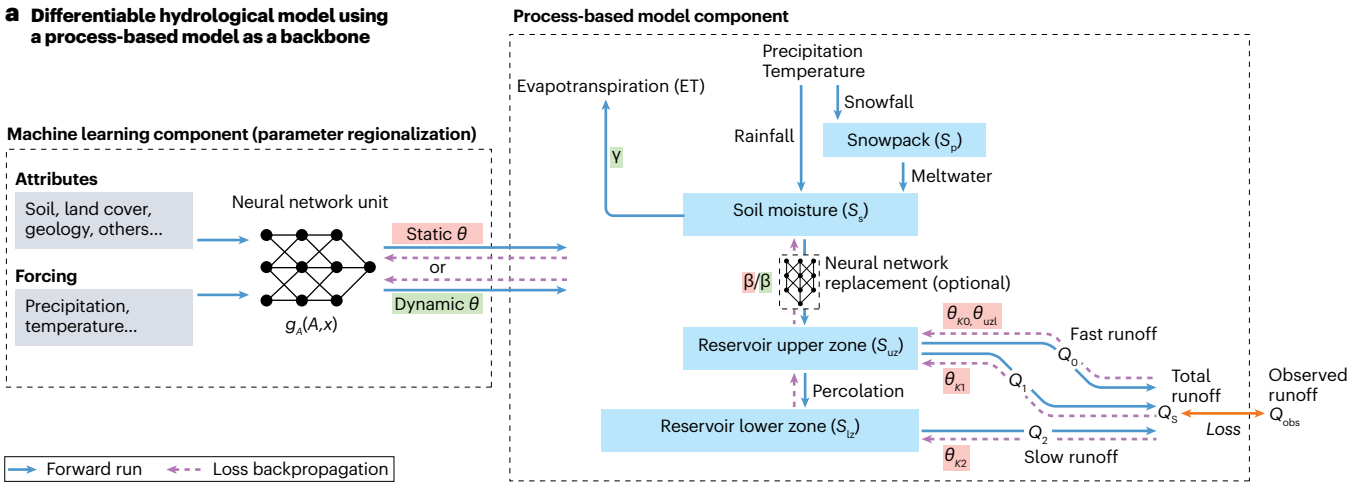
The cost of obtaining differentiable. Reimplementing a model into a differentiable form can incur non-trivial developmental costs. Mathematical changes might be required to adapt previously non-differentiable mathematical operations, for example by replacing

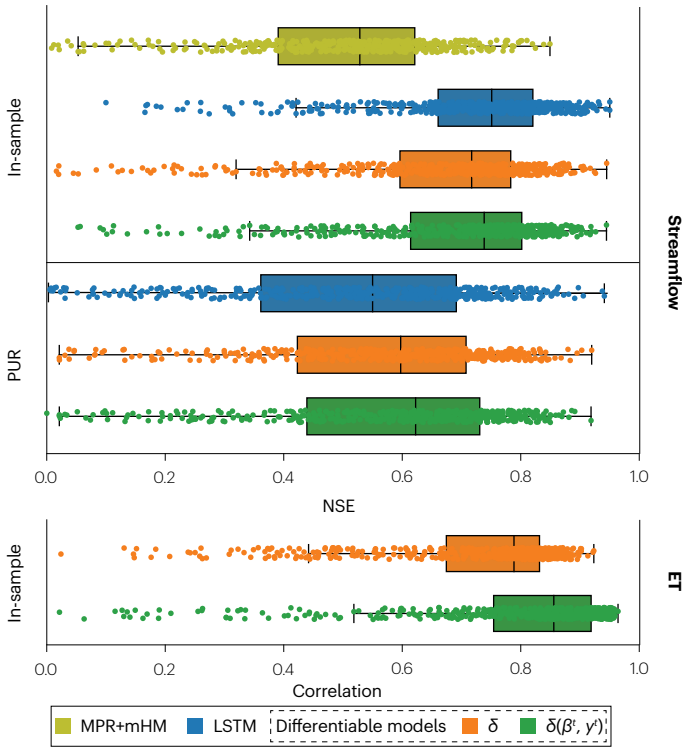
Box 2

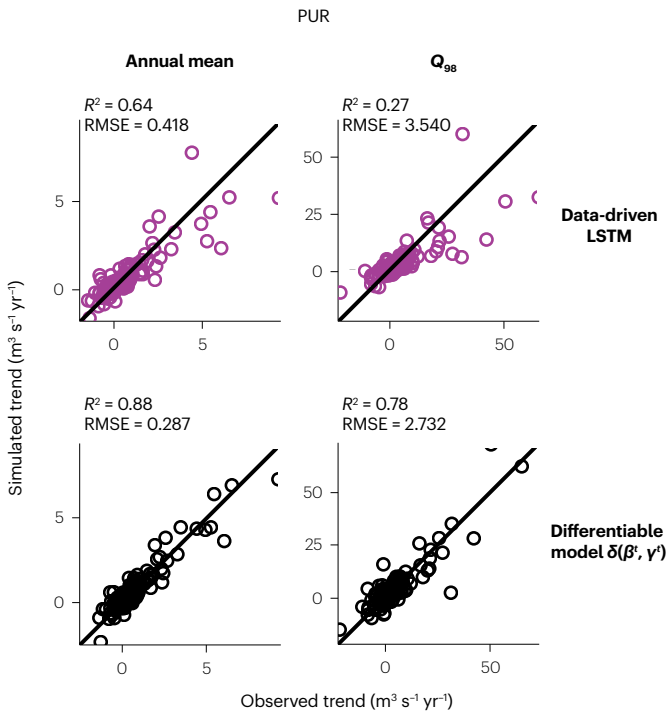
Geoscience questions that differentiable modelling could help to answer

Differentiable modelling (DM) could help almost all geoscientific domains in knowledge discovery and improving simulation quality. Some core domains and example questions are as follows.

- **What is the relationship between x and y?**
  - **Hydraulics:** How do we estimate floodplain hydraulic parameter values efficiently at large scales using new sensing data?
  - **Hydrology:** How does global groundwater-dominated baseflow respond to climate change?
- **What physics is missing from this differential equation?**
  - **Soil science:** How we find functional forms to express soil hydraulic properties (water retention and hydraulic conductivity) that describe non-equilibrium flow?
- **What should be the assumption here?**
  - **Ecosystems:** What is the main driver of reduced plant production: vapour pressure deficit or deficit in soil moisture?
  - **Hydrology:** What is a proper, scale-appropriate way to parameterize groundwater storage and flow at the global scale?
- **How does factor A influence parameter β?**
  - **Geohazards:** Can we use space-based observations of geohazards such as landslides to quantify subsurface properties (so that we can better predict future events)?
  - **Water quality:** 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 with intensified human modification?
- **Is a process causing phenomenon P?**
  - **Climate:** Is CO_2 fertilizing plants and increasing global photosynthesis?
- **What will happen under new environmental conditions?**
  - **Agriculture:** How can we predict crop phenology dynamics (for example, planting, shooting, flowering, harvesting) and assess potential production risk under future climate change, which involves interconnected biotic, abiotic, and human influences?
  - **Cryosphere:** How can we use both physics and data to create more accurate models for ice dynamics within the cryosphere and better constrain its fate under climate change?
- **What is the information content of datasets (inputs, training targets)?**
  - **Coastal:** How can we better leverage emerging sensing platforms while improving our model representations of sediment transport and nonlinear wave–wave interactions to infer nearshore bathymetry at large scales?







indexing with convolutions, or to improve parallel efficiency. Although DM does not have to run on graphical processing units (GPUs), enabling the use of GPUs would improve the computational efficiency by orders of magnitude compared with approaches that mainly use central processing units (CPUs), notwithstanding some current challenges. Our opinion is that in most cases, the cost incurred is worth the investment owing to the potential to interrogate the model, make changes, and learn physics. The reimplementation could also provide an opportunity to re-examine many of the common model assumptions or implementation choices.

41-42
2 notes:

Classes of DM in geosciences

NNs can be used in various ways for geoscientific problems, ranging from learning physical parameters[124] to updating structural assumptions in a component of the model[79], or estimating time-dependent ing terms of the natural systems. However, we emphasize that different from previous concepts introduced in physics-guided ML or not-fully-differentiable models in terms of methodology (the equations in DM must be fully differentiable), mission (DM aims to advance process understanding), and philosophy (whether DM treats physical laws as truth). This section briefly describes early explorations

Fig. 3 | An example differentiable hydrological model. a, Sketch of a differentiable hydrological model using the process-based HBV model as a backbone. The purple dashed lines illustrate the back-propagation paths used to train the two embedded neural networks (NNs), but back-propagation can update any component including making corrections to precipitation. Green symbol backgrounds indicate static parameters while pink symbol backgrounds indicate time-dependent parameters. **b**, Differentiable models (δ with static parameters and $\delta(\beta^t,\gamma^t)$ with two time-dependent parameters) can approach the performance of machine learning (ML) models such as long short-term memory (LSTM) networks and greatly outperform a traditional parameterization approach (multiscale parameter regionalization, MPR) applied to the mesoscale hydrological model (mHM) for the in-sample temporal test. They also outperform LSTM in a spatial extrapolation test for predictions in ungauged regions (PUR). Results are based on the basins from the Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset, where high Nash–Sutcliffe efficiency (NSE) values indicate better performance. Bottom, the differentiable models can output evapotranspiration with high accuracy whereas LSTM cannot at all. The accuracy was assessed using the correlation of the output with a satellite product. **c**, For PUR (representing spatial extrapolation: trained in some regions and tested in another large ungauged region), the performance of the differentiable models (black) surpasses that of the LSTM networks (pink) with higher R^2 and lower root-mean-square error (RMSE) for predictions of the decadal-scale trends in annual mean streamflow (left) or high flow (right, Q_{98}). This framework delivers high accuracy, robust generalizability, interpretability, multiphysical outputs, and computational efficiency at the same time. ET, evapotranspiration; P, precipitation; Q_0, quick flow; Q_1, shallow subsurface flow; Q_2, baseflow; Q_S, simulated streamflow; Q_{obs}, observed streamflow; S_p, snowpack water storage; S_s, soil water storage; S_{uz}, upper subsurface zone water storage; S_{lz}, lower subsurface zone water storage; T, temperature; θ_{uzl}, upper subsurface threshold for quick flow; θ_{K0} recession coefficient for quick flow; θ_{K1}, recession coefficient for shallow subsurface flow; θ_{K2}, recession coefficient for baseflow; β, shape coefficient of the runoff relationship; γ, newly added dynamic shape coefficient of the evapotranspiration relationship. $g_A(A,x)$ is the neural network unit to learn physical parameters, which is fed static attributes (A) as well as meteorological forcing data (x). Figure 3a,b adapted with permission from ref. 107, Wiley. Figure 3b,c adapted with permission from ref. 121, CC BY 4.0, http://creativecommons.org/licenses/by/4.0/.

and demonstrations of DM in geosciences, categorized by how the gradients are computed and used. The examples discussed in this section, which are not exhaustive, explain the concepts and are meant to inspire further innovation.

Differentiating through numerical models. The most straightforward DM approach is to differentiate through numerical models by leveraging ML platforms; this approach is also the most similar to traditional models. Both AD and customized backward functions (adjoint) can be used to keep track of gradients at relatively elementary levels of operations.

Automatic differentiation is the most obvious low-hanging fruit to adapt existing models into DM. For models without iterative solvers, ML platforms such as PyTorch, Julia, or JAX could be used to reimplement an existing physical model coded in Fortran or C/C++ to obtain a differentiable model version through AD and ensure reproducibility. The differentiable model can then be connected to NNs by simply suppling inputs to the NN and feeding its outputs to the rest of the model. The physical laws are enforced, providing an efficient forward simulator for any initial, boundary, and forcing conditions. The relationships learned using this approach could also be transferred to existing models to immediately support operational work such as flood and weather forecasting or food production estimates.

For problems that need iterative solvers, such as systems of nonlinear equations or stiff ODEs that require implicit time-stepping, direct AD can consume too much memory, but adjoint-based backward functions could be used instead at the iterative solver level (known as the discretize-then-optimize approach). This method can be mixed and matched with AD. Alternatively, adjoint functions could be written at the differential equation level, in which case the adjoint differential equation is solved backwards in time to compute the gradients (known as the optimize-then-discretize approach)[114]. Care must be taken with the optimize-then-discretize approach as sometimes low-accuracy gradients are obtained, which can interfere with the training of the model[129].

Adjoint methods have also been used to solve optimization problems governed by PDEs, by either deriving the adjoint equations manually[130] or more rarely with automated programmes[131]. Adjoint methods have the potential to be more computationally efficient than AD for certain problems. Adjoint solvers have long been successfully used for numerical weather predictions for the purpose of data assimilation, with systems such as 4DVar[132], and groundwater modelling[133], for the purpose of calibration. However, these approaches did not use NN training machinery, perhaps because the role of differentiable programming was not clear at the time.

A geoscientific example of differentiating through numerical models is the adaptation of the conceptual hydrological model Hydrologiska Byråns Vattenbalansavdelning (HBV), a system of ODEs. HBV was reimplemented on PyTorch and coupled to NNs which provide a regionalized parameterization[107] (with 'regionalized' meaning that the parameterization is trained by all sites simultaneously, which provides a strong constraint and improves the robustness of the model) (Fig. 3). Strikingly, the reimplemented HBV can simulate streamflow with accuracy close to that of LSTM[107]. The soil moisture–runoff relationship can be replaced with an NN to learn the relationship between soil moisture, precipitation, and runoff (similar to a constitutive relationship) for threshold-like watershed systems. This implementation also output untrained variables such as evapotranspiration and baseflow, which agreed well with alternative estimates. AD was the main method of obtaining gradients, but to improve the numerical accuracy and parameter robustness of the model[134,135] adjoint backward functions for implicit time-stepping can also be incorporated. Moreover, in spatial extrapolation cases, the differentiable model moderately outperformed ML models (LSTM in this case) with respect to daily metrics like the Nash–Sutcliffe model efficiency coefficient, as well as decadal-scale trends[121] (Fig. 3) owing to the structural constraints, demonstrating its potential for global hydrological modelling.

Similarly, the hydrological model EXP-HYDRO was encoded as a recurrent NN architecture and coupled with fully connected NNs, which served as the parameterization pipeline and the post-processor to improve runoff[125]. Integrating NNs with physics in these models led to robust transferability across basins. Additionally, hybrid neural ODE approaches, in which NNs replace the differential-equation-based hydrological model, produce more accurate predictions than single-basin LSTMs, but retain the interpretability of a mechanistic model[136]. Differentiable models have also been used for biogeophysical and ecosystem modelling, to improve parameterization for photosynthesis[137] at large scales.

Apart from models similar to ODEs, direct differentiation can also be applied to models that operate on graphs to represent natural systems, such as river networks. An advective dispersion equation implemented on a river graph to simulate streamwater temperature performed better than LSTM in data-sparse situations[138]. Similarly, a differentiable river routing model was trained on daily discharge at a gauge downstream of a river network to learn a parameterization scheme for Manning's roughness coefficient (n) (ref. 139). It learned a power-law-like curve for the relationship between n and catchment area, which is consistent with the expected behaviour from the literature.

Another, more adjoint-focused example uses NNs to replace unknown functions or operators in a PDE, before discretizing the PDE using a finite element method, and calculating the gradient with the adjoint method; this approach recovered a nonlinear coefficient for the Poisson and heat equations[130]. To overcome the challenge with Newton iteration convergence (an approach to find the root to systems of nonlinear equations) owing 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, whereas the other subproblem with NNs can be solved by integrating NNs using a Gaussian quadrature rule. This approach can be similarly applied to equations in geosciences such as subsurface reactive transport equations.

Connecting NNs with PBMs through surrogate models. If an NN is trained as a surrogate for a PBM (reproducing its behaviour), this surrogate model can then be connected to other NN components in a DM framework because the NN surrogate is programmatically differentiable. Many studies trained surrogate models for hydrological, hydraulic[140,141], and reactive transport models, and then further used the surrogate models for inversion[142] and optimization. In 2021, differentiable parameter learning (dPL) by Tsai et al.[124] was able to exploit differentiable nature of such a surrogate model for training. They connected a surrogate model of the Variable Infiltration Capacity (VIC) process-based hydrological model to a neural network (g) that estimates physical parameters of VIC ($\boldsymbol{\theta}$) using some widely available attributes ($\mathbf{A}$): $\boldsymbol{\theta} = g(\mathbf{A})$. 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 an ML problem, trained on all sites simultaneously using back-propagation and gradient descent (Fig. 4a). 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 coher[illegible] upplementary Fig. 2) and extrapolate better in space (Fig. 4b,c). [illegible]s hyperefficient: a job that normally takes a 100-CPU cluster two to three days now takes a single GPU one hour. dPL allows the combined model to output unobserved variables while alleviating the notorious problem of parameter equifinality[126], and exhibits a favourable scaling relationship (with increasing volumes of data).

The initial effort associated with the surrogate model approach is low compared with fully recoding a model; however, the surrogate models might need to be continuously retrained as the optimization goes to different regions of the parameter or state space. Furthermore,

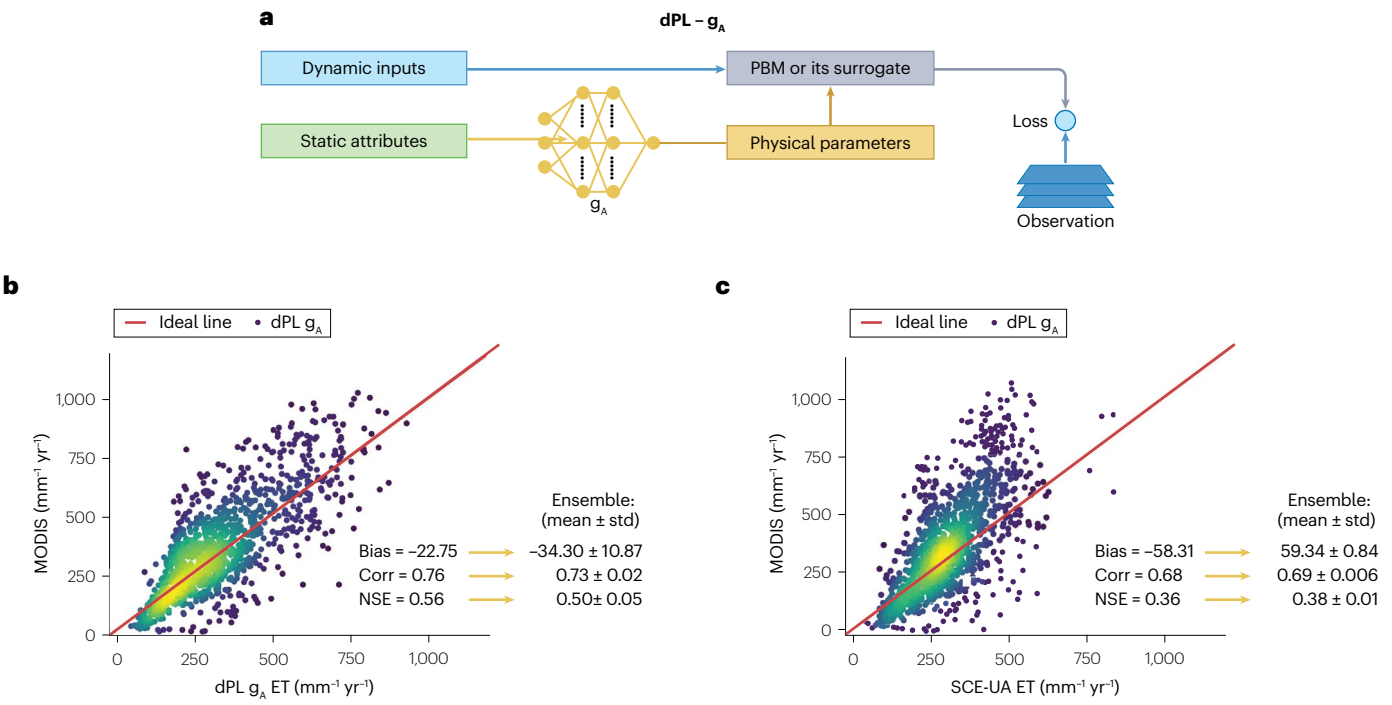


Fig. 4 | Differentiable parameter learning. a, Diagram showing the basic structure of a model using the differentiable parameter learning (dPL) framework called g_A. **b**,**c**, Comparisons of uncalibrated evapotranspiration (ET) rates (mm yr^{-1}) estimated from dPL and traditional algorithms. dPL has a stronger correlation with observational data from the moderate resolution imaging spectroradiometer (MODIS) satellite product than the traditional site-by-site calibrated shuffled complex evolutionary algorithm (SCE-UA). Each point on the plot is the temporal mean ET from 1/8 of a latitude–longitude–degree NLDAS-2 grid-cell. Yellow colour indicates higher density of points. The dPL algorithm works with both fully differentiable and surrogate models, and obtains good generalizability via a global loss function. This figure is adapted from ref. 124, CC BY 4.0, http://creativecommons.org/licenses/by/4.0/.

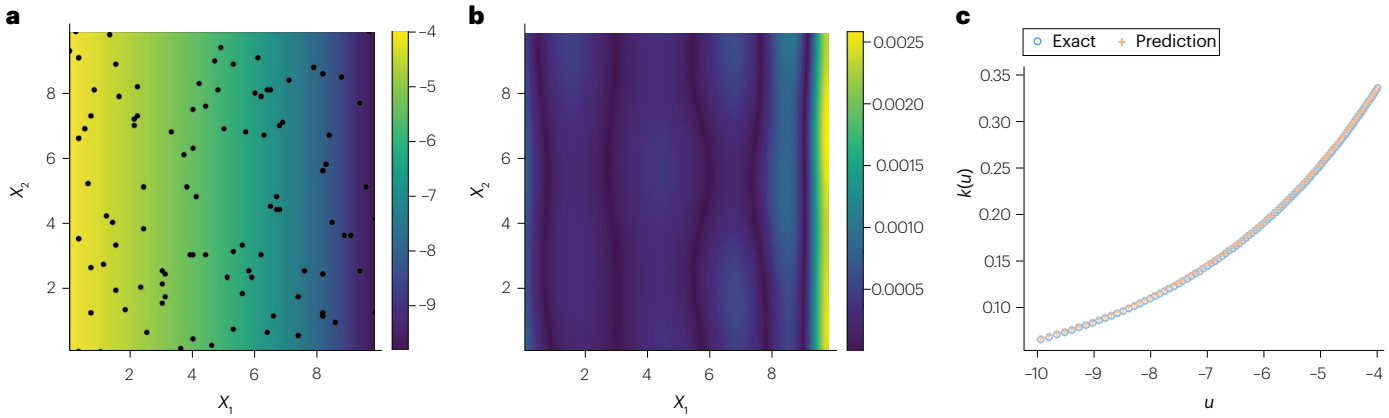


Fig. 5 | Physics-informed neural networks for learning constitutive relationships in unsaturated flow models. a, The locations of observational pressure head measurements (dots) for steady-state groundwater flow in the vadose zone of an unconfined aquifer system with an unknown pressure-dependent conductivity. The quantities of interest were the unsaturated conductivity k as a function of the pressure head u, and the pressure head field $u(x)$. The physics-informed neural network (PINN) method was used to estimate the unknowns with neural networks in a differentiable model of the unsaturated flow in the vadose zone and the reference pressure head field (yellow–blue background colour). **b**, The point errors in the estimated head field. **c**, The reference and estimated conductivities as functions of the pressure head. The PINN method can be perceived as a genre of differentiable modelling, as the gradient information is critically employed. It can learn both the state variable and the constitutive relationship very accurately. Figure is adapted with permission from ref. 109, Wiley.

48-49 2 notes: surrogate model does not allow direct changes to the model structure. Therefore, this approach is only recommended for highly complex and computationally expensive models that are challenging to reimplement. Such cases could arise in climate models, or hydraulics or 50 subsurface modelling, in which the governing PDEs that describe fluid dynamics and sediment transport must be solved with high spatial and temporal resolution requiring non-trivial computational code or complex boundary conditions. Although not the primary focus or philosophical theme of DM, surrogate models can certainly accelerate and aid its purpose. The possibility of solving PDEs with NNs has also attracted increasing attention, with NNs being used in many cases to approximate the numerical solution of the PDEs[143–145], such as the Richards equation, which describes the movement of water in soil[146]. Differentiable surrogate models can also be used to invert bathymetry measurements for 2D hydraulic simulations[147].

mingxi zhang

Physics-informed neural networks. Although first published in 2017, physics-informed neural networks (PINNs)[109,148–150] could be perceived as a category of DM as the gradient information is critically used. PINNs pose problems in a unique way, seeking to train a neural network with space–time coordinates as inputs, $h(t,x)$ where t is time and x represents spatial coordinates such that $h(t,x)$ agrees with known data points at (t,x). The derivatives dh/dx, dh/dt and so forth, which can be obtained via AD, are forced to satisfy the governing partial differential equations. Physical parameters could also be part of the inputs to the h network[150]. PINNs have been tested on applications in many domains, and there have been a number of good reviews of this work[145,151]. PINNs have many uses such data assimilation[149] and learning governing equations, but, as with other methods, there are also 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 easy 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] (Fig. 5). 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 pressure-dependent 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 5a shows the reference pressure head field and the locations of the measurements. Figure 5b shows the point errors in the estimated pressure head field. The reference and estimated unsaturated conductivity functions are shown in Fig. 5c. These figures demonstrate that the PINN method can learn both the state variable and the constitutive relationship very accurately.

Compared with DM methods, PINNs have a unique design that directly learns the problem-dependent space–time solution of states. As a result, PINNs focus on knowledge discovery rather than being efficient forward simulators.

51 mingxi zhang

ML-dominant hybrid models with limited physics. Another class of models that can be used for data-rich applications uses NNs for most of the modelling but inserts physical operators for imposing the restrictions from a limited number of physical laws. For example, LSTM has been used to estimate physical surface fluxes such as evaporation, runoff, and recharge constrained only by the mass balance equations[152]. The only control on the fluxes was the observations of discharge; therefore, it was uncertain whether the flux terms maintained their physical meaning. The system was later constrained using more observations[153]. A model learning from two data sources outperformed (raising the correlation with observations from 0.8 to 0.91) those learning from only one source when applied to the prediction of soil moisture using LSTM trained at 9-km resolution[154]. The solutions from the LSTM were fed into an averaging operation to obtain outputs with 36-km resolution, and loss functions were computed at both resolutions and compared with in situ and satellite-based observations. Overall, ML-dominant systems can be strong predictors and a beneficial option in DM; however, the interpretability and physical significance of the diagnostic intermediate variables must be carefully assessed to avoid misinterpreting the results.

Differentiable modelling and physics-guided ML

DM has the potential to tackle a diverse array of questions across various geoscientific domains (Box 2), pursuing ambitious goals ranging from high accuracy to knowledge discovery. Many strategies have been proposed to integrate ML with physics in a seemingly scattered manner with a range of complexities, such that it is difficult to classify these approaches[155]. It has not been sufficiently recognized that some of these algorithms work fundamentally because they leverage differentiable programming. The scattered nature of the literature makes the landscape of ML–physics integration daunting and confusing, while hindering innovations based on first principles. However, the concept of differentiability can help understand proposed methods by identifying whether 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 differentiable modelling in geosciences and some existing methods.

52 mingxi zhang

DM and physics-guided (or theory-guided, or knowledge-guided) machine learning (PGML)[156–158] both seek to combine physics with ML, but they differ in their approaches, purposes, and philosophies. PGML has been used to introduce physical constraints, for example, as regularization or pretraining, to ML methods to gain better generalizability with less training data. In theory, PGML does not need to use differentiable programming and usually enforces only a part of the physics. In contrast, DM uses numerical physical models as the backbone and demands that the entire workflow be differentiable.

53 mingxi zhang

The goal of PGML is to make the ML model more robust, whereas DM seeks to update assumptions or discover new knowledge. Philosophically, when a physical law is introduced in PGML, which often includes all the calculations and assumptions to support the law, it is treated as truth (albeit sometimes with some tolerance level[59]). In DM, the physical laws are not presumed to be correct, and opportunities to update existing knowledge are constantly sought.

There are many methods that are not fully differentiable that could also be valuable but are beyond the scope of DM[159]. For example, ML algorithms trained offline on datasets could be incorporated into physical models, such as training an NN on turbulent heat fluxes and inserting it into a hydrological model[160]; training pedotransfer functions that can infer soil parameters from soil hydraulic data[161]; training an atmospheric parameterization network on short-term cloud-resolving simulations[162]; or training ocean-mixing parameterizations on data and physical constraints[163]. Although this approach has the advantage that the physical meaning of the NN is clear, direct training data are needed for the variable of interest, and the network cannot evolve and adapt interactively, for instance to update the model following further observations. In the future, these NNs might be incorporated into DM. Other offline coupling methods include providing outputs of process-based models (PBMs) as inputs to NNs to integrate over spatiotemporal heterogeneity[164,165], or training ML models to predict the PBM residuals[166–168].

Summary and future perspectives

In this Perspective we discuss how DM allows varying amounts of structural priors to be flexibly used along with NNs, ranging from having just a few physically based operators to substantially physically based structures. Therefore, DM could dissolve the divide between ML and PBM. Understanding the role of differentiable programming makes it possible to break free from thinking about fixed methods or possibilities for integrating these two approaches, instead focusing on physical priors, uncertainty, unknown relationships, and data. DM is suitable for many geoscientific applications, can learn from multiple sources of data and multi-scale datasets, and can leverage the benefits of big or small data. Beyond the proof-of-concept stage of DM, the next stage can focus on science needs: updating inadequate assumptions, filling knowledge gaps, addressing long-term model deficiencies, and delivering practical, scalable benefits to science and society. Many geoscientific findings, habitual assumptions, and operational practices can be updated by DM.

Computational challenges will arise with DM and may spur advances in computer science. Memory usage and vanishing gradients are major issues when training NNs, especially in cases involving iterative numerical solvers. Keeping track of gradients requires storing information (this requirement is partly alleviated if checkpointing is applied but it is still a prominent issue) and thus uses memory, which is especially constrained with GPUs. 'Vanishing gradients' means that the parameters in deeper layers of the model have very small gradients, so they become difficult to train[169,170]. Vanishing gradients can occur with recurrent NNs, which are similar to differentiable models. Moreover, differentiable models might have heterogeneous operations (unlike NNs, which predominantly use matrix multiplications) so it might be challenging to optimize the use of the GPU. For the core DM algorithms, as DM opens up a new avenue, new challenges will emerge, driving new solutions to address them.

Although current differentiable computing platforms can readily accommodate numerical solvers for ODEs, solving PDEs could still be challenging because it requires substantial computation and memory, which makes it expensive (in terms of computational power and memory usage) to train the connected NNs with a batch of examples. Architectures suitable for big-data ML training often use massive parallelism, which reduces the range of numerical algorithms that can be used. Modellers working in DM now need to understand both forward and backward methods, adding to the mathematical learning

curve. Nevertheless, some differentiable numerical PDE solvers have been proposed and tested in computational fluid mechanics, and seem to be a suitable alternative to standard solvers[171]. In the future, suitable methods are needed to improve the efficiency of gradient tracking for complex numerical schemes.

DM makes it possible to learn processes; therefore, it is to be expected that the problem of 'process non-uniqueness', also called 'process equifinality', could arise. In traditional hydrological modelling, multiple working hypotheses must be proposed to test different model formulations coupled together[33]. With DM, we need systematic development approaches that can solve part of the problem or determine one process at a time to reduce interactions between the modules. Furthermore, more mature uncertainty quantification techniques are needed to help assess the successes and failures of hypotheses going beyond ensemble methods[166,172–174]. Finally, large and multivariate benchmarks and extrapolation tests that match the intended use cases to verify the validity and realism of the physical outputs will be useful. For example, models for assessing the impact of climate change must be tested for long-term projection fidelity and models for global-scale applications must pass rigorous spatial extrapolation tests[120].

The rise in the use of AI for big data and models has been astonishing[88,151]. We argue that both prediction accuracy and knowledge discovery in the geosciences can be improved by combining advanced AI model architectures with physics using a differentiable programming framework. Although it is perceived as a technological advance, DM can also lead to philosophical changes: for example, it could make it possible to ask new questions and test hypotheses on model structure or data usage, and therefore use data more effectively. We look forward to DM greatly advancing geosciences.

Published online: 11 July 2023

References

1. Ajami, N. K., Gupta, H., Wagener, T. & Sorooshian, S. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. *J. Hydrol.* **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. *J. Hydroinform.* **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. *J. Geophys. Res. Biogeosci.* **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. *Geosci. Model. Dev.* **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. *Comput. Electron. Agric.* **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. *Hydrol. Earth Syst. Sci.* **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. Li, L. et al. Toward catchment hydro-biogeochemical theories. *WIREs Water* **8**, e1495 (2021).
11. Steefel, C. I. et al. Reactive transport codes for subsurface environmental simulation. *Comput. Geosci.* **19**, 445–478 (2015).
12. Li, L. et al. Expanding the role of reactive transport models in critical zone processes. *Earth Sci. Rev.* **165**, 280–301 (2017).
13. Flato, G. M. Earth system models: an overview. *WIREs Clim. Change* **2**, 783–800 (2011).
14. Danabasoglu, G. et al. The Community Earth System Model version 2 (CESM2). *J. Adv. Modeling Earth Syst.* **12**, e2019MS001916 (2020).
15. Eyring, V. et al. Overview of the Coupled Model Intercomparison Project phase 6 (CMIP6) experimental design and organization. *Geosci. Model. Dev.* **9**, 1937–1958 (2016).
16. Calvin, K. et al. GCAM v5.1: representing the linkages between energy, water, land, climate, and economic systems. *Geosci. Model. Dev.* **12**, 677–698 (2019).
17. ISIMIP. The Inter-Sectoral Impact Model Intercomparison Project (ISIMIP). https://www.isimip.org/ (2022).
18. Lange, S. Trend-preserving bias adjustment and statistical downscaling with ISIMIP3BASD (v1.0). *Geosci. Model. Dev.* **12**, 3055–3070 (2019).
19. Weyant, J. et al. in *Climate Change 1995: Social and Economic Dimensions of Climate Change* (eds Bruce, J. P., Lee, H. & Haites, E. F.) 367–396 (IPCC, Cambridge Univ. Press, 1996).
20. IPCC. *Climate Change 2021: The Physical Science Basis* (eds Masson-Delmotte, V. et al.) (Cambridge Univ. Press, 2021).
21. Clark, M. P. et al. Improving the representation of hydrologic processes in Earth system models. *Water Resour. Res.* **51**, 5929–5956 (2015).
22. Geary, W. L. et al. A guide to ecosystem models and their environmental applications. *Nat. Ecol. Evol.* **4**, 1459–1471 (2020).
23. Fatichi, S. et al. An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. *J. Hydrol.* **537**, 45–60 (2016).
24. Wagener, T. et al. On doing hydrology with dragons: realizing the value of perceptual models and knowledge accumulation. *WIREs Water* **8**, e1550 (2021).
25. Hood, R. R. et al. The Chesapeake Bay program modeling system: overview and recommendations for future development. *Ecol. Model.* **456**, 109635 (2021).
26. Fan, Y. et al. Hillslope hydrology in global change research and Earth system modeling. *Water Resour. Res.* **55**, 1737–1772 (2019).
27. van Kampenhout, L. et al. Improving the representation of polar snow and firn in the Community Earth System Model. *J. Adv. Modeling Earth Syst.* **9**, 2583–2600 (2017).
28. Medlyn, B. E. et al. Using ecosystem experiments to improve vegetation models. *Nat. Clim. Change* **5**, 528–534 (2015).
29. Nearing, G. S. et al. What role does hydrological science play in the age of machine learning? *Water Resour. Res.* **57**, e2020WR028091 (2021).
30. Shen, C. et al. HESS Opinions: incubating deep-learning-powered hydrologic science advances as a community. *Hydrol. Earth Syst. Sci.* **22**, 5639–5656 (2018).
31. Hunt, R. J., Fienen, M. N. & White, J. T. Revisiting 'An exercise in groundwater model calibration and prediction' after 30 years: insights and new directions. *Groundwater* **58**, 168–182 (2020).
32. Addor, N. & Melsen, L. A. Legacy, rather than adequacy, drives the selection of hydrological models. *Water Resour. Res.* **55**, 378–390 (2019).
33. Clark, M. P., Kavetski, D. & Fenicia, F. Pursuing the method of multiple working hypotheses for hydrological modeling. *Water Resour. Res.* **47**, WR009827 (2011).
34. Jakeman, A. J. & Hornberger, G. M. How much complexity is warranted in a rainfall-runoff model? *Water Resour. Res.* **29**, 2637–2649 (1993).
35. Wagener, T., Wheater, H. S. & Gupta, H. V. in *Calibration of Watershed Models* 29–47 (Wiley, 2003).
36. Young, P., Jakeman, A. & McMurtrie, R. An instrumental variable method for model order identification. *Automatica* **16**, 281–294 (1980).
37. Shen, C. A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resour. Res.* **54**, 8558–8593 (2018).
38. Abbott, B. W. et al. Human domination of the global water cycle absent from depictions and perceptions. *Nat. Geosci.* **12**, 533–540 (2019).
39. Lemordant, L., Gentine, P., Swann, A. S., Cook, B. I. & Scheff, J. Critical impact of vegetation physiology on the continental hydrologic cycle in response to increasing CO_2. *Proc. Natl Acad. Sci. USA* **115**, 4093–4098 (2018).
40. Trancoso, R., Larsen, J. R., McVicar, T. R., Phinn, S. R. & McAlpine, C. A. CO_2–vegetation feedbacks and other climate changes implicated in reducing base flow. *Geophys. Res. Lett.* **44**, 2310–2318 (2017).
41. Yu, D. et al. Socio-hydrology: an interplay of design and self-organization in a multilevel world. *Ecol. Soc.* **25**, 22 (2020).
42. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
43. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
44. Yin, H., Guo, Z., Zhang, X., Chen, J. & Zhang, Y. RR-Former: rainfall-runoff modeling based on transformer. *J. Hydrol.* **609**, 127781 (2022).
45. Amanambu, A. C., Mossa, J. & Chen, Y.-H. Hydrological drought forecasting using a deep transformer model. *Water* **14**, 3611 (2022).
46. Sun, A. Y., Jiang, P., Mudunuru, M. K. & Chen, X. Explore spatio-temporal learning of large sample hydrology using graph neural networks. *Water Resour. Res.* **57**, e2021WR030394 (2021).
47. Krizhevsky, A., Sutskever, I. & Hinton, G. E. in *Advances in Neural Information Processing Systems* vol. 25 (eds Bartlett, P. et al.) 1097–1105 (Curran Associates, 2012).
48. Lecun, Y. & Bengio, Y. Convolutional networks for images, speech, and time-series. in *The Handbook of Brain Theory and Neural Networks* (ed. Arbib, M. A.) 255–258 (MIT Press, 1995).
49. Khaki, S. & Wang, L. Crop yield prediction using deep neural networks. *Front. Plant Sci.* **10**, 00621 (2019).
50. Wang, A. X., Tran, C., Desai, N., Lobell, D. & Ermon, S. Deep transfer learning for crop yield prediction with remote sensing data. In *Proceedings of the 1st ACM SIGCAS Conference on Computing and Sustainable Societies*, 1–5 (ACM, 2018).
51. Pan, B. et al. Improving seasonal forecast using probabilistic deep learning. *J. Adv. Model. Earth Syst.* **14**, e2021MS002766 (2022).

52. Shi, X. et al. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In *Proceedings of the 28th International Conference on Neural Information Processing Systems* Vol. 1, 802–810 (MIT Press, 2015).
53. Bhowmik, M., Singh, M., Rao, S. & Paul, S. DeepClouds.ai: deep learning enabled computationally cheap direct numerical simulations. Preprint at https://doi.org/10.48550/arXiv.2208.08956 (2022).
54. Lin, G.-Y., Chen, H.-W., Chen, B.-J. & Yang, Y.-C. Characterization of temporal PM2.5, nitrate, and sulfate using deep learning techniques. *Atmos. Pollut. Res.* **13**, 101260 (2022).
55. Varadharajan, C. et al. Can machine learning accelerate process understanding and decision-relevant predictions of river water quality? *Hydrol. Process.* **36**, e14565 (2022).
56. Jia, X. et al. Physics-guided recurrent graph model for predicting flow and temperature in river networks. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, 612–620 (Society for Industrial and Applied Mathematics, 2021).
57. Rahmani, F. et al. Exploring the exceptional performance of a deep learning stream temperature model and the value of streamflow data. *Environ. Res. Lett.* https://doi.org/10.1088/1748-9326/abd501 (2021).
58. Rahmani, F., Shen, C., Oliver, S., Lawson, K. & Appling, A. Deep learning approaches for improving prediction of daily stream temperature in data-scarce, unmonitored, and dammed basins. *Hydrol. Process.* **35**, e14400 (2021).
59. Read, J. S. et al. Process-guided deep learning predictions of lake water temperature. *Water Resour. Res.* **55**, 9173–9190 (2019).
60. Zhi, W. et al. From hydrometeorology to river water quality: can a deep learning model predict dissolved oxygen at the continental scale? *Environ. Sci. Technol.* **55**, 2357–2368 (2021).
61. Zhi, W., Ouyang, W., Shen, C. & Li, L. Temperature outweighs light and flow as the predominant driver of dissolved oxygen in US rivers. *Nat. Water* **1**, 249–260 (2023).
62. He, M., Wu, S., Huang, B., Kang, C. & Gui, F. Prediction of total nitrogen and phosphorus in surface water by deep learning methods based on multi-scale feature extraction. *Water* **14**, 1643 (2022).
63. Hrnjica, B., Mehr, A. D., Jakupović, E., Crnkić, A. & Hasanagić, R. Application of deep learning neural networks for nitrate prediction in the Klokot River, Bosnia and Herzegovina. In *7th International Conference on Control, Instrumentation and Automation (ICCIA)* 1–6 (IEEE, 2021).
64. Xiong, R. et al. Predicting dynamic riverine nitrogen export in unmonitored watersheds: leveraging insights of AI from data-rich regions. *Environ. Sci. Technol.* **56**, 10530–10542 (2022).
65. Shen, C., Chen, X. & Laloy, E. Editorial: broadening the use of machine learning in hydrology. *Front. Water*, https://doi.org/10.3389/frwa.2021.681023 (2021).
66. Fang, K., Shen, C., Kifer, D. & Yang, X. Prolongation of SMAP to spatiotemporally seamless coverage of continental U.S. using a deep learning neural network. *Geophys. Res. Lett.* **44**, 11030–11039 (2017).
67. Fang, K., Pan, M. & Shen, C. The value of SMAP for long-term soil moisture estimation with the help of deep learning. *IEEE Trans. Geosci. Remote. Sens.* **57**, 2221–2233 (2019).
68. Fang, K. & Shen, C. Near-real-time forecast of satellite-based soil moisture using long short-term memory with an adaptive data integration kernel. *J. Hydrometeorol.* **21**, 399–413 (2020).
69. Feng, D., Fang, K. & Shen, C. Enhancing streamflow forecast and extracting insights using long-short term memory networks with data integration at continental scales. *Water Resour. Res.* **56**, e2019WR026793 (2020).
70. Kratzert, F. et al. Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrol. Earth Syst. Sci.* **23**, 5089–5110 (2019).
71. Xiang, Z. & Demir, I. Distributed long-term hourly streamflow predictions using deep learning — a case study for State of Iowa. *Environ. Model. Softw.* **131**, 104761 (2020).
72. Alemohammad, S. H. et al. Water, energy, and carbon with artificial neural networks (WECANN): a statistically based estimate of global surface turbulent fluxes and gross primary productivity using solar-induced fluorescence. *Biogeosciences* **14**, 4101–4124 (2017).
73. Jung, M. et al. The FLUXCOM ensemble of global land-atmosphere energy fluxes. *Sci. Data* **6**, 74 (2019).
74. Zhao, W. L. et al. Physics-constrained machine learning of evapotranspiration. *Geophys. Res. Lett.* **46**, 14496–14507 (2019).
75. Afzaal, H., Farooque, A. A., Abbas, F., Acharya, B. & Esau, T. Groundwater estimation from major physical hydrology components using artificial neural networks and deep learning. *Water* **12**, 5 (2020).
76. Meyal, A. Y. et al. Automated cloud based long short-term memory neural network based SWE prediction. *Front. Water* **2**, 574917 (2020).
77. McDonnell, J. J. & Beven, K. Debates — the future of hydrological sciences: a (common) path forward? A call to action aimed at understanding velocities, celerities and residence time distributions of the headwater hydrograph. *Water Resour. Res.* **50**, 5342–5350 (2014).
78. Appling, A. P., Oliver, S. K., Read, J. S., Sadler, J. M. & Zwart, J. Machine learning for understanding inland water quantity, quality, and ecology. *earthArXiv* preprint at https://doi.org/10.1016/B978-0-12-819166-8.00121-3 (2022).
79. Fang, K., Kifer, D., Lawson, K., Feng, D. & Shen, C. The data synergy effects of time-series deep learning models in hydrology. *Water Resour. Res.* **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. *Environ. Data Sci.* **1**, e6 (2022).
81. Schölkopf, B. in *Probabilistic and Causal Inference: The Works of Judea Pearl* vol. 36, 765–804 (Association for Computing Machinery, 2022).
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. *Digit. Signal Process.* **73**, 1–15 (2018).
84. Toms, B. A., Barnes, E. A. & Ebert-Uphoff, I. Physically interpretable neural networks for the geosciences: applications to Earth system variability. *J. Adv. Modeling Earth Syst.* **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 Netw.* **4**, 251–257 (1991).
87. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**, 359–366 (1989).
88. Bubeck, S. et al. Sparks of artificial general intelligence: early experiments with GPT-4. Preprint at https://arxiv.org/abs/2303.12712 (2023).
89. Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* **6**, 182–197 (2002).
90. Duan, Q., Sorooshian, S. & Gupta, V. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.* **28**, 1015–1031 (1992).
91. Zitzler, E., Laumanns, M. & Thiele, L. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. *TIK Report* vol. 103 (ETH Zurich, 2001); https://www.research-collection.ethz.ch/handle/20.500.11850/145755.
92. Liu, S. et al. A hybrid approach of support vector regression with genetic algorithm optimization for aquaculture water quality prediction. *Math. Comput. Model.* **58**, 458–465 (2013).
93. Zambrano-Bigiarini, M. & Rojas, R. A model-independent particle swarm optimisation software for model calibration. *Environ. Model. Softw.* **43**, 5–25 (2013).
94. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **18**, 1–43 (2018).
95. Innes, M. et al. A differentiable programming system to bridge machine learning and scientific computing. Preprint at https://arxiv.org/abs/1907.07587 (2019).
96. Goodfellow, I., Bengio, Y. & Courville, A. in *Deep Learning* (ed. Goodfellow, I.) Ch. 4 (MIT Press, 2016).
97. Paszke, A. et al. Automatic differentiation in PyTorch. In *31st Conference on Neural Information Processing Systems (NIPS 2017)* (NIPS, 2017).
98. Bradbury, J. et al. JAX: Autograd and XLA. *Astrophysics Source Code Library* record ascl:2111.002 (Astrophysics Source Code Library, 2021).
99. Bezanson, J., Edelman, A., Karpinski, S. & Shah, V. B. Julia: a fresh approach to numerical computing. *SIAM Rev.* **59**, 65–98 (2017).
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? *Bull. Am. Meteorol. Soc.* **78**, 2577–2592 (1997).
102. Johnson, S. G. Notes on Adjoint Methods for 18.335 (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).
106. Rackauckas, C. et al. Universal differential equations for scientific machine learning. Preprint at https://doi.org/10.48550/arXiv.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 Resour. Res.* **58**, e2022WR032404 (2022).
108. Huang, D. Z., Xu, K., Farhat, C. & Darve, E. Learning constitutive relations from indirect observations using deep neural networks. *J. Comput. Phys.* **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 Resour. Res.* **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. *Sci. Adv.* **6**, eaay2631 (2020).
112. Ma, Y., Tsao, D. & Shum, H.-Y. On the principles of parsimony and self-consistency for the emergence of intelligence. *Front. Inf. Technol. Electron. Eng.* **23**, 1298–1323 (2022).
113. Haber, E. & Ruthotto, L. Stable architectures for deep neural networks. *Inverse Probl.* **34**, 014004 (2018).
114. 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, 2018).
115. Mahecha, M. D. et al. Earth system data cubes unravel global multivariate dynamics. *Earth Syst. Dyn.* **11**, 201–234 (2020).

116. Myneni, R., Knyazikhin, Y. & Park, T. MCD15A2H MODIS/Terra+Aqua Leaf Area Index/FPAR 8-day L4 Global 500m SIN Grid V006 (2015); https://doi.org/10.5067/MODIS/MCD15A2H.006.
117. *About SMOS — Soil Moisture and Ocean Salinity Mission* (ESA, 2022); https://earth.esa.int/eogateway/missions/smos.
118. 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); https://doi.org/10.5067/4DQ54OUIJ9DL.
119. Lin, Y.-S. et al. Optimal stomatal behaviour around the world. *Nat. Clim. Change* **5**, 459–464 (2015).
120. 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. *Geophys. Res. Lett.* **48**, e2021GL092999 (2021).
121. 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 (European Geoscience Union, 2022); https://doi.org/10.5194/hess-2022-245.
122. Wagener, T. et al. The future of hydrology: an evolving science for a changing world. *Water Resour. Res.* **46**, 1–10 (2010).
123. 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. *Geosci. Model. Dev.* **16**, 1553–1567 (2023).
124. 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).
125. Jiang, S., Zheng, Y. & Solomatine, D. Improving AI system awareness of geoscience knowledge: Symbiotic integration of physical approaches and deep learning. *Geophys. Res. Lett.* **47**, e2020GL088229 (2020).
126. Beven, K. A manifesto for the equifinality thesis. *J. Hydrol.* **320**, 18–36 (2006).
127. Pokhrel, P., Gupta, H. V. & Wagener, T. A spatial regularization approach to parameter estimation for a distributed watershed model. *Water Resour. Res.* **44**, WR006615 (2008).
128. 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).
129. Onken, D. & Ruthotto, L. Discretize-optimize vs. optimize-discretize for time-series regression and continuous normalizing flows. Preprint at https://arxiv.org/abs/2005.13420 (2020).
130. Mitusch, S. K., Funke, S. W. & Kuchta, M. Hybrid FEM-NN models: combining artificial neural networks with the finite element method. *J. Comput. Phys.* **446**, 110651 (2021).
131. 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).
132. 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).
133. Neupauer, R. M. & Wilson, J. L. Adjoint-derived location and travel time probabilities for a multidimensional groundwater system. *Water Resour. Res.* **37**, 1657–1668 (2001).
134. Clark, M. P. & Kavetski, D. Ancient numerical daemons of conceptual hydrological modeling: 1. Fidelity and efficiency of time stepping schemes. *Water Resour. Res.* **46**, W10510 (2010).
135. Kavetski, D. & Clark, M. P. Ancient numerical daemons of conceptual hydrological modeling: 2. Impact of time stepping schemes on model analysis and prediction. *Water Resour. Res.* **46**, W10511 (2010).
136. Höge, M., Scheidegger, A., Baity-Jesi, M., Albert, C. & Fenicia, F. Improving hydrologic models for predictions and process understanding using neural ODEs. *Hydrol. Earth Syst. Sci.* **26**, 5085–5102 (2022).
137. Aboelyazeed, D. et al. A differentiable ecosystem modeling framework for large-scale inverse problems: demonstration with photosynthesis simulations. *Biogeosci. Discuss.* https://doi.org/10.5194/bg-2022-211 (2022).
138. 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 (IEEE, 2021); https://doi.org/10.1109/ICDM51629.2021.00011.
139. 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).
140. Forghani, M. et al. Application of deep learning to large scale riverine flow velocity estimation. *Stoch. Environ. Res. Risk Assess.* **35**, 1069–1088 (2021).
141. Forghani, M. et al. Variational encoder geostatistical analysis (VEGAS) with an application to large scale riverine bathymetry. *Adv. Water Resour.* **170**, 104323 (2022).
142. 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 Resour. Res.* **51**, 5957–5973 (2015).
143. Blechschmidt, J. & Ernst, O. G. Three ways to solve partial differential equations with neural networks — a review. *GAMM-Mitteilungen* **44**, e202100006 (2021).
144. Lu, L., Meng, X., Mao, Z. & Karniadakis, G. E. DeepXDE: a deep learning library for solving differential equations. *SIAM Rev.* **63**, 208–228 (2021).
145. Takamoto, M. et al. PDEBENCH: an extensive benchmark for scientific machine learning. Preprint at https://arxiv.org/abs/2210.07182 (2022).
146. 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).
147. 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).
148. 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. *J. Comput. Phys.* **378**, 686–707 (2019).
149. He, Q., Barajas-Solano, D., Tartakovsky, G. & Tartakovsky, A. M. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Adv. Water Resour.* **141**, 103610 (2020).
150. Wang, N., Zhang, D., Chang, H. & Li, H. Deep learning of subsurface flow via theory-guided neural network. *J. Hydrol.* **584**, 124700 (2020).
151. Brown, T. B. et al. Language models are few-shot learners. In *Proc. of the 34th International Conference on Neural Information Processing Systems*, 1877–1901 (2020).
152. Kraft, B., Jung, M., Körner, M. & Reichstein, M. in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* vol. XLIII-B2-2020, 1537–1544 (Copernicus, 2020).
153. Kraft, B., Jung, M., Körner, M., Koirala, S. & Reichstein, M. Towards hybrid modeling of the global hydrological cycle. *Hydrol. Earth Syst. Sci.* **26**, 1579–1614 (2022).
154. Liu, J., Rahmani, F., Lawson, K. & Shen, C. A multiscale deep learning model for soil moisture integrating satellite and in situ data. *Geophys. Res. Lett.* **49**, e2021GL096847 (2022).
155. Karniadakis, G. E. et al. Physics-informed machine learning. *Nat. Rev. Phys.* **3**, 422–440 (2021).
156. Karpatne, A. et al. Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **29**, 2318–2331 (2017).
157. Khandelwal, A. et al. Physics guided machine learning methods for hydrology. https://arxiv.org/abs/2012.02854 (2020).
158. Pawar, S., San, O., Aksoylu, B., Rasheed, A. & Kvamsdal, T. Physics guided machine learning using simplified theories. *Phys. Fluids* **33**, 011701 (2021).
159. Reichstein, M. et al. Deep learning and process understanding for data-driven Earth system science. *Nature* **566**, 195–204 (2019).
160. Bennett, A. & Nijssen, B. Deep learned process parameterizations provide better representations of turbulent heat fluxes in hydrologic models. *Water Resour. Res.* **57**, e2020WR029328 (2021).
161. Schaap, M. G., Leij, F. J. & van Genuchten, M. Th. Rosetta: a computer program for estimating soil hydraulic parameters with hierarchical pedotransfer functions. *J. Hydrol.* **251**, 163–176 (2001).
162. Rasp, S., Pritchard, M. S. & Gentine, P. Deep learning to represent subgrid processes in climate models. *Proc. Natl Acad. Sci. USA* **115**, 9684–9689 (2018).
163. Zhu, Y. et al. Physics-informed deep-learning parameterization of ocean vertical mixing improves climate simulations. *Natl Sci. Rev.* **9**, nwac044 (2022).
164. 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).
165. 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).
166. 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 Resour. Res.* **57**, e2021WR029772 (2021).
167. Frame, J. M. et al. Post-processing the national water model with long short-term memory networks for streamflow predictions and model diagnostics. *J. Am. Water Resour. Assoc.* **57**, 885–905 (2021).
168. Sun, A. Y., Jiang, P., Yang, Z.-L., Xie, Y. & Chen, X. A graph neural network (GNN) approach to basin-scale river network learning: the role of physics-based connectivity and data fusion. *Hydrol. Earth Syst. Sci. Discuss.* **26**, 5163–5184 (2022).
169. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Syst.* **06**, 107–116 (1998).
170. Hochreiter, S., Bengio, Y., Frasconi, P. & Jürgen S. in *A Field Guide to Dynamical Recurrent Neural Networks* (eds Kremer, S. C. & Kolen, J. F.) 237–244 (IEEE, 2001).
171. Kochkov, D. et al. Machine learning–accelerated computational fluid dynamics. *Proc. Natl Acad. Sci. USA* **118**, e2101784118 (2021).
172. 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 Resour. Res.* **56**, e2020WR028095 (2020).
173. 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).
174. Krapu, C. & Borsuk, M. A differentiable hydrology approach for modeling with time-varying parameters. *Water Resour. Res.* **58**, e2021WR031377 (2022).

Acknowledgements

The authors are grateful for the discussion at 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. C.S., Y.S. and T.B. were supported by National Science Foundation Award EAR-2221880, 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, respectively. P.G. 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. M. Wernimont at the US Geological Survey (USGS) greatly improved the presentation of Figs. 1 and 2. A.P.A. was 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 US Government.

Author contributions

C.S. researched data for the article. C.S., A.P.A., P.G., T.B., H.G., Y.Z., A.T., M.B.-J., F.F., D.K., L.L., X.L. and W.R. contributed substantially to discussion of the content. C.S. wrote the initial article. All authors reviewed and/or edited the manuscript before submission.

Competing interests

K.L. and C.S. have financial interests in HydroSapient, Inc.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43017-023-00450-9.

Peer review information *Nature Reviews Earth & Environment* thanks A. McGovern, G. Varando and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

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]US 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]Energy Geoscience Divisions, Earth and Environmental Sciences Area, Lawrence Berkeley National Laboratory, Berkeley, 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]Climate and Ecosystem Sciences Divisions, Earth and Environmental Sciences Area, 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, Cambridge, MA, USA. [19]Department of Biological and Agricultural Engineering, Texas A&M University, College Station, TX, USA. [20]Civil and Environmental Engineering, University of Nebraska-Lincoln, Lincoln, NE, USA. [21]Earth and Environmental Sciences Division, Los Alamos National Laboratory, New Mexico, NM, USA.

Differentiable modelling to unify machine learning and physical models for geosciences

Shen, Chaopeng; Appling, Alison P.; Gentine, Pierre; Bandai, Toshiyuki; Gupta, Hoshin; Tartakovsky, Alexandre; Baity-Jesi, Marco; Fenicia, Fabrizio; Kifer, Daniel; Li, Li; Liu, Xiaofeng; Ren, Wei; Zheng, Yi; Harman, Ciaran J.; Clark, Martyn; Farthing, Matthew; Feng, Dapeng; Kumar, Praveen; Aboelyazeed, Doaa; Rahmani, Farshid; Song, Yalan; Beck, Hylke E.; Bindas, Tadd; Dwivedi, Dipankar; Fang, Kuai; Höge, Marvin; Rackauckas, Chris; Mohanty, Binayak; Roy, Tirthankar; Xu, Chonggang; Lawson, Kathryn

01 mingxi zhang Page 1
18/7/2023 1:48

02 mingxi zhang Page 1
18/7/2023 2:35

03 mingxi zhang Page 3
18/7/2023 1:57

04 mingxi zhang Page 3
18/7/2023 1:58

05 mingxi zhang Page 3
18/7/2023 1:53

06 mingxi zhang Page 3
18/7/2023 2:10

07 mingxi zhang Page 3
18/7/2023 1:55

08 mingxi zhang Page 3

18/7/2023 2:12

09 mingxi zhang Page 3

18/7/2023 2:13

10 mingxi zhang Page 3

18/7/2023 2:12

11 mingxi zhang Page 3

18/7/2023 2:16

12 mingxi zhang Page 3

18/7/2023 2:18

13 mingxi zhang Page 3

18/7/2023 2:28

14 mingxi zhang Page 4

18/7/2023 2:28

15 mingxi zhang Page 4

18/7/2023 2:28

16 mingxi zhang Page 4

18/7/2023 2:33

17 mingxi zhang Page 4

18/7/2023 2:33

18 mingxi zhang Page 4

18/7/2023 2:46

19 mingxi zhang Page 4

18/7/2023 2:50

20 mingxi zhang Page 4

18/7/2023 2:52

21 mingxi zhang Page 5

18/7/2023 2:52

22 mingxi zhang Page 5

18/7/2023 7:52

23 mingxi zhang Page 5

18/7/2023 7:52

24 mingxi zhang Page 5

18/7/2023 3:01

25 mingxi zhang Page 5

18/7/2023 3:09

26 mingxi zhang Page 5

18/7/2023 3:09

27 mingxi zhang Page 5

18/7/2023 3:11

28 mingxi zhang Page 5

18/7/2023 3:11

29 mingxi zhang Page 6

18/7/2023 8:06

30 mingxi zhang Page 6

18/7/2023 8:02

31 mingxi zhang Page 6

18/7/2023 8:03

32 mingxi zhang Page 6

18/7/2023 8:03

33 mingxi zhang Page 7

18/7/2023 3:31

34 mingxi zhang Page 7

18/7/2023 3:32

35 mingxi zhang Page 7

18/7/2023 3:28

36 mingxi zhang Page 7

18/7/2023 3:30

37 mingxi zhang Page 7

18/7/2023 3:37

38 mingxi zhang Page 7

18/7/2023 3:18

39 mingxi zhang Page 7

18/7/2023 3:18

40 mingxi zhang Page 7

18/7/2023 3:17

41 mingxi zhang Page 8

18/7/2023 3:38

42 mingxi zhang Page 8

18/7/2023 3:40

43 mingxi zhang Page 9

18/7/2023 3:46

44 mingxi zhang Page 9

18/7/2023 7:12

45 mingxi zhang Page 9

18/7/2023 8:44

46 mingxi zhang Page 10

18/7/2023 8:47

47 mingxi zhang Page 10

18/7/2023 8:48

48 mingxi zhang Page 11

18/7/2023 8:49

49 mingxi zhang Page 11

18/7/2023 8:15

50 mingxi zhang Page 11

18/7/2023 8:50

51 mingxi zhang Page 12

18/7/2023 8:27

52 mingxi zhang Page 12

18/7/2023 8:32

53 mingxi zhang Page 12

18/7/2023 8:30